Statistical Data Analytics

Foundations for Data Mining, Informatics, and Knowledge Discovery

Walter W. Piegorsch

WILEY

Statistical Data Analytics

Statistical Data Analytics

Foundations for Data Mining, Informatics, and Knowledge Discovery

Walter W. Piegorsch

University of Arizona, USA

WILEY

This edition first published 2015

Registered office
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

Library of Congress Cataloging-in-Publication Data

Piegorsch, Walter W.
Statistical data analytics : foundations for data mining, informatics, and knowledge discovery / Walter W. Piegorsch.
pages cm
Includes bibliographical references and index.
ISBN 978-1-118-61965-0 (cloth : alk. paper) 1. Data mining–Mathematics. 2. Mathematical statistics. I. Title.
QA76.9.D343P535 2015
006.3′12—dc23

2015015327

A catalogue record for this book is available from the British Library.

Typeset in 10/12pt TimesLTStd by SPi Global, Chennai, India

1 2015

To Karen

Contents

Preface **xiii**

Part I Background: Introductory Statistical Analytics **1**

1 Data analytics and data mining 3
- 1.1 Knowledge discovery: finding structure in data 3
- 1.2 Data quality versus data quantity 5
- 1.3 Statistical modeling versus statistical description 7

2 Basic probability and statistical distributions 10
- 2.1 Concepts in probability 10
  - 2.1.1 Probability rules 11
  - 2.1.2 Random variables and probability functions 12
  - 2.1.3 Means, variances, and expected values 17
  - 2.1.4 Median, quartiles, and quantiles 18
  - 2.1.5 Bivariate expected values, covariance, and correlation 20
- 2.2 Multiple random variables* 21
- 2.3 Univariate families of distributions 23
  - 2.3.1 Binomial distribution 23
  - 2.3.2 Poisson distribution 26
  - 2.3.3 Geometric distribution 27
  - 2.3.4 Negative binomial distribution 27
  - 2.3.5 Discrete uniform distribution 28
  - 2.3.6 Continuous uniform distribution 29
  - 2.3.7 Exponential distribution 29
  - 2.3.8 Gamma and chi-square distributions 30
  - 2.3.9 Normal (Gaussian) distribution 32
  - 2.3.10 Distributions derived from normal 37
  - 2.3.11 The exponential family 41

3 Data manipulation 49
3.1 Random sampling 49
3.2 Data types 51
3.3 Data summarization 52
3.3.1 Means, medians, and central tendency 52
3.3.2 Summarizing variation 56
3.3.3 Summarizing (bivariate) correlation 59
3.4 Data diagnostics and data transformation 60
3.4.1 Outlier analysis 60
3.4.2 Entropy* 62
3.4.3 Data transformation 64
3.5 Simple smoothing techniques 65
3.5.1 Binning 66
3.5.2 Moving averages* 67
3.5.3 Exponential smoothing* 69

4 Data visualization and statistical graphics 76
4.1 Univariate visualization 77
4.1.1 Strip charts and dot plots 77
4.1.2 Boxplots 79
4.1.3 Stem-and-leaf plots 81
4.1.4 Histograms and density estimators 83
4.1.5 Quantile plots 87
4.2 Bivariate and multivariate visualization 89
4.2.1 Pie charts and bar charts 90
4.2.2 Multiple boxplots and QQ plots 95
4.2.3 Scatterplots and bubble plots 98
4.2.4 Heatmaps 102
4.2.5 Time series plots* 105

5 Statistical inference 115
5.1 Parameters and likelihood 115
5.2 Point estimation 117
5.2.1 Bias 118
5.2.2 The method of moments 118
5.2.3 Least squares/weighted least squares 119
5.2.4 Maximum likelihood* 120
5.3 Interval estimation 123
5.3.1 Confidence intervals 123
5.3.2 Single-sample intervals for normal (Gaussian) parameters 124
5.3.3 Two-sample intervals for normal (Gaussian) parameters 128
5.3.4 Wald intervals and likelihood intervals* 131
5.3.5 Delta method intervals* 135
5.3.6 Bootstrap intervals* 137
5.4 Testing hypotheses 138
5.4.1 Single-sample tests for normal (Gaussian) parameters 140
5.4.2 Two-sample tests for normal (Gaussian) parameters 142

5.4.3 Walds tests, likelihood ratio tests, and 'exact' tests* 145
5.5 Multiple inferences* 148
5.5.1 Bonferroni multiplicity adjustment 149
5.5.2 False discovery rate 151

Part II Statistical Learning and Data Analytics 161

6 Techniques for supervised learning: simple linear regression 163
6.1 What is "supervised learning?" 163
6.2 Simple linear regression 164
6.2.1 The simple linear model 164
6.2.2 Multiple inferences and simultaneous confidence bands 171
6.3 Regression diagnostics 175
6.4 Weighted least squares (WLS) regression 184
6.5 Correlation analysis 187
6.5.1 The correlation coefficient 187
6.5.2 Rank correlation 190

7 Techniques for supervised learning: multiple linear regression 198
7.1 Multiple linear regression 198
7.1.1 Matrix formulation 199
7.1.2 Weighted least squares for the MLR model 200
7.1.3 Inferences under the MLR model 201
7.1.4 Multicollinearity 208
7.2 Polynomial regression 210
7.3 Feature selection 211
7.3.1 R_p^2 plots 212
7.3.2 Information criteria: AIC and BIC 215
7.3.3 Automated variable selection 216
7.4 Alternative regression methods* 223
7.4.1 Loess 224
7.4.2 Regularization: ridge regression 230
7.4.3 Regularization and variable selection: the Lasso 238
7.5 Qualitative predictors: ANOVA models 242

8 Supervised learning: generalized linear models 258
8.1 Extending the linear regression model 258
8.1.1 Nonnormal data and the exponential family 258
8.1.2 Link functions 259
8.2 Technical details for GLiMs* 259
8.2.1 Estimation 260
8.2.2 The deviance function 261
8.2.3 Residuals 262
8.2.4 Inference and model assessment 264
8.3 Selected forms of GLiMs 265
8.3.1 Logistic regression and binary-data GLiMs 265

8.3.2 Trend testing with proportion data 271
8.3.3 Contingency tables and log-linear models 273
8.3.4 Gamma regression models 281

9 Supervised learning: classification 291
9.1 Binary classification via logistic regression 292
9.1.1 Logistic discriminants 292
9.1.2 Discriminant rule accuracy 296
9.1.3 ROC curves 297
9.2 Linear discriminant analysis (LDA) 297
9.2.1 Linear discriminant functions 297
9.2.2 Bayes discriminant/classification rules 302
9.2.3 Bayesian classification with normal data 303
9.2.4 Naïve Bayes classifiers 308
9.3 *k*-Nearest neighbor classifiers 308
9.4 Tree-based methods 312
9.4.1 Classification trees 312
9.4.2 Pruning 314
9.4.3 Boosting 321
9.4.4 Regression trees 321
9.5 Support vector machines* 322
9.5.1 Separable data 322
9.5.2 Nonseparable data 325
9.5.3 Kernel transformations 326

10 Techniques for unsupervised learning: dimension reduction 341
10.1 Unsupervised versus supervised learning 341
10.2 Principal component analysis 342
10.2.1 Principal components 342
10.2.2 Implementing a PCA 344
10.3 Exploratory factor analysis 351
10.3.1 The factor analytic model 351
10.3.2 Principal factor estimation 353
10.3.3 Maximum likelihood estimation 354
10.3.4 Selecting the number of factors 355
10.3.5 Factor rotation 356
10.3.6 Implementing an EFA 357
10.4 Canonical correlation analysis* 361

11 Techniques for unsupervised learning: clustering and association 373
11.1 Cluster analysis 373
11.1.1 Hierarchical clustering 376
11.1.2 Partitioned clustering 384
11.2 Association rules/market basket analysis 395
11.2.1 Association rules for binary observations 396
11.2.2 Measures of rule quality 397

11.2.3 The Apriori algorithm 398
11.2.4 Statistical measures of association quality 402

A Matrix manipulation 411
A.1 Vectors and matrices 411
A.2 Matrix algebra 412
A.3 Matrix inversion 414
A.4 Quadratic forms 415
A.5 Eigenvalues and eigenvectors 415
A.6 Matrix factorizations 416
A.6.1 QR decomposition 417
A.6.2 Spectral decomposition 417
A.6.3 Matrix square root 417
A.6.4 Singular value decomposition 418
A.7 Statistics via matrix operations 419

B Brief introduction to **R** 421
B.1 Data entry and manipulation 422
B.2 A turbo-charged calculator 426
B.3 **R** functions 427
B.3.1 Inbuilt **R** functions 427
B.3.2 Flow control 429
B.3.3 User-defined functions 429
B.4 **R** packages 430

References **432**

Index **453**

Preface

Every data set tells a story. *Data analytics*, and in particular the statistical methods at their core, piece together that story's components, ostensibly to reveal the underlying message. This is the target paradigm of *knowledge discovery*: distill via statistical calculation and summarization the features in a data set/database that teach us something about the processes affecting our lives, the civilization which we inhabit, and the world around us. This text is designed as an introduction to the statistical practices that underlie modern data analytics.

Pedagogically, the presentation is separated into two broad themes: first, an introduction to the basic concepts of probability and statistics for novice users and second, a selection of focused methodological topics important in modern data analytics for those who have the basic concepts in hand. Most chapters begin with an overview of the theory and methods pertinent to that chapter's focal topic and then expand on that focus with illustrations and analyses of relevant data. To the fullest extent possible, data in the examples and exercises are taken from real applications and are not modified to simplify or "clean" the illustration. Indeed, they sometimes serve to highlight the "messy" aspects of modern, real-world data analytics. In most cases, sample sizes are on the order of 10^2–10^5, and numbers of variables do not usually exceed a dozen or so. Of course, far more massive data sets are used to achieve knowledge discovery in practice. The choice here to focus on this smaller range was made so that the examples and exercises remain manageable, illustrative, and didactically instructive. Topic selection is intended to be broad, especially among the exercises, allowing readers to gain a wider perspective on the use of the methodologies. Instructors may wish to use certain exercises as formal examples when their audience's interests coincide with the exercise topic(s).

Readers are assumed to be familiar with four semesters of college mathematics, through multivariable calculus and linear algebra. The latter is less crucial; readers with only an introductory understanding of matrix algebra can benefit from the refresher on vector and matrix relationships given in Appendix A. To review necessary background topics and to establish concepts and notation, Chapters 1–5 provide introductions to basic probability (Chapter 2), statistical description (Chapters 3 and 4), and statistical inference (Chapter 5). Readers familiar with these introductory topics may wish to move through the early chapters quickly, read only selected sections in detail (as necessary), and/or refer back to certain sections that are needed for better comprehension of later material. Throughout, sections that address more advanced material or that require greater familiarity with probability and/or calculus are highlighted with asterisks (*). These can be skipped or selectively perused on a first reading, and returned to as needed to fill in the larger picture.

The more advanced material begins in earnest in Chapter 6 with techniques for supervised learning, focusing on simple linear regression analysis. Chapters 7 and 8 follow with multiple linear regression and generalized linear regression models, respectively. Chapter 9 completes the tour of supervised methods with an overview of various methods for classification. The final two chapters give a complementary tour of methods for unsupervised learning, focusing on dimension reduction (Chapter 10) and clustering/association (Chapter 11).

Standard mathematical and statistical functions are used throughout. Unless indicated otherwise – usually by specifying a different base – log indicates the natural logarithm, so that $\log(x)$ is interpreted as $\log_e(x)$. All matrices, such as $\mathbf{X}$ or $\mathbf{M}$, are presented in bold uppercase. Vectors will usually display as bold lowercase, for example, $\mathbf{b}$, although some may appear as uppercase (typically, vectors of random variables). Most vectors are in column form, with the operator $^{\mathrm{T}}$ used to denote transposition to row form. In selected instances, it will be convenient to deploy a vector directly in row form; if so, this is explicitly noted.

Much of modern data analytics requires appeal to the computer, and a variety of computer packages and programming languages are available to the user. Highlighted herein is the **R** statistical programming environment (R Core Team 2014). **R**'s growing ubiquity and statistical depth make it a natural choice. Appendix B provides a short introduction to **R** for beginners, although it is assumed that a majority of readers will already be familiar with at least basic **R** mechanics or can acquire such skills separately. Dedicated introductions to **R** with emphasis on statistics are available in, for example, Dalgaard (2008) and Verzani (2005), or online at the Comprehensive **R** Archive Network (CRAN): http://cran.r-project.org/. Also see Wilson (2012).

Examples and exercises throughout the text are used to explicate concepts, both theoretical and applied. All examples end with a ❑ symbol. Many present sample **R** code, which is usually intended to illustrate the methods and their implementation. Thus the code may not be most efficient for a given problem but should at least give the reader some inkling into the process. Most of the figures and graphics also come from **R**. In some cases, the **R** code used to create the graphic is also presented, although, for simplicity, this may only be "base" code without accentuations/options used to stylize the display.

Throughout the text, data are generally presented in reduced tabular form to show only a few representative observations. If public distribution is permitted, the complete data sets have been archived online at http://www.wiley.com/go/piegorsch/data_analytics or their online source is listed. A number of the larger data sets came from from the University of California–Irvine (UCI) Machine Learning Repository at http://archive.ics.uci.edu/ml (Frank and Asuncion, 2010); appreciative thanks are due to this project and their efforts to make large-scale data readily available.

Instructors may employ the material in a number of ways, and creative manipulation is encouraged. For an intermediate-level, one-semester course introducing the methods of data analytics, one might begin with Chapter 1, then deploy Chapters 2–5, and possibly Chapter 6 as needed for background. Begin in earnest with Chapters 6 or 7 and then proceed through Chapters 8–11 as desired. For a more complete, two-semester sequence, use Chapters 1–6 as a (post-calculus) introduction to probability and statistics for data analytics in the first semester. This then lays the foundations for a second, targeted-methods semester into the details of supervised and unsupervised learning via Chapters 7–11. Portions of any chapter (e.g., advanced subsections with asterisks) can be omitted to save time and/or allow for greater focus in other areas.

Experts in data analytics may canvass the material and ask, how do these topics differ from any basic selection of statistical methods? Arguably, they do not. Indeed, whole books can be (and have been) written on the single theme of essentially every chapter. The focus in this text, however, is to highlight methods that have formed at the core of data analytics and statistical learning as they evolved in the twenty-first century. Different readers may find certain sections and chapters to be of greater prominence than others, depending on their own scholarly interests and training. This eclectic format is unavoidable, even intentional, in a single volume such as this. Nonetheless, it is hoped that the selections as provided will lead to an effective, unified presentation.

Of course, many important topics have been omitted or noted only briefly, in order to make the final product manageable. Omissions include methods for missing data/imputation, spurious data detection, novelty detection, robust and ordinal regression, generalized additive models, multivariate regression, and ANOVA (analysis of variance, including multivariate analysis of variance, MANOVA), partial least squares, perceptrons, artificial neural networks and Bayesian belief networks, self-organizing maps, classification rule mining, and text mining, to name a few. Useful sources that consider some of these topics include (a) for missing data/imputation, Abrahantes et al. (2011); (b) for novelty detection, Pimentel et al. (2014); (c) for generalized additive models, Wood (2006); (d) for MANOVA, Huberty and Olejnik (2006); (e) for partial least squares, Esposito Vinzi and Russolillo (2013); (f) for neural networks, Stahl and Jordanov (2012); (g) for Bayesian belief networks, Phillips (2005); (h) for self-organizing maps, Wehrens and Buydens (2007); and (i) for text mining, Martinez (2010), and the references all therein. Many of these topics are also covered in a trio of dedicated texts on statistical learning – also referenced regularly throughout the following chapters – by Hastie et al. (2009), Clarke et al. (2009), and James et al. (2013). Interested readers are encouraged to peruse all these various sources, as appropriate.

By way of acknowledgments, sincere and heartfelt thanks are due numerous colleagues, including Alexandra Abate, Euan Adie, D. Dean Billheimer and the statisticians of the Arizona Statistical Consulting Laboratory (John Bear, Isaac Jenkins, and Shripad Sinari), Susan L. Cutter, David B. Hitchcock, Fernando D. Martinez, James Ranger-Moore, Martin Sill, Debra A. Stern, Hao Helen Zhang, and a series of anonymous reviewers. Their comments and assistance helped to make the presentation much more accessible. Of course, despite the fine efforts of all these individuals, some errors may have slipped into the text and these are wholly my own responsibility. I would appreciate hearing from readers who identify any inconsistencies that they may come across.

Most gracious thanks are also due the editorial team at John Wiley & Sons – Prachi Sinha Sahay, Kathryn Sharples, Heather Kay, and Richard Davies – and their LaTeX support staff led by Alistair Smith. Their patience and professionalism throughout the project's development were fundamental in helping it achieve fruition.

Walter W. Piegorsch
Tucson, Arizona
October 2014

Part I

BACKGROUND: INTRODUCTORY STATISTICAL ANALYTICS

1

Data analytics and data mining

1.1 Knowledge discovery: finding structure in data

The turn of the twenty-first century has been described as the beginning of the (or perhaps "an") Information Age, a moniker that is difficult to dismiss and likely to be understated. Throughout the period, contemporary science has evolved at a swift pace. Ever-faster scanning, sensing, recording, and computing technologies have developed which, in turn, generate data from ever-more complex phenomena. The result is a rapidly growing amount of "information." When viewed as quantitative collections, the term heard colloquially is "Big Data," suggesting a wealth of information – and sometimes disinformation – available for study and archiving. Where once computer processing and disk storage were relegated to the lowly kilobyte (1024 bytes) and megabyte (1024 KB) scales, we have moved past routine gigabyte- (1024 MB) and terabyte- (1024 GB) scale computing and now collect data on the petabyte (1024 TB) and even the exabyte (1024 PB) scales. Operations on the zettabyte scale (1024 EB) are growing, and yottabyte- (1024 ZB) scale computing looms on the horizon. Indeed, one imagines that the brontobyte (1024 YB) and perhaps geopbyte (1024 BB) scales are not far off (and may themselves be common by the time you read this).

Our modern society seems saturated by the "Big Data" produced from these technological advances. In many cases, the lot can appear disorganized and overwhelming – and sometimes it is! – engendering a sort of "quantitative paralysis" among decision makers and analysts. But we should look more closely: through clever study of the features and latent patterns in the underlying information, we can enhance decision- and policy-making in our rapidly changing society. The key is applying careful and proper analytics to the data.

At its simplest, and no matter the size, *data* are the raw material from which *information* is derived. This is only a first step, however: the information must itself be studied and its patterns analyzed further, leading to *knowledge discovery*. [An earlier term was *knowledge discovery in databases*, or "KDD" (Elder and Pregibon 1996), because the data often came from a

Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery,
First Edition. Walter W. Piegorsch.

www.wiley.com/go/piegorsch/data_analytics

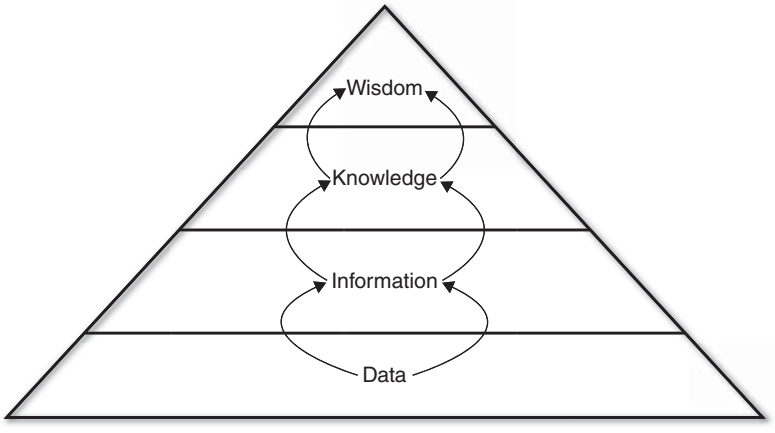

Figure 1.1 The DIKW pyramid.

database repository.] A capstone step in the process integrates and synthesizes the knowledge that has been gained on the phenomenon of interest, to produce true *wisdom* and advance the science. Thus from Data we derive Information, gaining Knowledge and producing Wisdom: D → I → K → W. The effort is sometimes described as a *DIKW pyramid* or *DIKW hierarchy* (Rowley 2007), because each step in the process represents a further refinement on the previous advance. (Some authors have derided the term, suggesting that it underemphasizes and misrepresents the complexities of synthesizing knowledge from information. Nonetheless, the DIKW pyramid still gives a useful framework for conceptualizing the knowledge- and wisdom-discovery process.) Figure 1.1 abstracts the concept.

The DIKW paradigm is by nature a multidisciplinary endeavor: computer scientists construct algorithms to manipulate and organize the data, aided by statisticians and mathematicians who instruct on development and application of quantitative methodology. Then, database experts collect and warehouse the data, software designers write programs that apply the analytic algorithms to the data, engineers build electronics and hardware to implement the programming, and subject-matter/domain experts – that is, biologists, chemists, economists, and social scientists – interpret the findings. To be successful, no one discipline or contributor can operate in a vacuum: each step in the process is enhanced by the interaction and interplay of all participants. (Indeed, the more each contributing discipline informs itself about and involves itself with the others, the more productive the DIKW effort becomes.) It is true interdisciplinarity at work, driving us to the targets, knowledge and wisdom, at the top of the pyramid.

At the base of the pyramid lies the foundation: the data. To advance successfully through each DIKW step, we must apply effective data collection, description, analysis, and interpretation. These all are the purview of statistical science, and it is the methods of modern statistical analysis that lie at the core of data analytics. Thus experience and familiarity with *statistical data analytics* has become a fundamental, necessary skill for any modern scientist dealing with Big Data. Since these methods are applied at the base of the pyramid – and often also throughout the advancing steps – this textbook views them as foundations for the DIKW process. When applied properly, and within the context of the larger interdisciplinary endeavor, features and structures in the data are revealed: for example, clinicians identify susceptible subpopulations in large databases of breast cancer patients, economists study credit card

records for possible trends in purchasing behavior, sociologists track how networks develop among users of social media, and geographers catalog data on natural hazards and highlight localities with increased risk.

It is important to warn that domain-aided interpretation is a necessary component in this process: large data sets can contain structural features which when studied in greater depth represent nothing more than random noise. Teasing out real patterns from any apparent structure is often as much art as science. Nonetheless, when the analytics are successful, they facilitate our ultimate goal of knowledge discovery and advancement in wisdom.

The effort to bore through a large database studying possible patterns of response is often called *data mining*. The term conjures imagery of a miner digging through masses of rock in search of precious stones and is a surprisingly useful metaphor here. A more formal definition is "the process of seeking interesting or valuable information within large data sets" (Hand et al. 2000, p. 111). Larger still (although the two areas need not overlap) is the field of *informatics*, the study and development of quantitative tools for information processing. Many informatic subfields have emerged as data miners and analysts have specialized their focus. Examples include bioinformatics and medical informatics, ecoinformatics, geoinformatics, socioinformatics; the list grows daily! In all these areas, the data-analytic effort relies heavily on proper description, summarization, visualization, and, when necessary, inferential analysis of the collected data mass. The foundational statistical techniques for doing so are the basis of the material presented in this textbook. Some of the focus will be on issues associated with data mining, that is, how one explores collections of data statistically to identify important patterns and structure. Where pertinent, however, connections and extensions to larger applications in informatic science will also gain attention. The material is presented primarily at an introductory level, although the later chapters also give guidance on intermediate and (occasionally) advanced topics.

1.2 Data quality versus data quantity

An often-overlooked issue in data mining and data analytics is the need for sufficiently high quality in the data under study. Data miners regularly remind themselves of the GIGO principle: "if **G**arbage goes **I**n, **G**arbage come **O**ut" (Hand et al. 2001, Section 2.6). That is, the quality and value of any data mining or informatic analysis is contingent upon the quality of the underlying data. In and of itself, a large database is oftentimes an important resource; however, quantity of data does not always equate with quality of information. The data must themselves possess a level of quality commensurate with the questions they are asked to address.

This concern is worth emphasizing at an early stage in any data-analytic effort and should be kept in mind as the calculations and analyses unfold. Many informatic projects utilize data stores not under the control of the analyst or involve secondary analyses of existing databases. Thus it may not be possible to avoid or control data entry errors, coding inaccuracies, measurement mistakes, subject misidentifications, etc. If direct access and oversight is available, then some level of quality assurance/quality control ("QA/QC") should be imposed on the data entry and curation process; see, for example, Fong (2001) or Pierchala and Surti (2009). Otherwise, potential data entry missteps or other errors in an existing database can sometimes be identified via statistical analysis, including single- or multi-dimensional graphical displays, data summarization techniques, or other forms of comparative statistical testing. (Many of these methods have more-general uses in statistical data analytics as well; these are described

in the following chapters.) Of course, the analyst must also be wary of going too far: over-correction of, say, missing data by imputing the missing values from the remainder of the database might just as quickly smooth away the very patterns the mining exercise is intended to detect.

Hand et al. (2000) distinguish between two general forms of data quality distortion: individual and collective. The first ("individual") occurs when the larger database is generally sound, but particular records in the database are affected by errors in collection, entry, or some other form of disruption. Classical examples include misplaced decimal points, transposed digits, measurement rounding errors, missing data records, and impossible combinations in classification fields (think: pregnant = "yes"/sex = "male"). These sorts of errors are often difficult to control, and the problem is common to almost any large collection of data: even the best data quality assurance program will on occasion let errors slip by. Data miners must be aware that sometimes, a feature or pattern uncovered in a large database could simply be the consequence of (a series of) individual-level data distortions. When examined in greater depth, these likely will be recognized as such and usually are afforded little value. Indeed, Hand et al. mention, only partly with tongue-in-cheek, that a large database found to be free of any errors may call into suspicion the quality of the database as a whole! More seriously though, they also note that certain patterns of distortion may in fact be of actual interest to the data miner; for example, large blocks of missing data can sometimes indicate a real, predictive classification feature in the population under study. Obviously, a kind of balancing act is required here: while outlying observations might be the purposeful target of an exercise in, say, credit-fraud detection, they more often hinder proper pattern detection in a typical data mining project (Hand et al. 2001, Section 2.7).

The second form of distortion ("collective") occurs when the larger collection suffers irregularities in the selection mechanisms under which the data were identified or sampled. Technically, data scientists define the *sampling frame* as the population of units from which a data set or database has been drawn and upon which measurements are taken/recorded. It is to this population that any inferences made from the data apply. For instance, suppose an analyst mines a database of patients suffering from a particular respiratory disease, such as asthma, in the warm and arid US Southwest. Any patterns of disease associated with, say, low-pressure weather systems gleaned from those records might not – indeed, likely will not – apply to asthma patients in more-humid, cooler north Britain/Scotland.

More generally, when data are inaccurately registered in a systematic manner, they contaminate the database and confuse the underlying sampling frame. A form of collective-scale data distortion ensues. To help to avoid collective sampling frame distortions, statistical practice encourages application of formal sampling strategies to the target population in order to construct the database. Preferred, at least at a basic level, is *simple random sampling*, where the units are sampled independently and with equally likely probabilities (see Section 3.1). By contrast, in selected instances, the database is large enough to contain the *entire* population of interest; for example, a grocery chain may collect shopping records of all its customers over a 6-month period. If so, the data now represent a full *census* of the population, and issues of sampling are less urgent. Complete enumerations of this sort are typically necessary if the informatic goal is one of fine-pattern detection across the target population.

More complex forms of probability-based sampling are also possible, although these exceed the scope here. For a deeper introduction to the theory and application of sampling methodology, see Thompson (2012) or Lohr (2010).

Of course, it is not always possible to control the sampling/selection process. In many cases, the data are recorded simply as the opportunity allows, at the convenience of the team building the database and with limited or no regard to sampling theory guidelines. This is called *convenience sampling* or *opportunity sampling*. Or, the selection process may by its very nature favor certain subjects; for instance, patients recruited for a study of genetic susceptibility to lung cancer may already be in the clinic for other disease-related reasons. (In the worst case, they all might be cigarette smokers under treatment for another, noncancerous disease such as emphysema, confounding study of the factors that lead to disease onset or progression. Upon reflection, it is perhaps obvious here that the subjects are being sampled preferentially; still, it is also surprising how often a selective process such as this goes unrecognized in practice.) The effect is known as *selection bias*, where the inclusion of a record in the database depends on what value(s) the variables take, or on some other, external, nonrandom feature (Wei and Cowan 2006). Selection bias can have a substantial confounding or contaminating effect on a large database.

Other forms of collection-level distortion include drift in the target population's attributes over time (e.g., oxygenation levels in an ecosystem's lakes may exhibit unrecognized changes due to increasing climate temperature) or overzealous data screening to expunge distortions that ends up excluding perfectly valid records. In the end, one can control for (some) data distortions via statistical adjustments in the data and/or in the analyses applied to them, but this is not always possible. At a minimum, the analyst must be aware of distorting influences on data quality in order to avoid falling victim to their ills. See Hand et al. (2000, Section 4) or Hand et al. (2001, Section 2.7) for more details and some instructive examples.

1.3 Statistical modeling versus statistical description

An important component in statistical analytics, and one that has exhibited the power of statistical science over the past century, is that of *statistical inference* (Casella and Berger 2002; Hogg and Tanis 2010). Statistical inference is the derivation of conclusions about a population from information in a random sample of that population. In many cases, formal statistical models are required to implement the inferential paradigm, using probability theory. By contrast, statistical *description* is the process of summarizing quantitative and qualitative features in a sample or population. The description process is often represented as simpler than the modeling/inferential process, but in fact, both require a level of skill and expertise beyond that of simple statistical arithmetic. A better distinction might be that inference is designed to make deductions about a feature of the population, while description is designed to bring features of a population to light.

Statistical description and statistical inference are typically applied in tandem, and the inferential process often contains descriptive aspects. They can also be employed separately, however, and it is not unusual in a data mining exercise to focus on only one of the two. For instance, an exploratory investigation of radio-transmitter data from tagged animals of a certain species may only involve simple description of their tracks and trajectories throughout a wildlife preserve. Alternatively, an inferential study on how two different species traverse the preserve might determine if a significant difference was evidenced in their trajectory patterns. In the former case, we call the effort one of *exploratory data analysis*, or "EDA," a statistical archetype popularized by Tukey (1977); more recently, see Gelman (2004) or Buja

et al. (2009). The EDA approach shares similarities with many descriptive statistical methods employed in data analytics, and the two paradigms often overlap (Myatt 2007). As a result, the focus in this text will be on exploratory aspects of the data mining and knowledge discovery process, driven by statistical calculation. To provide a broader panorama, however, associated methods of statistical inference will also be considered. Chapter 2 begins with an introduction to basic probability models useful in statistical inference. Chapters 3 and 4 follow with an introduction to methods of statistical description, data manipulation, and data visualization. On the basis of these methods of probability and data description, Chapter 5 then formally introduces the inferential paradigm. Readers familiar with introductory concepts in the earlier chapters may wish to skip forward to Chapter 6 on regression techniques for supervised learning or on to further chapters where specific foundational statistical methods for data analytics and selected informatic applications are presented.

Exercises

1.1 Use an online search engine or any other means of textual search to give a list of at least three more specialized areas of "informatics", beyond those mentioned (bioinformatics, ecoinformatics, etc.) in Section 1.1.

1.2 Give an application (from your own field of study, as appropriate) where data mining is used, and indicate instances of knowledge discovery generated from it.

1.3 Describe the nature of the database(s) from Exercise 1.2 on which the data mining was performed. What quantities were measured? What was the target population? What was/were the sampling frame/s?

1.4 Give an application (from your own field of study, as appropriate) where data distortion can occur for

(a) individual-level distortion.

(b) collective-level distortion.

1.5 As mentioned in Section 1.2, a grocery chain constructed a large database from shopping records of all its customers between January 1 and June 30 in a given year. The data only recorded each customer's purchase(s) of (i) any cheese products at least once every month, (ii) any meat products at least once every month, and (iii) any seafood products at least once every month. It was not recorded whether the customers considered themselves vegetarians, however. Is this a form of individual-level data distortion or collective-level distortion? Justify your answer.

1.6 (Hand et al., 2000) A large database was constructed on male adult diastolic blood pressures (in mmHg). When graphed, the data showed that measurements ending in odd numbers were much more common at higher blood pressure readings. Upon deeper investigation, it was found that the pressures were taken with a digital instrument that could only display even values. When a male subject's reading was exceptionally high, however, the technician repeated the measurement and recorded the average of the two readings. Thus although both original readings had to be even, the averaged reading could be odd. Is this a form of individual-level data distortion or collective-level distortion? Justify your answer.

1.7 To gauge students' opinions on proposed increases in statewide taxes, a polling firm sent operatives to every public college or university in their state. At each campus, the operatives stood outside the Student Union or main cafeteria just before lunch. For 30 minutes, they asked any student entering the building if he or she supported or opposed the tax increases. They also recorded the student's age, sex, and class standing (freshman, sophomore, etc.). Describe in what way(s) this can be viewed as a form of convenience sampling. Can you imagine aspects that could be changed to make it more representative and less opportunistic?

1.8 A financial firm builds a large database of its customers to study their credit card usage. In a given month, customers who had submitted at least their minimum monthly payment but less than the total amount due on that month's statement were included. By how much, if at all, the monthly payment exceeded the minimum payment level was recorded. These values were then mined for patterns using the customers' ages, lengths of patronage, etc. Is there any selection bias evident in this approach? Why or why not?

1.9 As mentioned in Section 1.2, a physician collected a large database of records on asthma patients in the US Southwest. He determined whether temporal patterns occurred in the patients' asthma onset when low-pressure weather fronts passed through the region. Is this a question of statistical description or statistical inference? Justify your answer.

1.10 Return to the asthma study in Exercise 1.9. The physician there also mined the database for associative patterns of patient proximity to construction sites where large amounts of airborne particulates were generated. Is this a question of statistical description or statistical inference? Again, justify your answer.

1.11 A geographer constructs a database on the county-by-county occurrence of natural disasters in the US Southeast over a 40-year period, along with corresponding county-level information on concurrent property damage (in $). She then uses the database to determine statistically if a difference exists in property losses due to a particular form of disaster (floods) among counties in two adjoining US states. Is this a question of statistical description or statistical inference? Why or why not?

2

Basic probability and statistical distributions

The elements of probability theory serve as a cornerstone to most, if not all, statistical operations, be they descriptive or inferential. In this chapter, a brief introduction is given to these elements, with focus on the concepts that underlie the foundations of statistical informatics. Readers familiar with basic probability theory may wish to skip forward to Section 2.3 on special statistical distributions or farther on to Chapter 3 and its introduction to basic principles of data manipulation.

2.1 Concepts in probability

Data are generated when a random process produces a quantifiable or categorical outcome. We collect all possible outcomes from a particular random process together into a set, $\mathcal{S}$, called the *sample space* or *support space*. Any subcollection of possible outcomes, including a single outcome, is called an *event*, $\mathcal{E}$. Notice that an event is technically a subset of the sample space $\mathcal{S}$. Standard set notation for this is $\mathcal{E} \subset \mathcal{S}$.

Probabilities of observing events are defined in terms of their long-term frequencies of occurrence, that is, how frequent the events (or combinations of events) occur relative to all other elements of the sample space. Thus if we generate a random outcome in a repeated manner and count the number of occurrences of an event $\mathcal{E}$, then the ratio of this count to the total number of times the outcome could occur is the probability of the event of interest. This is the *relative frequency interpretation* of probability. The shorthand for P[Observe event $\mathcal{E}$] is $P[\mathcal{E}]$ for any $\mathcal{E} \subset \mathcal{S}$. To illustrate, consider the following simple, if well recognized, example.

Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery,
First Edition. Walter W. Piegorsch.

www.wiley.com/go/piegorsch/data_analytics

Example 2.1.1 Six-sided die roll. Roll a fair, six-sided die and observe the number of 'pips' seen on that roll. The sample space is the set of all possible outcomes from one roll of that die: $\mathcal{S} = \{1, 2, \ldots, 6\}$. Any individual event is a single number, say, $\mathcal{E} = \{6\} = \{\text{a roll showing 6 pips}\}$. Clearly, the single event $\mathcal{E} = \{6\}$ is contained within the larger sample space $\mathcal{S}$.

If the die is fair, then each individual event is equally likely. As there are six possible events in $\mathcal{S}$, to find $P[\mathcal{E}]$, divide 1 (for the single occurrence of $\mathcal{E}$) by 6 (for the six possible outcomes): $P[\mathcal{E}] = \frac{1}{6}$, that is, in one out of every six tosses, we expect to observe a $\{6\}$. □

2.1.1 Probability rules

A variety of fundamental axioms are applied in the interpretation of a probability $P[\mathcal{E}]$. The most well known are

(1a) $0 \leq P[\mathcal{E}] \leq 1$, and

(1b) $P[\mathcal{S}] = 1$.

In addition, a number of basic rules apply for combinations of two events, $\mathcal{E}_1$ and $\mathcal{E}_2$. These are

(2a) *Addition Rule*. $P[\mathcal{E}_1 \text{ or } \mathcal{E}_2] = P[\mathcal{E}_1] + P[\mathcal{E}_2] - P[\mathcal{E}_1 \text{ and } \mathcal{E}_2]$.

(2b) *Conditionality Rule*. $P[\mathcal{E}_1 \text{ given } \mathcal{E}_2] = P[\mathcal{E}_1 \text{ and } \mathcal{E}_2]/P[\mathcal{E}_2]$ for any event $\mathcal{E}_2$ such that $P[\mathcal{E}_2] > 0$. For notation, conditional probabilities are written with the symbol '|', for example, $P[\mathcal{E}_1|\mathcal{E}_2] = P[\mathcal{E}_1 \text{ given } \mathcal{E}_2]$.

(2c) *Multiplication Rule*. $P[\mathcal{E}_1 \text{ and } \mathcal{E}_2] = P[\mathcal{E}_1|\mathcal{E}_2]\, P[\mathcal{E}_2]$.

Special cases of these rules occur when the events in question relate in a certain way. For example, two events $\mathcal{E}_1$ and $\mathcal{E}_2$ that can never occur simultaneously are called *disjoint* (or equivalently, *mutually exclusive*). In this case, $P[\mathcal{E}_1 \text{ and } \mathcal{E}_2] = 0$. Notice that if two events are disjoint, the Addition Rule in (2a) simplifies to $P[\mathcal{E}_1 \text{ or } \mathcal{E}_2] = P[\mathcal{E}_1] + P[\mathcal{E}_2]$. Two disjoint events, $\mathcal{E}_1$ and $\mathcal{E}_2$, are *complementary* if the joint event $\{\mathcal{E}_1 \text{ or } \mathcal{E}_2\}$ makes up the entire sample space $\mathcal{S}$. Notice that this implies $P[\mathcal{E}_1 \text{ or } \mathcal{E}_2] = 1$. If two events, $\mathcal{E}_1$ and $\mathcal{E}_2$, are complementary so are their probabilities. This is known as the *Complement Rule*:

(2d) **Complement Rule**: If, for two disjoint events $\mathcal{E}_1$ and $\mathcal{E}_2$, the joint event $\{\mathcal{E}_1 \text{ and } \mathcal{E}_2\}$ equals the entire sample space $\mathcal{S}$, then $P[\mathcal{E}_1] = 1 - P[\mathcal{E}_2]$ and $P[\mathcal{E}_2] = 1 - P[\mathcal{E}_1]$.

Example 2.1.2 Six-sided die roll (Example 2.1.1, continued). Return to the roll of a fair, six-sided die. As seen in Example 2.1.1, the sample space is $\mathcal{S} = \{1, 2, \ldots, 6\}$. As the die is only rolled once, no two singleton events can occur together, so, for example, observing a $\{6\}$ and observing a $\{4\}$ are disjoint events. Thus from the Addition Rule (2a) with disjoint events, $P[4 \text{ or } 6] = P[4] + P[6] = \frac{1}{6} + \frac{1}{6} = \frac{1}{3}$.

More involved constructions are also possible. For instance, from the Complement Rule (2d), $P[\text{not observing a } 6] = P[1 \text{ or } 2 \text{ or } 3 \text{ or } 4 \text{ or } 5] = 1 - P[6] = 1 - \frac{1}{6} = \frac{5}{6}$. □

The case where disjoint events completely enumerate the sample space $\mathcal{S}$ has a special name: it is called a *partition*. One need not be restricted to only two events, however. If a set of $h \geq 2$ events, $\{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_h\}$, exists such that (i) all the h events are disjoint from

each other – technically, if every pair of events $\mathcal{E}_i$ and $\mathcal{E}_j$, $i \neq j$, is disjoint – and (ii) the collective event $\{\mathcal{E}_1$ and $\mathcal{E}_2$ and $\cdots$ and $\mathcal{E}_h\}$ equals $\mathcal{S}$, we say the set forms a partition of $\mathcal{S}$. (Mathematically, the partition can even consist of a countably infinite set of pairwise-disjoint events $\{\mathcal{E}_1, \mathcal{E}_2, \ldots\}$ if they satisfy these conditions.) This leads to another important rule from probability theory:

(2e) *The Law of Total Probability*. For any event $\mathcal{B} \subset \mathcal{S}$ and any partition, $\{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_h\}$, of $\mathcal{S}$,

$$P[\mathcal{B}] = \sum_{i=1}^{h} P[\mathcal{B}|\mathcal{E}_i]P[\mathcal{E}_i].$$

The Law of Total Probability in (2e) is an important building block in another famous result from probability theory, known as *Bayes' rule*. It describes how the probability of an event $P[\mathcal{E}]$ can be 'updated' using external information. The result is credited to eighteenth century Presbyterian minister Sir Thomas Bayes (Bayes 1763), although, see Stigler (1983) regarding the particulars behind that assignment. In its simplest form, Bayes' rule also shows how conditional probabilities can be reversed: by recognizing that $P[\mathcal{B}$ and $\mathcal{E}] = P[\mathcal{E}$ and $\mathcal{B}]$ and manipulating the Multiplication Rule (2c) with this fact in mind, one can show (Exercise 2.5) that

$$P[\mathcal{E}|\mathcal{B}] = P[\mathcal{B}|\mathcal{E}]\frac{P[\mathcal{E}]}{P[\mathcal{B}]}. \tag{2.1}$$

More generally, the result can be applied to full partitions of the sample space:

(2f) *Bayes' Rule*. For any event $\mathcal{B} \subset \mathcal{S}$ and any partition, $\{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_h\}$, of $\mathcal{S}$,

$$P[\mathcal{E}_i|\mathcal{B}] = \frac{P[\mathcal{B}|\mathcal{E}_i]P[\mathcal{B}]}{\sum_{i=1}^{h} P[\mathcal{B}|\mathcal{E}_i]}$$

for every $i = 1, \ldots, h$.

A different relationship occurs between two events if they do not impact each other in any way. Suppose the knowledge that one event $\mathcal{E}_1$ occurs has absolutely no impact on the probability that a second event $\mathcal{E}_2$ occurs and that the reverse is also true. Two such events are called *independent*. In effect, independent events modify the Conditionality Rule (2b) into $P[\mathcal{E}_1|\mathcal{E}_2] = P[\mathcal{E}_1]$ and $P[\mathcal{E}_2|\mathcal{E}_1] = P[\mathcal{E}_2]$. More importantly, for two independent events, the Multiplication Rule (2c) simplifies to $P[\mathcal{E}_1$ and $\mathcal{E}_2] = P[\mathcal{E}_1]P[\mathcal{E}_2]$.

2.1.2 Random variables and probability functions

Suppose a random outcome can be quantified formally, either because (i) it is an actual measurement or count or (ii) it is a qualitative outcome that has been unambiguously coded into a numeric value. Such a quantified random outcome is called a *random variable*. Standard notation for random variables is uppercase Roman letters, such as X or Y. To distinguish between a conceptual random variable and one that has already been realized in practice, the realized value is denoted by a lowercase Roman character: x or y. The basic probability rules for events as discussed in Section 2.1.1 can then be expanded to describe random variables.

At the core of the operation is the notion of a *probability function*. Probability functions are unifying mathematical descriptions of how a random outcome varies. They are used to characterize two basic types of random variables: discrete and continuous. A *discrete random variable* takes on only discrete values; examples include simple binary variates (say 'damaged' = 1 vs. 'operating' = 0 in a component reliability study), counts of occurrences (numbers of customers who purchase a sale item or numbers of adverse-event reports with a new drug), or even studies that result in an infinite, yet countable, number of outcomes (i.e., counts without a clear upper bound, such as the number of different insect species in a tropical forest). A *continuous random variable* takes on values over a continuum (mass or length, stock market averages, blood levels of a chemical, etc.). Discrete random variables often arise from counting or classification processes, while continuous random variables often arise from some sort of measurement process. In either case, the probability functions will depend on the nature of the random outcome.

Suppose the random variable X is discrete and consider the 'event' that X takes on some specific value, say $X = m$. Then, the values of $P[X = m]$ over all possible values of m describe the *probability distribution* of X. Standard notation here is $f_X(m) = P[X = m]$; this is called the *probability mass function* (or p.m.f.) of X. As $f_X(m)$ is a probability, it must satisfy the various axioms and rules from Section 2.1.1. Thus, for example, $0 \leq f_X(m) \leq 1$ for all arguments m and $\sum_{m \in \mathcal{S}} f_X(m) = 1$, where the sum is taken over all possible values of m in the sample space $\mathcal{S}$. (The symbol '$\in$' is read 'is an element of.')

Summing the discrete p.m.f. over increasing values up to m produces what is called the *cumulative distribution function* (or c.d.f.) of X:

$$F_X(m) = P[X \leq m] = \sum_{i \leq m} f_X(i).$$

As the c.d.f. is itself a probability, it must also satisfy the probability rules from Section 2.1.1, in particular, $0 \leq F_X(m) \leq 1$ for any m. Or, from the Complement Rule (2d), $P[X > m] = 1 - P[X \leq m] = 1 - F_X(m)$.

As it gives cumulative probabilities, the c.d.f. must be a nondecreasing function. In fact, for discrete random variables, the c.d.f. will typically have the appearance of a nondecreasing step function.

Example 2.1.3 **Six-sided die roll (Example 2.1.1, continued).** Roll a fair, six-sided die and now formally define the random variable X as the number of 'pips' seen on that roll. As seen in Example 2.1.1, the sample space is $\mathcal{S} = \{1, 2, \ldots, 6\}$ and because the die is fair, the p.m.f. is $P[X = m] = f_X(m) = 1/6$ for any $m \in \mathcal{S}$.

The c.d.f. $F_X(m) = P[X \leq m]$ is simply the cumulative sum of these uniform probabilities up to and including the argument m. So, for example,

$$F_X(4) = \sum_{m=1}^{4} f_X(m) = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{2}{3}.$$

Notice, however, that if the argument to the c.d.f. is not an element of the sample space $\mathcal{S}$, then the cumulative probability calculation will stop at the previous element in $\mathcal{S}$. So, for example, $F_X(4.2) = P[X \leq 4.2]$ is equal to $P[X \leq 4] = 2/3$, because the event $\{X \leq 4.2\}$ is identical to the event $\{X \leq 4\}$ for this discrete random variable. This effect gives the c.d.f. here a 'step function' appearance, as in Figure 2.1. Solid dots in the figure indicate the 'jumps' in probability mass at each value of m in $\mathcal{S}$. □

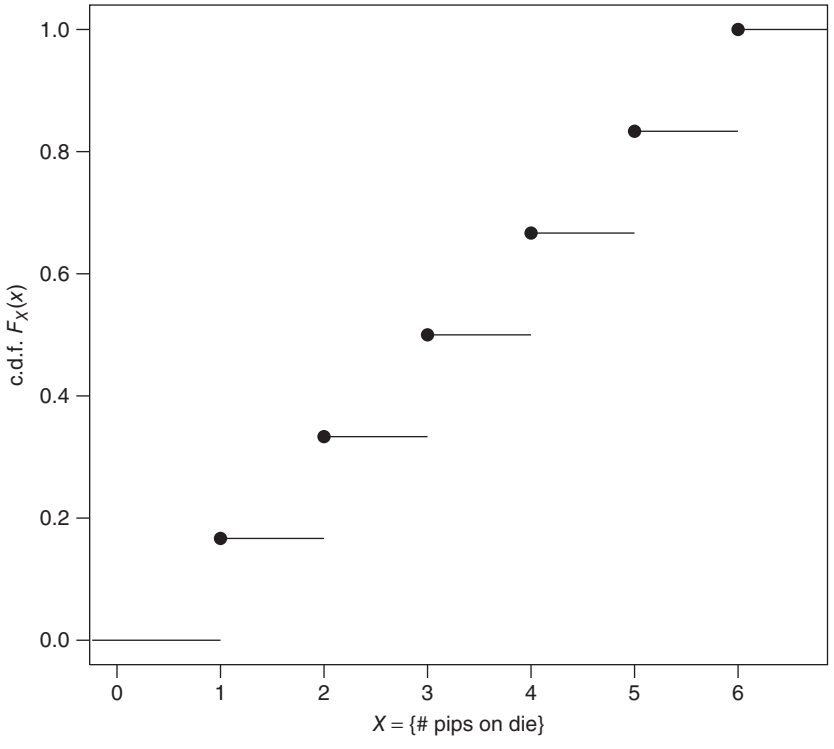

Figure 2.1 Cumulative distribution function (c.d.f.) for a discrete random variable: X ={number of pips on roll of six-sided die} in Example 2.1.3.

Conversely, for continuous random variables, the probability function, $f_X(x)$, describes the random variable's continuous density of probability. Thus the terminology changes, and $f_X(x)$ is now called the *probability density function* (or p.d.f.) of the continuous random variable X. It is used in expressing probabilities over interval subsets of the real numbers via definite integrals, for example,

$$P[a \leq X \leq b] = \int_a^b f_X(x)dx.$$

(Readers unfamiliar with concepts of integration and derivatives should refer to introductory texts in calculus (Hughes-Hallett et al. 2013); readers requiring only a refresher may find targeted texts such as Khuri (2003) helpful.) Similarly, the c.d.f. of a continuous random variable is the area under the p.d.f. integrated from $-\infty$ to the argument, x, of the function:

$$F_X(x) = P[X \leq x] = \int_{-\infty}^{x} f_X(u)\, du. \tag{2.2}$$

The definition in (2.2) relates cumulative probabilities to areas under p.d.f. curves. This produces some interesting consequences: notice that $P[a \leq X \leq b] = F_X(b) - F_X(a)$ and so $P[X = a] = P[a \leq X \leq a] = F_X(a) - F_X(a) = 0$ for any a. That is, for a continuous random variable, nonzero probability can only be assigned to events that correspond to intervals of

values. As a result, if X is continuous, $P[X \leq a] = P[X < a] + P[X = a] = P[X < a] + 0 = P[X < a]$. From (2.2), one also finds that a continuous c.d.f. must possess strict limiting values: $\lim_{x \to -\infty} F_X(x) = 0$ and $\lim_{x \to \infty} F_X(x) = 1$. For discrete random variables, however, nonzero probability can be assigned to events that correspond to particular outcomes, so that statements such as $P[Y = a]$ could evaluate as numbers between 0 and 1, depending on the underlying p.m.f.

Similar to the case with discrete random variables and their p.m.f.s, the p.d.f. from a continuous random variable satisfies two basic axioms: (i) $f_X(x) \geq 0$ for all arguments x and (ii) $\int_{-\infty}^{\infty} f_X(x)\, dx = 1$. Notice also that if $F_X(x)$ is a differentiable function, then from the fundamental theorem of calculus (Khuri 2003, Section 6.4), its derivative at the point x is the p.d.f.: $dF_X(x)/dx = f_X(x)$.

A probability function (p.m.f. or p.d.f.) that when graphed rises to a single peak and then falls back is called *unimodal*. The 'mode' of the function is the value of x at which the single peak is attained. A probability function with more than one mode is called *multimodal*.

When a probability function is arranged such that its heights are equal both to the left and right of some central point, say $x = b$, it has a special structure: a p.m.f. or p.d.f. is *symmetric* about a point b if $f_X(b + \epsilon) = f_X(b - \epsilon)$ for all $\epsilon > 0$. By contrast, unimodal probability functions (and their underlying random variables) that deviate from symmetry are called *skewed*. A probability function that tails off faster to the right than to the left is 'skewed right;' one that tails off faster to the left is 'skewed left.' Figure 2.2 plots a typical, right-skewed, unimodal p.d.f. for a positive random variable X, along with a typical, symmetric, unimodal p.d.f. for a continuous random variable Y.

Right skew is not uncommon with many positive random variables such as blood concentrations or income levels: the hard lower bound at $x = 0$ often causes the probability mass or density to crowd together as x approaches 0, and/or the open upper range allows for extremely large values of x, with correspondingly low probabilities of concurrence.

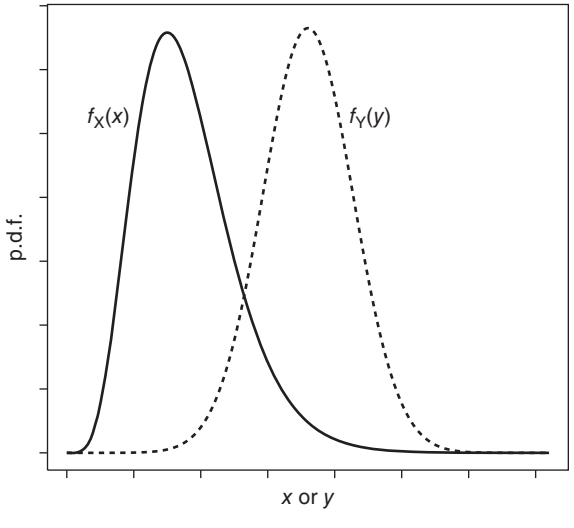

Figure 2.2 Unimodal probability density functions (p.d.f.s) for a continuous, right-skewed random variable X (solid curve, ——) and a continuous, symmetric random variable Y (dashed curve, - - -). Functions are offset such that the central value of Y is twice that of X.

In certain instances, it may be desirable to modify a skewed probability function in order to make it appear more symmetric. To do so, one can manipulate the random variable X via a functional transformation, say $Y = g(X)$ for some given function, $g(\cdot)$. For example, the (natural) logarithmic transform $g(X) = \log(X)$ is often employed to reduce heavy right skew in a positive-valued random variable.

Bivariate and multivariate extensions of probability functions are also possible. For instance, with two discrete random variables, X and Y, the *joint bivariate p.m.f.* is $f_{X,Y}(k, m) = P[X = k \text{ and } Y = m]$ and the *joint bivariate c.d.f.* is $F_{X,Y}(k, m) = P[X \leq k \text{ and } Y \leq m]$. Individually, X is itself a random variable; its *marginal p.m.f.* is derived from the joint p.m.f. by summing over all possible values of $Y = m$:

$$f_X(k) = \sum_{m \in S} f_{X,Y}(k, m).$$

(For the more general multivariate case, see Section 2.2.)

Using conditional probabilities, one can also introduce the concept of a *conditional p.m.f.*, that is, the probability that X takes the value k *given* that $Y = m$. Denote this as $P[X = k|Y = m] = f_{X|Y}(k|m)$. Mimicking the construction from the Conditionality Rule in (2b), the conditional p.m.f. of X given Y is formally

$$P[X = k \mid Y = m] = f_{X|Y}(k|m) = \frac{f_{X,Y}(k, m)}{f_Y(m)}. \tag{2.3}$$

For two continuous random variables, analogous results apply. The joint p.d.f. is $f_{X,Y}(x, y)$ and the joint c.d.f. is $F_{X,Y}(x, y) = P[X \leq x \text{ and } Y \leq y]$. The marginal p.d.f. of X is found by integrating Y out of the joint p.d.f.:

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x, y)\, dy.$$

(Reverse the process for the marginal p.d.f. of Y.) Given the marginal p.d.f.s, the conditional p.d.f.s follow naturally, for example,

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x, y)}{f_Y(y)}.$$

With this, it is possible to describe a version of Bayes' rule from (2.1) for p.d.f.s:

$$f_{X|Y}(x|y) = \frac{f_{Y|X}(y|x) f_X(x)}{f_Y(y)}. \tag{2.4}$$

A similar version of Bayes' rule is available for p.m.f.s by manipulating the relationships in (2.3).

One can also extend the concept of independent events to random variables. Suppose two random variables, X_1 and X_2, exist such that the value of X_1 has absolutely no impact on X_2, and vice versa. If so, the two variables are *independent*. Extending the Multiplication Rule in (2c) for independent events, the joint probability function for two independent random variables factors into the marginal components: $f_{X,Y}(x, y) = f_X(x) f_Y(y)$.

2.1.3 Means, variances, and expected values

The various probability functions for discrete and continuous random variables introduced in Section 2.1.2 provide general characterizations of a variable's probability structure. In many cases, however, it is useful to construct summary measures of the random variable that encapsulate its various features. These can be derived from the underlying p.m.f. or p.d.f. The general form of such a summary measure is called an *expected value*, and it is based on a mathematical construct known as an *expectation operator*, $E[\cdot]$. In its most general usage, the expected value of a function of a random variable, $g(X)$, is defined as

$$E[g(X)] = \sum_{m \in S} g(m) f_X(m) \tag{2.5}$$

for a p.m.f. $f_X(m)$ and

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x) dx \tag{2.6}$$

for a p.d.f. $f_X(x)$. That is, expectation involves summation for discrete random variables and integration for continuous random variables. In effect, the expected value of a function $g(X)$ is a weighted average of $g(X)$, with weights taken as the probability mass or density of the random variable X.

Some special features of the expectation operator include

- if $g(X) \geq 0$ for all X, then $E[g(X)] \geq 0$,
- if $g(X) \leq h(X)$ for two functions $g(X)$ and $h(X)$, then $E[g(X)] \leq E[h(X)]$,
- if $a \leq g(X) \leq b$ for two constants $a \leq b$, then $a \leq E[g(X)] \leq b$.

See Casella and Berger (2002, Section 2.2).

The simplest example of an expected value occurs for $g(X) = X$, that is, the expected value of X. This is known as the *population mean* of X and is usually given the special notation $\mu = E[X]$. (This is the Greek lowercase letter μ; Greek letters commonly are employed to denote special statistical parameters.) From (2.5), for a discrete random variable, this is just $\mu = \sum_{m \in S} m f_X(m)$, while from (2.6), for a continuous random variable, it is $\mu = \int_{-\infty}^{\infty} x f_X(x)\, dx$.

The mean quantifies the *central tendency* for X and serves as a single, common, summary descriptor for the value we expect X to take on. (Of course, X is a random variable, so it can realize any value in the sample space S. Indeed, although the mean μ lies between the minimum and maximum of S, it does not necessarily have to equal any specific value in S.) When necessary, a subscript can be used to further identify a random variable with its mean: $E[X] = \mu_X$.

Example 2.1.4 Six-sided die roll (Example 2.1.3, continued). Roll a fair, six-sided die and let X be the observed number of 'pips' as in Example 2.1.3. The p.m.f. was seen to be $f_X(m) = \frac{1}{6}$ for any $m \in \{1, 2, \ldots, 6\}$. To find the population mean, appeal to (2.5):

$$\mu_X = \sum_{m=1}^{6} m\, f_X(m) = \frac{1}{6} \sum_{m=1}^{6} m.$$

Using the well-known relationship $\sum_{m=1}^{n} m = \frac{1}{2}n(n+1)$, we find $\mu_X = \left(\frac{1}{6}\right)\left(\frac{1}{2}\right)(6)(7) = 3.5$. As per its design, μ_X gives a measure of the 'central' value of X, although here $\mu_X = 3.5$ is not actually an element of $\mathcal{S}$. It does, however, rest (precisely) in between the smallest and largest elements of $\mathcal{S}$. □

In Exercise 2.7, it is shown that $E[a] = a$ for any constant a and that constants can be brought out of expected value operations, that is, $E[aX] = aE[X]$ for any random variable X. Also, for the special case of a linear transformation of X, say $g(X) = a + bX$ for any two constants a and b, the expectation operator is linear: $E[a + bX] = a + bE[X] = a + b\mu_X$.

While the mean quantifies central tendency of a random variable X, another special form of expectation quantifies the inherent variation of X. Given $E[X] = \mu_X$, let $g(X) = (X - \mu_X)^2$ be the squared deviation from the mean. Then, the *population variance* of X is the expected value of this squared deviation: $\text{Var}[X] = E[(X - \mu_X)^2]$. Notice that because this is an expected value of a nonnegative function, it must also be nonnegative. Standard notation for $\text{Var}[X]$ employs another Greek letter σ. To emphasize the nonnegative aspect the parameter carries a square: $\sigma_X^2 = \text{Var}[X] = E[(X - \mu_X)^2]$. In practice, it is often more useful to operate with the equivalent expression $\sigma_X^2 = E[X^2] - \mu_X^2$ (see Exercise 2.8).

The expectation $E[X^N]$ is called the *Nth moment of X*, so that the variance can be described as the difference between the second moment and the first squared moment of a random variable. In Exercise 2.9, it is shown that $\text{Var}[a] = 0$ for any constant a and that when brought out of variance operations, a constant is squared: $\text{Var}[aX] = a^2\text{Var}[X]$ for any random variable X whose variance exists.

For measuring variation on the original scale of X, the *population standard deviation* is defined as the positive square root of the variance: $\sigma_X = \sqrt{\sigma_X^2}$.

An alternative summary descriptor for a distribution's spread is known as the *entropy*. If X has p.m.f. or p.d.f. $f_X(x)$, then the entropy for f_X is given by

$$H(f_X) = -E[\log\{f_X(X)\}], \tag{2.7}$$

where the expected value is taken with respect to the probability function $f_X(x)$. Thus, for example, if X is continuous, $H(f_X) = -\int_{-\infty}^{\infty} f_X(x)\log\{f_X(x)\}dx$. The continuous version is often called *differential entropy*. The discrete case is similar, although many authors will then use $\log_2$ instead of the natural logarithm in the expected value and drop use of the 'differential' adjective. (In this case, the entropy is said to be measured in 'bits.' If using the natural log, it is measured in 'nats.')

The term 'entropy' was coined by Clausius (1865) to describe the amount of disorder – a better term might be 'dispersion' – in a thermodynamic system. Shannon (1948) later developed and popularized the information-theoretic features associated with (2.7) for describing loss of data ('disorder') in information transmission.

Entropy can be viewed as the extent to which the probability mass or density of X is localized at a few separated values or dispersed over a wider range. As defined in (2.7), it increases as $f_X(x)$ increases in variability. Thus is it often taken as a measure of dispersion or heterogeneity.

2.1.4 Median, quartiles, and quantiles

Another useful summarization for a random variable X relates the values it achieves to the c.d.f., $F_X(x)$, at those values. Consider, for example, the c.d.f. at its middle value, 50%.

One might ask what value of x in the sample space $\mathcal{S}$ satisfies $F_X(x) = 1/2$? The point that does so is called the *median* of X. More formally, the median is the quantity $Q_2 \in \mathcal{S}$ where $P[X \leq Q_2] \geq \frac{1}{2}$ and $P[X \geq Q_2] \geq \frac{1}{2}$. That is, Q_2 is the point below which *and* above which at least 50% of the probability mass or probability density rests. (The notation 'Q_2' will be explained below.)

For most continuous distributions, the median is unique. For many discrete distributions, however, its defining equations may not be unambiguously satisfied. For example, for a discrete random variable, X may have two adjacent values $m_1 < m_2$ such that $P[X \leq m_1] = \frac{1}{2}$ and $P[X \geq m_2] = \frac{1}{2}$. In this case, any value of X between m_1 and m_2 could be called the median of X. If this occurs in practice, Q_2 is set equal to the midpoint of the interval, $Q_2 = \frac{1}{2}(m_1 + m_2)$.

Example 2.1.5 Six-sided die roll (Example 2.1.4, continued). Roll a fair, six-sided die, and let X be the number of 'pips' seen on that roll. As seen in Example 2.1.4, the sample space is $\mathcal{S} = \{1, 2, \ldots, 6\}$ and the p.m.f. is $f_X(m) = \frac{1}{6}$ for any $m \in \mathcal{S}$.

To find the population median, recognize that $P[X \leq 3] = P[X = 1 \text{ or } X = 2 \text{ or } X = 3] = P[X = 1] + P[X = 2] + P[X = 3] = 0.5$, because the events are disjoint. Similarly, $P[X \geq 4] = P[X = 4 \text{ or } X = 5 \text{ or } X = 6] = 0.5$. Thus any value of Q_2 between (but not including) $m = 3$ and $m = 4$ would satisfy the definition of a median for X. Simplest here is to let Q_2 be the midpoint: $Q_2 = \frac{1}{2}(3 + 4) = 3.5$. Notice that this is not an element of $\mathcal{S}$. As with the population mean, the population median need not be an element of the sample space. □

As it characterizes the 'center' of a distribution, Q_2 is used as an alternative to the mean $E[X]$ to measure X's central tendency. In fact, the two values can be equal – as in Examples 2.1.4 and 2.1.5 – although this is not guaranteed. When a random variable exhibits a large skew, the median will be less influenced than the population mean by the extreme values in the skewed tail of the distribution. Thus it can be particularly useful for measuring central tendency with skewed distributions.

One can extend the concept of a median – that is, the 50% point of a distribution – to any desired probability point along the range of $F_X(x)$. Two obvious values are the 25% and 75% points. These are known as the first (or lower) and third (or upper) *quartiles* of the distribution and are denoted as Q_1 and Q_3, respectively. (The second or middle quartile is just the median, Q_2, which explains its notation.) Formally, the first (lower) quartile is defined as the point $Q_1 \in \mathcal{S}$ such that $P[X \leq Q_1] \geq 0.25$ and $P[X \geq Q_1] \geq 0.75$. Similarly, the third (upper) quartile is defined as the point Q_3 such that $P[X \leq Q_3] \geq 0.75$ and $P[X \geq Q_3] \geq 0.25$.

From Q_1 and Q_3, it is possible to derive another measure of variability in the population, known as the *interquartile range*: $\text{IQR} = Q_3 - Q_1$. This is different in structure and interpretation from the variance, $\text{Var}[X]$: the variance measures average squared deviation from the center (i.e., the mean) of a distribution, while the IQR gives the length of that portion of the sample space in which the middle half of the probability mass or density lies. While fundamentally different, the two measures do share the feature that as they grow larger, the p.m.f. or p.d.f. is more dispersed.

Quartiles act to separate the distribution of a random variable into equal-probability fourths (hence, their name). This concept can be applied to any desired separation, so that, for example, *quintiles* separate $\mathcal{S}$ into fifths, *deciles* separate $\mathcal{S}$ into tenths, and *percentiles* into hundredths. Fully generalized to any desired probability point, the pth *quantile* of a distribution is the point q_p that satisfies $P[X \leq q_p] \geq p$ and $P[X \geq q_p] \geq 1 - p$, for $0 < p < 1$. If the c.d.f. of X, $F_X(x)$, is continuous and strictly increasing such that it has an inverse function $F_X^{-1}(\cdot)$, the quantiles can be defined by inverting $F_X(x)$: $q_p = F_X^{-1}(p)$.

2.1.5 Bivariate expected values, covariance, and correlation

The expectation operator also can be applied to *pairs* of random variables. For instance, suppose two random variables X and Y possess a joint p.m.f. $f_{X,Y}(k,m)$. Then for any bivariate function $g(X,Y)$, the expected value of $g(X,Y)$ is

$$E[g(X,Y)] = \sum_{(k,m)\in S}\sum g(k,m) f_{X,Y}(k,m).$$

If instead X and Y possess a joint p.d.f. $f_{X,Y}(x,y)$, the bivariate expected value is

$$E[g(X,Y)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y) f_{X,Y}(x,y)\, dx\, dy.$$

Marginal and conditional expectations can also be defined, based on their respective p.m.f.s or p.d.f.s. For instance, in the continuous case, the marginal mean of Y can be recovered from the joint probability function of X and Y: simply take $g(X,Y)$ as the univariate function $g(X,Y) = Y$ and evaluate the joint expectation. For instance, in the joint continuous case,

$$\begin{aligned} E[Y] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} y f_{X,Y}(x,y)\, dx\, dy \\ &= \int_{-\infty}^{\infty} y \int_{-\infty}^{\infty} f_{X,Y}(x,y)\, dx\, dy = \int_{-\infty}^{\infty} y f_Y(y)\, dy, \end{aligned} \tag{2.8}$$

because $f_Y(y) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)\, dx$. Equation (2.8) ends with the desired expression for μ_Y. (The joint discrete case is similar: simply replace the integrals with corresponding sums.)

One can also develop conditional expected values. For instance, the mean of Y conditional on $X = x$ is

$$E[Y|x] = \int_{-\infty}^{\infty} y\, f_{Y|X}(y|x)\, dy. \tag{2.9}$$

(Notice that, by construction, when the calculation in (2.9) is finished, $E[Y|x]$ will be a function of x.) Similarly, the conditional variance of Y, given $X = x$, can be found as $\mathrm{Var}[Y|x] = E[Y^2|x] - E^2[Y|x]$, where $E^2[Y|x]$ is the square of the conditional mean. An analogous set of expressions is available for $X|y$.

An expected value that quantifies the joint variability between two random variables is known as the *covariance*. An extension of the univariate variance, the covariance between X and Y, is defined as $\sigma_{XY} = \mathrm{Cov}[X,Y] = E[(X-\mu_X)(Y-\mu_Y)]$, where $\mu_X = E[X]$ and $\mu_Y = E[Y]$. Notice that $\mathrm{Cov}[X,Y] = \mathrm{Cov}[Y,X]$ and in particular $\mathrm{Cov}[X,X] = \mathrm{Var}[X]$. This motivates the use of the notation σ_{XY}. Exercise 2.11 shows that if X and Y are independent, $\mathrm{Cov}[X,Y] = 0$. When $\mathrm{Cov}[X,Y] > 0$, increasing values of X tend to associate with increasing values of Y, while when $\mathrm{Cov}[X,Y] < 0$, the reverse is true.

Related to the covariance is the *correlation coefficient* between two random variables. This is defined as the ratio of the covariance to the product of the marginal standard deviations:

$$\mathrm{Corr}[X,Y] = \frac{\mathrm{Cov}[X,Y]}{\sigma_X\, \sigma_Y}. \tag{2.10}$$

Traditional notation for the correlation coefficient is $\rho = \mathrm{Corr}[X,Y]$ or, if greater specificity is required, ρ_{XY}. It can be shown that while $\mathrm{Cov}[X,Y]$ can take on any real value, ρ is restricted to the interval $-1 \le \rho \le 1$ (see Exercise 2.12).

As it is a scaled covariance, the correlation coefficient ρ measures association between two random variables in much the same way as σ_{XY}: when $\rho > 0$, increasing values of X tend to associate with increasing values of Y, while when $\rho < 0$, the reverse is true. And, if X and Y are independent, then $\rho = 0$. The converse is somewhat more complicated, however. The correlation is a *linear* measure of association, so when $\rho = 0$, no linear association is evidenced between X and Y. The two variables may still be related in, however, say, a quadratic or other curvilinear manner and, hence, may not be independent. When employing ρ to measure association between two variables, it is best in practice to be aware of both its strengths and its limitations.

2.2 Multiple random variables*

All of the operations described in Section 2.1 can be applied to a set of $n \geq 2$ random variables, $X_1, X_2, \ldots, X_n$. Generically, the corresponding joint p.m.f. or p.d.f. is denoted by $f_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n)$. Any subset of these n variates, say $X_1, \ldots, X_j$ for $j < n$, possesses a corresponding joint marginal p.m.f. or p.d.f. $f_{X_1,\ldots,X_j}(x_1, \ldots, x_j)$, found by summing or integrating over the remaining variables. For example, in the continuous case, one has

$$f_{X_1,\ldots,X_j}(x_1, \ldots, x_j) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n)\, dx_{j+1} \cdots dx_n.$$

Joint conditional probability functions are also possible; for example, given $X_1 = x_1, \ldots, X_j = x_j$, the joint conditional p.m.f. or p.d.f. of $X_{j+1}, \ldots, X_n$ is

$$f_{X_{j+1},\ldots,X_n|X_1,\ldots,X_j}(x_{j+1}, \ldots, x_n|x_1, \ldots, x_j) = \frac{f_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n)}{f_{X_1,\ldots,X_j}(x_1, \ldots, x_j)}.$$

If the n random variables $X_1, X_2, \ldots, X_n$ are independent, an extension of the Multiplication Rule (2c) provides their joint p.m.f. or p.d.f., $f_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n) = f_{X_1}(x_1) f_{X_2}(x_2) \cdots f_{X_n}(x_n)$, or simply

$$f_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} f_{X_i}(x_i). \tag{2.11}$$

A convenient notational device for multivariate random variables is to employ vector and matrix notation. (See Appendix A for a refresher on vector and matrix terminology.) The n-variate random vector is $\mathbf{X} = [X_1 \cdots X_n]^{\mathrm{T}}$ and its corresponding vector of population means is $\boldsymbol{\mu} = [\mu_1 \cdots \mu_n]^{\mathrm{T}}$. Here, superscript T denotes the transpose of a vector. For assembling together the collection of n variances σ_i^2 and the $\frac{1}{2}n(n-1)$ distinct covariances $\sigma_{ij} (i \neq j)$, use the *variance–covariance matrix* (or just *covariance matrix*)

$$\mathrm{Var}[\mathbf{X}] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_n^2 \end{bmatrix}, \tag{2.12}$$

where the variances are placed on the main (left-to-right) diagonal and the covariances lie in the corresponding off-diagonal positions. Notice that because $\sigma_{ij} = \sigma_{ji}$, $\mathbf{V} = \text{Var}[\mathbf{X}]$ is a symmetric matrix in that $\mathbf{V} = \mathbf{V}^{\text{T}}$.

To illustrate selected operations with multiple random variables, consider the following example with sums of random variables.

Example 2.2.1 Linear combinations. We often have reason to construct linear combinations of random variables, say $L = \sum_{i=1}^{n} w_i X_i$, for some real values w_i. For instance, when the w_is satisfy (i) $w_i \geq 0$ and (ii) $\sum_{i=1}^{n} w_i = 1$, L is called a *weighted average* of the X_is, with weights w_i. Using standard notation, let $\mu_i = E[X_i]$ and $\sigma_i^2 = \text{Var}[X_i]$, $i = 1, \ldots, n$.

As expectation is a linear operator, any linear combination L has expected value

$$E[L] = E\left[\sum_{i=1}^{n} w_i X_i\right] = \sum_{i=1}^{n} E[w_i X_i] = \sum_{i=1}^{n} w_i E[X_i] = \sum_{i=1}^{n} w_i \mu_i, \tag{2.13}$$

When the w_is are weights, this shows that the expected value of a weighted average is a weighted average of the expected values.

In the special case where $n = 2$ and $w_1 = w_2 = 1$, L is simply the sum of two random variables. Clearly then, (2.13) reduces to $E[X_1 + X_2] = \mu_1 + \mu_2$. Similarly, if $n = 2$ and $w_1 = -w_2 = 1$, L is the difference between two random variables, with expected value $E[X_1 - X_2] = \mu_1 - \mu_2$.

The variance of a linear combination is somewhat more difficult to express. Suppose for the moment that the covariance between X_i and X_j is $\text{Cov}[X_i, X_j] = \sigma_{ij}$ $(i \neq j)$. Then the variance of the sum $X_i + X_j$ can be shown (Exercise 2.13) to be

$$\text{Var}[X_i + X_j] = \text{Var}[X_i] + \text{Var}[X_j] + 2\text{Cov}[X_i, X_j] = \sigma_i^2 + \sigma_j^2 + 2\sigma_{ij}. \tag{2.14}$$

Including weights w_i and w_j in (2.14) produces

$$\text{Var}[w_i X_i + w_j X_j] = w_i^2 \sigma_i^2 + w_j^2 \sigma_j^2 + 2 w_i w_j \sigma_{ij}, \tag{2.15}$$

so that, for example, $\text{Var}[X_i - X_j] = \sigma_i^2 + \sigma_j^2 - 2\sigma_{ij}$ (notice where the minus sign enters into the expression).

Expanding (2.15) to any n-variate linear combination yields

$$\text{Var}[L] = \text{Var}\left[\sum_{i=1}^{n} w_i X_i\right] = \sum_{i=1}^{n} w_i^2 \sigma_i^2 + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_i w_j \sigma_{ij}. \tag{2.16}$$

If the X_is are independent, then we know the covariances are zero: $\sigma_{ij} = 0$. Thus (2.16) simplifies to

$$\text{Var}[L] = \sum_{i=1}^{n} w_i^2 \sigma_i^2.$$

If $w_i = 1$ for all i and *if the random variables are independent*, then we see the variance of a sum is the sum of the variances. □

It is also possible to study the limiting or *asymptotic* behavior of a sequence of n random variables as $n \to \infty$. For instance, a sequence of random variables $X_1, \ldots, X_n$ is said to

converge in distribution to a target random variable T if the c.d.f.s of the X_is converge in the limit to the c.d.f. of T. Formally, this is $\lim_{n\to\infty} F_{X_n}(t) = F_T(t)$ at all real values t where $F_T(t)$ is continuous. This concept can be applied to a weighted average (or any sum) of n random variables as well, in order to study the distribution of the average as its number of components grows. This will be explored further in Section 2.3.9.

A full study of convergence in probability is beyond the scope here, and interested readers are referred to advanced treatments in Lehmann and Casella (1998, Section 1.8) or Casella and Berger (2002, Section 5.5). Indeed, greater detail on all the concepts reviewed above is available in textbooks on probability and statistics such as Horgan (2009) or Hogg and Tanis (2010), or the classic theory text by Feller (1968).

2.3 Univariate families of distributions

When a random variable X occurs with a regular structure, it is useful to refer it to a specific distributional pattern. If this pattern can be formulated as a mathematical function, the distribution is said to belong to a family of such functions. These families typically possess one or more unknown *parameters*, such as the population mean μ and variance σ^2, that describe the distribution's characteristics. Thus it is common to refer to a parametric family of distributions when specifying the p.m.f. or p.d.f. of X. This section summarizes some important families, beginning with discrete distributions and moving on to continuous forms. More general descriptions on families of statistical distributions are available in dedicated texts such as Forbes et al. (2010) or the series by Johnson et al. (2005, 1994, 1995, 1997) and Kotz et al. (2000).

2.3.1 Binomial distribution

The *binomial distribution* is a basic discrete family used to describe data in the form of proportions. A binomial random variable X is generically constructed as the number of positive outcomes (or 'successes') among N statistically independent, binary 'trials.' Each trial is assumed to produce a success with (constant) probability $\pi \in (0, 1)$. The corresponding p.m.f. takes the form

$$f_X(m) = \binom{N}{m} \pi^m \, (1-\pi)^{N-m} I_{\{0,1,\ldots,N\}}(m), \tag{2.17}$$

where

$$\binom{N}{m} = \frac{N!}{m!\,(N-m)!} \tag{2.18}$$

is the *binomial coefficient* (the number of ways of selecting m items from a collection of N elements) and $m!$ is the *factorial operator*

$$m! = m(m-1)(m-2)\cdots(2)(1). \tag{2.19}$$

for any positive integer m. Also, define $0! = 1$. Notice in (2.17) use of the notation $I_S(m)$; this represents the *indicator function* over the set S,

$$I_S(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{otherwise,} \end{cases} \tag{2.20}$$

and is an efficient way to specify in the expression for a p.m.f. or p.d.f. the particular sample space for X.

The notation to indicate quickly that X possesses a binomial p.m.f. is $X \sim \text{Bin}(N, \pi)$. (The tilde symbol, $\sim$, is read as 'is distributed as.') If $X \sim \text{Bin}(N, \pi)$, its population mean is $E[X] = N\pi$ and its population variance is $\text{Var}[X] = N\pi(1 - \pi)$. A special case occurs with $N = 1$, that is, a single trial that produces a dichotomous, binary outcome for X. If so, the binomial reduces to what is known as a *Bernoulli distribution*. The singleton event producing the binary count X is often called a *Bernoulli trial*.

Calculating the binomial p.m.f. in (2.17) or the c.d.f., $F_X(m) = P[X \leq m]$ is not particularly onerous, although when N is larger than about 20, the factorial computations can become challenging. To facilitate the effort, some statistics textbooks present tables of the factorial operator and the p.m.f. or c.d.f. The calculations may also be performed conveniently by computer, however, and many statistical computing languages and packages provide internal functions for binomial probabilities. Most popular are the **R** language (http://cran.r-project.org/), the similar S-Plus language (http://csan.insightful.com/), the SAS® system (http://www.sas.com/), and the IBM SPSS® package (http://www.ibm.com/software/analytics/spss/). Each has its own advantages and disadvantages for statistical data analytics; Tufféry (2011, Chapter 5) gives a useful overview. Throughout this textbook, the **R** language (R Core Team 2014) is highlighted. (See Appendix B for a short introduction to **R**.)

Example 2.3.1 Binomial distribution. To compute the binomial p.m.f., **R** provides the `dbinom(m, size, prob)` function, where `m` is the function argument, `size` is the sample size parameter N, and `prob` is the success probability π in (2.17). The corresponding binomial c.d.f. is available via **R**'s `pbinom(m, size, prob)` function.

To illustrate, suppose $X \sim \text{Bin}(50, 0.05)$. With N as large as 50 and $\pi = 0.05$ so close to zero, direct calculation here of the quantities in (2.17) is challenging. **R**'s `*binom` functions can facilitate the effort. To compute, say, $P[X = 6]$, use `dbinom(6, 50, .05)`. This gives $P[X = 6] = 0.0260$. □

For intermediate calculations, **R** also provides the `choose(n,m)` function for the binomial coefficient in (2.18) and the `factorial(m)` function for $m!$. With large m, use the `lfactorial(m)` function, giving $\log(m!) = \exp\{\log(m!)\}$.

Example 2.3.2 Purchasing probability. A retail outlet samples $N = 1024$ of its affinity customers to ascertain if temporary price reductions ('sales') lead to increased purchases. Let X be the number of those customers who purchase an item during a sale. Suppose the true probability that a customer would make a sale purchase is $\pi = 0.50$. The outlet manager wants to know the probability that at least half of those customers will make a purchase.

Assuming the $N = 1024$ customers make purchases independently of each other and that π remains constant among them, X can be taken as binomial: $X \sim \text{Bin}(1024, 0.5)$. The manager wishes to find $P[X \geq 512]$. From the Complement Rule (2d), this is $P[X \geq 512] = 1 - P[X < 512]$. As X is discrete, however, one cannot calculate this as 1 minus the c.d.f. at $m = 512$. Instead, recognize that for a binomial random variable, the event $\{X < 512\}$ is equivalent to the event $\{X \leq 511\}$. As a result,

$$P[X \geq 512] = 1 - P[X \leq 511] = 1 - F_X(511),$$

where $F_X(m)$ is the c.d.f. of $X \sim \text{Bin}(1024, 0.5)$.

With N so large, the c.d.f. here is unwieldy. The calculation is most efficiently performed by computer. In **R**, this is available via the `pbinom` function for the binomial c.d.f.: for $F_X(511)$, use `pbinom(511, 1024, 0.5)`. Then to find $P[X \geq 512]$, subtract the result from 1. This is $1 - 0.4875$, or about a 51.2% chance that at least half of these 1024 customers will make a purchase.

If the sample is drawn down to, say, only $N = 20$ affinity customers, the calculation is somewhat more manageable: for $X \sim \text{Bin}(20, 0.5)$, find $P[X \geq 10] = 1 - P[X < 10] = 1 - P[X \leq 9] = 1 - F_X(9)$, that is,

$$P[X \geq 10] = 1 - F_X(9) = 1 - \sum_{m=0}^{9} \binom{20}{m} \left(\frac{1}{2}\right)^m \left(1 - \frac{1}{2}\right)^{20-m}.$$

While this could be accomplished by hand, the computer is still a speedier alternative: in **R**, use `pbinom(9, 20, 0.5)` and, to find $P[X \geq 10]$, subtract the result from 1. This is $1 - 0.4119$, or now about a 58.8% chance that at least half of these 20 customers will make a purchase.

Figure 2.3 plots the Bin(20, 0.5) p.m.f. and shades in the area corresponding to $P[X \geq 10]$.

Notice that the p.m.f. is unimodal and symmetric about $m = 10$, which here is also the population mean: $\mu = N\pi = (20)(0.5) = 10$. □

The binomial model is a popular choice for settings in which the response is the number of Bernoulli trials that exhibit some characteristic of interest – generically, a Bernoulli 'success' – such as whether or not a customer will make a sale purchase. (Technically, the possible outcomes are nonnegative counts bounded above by some known integer, N.) In most instances, the number of trials, N, is fixed in advance while the number of successes, X,

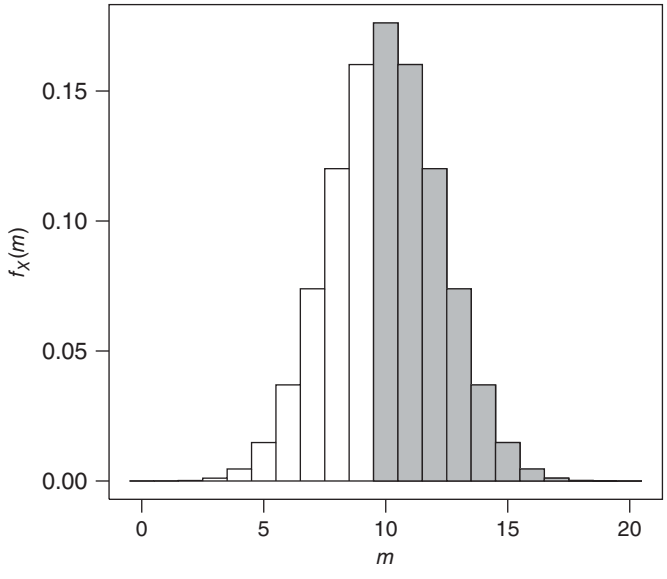

Figure 2.3 Probability mass function (p.m.f.) for $X \sim \text{Bin}(20, 0.5)$ over possible outcomes $m = 0, 1, \ldots, 20$. Shaded area is $P[X \geq 10]$ as in Example 2.3.2.

is the random variable of interest. The response is often reported as a proportion of successful outcomes, X/N.

Binomial random variables exhibit an important feature: the family is closed under addition. That is, suppose $X_1, X_2, \ldots, X_n$ are independent binomials with sample-size parameters N_i and (common) response probability π. Write this as $X_i \sim$ indep. Bin(N_i, π), $i = 1, \ldots, n$. Then, their sum is also binomial: $\sum_{i=1}^{n} X_i \sim \text{Bin}(N_+, \pi)$ where

$$N_+ = \sum_{i=1}^{n} N_i \, .$$

(The '+' in the subscript is used throughout as shorthand notation for summation over a subscripted index.)

2.3.2 Poisson distribution

For data in the form of discrete, unbounded counts, that is, observations taken over the space of nonnegative integers $\mathbb{Z}_{(\geq 0)} = \{0, 1, \ldots\}$, a popular probability model is known as the *Poisson distribution*. Given a random variable X as the number of outcomes of some random process, X has a Poisson distribution if its p.m.f. takes the form

$$f_X(m) = \frac{\lambda^m \, \mathrm{e}^{-\lambda}}{m!} \, I_{\{0,1,\ldots\}}(m), \tag{2.21}$$

where $\lambda > 0$ is the rate parameter of the distribution. The Poisson p.m.f. is based on a set of elementary conditions called the *Poisson postulates*. These describe how random events occur or 'arrive' in a fixed temporal or spatial region at a rate of λ events per unit time or unit area, respectively. They are summarized as follows:

- Start with no event occurrences/arrivals in the region.
- Assume occurrences in disjoint temporal/spatial subregions are independent.
- Allow the number of occurrences in disjoint subregions to depend only on each subregion's length, area, or volume (as appropriate).
- Set the occurrence probability proportional to the temporal/spatial length or area (in a limiting sense, as the length or area goes to zero).
- Allow for no exactly simultaneous occurrences.

If these five postulates hold, then the Poisson p.m.f. can be derived as the probability of observing a nonnegative random count, X, of occurrences per unit time or space. (For more on the Poisson postulates from the temporal or spatial perspective, see, for example, Casella and Berger (2002, Section 3.8) or Piegorsch and Bailer (1997, Section 1.2.5), respectively.)

The mean and variance of a Poisson distribution are $E[X] = \text{Var}[X] = \lambda$. (Note the mean-to-variance equality! This is a stringent consequence of the Poisson sampling assumption.) The reference notation is $X \sim \text{Poisson}(\lambda)$.

As with the binomial model, direct calculation of the Poisson p.m.f. or c.d.f. can grow difficult for large values of m and one often turns to the computer. In **R**, the pertinent functions are `dpois(m, lambda)` and `ppois(m, lambda)`. These are illustrated in the following example.

Example 2.3.3 Microprocessor failure. A computer manufacturer uses microprocessor chips that fail with a low event rate of $\lambda = 2.7$ per 100 000 h (or about 11.4 years) of use. The manufacturer wishes to know if this will lead to high component failure in its machines. In particular, find $P[X > 0]$, where $X = \{\text{number of chip failures in 100 000 h of use}\}$ is assumed to take a Poisson distribution with rate parameter $\lambda = 2.7$. To find $P[X > 0]$, apply the Complement Rule (2d): $P[X > 0] = 1 - P[X \leq 0] = 1 - P[X = 0]$. (The latter equality holds because the events $\{X \leq 0\}$ and $\{X = 0\}$ are identical – Poisson random variables are only defined over nonnegative integers.) From (2.21), we have $1 - P[X = 0] = 1 - (2.7)^0 e^{-2.7}/0! = 1 - e^{-2.7} = 0.9328$. Thus, there is about a 93% chance that at least one microprocessor will lead to component failure over 100 000 h of use.

In **R**, this calculation is available via `1 - dpois(0, 2.7)`, which again returns a probability of slightly over 93%. □

Similar to the binomial distribution, the Poisson family is closed under addition: if $X_i \sim$ indep. Poisson(λ_i), $i = 1, \ldots, n$, then $\sum_{i=1}^n X_i \sim$ Poisson(λ_+), where $\lambda_+ = \sum_{i=1}^n \lambda_i$.

2.3.3 Geometric distribution

Return to the setting where a series of independent Bernoulli trials are observed, each with binary outcome equal to either 1 ('success') or 0 ('failure'). Assume that the probability of 'success' on any trial is held constant at $\pi \in (0, 1)$. Now, however, instead of fixing the number of trials at a known upper limit N – as in Section 2.3.1 – allow the trials to continue until the first success is observed. Take X as the number of trials up to (but *not* including) that first success. The p.m.f. of X is then

$$f_X(m) = \pi\ (1 - \pi)^m\ I_{\{0,1,\ldots\}}(m). \tag{2.22}$$

This is known as the *geometric distribution*. The reference notation is $X \sim$ Geom(π).

The mean of a geometric random variable is $E[X] = (1 - \pi)/\pi$ and the variance is $\text{Var}[X] = (1 - \pi)/\pi^2$. Given the simple form of the p.m.f. in (2.22), the c.d.f. here is especially easy to derive: $F_X(m) = P[X \leq m] = \sum_{i=0}^m \pi(1 - \pi)^i = \pi \sum_{i=0}^m (1 - \pi)^i$. Recall, however, the formula for a finite geometric series:

$$\sum_{i=0}^{m} \psi^i = \frac{1 - \psi^{m+1}}{1 - \psi} \tag{2.23}$$

for any $|\psi| < 1$. Applying this to the geometric c.d.f., one finds $F_X(m) = 1 - (1 - \pi)^{m+1}$ for any $m = \{0, 1, 2, \ldots\}$.

A warning: some authors alternatively define the geometric as the number of failures up to *and including* the first success. Thus the random variable is now defined as a strictly positive count. (In effect, it is the transformed variable $Y = X + 1$ in the notation above.) This changes the form of (2.22) and all its consequent expressions. See, for example, Casella and Berger (2002, Section 3.2).

2.3.4 Negative binomial distribution

A natural extension of the geometric sampling construction in Section 2.3.3 is to instead let X be the number of trials up to (but *not* including) the rth success, for $r \geq 1$. (Continue to

hold the probability of success on any trial constant at π.) In this case, the p.m.f. in (2.22) generalizes to

$$f_X(m) = \binom{r+m-1}{m} \pi^r (1-\pi)^m I_{\{0,1,\dots\}}(m). \tag{2.24}$$

This is known as the *negative binomial distribution*. The reference notation is $X \sim \text{NB}(r, \pi)$. The negative binomial mean is $E[X] = r(1-\pi)/\pi$ and the variance is $\text{Var}[X] = r(1-\pi)/\pi^2$. Clearly, when $r = 1$, the negative binomial reduces to the simpler geometric form. Viewed this way, the negative binomial family exhibits a special form of closure under addition: if $X_i \sim$ indep. $\text{Geom}(\pi) = \text{NB}(1, \pi)$, then $\sum_{i=1}^{r} X_i \sim \text{NB}(r, \pi)$.

Notice that both the geometric distribution and the negative binomial distribution are defined for nonnegative counts on unbounded sample spaces. Thus in some sense, they serve as competitors to the Poisson distribution for modeling unbounded count data. This can be recognized more clearly by redefining their parametric structure. To wit, in (2.24), write the negative binomial mean as $\mu = r(1-\pi)/\pi$ and also let $\delta = 1/r$. Then, the p.m.f. becomes

$$f_X(m) = \frac{\Gamma(\delta^{-1}+m)}{\Gamma(\delta^{-1})\, m!} \left(\frac{\delta\mu}{1+\delta\mu}\right)^m \frac{1}{(1+\delta\mu)^{1/\delta}} I_{\{0,1,\dots\}}(m), \tag{2.25}$$

where

$$\Gamma(a) = \int_0^\infty t^{a-1} \mathrm{e}^{-t}\, dt \tag{2.26}$$

is the *gamma function*. In passing, it is useful to note a recursive relationship that exists for $\Gamma(a)$: one can show (via integration by parts from univariate calculus) that $\Gamma(a+1) = a\Gamma(a)$ for any $a > 0$. As a consequence, for any positive integer n the gamma function in (2.26) relates to the factorial operator: $n! = \Gamma(n+1)$.

Clearly, in (2.25), both μ and δ are positive-valued parameters. The population mean of X is of course now $E[X] = \mu$, and the variance becomes $\text{Var}[X] = \mu + \delta\mu^2$. This provides a more flexible mean-to-variance relationship than that exhibited under the Poisson model. As $\delta \to 0$, the p.m.f. in (2.25) converges to the Poisson p.m.f. in (2.21).

By the way, at $\delta = 1$ (i.e., $r = 1$), one recovers a corresponding reexpression for the geometric p.m.f. in (2.22) based on its mean μ:

$$f_X(m) = \left(\frac{\mu}{1+\mu}\right)^m \frac{1}{(1+\mu)} I_{\{0,1,\dots\}}(m). \tag{2.27}$$

One finds, as expected, that $E[X] = \mu$. More interestingly, however, the geometric variance now can be expressed as a quadratic function of its mean: under (2.27), $\text{Var}[X] = \mu(\mu+1)$.

2.3.5 Discrete uniform distribution

One of the simplest discrete distributions is the (discrete) *uniform distribution*. This occurs when observations are taken with equal probability over a discrete sample space. In its canonical form, the discrete uniform samples over the first N positive integers, so the p.m.f. assigns equal probability to every element of $\mathcal{S} = \{1, 2, \dots, N\}$. This is $f_X(m) = (1/N) I_{\{1,2,\dots,N\}}(m)$.

The reference notation for a discrete uniform p.m.f. is $X \sim \text{Unif}\{N\}$. The mean is quickly found to be $E[X] = \frac{1}{2}(N+1)$, while the variance is $\text{Var}[X] = (N^2-1)/12$.

Referring back to the simple six-sided die roll in Example 2.1.3, we see that the random variable $X = \{\text{Number of pips}\} \sim \text{Unif}\{6\}$. As in Example 2.14, $E[X] = \frac{1}{2}(6+1) = 3.5$. Now, however, we also find $\text{Var}[X] = (6^2 - 1)/12 = 35/12$.

2.3.6 Continuous uniform distribution

A continuous analog to the discrete uniform distribution in Section 2.3.5 is the *continuous uniform distribution* over any interval (a, b). The interval serves as the sample space. The associated p.d.f. assigns uniform probability density across all of (a, b):

$$f_X(x) = \frac{1}{b-a}\, I_{(a,b)}(x).$$

The reference notation to indicate that X possesses a continuous uniform p.d.f. is $X \sim U(a, b)$. The mean is $E[X] = \frac{1}{2}(a+b)$, while the variance is $\text{Var}[X] = (b-a)^2/12$. A special case of $U(a, b)$ occurs when $a = 0$ and $b = 1$, producing a uniform p.d.f. over the unit interval.

The continuous uniform distribution is also referred to by some authors as the *rectangular distribution*.

2.3.7 Exponential distribution

After the uniform, perhaps the simplest continuous p.d.f. is associated with the *exponential distribution*:

$$f_X(x) = \frac{1}{\beta}\, \mathrm{e}^{-x/\beta}\, I_{(0,\infty)}(x), \tag{2.28}$$

where the rate parameter β is constrained to be positive. The exponential is a common choice for positive random variables that represent waiting times between events or certain forms of lifetime data. The reference notation is $X \sim \text{Exp}(\beta)$.

The exponential mean is $E[X] = \beta$, while the variance is $\text{Var}[X] = \beta^2$. Notice that this is yet another quadratic relationship between variance and mean. An interesting property arises in this case: the ratio of an exponential variable's population standard deviation to its population mean is constant. (This ratio is known as the *coefficient of variation*, or often just 'cv.') With the exponential p.d.f., $\text{cv} = \sqrt{\text{Var}[X]}/E[X] = \sqrt{\beta^2}/\beta = 1$.

The exponential distribution possesses an intriguing feature when calculating certain conditional probabilities. Recognize that the exponential c.d.f. is

$$\begin{aligned} F_X(x) &= \int_0^x \frac{1}{\beta}\, \mathrm{e}^{-t/\beta} dt \\ &= -\mathrm{e}^{-t/\beta}\Big|_0^x = 1 - \mathrm{e}^{-x/\beta} \end{aligned}$$

for any $x > 0$. Notice then that the area under the upper tail of the p.d.f. beyond a point $t > 0$ – often called an 'upper-tail probability' – is

$$P[X > t] = 1 - P[X \le t] = 1 - F_X(t) = \mathrm{e}^{-t/\beta}, \tag{2.29}$$

by application of the Complement Rule (2d). In many reliability and time-to-event applications, this is viewed as a kind of 'survival probability,' that is, the probability of operating or surviving past a given time t. Now, consider the further, *conditional* probability that X

exceeds some value t, given that it has already exceeded some lesser value $u > 0$. This is $P[X > t|X > u]$ which, from the Conditionality Rule (2b), becomes

$$P[X > t|X > u] = \frac{P[X > t \text{ and } X > u]}{P[X > u]}. \tag{2.30}$$

But because $t > u$, the joint event $\{X > t \text{ and } X > u\}$ is identical to the event $\{X > t\}$. Thus (2.30) is just $P[X > t|X > u] = P[X > t]/P[X > u]$. Now, if X is exponentially distributed, (2.29) gives $P[X > t] = \mathrm{e}^{-t/\beta}$ and the conditional probability reduces to

$$\begin{aligned} P[X > t|X > u] &= \frac{P[X > t]}{P[X > u]} \\ &= \frac{\mathrm{e}^{-t/\beta}}{\mathrm{e}^{-u/\beta}} = \mathrm{e}^{-(t-u)/\beta} \end{aligned}$$

for any $t > u$ (> 0). Lastly, notice from (2.29) that $\mathrm{e}^{-(t-u)/\beta}$ is just $P[X > t - u]$, leading to

$$P[X > t|X > u] = P[X > t - u] \tag{2.31}$$

for any $X \sim \text{Exp}(\beta)$. In other words, the probability that an exponential random variable will exceed some survival time t, given that it has already exceeded an earlier survival time u, depends on the time between t and u but otherwise not directly on u. In effect, an exponential random variable 'forgets' where it has been when considering how much farther it can operate or last. This is known as the *memoryless property* of the exponential distribution.

2.3.8 Gamma and chi-square distributions

An extension of the exponential p.d.f. from Section 2.3.7 into a richer, more flexible family is known as the *gamma distribution*. The gamma p.d.f. is

$$f_X(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha}\, x^{\alpha-1}\, \mathrm{e}^{-x/\beta}\, I_{(0,\infty)}(x), \tag{2.32}$$

where $\alpha > 0$ is the *shape parameter* of the distribution and $\beta > 0$ is the *scale parameter* (Casella and Berger 2002, Section 3.3). The reference notation is $X \sim \text{Gamma}(\alpha, \beta)$. (Some authors write the gamma p.d.f. in terms of the scale parameter α and an alternative *rate* parameter which in the notation above is $1/\beta$. Users should be careful to identify the parameterization under which they operate.)

The gamma extends (i) the exponential mean to $E[X] = \alpha\beta$ and (ii) the exponential variance to $\text{Var}[X] = \alpha\beta^2$. Clearly, when $\alpha = 1$, we recover the simpler exponential form.

Example 2.3.4 **Ecological monitoring.** Ecologists study the health of organisms that inhabit coastal marine ecosystems. One measure of aquatic health is the biomass of green algae (*Selenastrum capricornutum*) at or near sites of ecological contamination. Signs of deceasing biomass in the algae can indicate potential ecosystem damage.

Observations on algal biomass are positive valued and can skew to the right, for which a gamma distribution provides a reasonable model (Bailer and Oris 1997). Suppose that under static conditions, $X = \{\text{Green algae biomass (in cells/mm)}\} \sim \text{Gamma}(100,12)$. When algal biomass drops below a 'sentinel' level of about 1000 cells/mm, however, this may indicate

ecological damage. To find $P[X \leq 1000]$, appeal to the gamma c.d.f.:

$$P[X \leq 1000] = F_X(1000) = \int_0^{1000} \frac{1}{\Gamma(100)(12)^{100}} \, x^{99} \, \mathrm{e}^{-x/12} \, dx \, . \tag{2.33}$$

The integral in (2.33) could be evaluated directly, although it is just as effective to call directly on the computer. In **R**, the gamma c.d.f. is available via the

```
> pgamma( x, shape=, scale= )
```

function, where `x` is the argument of the c.d.f., `shape=` specifies the shape parameter α, and `scale=` specifies the scale parameter β. (The `scale=` specification should always be written out; **R** includes an optional `rate=` specification, ordered before `scale=`, for users who wish to parameterize the gamma in terms of shape and rate, α and $1/\beta$, instead of shape and scale.)

As applied to (2.33), `pgamma(1000, shape=100, scale=12)` gives $P[X \leq 1000] = 0.0413$. Thus there is less than a 5% chance of seeing algal biomass levels this low under normal conditions. A drop in biomass to these levels may well be indicative of a toxic or otherwise hazardous ecological impact.

Figure 2.4 plots the corresponding p.d.f. The shaded area in the plot represents the target probability $P[X \leq 1000]$. The p.d.f. is unimodal, with a slight right skew. (The skew is difficult to see on this scale, but it is present.) □

The gamma family possesses a particular form of closure under addition: if $X_i \sim$ indep. Gamma(α_i, β), $i = 1, \ldots, n$, then $\sum_{i=1}^n X_i \sim$ Gamma(α_+, β), where $\alpha_+ = \sum_{i=1}^n \alpha_i$. In the special case of $\alpha_i = 1$ for all i, the gamma variates collapse to exponentials: $X_i \sim$ indep.

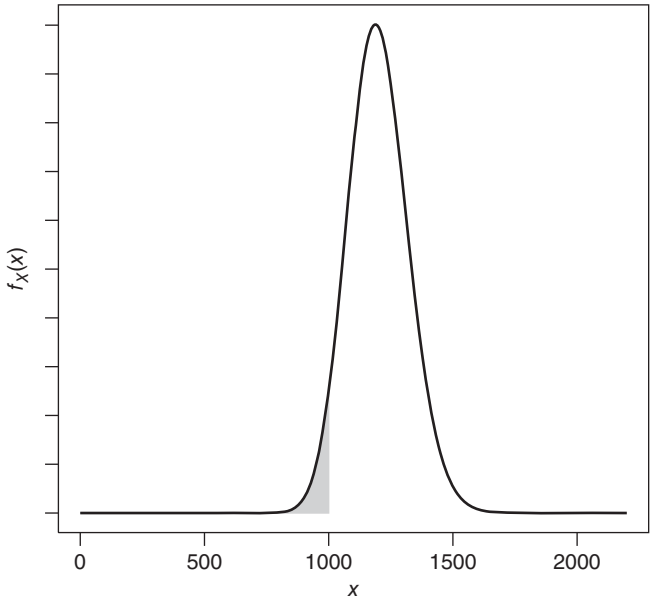

Figure 2.4 Probability density function (p.d.f.) for $X \sim$ Gamma(100, 12) in Example 2.3.4. Shaded area is $P[X \leq 1000] = 0.0413$.

Gamma$(1, \beta)$ = Exp(β). Then, sums of independent exponentials are gamma distributed: $\sum_{i=1}^{n} X_i \sim$ Gamma(n, β).

An important special case of the gamma p.d.f. occurs when $X \sim$ Gamma $\left(\frac{\nu}{2}, 2\right)$, that is, when $\alpha = \nu/2$ and $\beta = 2$ in (2.32). This is known as the χ^2 (or *chi-square*) *distribution*, where the parameter ν is usually a positive integer and is referred to as the *degrees of freedom* ('d.f.') of the distribution. The reference notation becomes $X \sim \chi^2(\nu)$. The chi-square mean is simply the degrees of freedom, $E[X] = \nu$, while the variance is twice the mean, Var$[X] = 2\nu$. Also, if $X_i \sim$ indep. $\chi^2(\nu_i)$, $i = 1, \ldots, n$, then $\sum_{i=1}^{n} X_i \sim \chi^2(\nu_+)$, for $\nu_+ = \sum_{i=1}^{n} \nu_i$. This follows from the additive closure (under fixed scale) of the larger gamma family. The χ^2 distribution can also be derived from another important, continuous random variable known as the normal (or Gaussian) distribution. The normal is introduced in the next subsection, while further features of the χ^2 are discussed in Section 2.3.10.

2.3.9 Normal (Gaussian) distribution

One of the most important continuous distributions used in data-analytic practice is the *normal distribution*, also called the *Gaussian distribution*. It has p.d.f.

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} I_{(-\infty,\infty)}(x), \tag{2.34}$$

where the parameter μ is any real number and the parameter σ^2 is strictly positive. Here, the fixed constant π is the ratio of a circle's circumference to its diameter, 3.14159265 …

The parameters μ and σ^2 also explicitly describe the population mean and variance, respectively, of the distribution: $E[X] = \mu$ and Var$[X] = \sigma^2$. Standard reference notation is $X \sim$ N(μ, σ^2). (Although, readers should be vigilant: some authors denote the normal as N(μ, σ), i.e., in terms of the mean and the standard deviation, not the mean and variance as done here.)

Normal distributions have unimodal, symmetric p.d.f.s that possess a 'bell' shape, centered at μ, and with spread governed by σ^2. They also share an important, unifying feature: any normal random variable can be standardized into a central form. Specifically, if $X \sim$ N(μ, σ^2), then $Z = (X - \mu)/\sigma \sim$ N$(0, 1)$ for any μ and any $\sigma > 0$. In this case, Z is known as the *standard normal random variable*. The c.d.f. of the standard normal is given a special notation:

$$\Phi(z) = P[Z \le z] = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}x^2} dx \,. \tag{2.35}$$

Unfortunately, the integral in (2.35) is intractable, and so it is obtained via numerical approximation. The resulting values are quite accurate and widely tabulated in many statistics books. They are also readily available via most statistical computing programs. For example, in **R**, the `pnorm(z)` function directly calculates $\Phi(z)$ in (2.35). In fact, `pnorm` operates like any general **R** `p*` function for c.d.f.s. Thus it can be extended to calculate the c.d.f of any $X \sim$ N(μ, σ^2): simply use `pnorm(x, mean=`μ`, sd=`σ`)`.

Also of value in many calculations is the N$(0, 1)$ upper-tail probability, that is, the area under the upper tail of the standard normal p.d.f. This is just $P[Z > z]$, which from the Complement Rule (2d) is found as $1 - P[Z \le z] = 1 - \Phi(z)$ for any real argument z. (Obviously, for a lower-tail probability, one simply appeals directly to the c.d.f.)

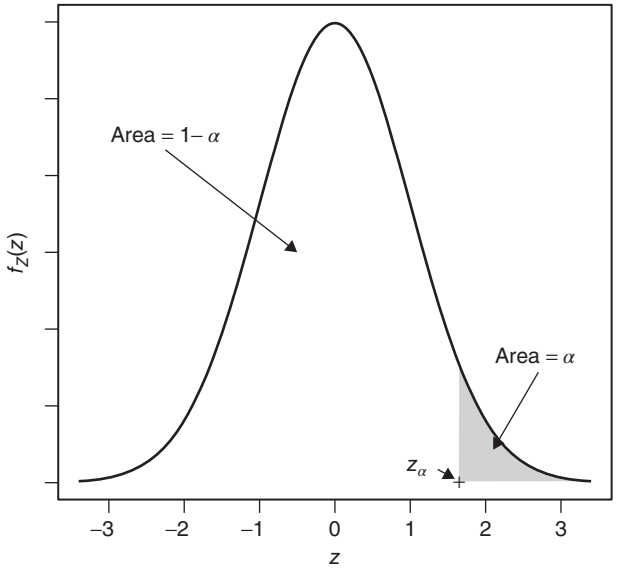

Figure 2.5 Standard normal probability density function (p.d.f.) and upper-α critical point z_α.

The standard normal c.d.f. $\Phi(z)$ is continuous and strictly increasing. This allows quantiles to be found by appealing to the inverse c.d.f., $\Phi^{-1}(p)$, as described in Section 2.1.4. Often more useful in practice, however, are the points *above which* a specified area under the p.d.f. is achieved, that is, the points that produce a given upper-tail probability. These values are known as the *upper-α critical points* of the standard normal p.d.f. and denoted by z_α. Mathematically, they satisfy the relationship $P[Z > z_\alpha] = \alpha$ for some $\alpha \in (0, 1)$. Similar to the distribution's quantiles, upper-α critical points can be found by inverting the c.d.f., if done carefully: area equal to α rests to the right of z_α, so from the Complement Rule (2d), area equal to $1 - \alpha$ must rest to the left. Thus z_α also satisfies $z_\alpha = \Phi^{-1}(1 - \alpha)$. Figure 2.5 illustrates some of these features.

In **R**, upper-α critical points from N(0, 1) are found using the command

```
> qnorm( 1-alpha, mean=0, sd=1 )
```

where `alpha` is the desired upper-tail probability. (`mean=0` and `sd=1` are the default values in `qnorm`, so `qnorm(1 - alpha)` would be sufficient.) One can also force **R** to perform the upper-tail calculation directly, via

```
> qnorm( alpha, mean=0, sd=1, lower.tail=FALSE )
```

As all normals are unimodal and symmetric about their means, so is the standard normal. In particular, it is symmetric about its mean, 0. This allows one to manipulate the c.d.f. and the critical points in a particular manner. For example, if z_α is chosen so that area α rests to its right on the z-scale, then from the symmetry about 0, area α must also rest to the *left* of $-z_\alpha$ on the z-scale. In effect, this establishes the relationship $\Phi^{-1}(1 - \alpha) = -\Phi^{-1}(\alpha)$, for any $\alpha \in (0, 1)$.

Another interesting feature of the normal distribution connects its standard deviation σ with its interquartile range (cf. Section 2.1.4): if $X \sim \text{N}(\mu, \sigma^2)$, then $\text{IQR}_X = 2\sigma\Phi^{-1}(3/4)$, which calculates to $\text{IQR} \approx 1.349\sigma$ (Stuart and Ord 1994, Section 10.11).

Readers are encouraged to explore other, similar facets of the normal distribution's symmetry, its p.d.f., and its c.d.f.

Example 2.3.5 Women's heights. Random variation in human height is often normally distributed, at least to a good approximation. Suppose the heights (in inches) of a population of college-age women are recorded as the random variable X. Take $X \sim \text{N}(65, 6.25)$ with population mean $\mu = 65$ inches (5 ft, 5 inches) and population standard deviation $\sigma = 2.5$ inches. If a volleyball coach is recruiting women taller than 72 inches (6 ft) to staff a new team, how likely is she to encounter such a woman in this population?

To answer this question, the target probability is $P[X > 72]$. Standardizing gives $Z = (X - \mu)/\sigma = (X - 65)/2.5 \sim \text{N}(0, 1)$, so that

$$P[X > 72] = P\left[\frac{X - 65}{2.5} > \frac{72 - 65}{2.5}\right] = P\left[Z > \frac{7}{2.5}\right] = P[Z > 2.8].$$

The Complement Rule (2d) then gives $P[Z > 2.8] = 1 - P[Z \leq 2.8] = 1 - \Phi(2.8)$. In **R**, this is simply `1 - pnorm(2.8)` which produces $P[Z > 2.8] = 0.0026$. Alternatively, one can force **R** to perform the upper-tail calculation directly, via

```
> pnorm( 2.8, lower.tail=FALSE )
```

or even avoid the standardization entirely and use

```
> pnorm( 72, mean=65, sd=2.5, lower.tail=FALSE )
```

All these give the same 0.0026 probability. In any case, we find that there is less than a three-tenths of 1% chance in finding a woman taller than 72 inches within this population.

The coach could also reverse the calculation and ask, for what height would he/she find women at or above the 95th percentile of this population (the 'top 5%')? That is, using the quantile notation from Section 2.1.4, find $q_{0.95}$ such that $P[X > q_{0.95}] = 0.05$. Standardizing here gives

$$P[X > q_{0.95}] = P\left[\frac{X - 65}{2.5} > \frac{q_{0.95} - 65}{2.5}\right] = 0.05. \tag{2.36}$$

But now, recall the notation for the upper-α critical point for Z: $P[Z > z_{0.05}] = 0.05$. Applying this to (2.36) shows $z_{0.05} = (q_{0.95} - 65)/2.5$, and solving for $q_{0.95}$ gives $q_{0.95} = 2.5z_{0.05} + 65$. To find $z_{0.05}$, we can use

```
> qnorm( 0.05, lower.tail=FALSE )
```

in **R**, producing $z_{0.05} = 1.64485$. Substituting this into the expression for $q_{0.95}$ leads to $q_{0.95} = (2.5)(1.64485) + 65 = 69.1121$ inches (or slightly over 5 ft, 9 inches) tall. Women taller than 69.1121 inches are in the upper 5th percentile of this distribution of heights. □

Similar to many of the distributions in this section, normal random variables possess closure properties under addition:

if $X_i \sim$ indep. $\text{N}(\mu_i, \sigma_i^2)$, $i = 1, \ldots, n$,
then $\sum_{i=1}^n X_i \sim \text{N}\left(\sum_{i=1}^n \mu_i, \sum_{i=1}^n \sigma_i^2\right)$.

In fact, this extends to any linear combination of independent normals (cf. Example 2.2.1):

if $X_i \sim$ indep. $\mathrm{N}(\mu_i, \sigma_i^2)$, $i = 1, \ldots, n$,
then $L = \sum_{i=1}^n w_i X_i \sim \mathrm{N}\left(\sum_{i=1}^n w_i \mu_i, \sum_{i=1}^n w_i^2 \sigma_i^2\right)$.

Suppose now that each normal is distributed identically as all the others, so that $X_i \sim$ indep. $\mathrm{N}(\mu, \sigma^2)$, $i = 1, \ldots, n$. Then, one has

$$L = \sum_{i=1}^n w_i X_i \sim \mathrm{N}\left(\mu \sum_{i=1}^n w_i, \sigma^2 \sum_{i=1}^n w_i^2\right). \tag{2.37}$$

In the special case of an equally weighted average, the w_is are all constant and sum to 1: $w_i = 1/n$, $\sum_{i=1}^n w_i = 1$, and $\sum_{i=1}^n w_i^2 = 1/n$. L is then the arithmetic average of the X_is. When associated with a random sample of n observations, this is also known as the *sample mean* and is given a special notation:

$$\overline{X} = \frac{1}{n} \sum_{i=1}^n X_i. \tag{2.38}$$

Clearly then, for n independent, identically distributed ('i.i.d.') normals $X_i \sim$ i.i.d. $\mathrm{N}(\mu, \sigma^2)$, one finds $\overline{X} \sim \mathrm{N}(\mu, \sigma^2/n)$. That is, the sample mean from a normal random sample is also normally distributed, with $E[\overline{X}]$ equal to the original mean μ but with $\mathrm{Var}[\overline{X}]$ equal to the original variance divided by n. In effect, $\overline{X}$ recovers the same location information, μ, as any member of the sample, but it does so with variability reduced by a factor of n (on the variance scale).

A powerful feature of the normal distribution, and one reason it is so common in data analytics, is that normality for the sample mean extends to more than just normal samples. That is, the sample mean of an i.i.d. sample will often possess a normal distribution, at least approximately. As seen above, this is exactly true for a sample from the normal itself, but it also encompasses essentially any i.i.d. sample. The result applies as the number of elements in the sample, n, grows large and is a form of convergence in distribution as described in Section 2.2. Known as the *Central Limit Theorem* or 'CLT,' it is best stated formally:

The central limit theorem (Lehmann and Casella 1998, Section 1.8). Take an i.i.d. random sample of observations, $X_i \sim$ i.i.d. $f_X(x)$, from some p.m.f. or p.d.f. $f_X(x)$ with finite mean $E[X_i] = \mu$ and finite variance $\mathrm{Var}[X_i] = \sigma^2$, $i = 1, \ldots, n$. The distribution of the sample mean $\overline{X}$ from (2.38) will converge to that of a normal distribution with mean $E[\overline{X}] = \mu$ and variance $\mathrm{Var}[\overline{X}] = \sigma^2/n$ as $n \to \infty$.

In practice, the CLT tells us that as n grows large,

$$\overline{X} \mathrel{\dot\sim} \mathrm{N}\left(\mu, \frac{\sigma^2}{n}\right),$$

where the symbol $\dot\sim$ is read 'is approximately distributed as.' The approximation's quality will vary case by case; for continuous p.d.f.s that are roughly symmetric and unimodal, the CLT approximation could be roughly valid for sample sizes as low as $n = 10$. For parent distributions that are more skewed and/or discrete, however, n must grow much larger, upwards of

50 or 60, for the approximation to take hold. (In extreme cases with extremely skewed parent distributions or with highly limited, discrete p.m.f.s, the sample size requirement can grow to over $n = 100$.)

When a sequence of random variables is known to converge to a normal distribution, it is often of interest to determine if some functional transformation of that random sequence also converges to normal. In many cases, the answer is yes, and the result is another convergence theorem from probability theory known as the *delta method*.

The delta method (Casella and Berger 2002, Section 5.5). Suppose a sequence of random variables X_n exists such that $\sqrt{n}(X_n - \theta)$ converges in distribution to $\mathrm{N}(0, \sigma^2)$ for some constant θ and some positive variance term σ^2. Then for any function $h(\theta)$ whose first derivative $h'(\theta)$ exists and is not equal to zero, the distribution of $\sqrt{n}\{h(X_n) - h(\theta)\}$ converges in distribution to $\mathrm{N}(0, \{h'(\theta)\}^2\sigma^2)$.

In effect, the delta method gives us an asymptotic approximation for any function of a centrally converging sequence of random variables. It is often useful when considering certain large-sample estimation problems, as discussed in Section 5.3.5.

The univariate normal distribution can also be extended into bivariate and multivariate forms. In the bivariate case, if two continuous random variables, X_1 and X_2, are jointly distributed as bivariate normal, then their marginal (univariate) distributions are $X_i \sim \mathrm{N}(\mu_i, \sigma_i^2)$, $i = 1, 2$, with $\mathrm{Cov}[X_1, X_2] = \sigma_{12}$. (The covariance, σ_{12}, can be any real number. When it is zero, X_1 and X_2 are statistically independent, and, of course, vice versa. The former feature is not true in general.) Thus the bivariate normal is fully described by these five separate parameters, $\mu_1, \mu_2, \sigma_1^2, \sigma_2^2$, and σ_{12}.

The correlation between X_1 and X_2 is $\rho_{12} = \mathrm{Corr}[X_1, X_2] = \sigma_{12}/(\sigma_1\sigma_2)$. As such, the bivariate normal distribution may alternatively be described in terms of the five parameters $\mu_1, \mu_2, \sigma_1^2, \sigma_2^2$, and ρ_{12}. The bivariate p.d.f. is then written as

$$f(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho_{12}^2}} \times \exp\left\{ -\frac{1}{2(1-\rho_{12}^2)} \left[\frac{(x_1-\mu_1)^2}{\sigma_1^2} - 2\rho_{12}\frac{(x_1-\mu_1)(x_2-\mu_2)}{\sigma_1\sigma_2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2} \right] \right\} I_{(-\infty,\infty)}(x_1) I_{(-\infty,\infty)}(x_2). \tag{2.39}$$

Note that any linear combination of the bivariate normal components, $L = w_1X_1 + w_2X_2$, itself possesses a normal distribution. This is $L \sim \mathrm{N}(\mu_L, \sigma_L^2)$ with $\mu_L = w_1\mu_1 + w_2\mu_2$ and $\sigma_L^2 = w_1^2\sigma_1^2 + 2w_1w_2\rho_{12}\sigma_1\sigma_2 + w_2^2\sigma_2^2$, using (2.13) and (2.16), respectively.

As in Section 2.2, one can collect these various components together into the bivariate vector of variates $\mathbf{X} = [X_1\ X_2]^{\mathrm{T}}$, with mean vector $\mu = [\mu_1\ \mu_2]^{\mathrm{T}}$ and covariance matrix

$$\mathbf{V} = \mathrm{Var}[\mathbf{X}] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}.$$

The reference notation becomes $\mathbf{X} \sim \mathrm{N}_2(\boldsymbol{\mu}, \mathbf{V})$. This construction can be extended to any n-vector of normal random variables, $\mathbf{X} = [X_1 \cdots X_n]^{\mathrm{T}}$, now with mean vector

$\boldsymbol{\mu} = [\mu_1 \cdots \mu_n]^{\mathrm{T}}$ and covariance matrix

$$\mathbf{V} = \mathrm{Var}[\mathbf{X}] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_n^2 \end{bmatrix}.$$

In this case, we say X has a *multivariate normal distribution* or *n-variate normal distribution* and use the reference notation $\mathbf{X} \sim \mathrm{N}_n(\boldsymbol{\mu}, \mathbf{V})$. The joint p.d.f. is written using matrix and vector notation as

$$f(x_1, x_2, \ldots, x_n) = \frac{1}{(2\pi)^{n/2}|\mathbf{V}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{V}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}, \tag{2.40}$$

where $|\mathbf{V}|$ is the determinant of $\mathbf{V}$. The external **R** package *mvtnorm* is useful for computing multivariate normal probabilities.

2.3.10 Distributions derived from normal

The normal distribution possesses many important applications in data analytics beyond those introduced in Section 2.3.9. For instance, it is the basis for a number of other distributions. Among these is the χ^2 random variable mentioned in Section 2.3.8. Although the χ^2 is by definition a special case of the gamma distribution, it can also be constructed from the normal. To do so, start with a single standard normal variate $Z_1 \sim \mathrm{N}(0, 1)$. It can be shown that the square of a standard normal is distributed as χ^2 with 1 d.f., that is, $Z_1^2 \sim \chi^2(1)$. Extending this to ν independent, squared, standard normal variates and applying the closure of the χ^2 under addition, one finds $X = \sum_{i=1}^{\nu} Z_i^2 \sim \chi^2(\nu)$.

Tail probabilities from the χ^2 often prove valuable in data analytics. Calculations with the χ^2 c.d.f. can be difficult to manipulate directly, however, so as with the standard normal, the computer is employed. In the **R** language, the `pchisq(x, df)` function gives χ^2 c.d.f. values, $P[X \leq x]$, where `df` are the pertinent χ^2 degrees of freedom. The `lower.tail=FALSE` option produces upper-tail areas $P[X > x]$; equivalently, appeal to the Complement Rule (2d) leads to

```
> 1 - pchisq( x, df )
```

for calculating $P[X > x]$.

One can also invert the upper-tail calculation to find upper-α critical points of the χ^2 distribution. Mimicking the notation from the standard normal, denote these as the points $\chi^2_\alpha(\nu)$ satisfying $P[X > \chi^2_\alpha(\nu)] = \alpha$ for $X \sim \chi^2(\nu)$. These χ^2 critical points are tabulated in many statistical sources but are also conveniently available via computer. To find $\chi^2_\alpha(\nu)$ in **R**, use

```
> qchisq( alpha, df, lower.tail=FALSE )
```

Figure 2.6 illustrates these features.

Example 2.3.6 χ^2 distribution. Suppose $X \sim \chi^2(13)$ and it is of interest to calculate $P[X > 20.1520]$. In **R**, this is simply

```
> pchisq( 20.152, df=13, lower.tail=FALSE )
```

producing $P[X > 20.1520] = 0.0915$. Or, to find the upper-5% critical point from a χ^2 distribution with 13 d.f., the **R** function is

```
> qchisq( .05, df=13, lower.tail=FALSE )
```

This yields $\chi^2_{0.05}(13) = 22.3620$. □

By combining the standard normal distribution with the χ^2, another important, heavily used, and historically famous statistical distribution is derived. Suppose a standard normal variate $Z \sim \mathrm{N}(0, 1)$ is independent of a separate χ^2 variate $W \sim \chi^2(\nu)$. Then, the ratio

$$T = \frac{Z}{\sqrt{W/\nu}} \tag{2.41}$$

is distributed as per the p.d.f.

$$f_{\mathrm{T}}(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\nu\pi}}\left(1+\frac{t^2}{\nu}\right)^{-(\nu+1)/2} I_{(-\infty,\infty)}(t)\,,$$

where ν are the d.f. of the p.d.f. and here the fixed constant π is 3.14159265 … This is known as *Student's t-distribution* after the work of W.S. Gosset, who wrote under the pseudonym 'Student' (Student 1908). (Gosset's use of the pseudonym and his larger contribution with the t-distribution has a colorful history; see Zabell (2008).)

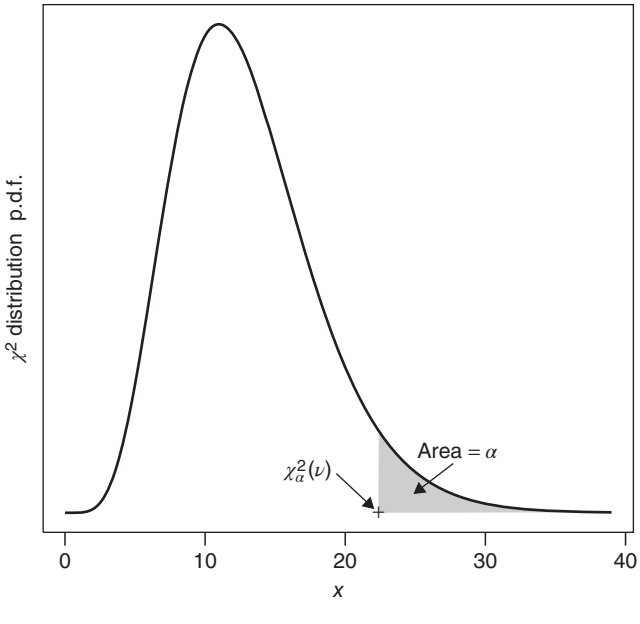

Figure 2.6 χ^2 probability density function (p.d.f.) and upper-α critical point $\chi^2_\alpha(\nu)$.

Reference notation for the t-distribution is $T \sim t(\nu)$. The mean and variance are $E[T] = 0$ (if $\nu > 1$) and $\text{Var}[T] = \nu/(\nu - 2)$ (if $\nu > 2$), respectively. The $t(\nu)$ p.d.f. graphs very similar to the standard normal p.d.f.; both are centered at zero with symmetric, 'bell' shapes. The $t(\nu)$ p.d.f. has heavier tails than the standard normal, however. As $\nu \to \infty$, $t(\nu)$ converges to N(0,1).

Extensive tables exist to give probabilities and/or critical points from $t(\nu)$, although these may also be calculated using computer software. In **R**, t-distribution tail areas are available via the `pt(t, df)` function, where `df` are the pertinent d.f. By default, this gives lower-tail areas $P[T \leq t]$. For upper-tail areas, $P[T > t]$ insert the `lower.tail=FALSE` option or appeal to the Complement Rule (2d) and use

```
> 1 - pt( t, df )
```

Upper-α critical points follow similarly. That is, to find the point $t_\alpha(\nu)$ such that $P[T > t_\alpha(\nu)] = \alpha$, one can use

```
> qt( alpha, df, lower.tail=FALSE )
```

Here, `alpha` is the targeted upper-tail area, `df` are the pertinent d.f., and the `lower.tail=FALSE` option forces **R** to calculate the upper-tail critical point. Figure 2.7 illustrates these features.

Example 2.3.7 ***t*-distribution.** Suppose $T \sim t(29)$ and we wish to find the upper-tail probability $P[T > 1.741] = 1 - P[T \leq 1.741]$. The **R** operation for this is simply

```
> 1 - pt( 1.741, 29 )
```

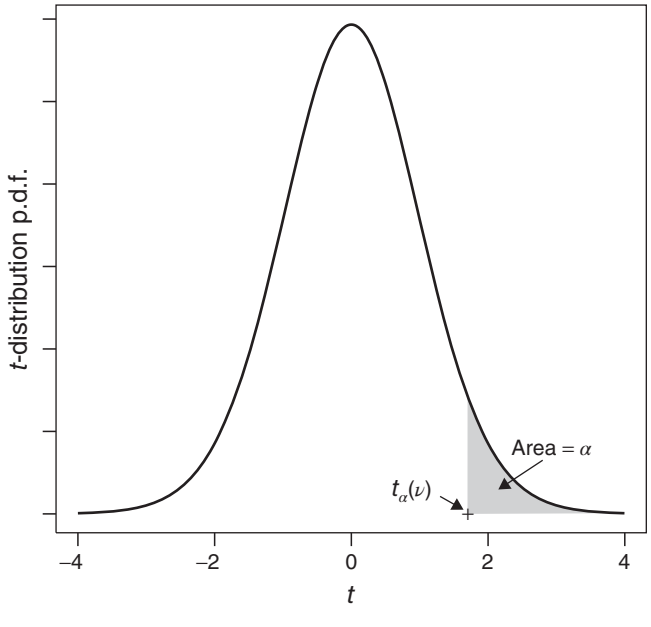

Figure 2.7 t-Distribution probability density function (p.d.f.) and upper-α critical point $t_\alpha(\nu)$.

producing $P[T > 1.741] = 0.0461$.

For upper-α critical points, suppose again $T \sim t(29)$ and now we wish to find its upper-5% critical point. Use

```
> qt( .05, 29, lower.tail=FALSE )
```

to find $t_{0.05}(29) = 1.6991$, that is, $P[T > 1.6991] = 0.05$. □

One final derived distribution important in statistical analytics is the ratio of two (scaled) independent χ^2 variates. That is, suppose the random variable $W_1 \sim \chi^2(\nu_1)$ is independent of $W_2 \sim \chi^2(\nu_2)$. Then we say the ratio

$$F = \frac{(W_1/\nu_1)}{(W_2/\nu_2)} = \frac{\nu_2 W_1}{\nu_1 W_2}$$

has an *F-distribution* with ν_1 and ν_2 d.f. The reference notation is $F \sim F(\nu_1, \nu_2)$.

The mean of an F-distribution is $E[F] = \nu_2/(\nu_2 - 2)$ for $\nu_2 > 2$ and for any $\nu_1 \geq 1$, while the variance is $\text{Var}[F] = 2\nu_2^2(\nu_1 + \nu_2 - 2)/\{\nu_1(\nu_2 - 2)^2(\nu_2 - 4)\}$ for $\nu_2 > 4$. The F-distribution is defined only for positive values and its p.d.f. is skewed to the right.

Note that the ordering of the d.f. is critical here: the d.f. associated with the numerator in the original ratio are listed first (and are sometimes called the *numerator degrees of freedom*), while the d.f. associated with the denominator in the original ratio are listed last, (and are sometimes called the *denominator degrees of freedom*). In fact, if $F \sim F(\nu_1, \nu_2)$, then $1/F \sim F(\nu_2, \nu_1)$; see Exercise 2.24.

Standard tables exist to give probabilities and/or critical points from $F(\nu_1, \nu_2)$, although as above these may be quickly calculated using the computer. In **R**, F-distribution tail areas are available via the `pf(x, df1, df2)` function, where `df1` sets the numerator d.f. and `df2` sets the denominator d.f. As with **R**'s other `p*` functions, this gives the c.d.f. $P[F \leq x]$. So, to find the upper-tail area $P[F > x]$ appeal to the Complement Rule (2d) and employ `1 - pf(x,df1,df2)`. Alternatively, one can force **R** to perform the upper-tail calculation directly via

```
> pf( x, df1, df2, lower.tail=FALSE )
```

Upper-α critical points follow similarly: to find the point $F_\alpha(\nu_1, \nu_2)$ such that $P[F > F_\alpha(\nu_1, \nu_2)] = \alpha$, use

```
> qf( alpha, df1, df2, lower.tail=FALSE )
```

Here, `alpha` is the targeted upper-tail area, `df1` and `df2` are the numerator and denominator d.f., respectively, and `lower.tail=FALSE` forces **R** to calculate the upper-tail critical point. Figure 2.8 illustrates these features.

Example 2.3.8 ***F*-distribution.** Suppose $F \sim F(7, 22)$ and we wish to find the upper-tail probability $P[F > 3.501] = 1 - P[F \leq 3.501]$. The **R** operation for this is simply

```
> pf( 3.501, 7, 22, lower.tail=FALSE )
```

This produces $P[F > 3.501] = 0.0112$.

For upper-α critical points, suppose again $F \sim F(7, 22)$ and now we wish to find its upper-1% critical point. Then,

```
> qf( .01, 7, 22, lower.tail=FALSE )
```

yields $F_{0.01}(7, 22) = 3.5867$, i.e. $P[F > 3.5867] = 0.01$. □

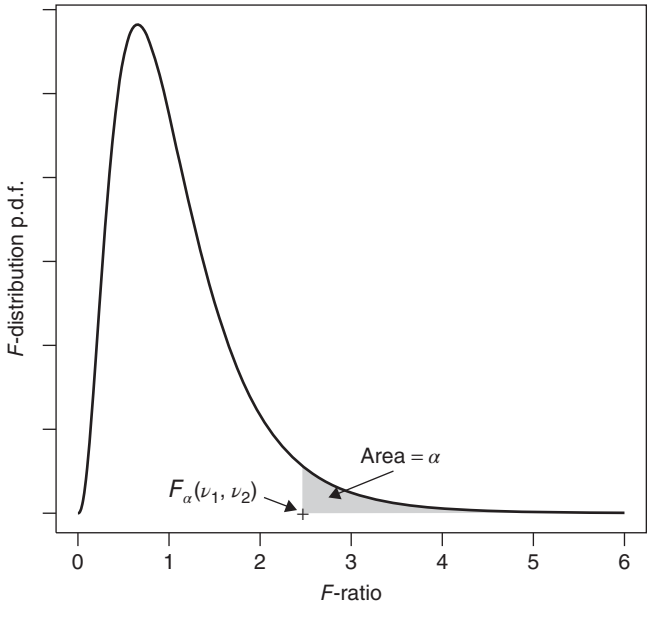

Figure 2.8 F-distribution probability density function (p.d.f.) and upper-α critical point $F_\alpha(\nu_1, \nu_2)$.

As might be expected with these derived distributions, the F, t, and χ^2 random variables are all interrelated. Besides the obvious connections evident in their definitions, one useful result is that for $T \sim t(\nu)$, $T^2 \sim F(1, \nu)$. Or, as $\nu_2 \to \infty$, $(\nu_1 F)$ converges to $\chi^2(\nu_1)$. For other interesting connections, see Leemis (1986) or Casella and Berger (2002, p. 627)

2.3.11 The exponential family

Many of the distributions described in this chapter may be assembled within a single class of probability functions known as the *exponential family of distributions*. This is a rich collection that can accommodate both discrete and continuous probability functions.

The exponential family is characterized by a summary form for the probability function. A random variable belongs to the family if its p.d.f. or p.m.f., $f_X(x)$, may be written as

$$f_X(x) = \exp\left\{\frac{x\theta - b(\theta)}{a(\varphi)} + c(x, \varphi)\right\}, \tag{2.42}$$

where $a(\varphi)$, $b(\theta)$, and $c(x, \varphi)$ are functions of known form. The parameter θ is the (unknown) *natural parameter* of the distribution, and the parameter $\varphi > 0$ is an additional *dispersion parameter* (sometimes alternatively called a *scale parameter*).

An important, additional constraint on the class in (2.42) is that the support space, $\mathcal{S}$, of X cannot depend on θ or, if it is not a known constant, φ. This is usually indicated by incorporating an indicator function (2.20) into $c(x, \varphi)$, such that the indicator does not depend on θ.

Equation (2.42) is actually a special case of a larger family of probability functions of the form

$$f_X(x) = \exp\left\{\frac{t(x)\theta - b(\theta)}{a(\varphi)} + c(x,\varphi)\right\}.$$

When $t(x) = x$, as in (2.42), we say the function is in *canonical form*. Also, for many models the function $a(\varphi)$ simplifies to $a(\varphi) = \varphi/w$, where $w > 0$ is a known constant.

In (2.42), the mean $\mu = E[X]$ is related to the natural parameter θ via the partial derivative $\mu = \partial b(\theta)/\partial\theta$. Similarly, the variance is expressible in terms of θ and φ:

$$\text{Var}[X] = a(\varphi)\,\frac{\partial^2 b(\theta)}{\partial\theta^2}.$$

When the dispersion parameter φ is known, the quantity $\partial^2 b(\theta)/\partial\theta^2$ is called the *variance function* of X, because it incorporates all the unknown aspects of the variance term. With this, we write

$$V(\mu) = \frac{\partial^2 b(\theta)}{\partial\theta^2} = \frac{\partial}{\partial\theta}\left(\frac{\partial b(\theta)}{\partial\theta}\right) = \frac{\partial\mu}{\partial\theta}$$

to highlight that the variance can be a function of the mean for an exponential family p.m.f. or p.d.f. Here are a few examples.

Example 2.3.9 Exponential family: Normal distribution. Given its central position in the pantheon of statistical distributions, it is natural to ask: is the normal p.d.f. from (2.34) a member of the exponential family in (2.42)? The answer is yes. To see how, write the p.d.f. as

$$\begin{aligned} f_X(x) &= \frac{1}{\sigma\sqrt{2\pi}}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} I_{(-\infty,\infty)}(x) \\ &= \exp\left\{-\frac{\log(2\pi\sigma^2)}{2} + \log[I_{(-\infty,\infty)}(x)]\right\}\exp\left\{-\frac{x^2 - 2x\mu + \mu^2}{2\sigma^2}\right\} \\ &= \exp\left\{\frac{x\mu - \frac{1}{2}\mu^2}{\sigma^2} - \frac{x^2}{2\sigma^2} - \frac{\log(2\pi\sigma^2)}{2} + \log[I_{(-\infty,\infty)}(x)]\right\}. \end{aligned}$$

Decomposed into this form, the natural parameter is $\theta = \mu$ and the dispersion parameter is $\varphi = \sigma^2$ so that $a(\varphi) = \varphi$, $b(\theta) = \frac{1}{2}\theta^2$, and

$$c(x,\varphi) = -\frac{1}{2}[\varphi^{-1}x^2 + \log(2\pi\varphi)] + \log[I_{(-\infty,\infty)}(x)].$$

Hence, the normal p.d.f. satisfies the class requirement given by (2.42).

Notice here that $E[X] = b'(\theta) = \theta = \mu$ and $\text{Var}[X] = a(\varphi)b''(\theta) = (\varphi)(1) = \sigma^2$, as expected. □

In Example 2.3.9, the indicator function describing the support space for $X \sim \text{N}(\mu, \sigma^2)$ was $I_{(-\infty,\infty)}(x)$ and was used as an explicit component in the function $c(x,\varphi)$. This was crucial for identifying the normal p.d.f. as an exponential family form. More generally, when the indicator function is used to write the p.m.f. or p.d.f. in its fully expressed form, one can check quickly whether the support of the p.m.f. or p.d.f. is dependent on any unknown parameters.

If it is, then as noted above X cannot be part of the exponential family. For example, suppose $X \sim U(0,\theta)$ for $\theta > 0$. Then the p.d.f. is $f_X(x) = \theta^{-1} I_{(0,\theta)}(x)$ and this cannot be written to satisfy (2.42). This uniform distribution – indeed, *any* uniform distribution with an unknown lower or upper limit – is not a member of the exponential family of distributions.

Example 2.3.10 Exponential family: Binomial distribution. The exponential family is not restricted to p.d.f.s. Consider the binomial model from Section 2.3.1. The p.m.f. is

$$\begin{aligned} f_X(x) &= \binom{N}{x} \pi^x (1-\pi)^{N-x} I_{\{0,1,\ldots,N\}}(x) \\ &= \exp\left\{ x\log(\pi) + (N-x)\log(1-\pi) + \log\left[\binom{N}{x} I_{\{0,1,\ldots,N\}}(x) \right] \right\} \\ &= \exp\left\{ x\log\left(\frac{\pi}{1-\pi}\right) + N\log(1-\pi) + \log\left[\binom{N}{x} I_{\{0,1,\ldots,N\}}(x) \right] \right\}. \end{aligned}$$

Decomposed into this form, the binomial's natural parameter is $\theta = \log\{\pi/(1-\pi)\} = \text{logit}(\pi)$ and the dispersion parameter is (trivially) fixed at $\varphi = 1$. Then, $f_X(x)$ does satisfy (2.42), with $a(\varphi) = 1$, $b(\theta) = -N\log(1-\pi) = N\log(1+\mathrm{e}^{\theta})$, and

$$c(x,1) = \log\left[\binom{N}{x} I_{\{0,1,\ldots,N\}}(x) \right].$$

Hence, the binomial p.m.f. is a member of the exponential family.

A technical caveat: when x is in the set $\mathcal{S} = \{0, 1, \ldots, N\}$, the indicator function $I_{\{0,1,\ldots,N\}}(x)$ equals 1 and the function $c(x, 1)$ in the binomial decomposition is well defined. When x is not in this support set, however, the indicator function is 0, and thus the function $c(x, 1)$ here attempts to evaluate the natural logarithm of zero. Although this is technically impossible, we can appeal to a limiting argument for the evaluation: recognize that as its argument approaches 0, the natural logarithm approaches $-\infty$. Evaluated in the exponent of the p.m.f., this drives $f_X(x)$ to an infinitesimal value, the limiting value of which is itself 0. This is precisely what the probability mass should be when x is not in the support set. □

The exponential family plays a central role in many statistical calculations, and this brief introduction only scratches its surface. For further explorations into the family, including ways to extend it for more complex analytic operations, see Brown (1986) or Casella and Berger (2002, Section 3.4).

Exercises

2.1 Describe the sample space for the following settings:

(a) Record the number of items purchased during a trip to a grocery store by male consumers.

(b) Observe thickness (in mm) of eggshells for a certain bird species exposed to a pesticide.

(c) Sample the annual wages paid to employees in a chain of convenience stores.

(d) Record the longitude and latitude where mobile phone calls are initiated.

2.2 Identify if the following random variables are discrete or continuous:

(a) Number of items purchased during a trip to a grocery store by a male consumer.

(b) Thickness (in mm) of eggshells for a bird exposed to a pesticide.

(c) Annual wages paid to an employee in a retail store.

(d) Blood concentration (in mmol/L) of glucose in a diabetic hospital patient.

(e) Whether or not a consumer with a certain credit score is awarded a loan.

(f) Time for an electronic component to recover full operating capacity after exposure to cold.

(g) Initial longitude and latitude of a mobile phone signal.

2.3 Are the following functions valid p.m.f.s? Explain why or why not.

(a)

m	1	1.3	1.9	2.1
$f(m)$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{4}$

(b)

m	-1	1	4	5	9
$f(m)$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{4}$

(c)

m	1	3	6	7	13
$f(m)$	0.2	0.5	-0.3	0.4	0.2

(d) For some positive integer N, $f(m) = \dfrac{6m^2}{N(N+1)(2N+1)} I_{\{1,2,\ldots,N\}}(m)$.

(e) For some probability $\pi \in (0,1)$,

$$f(m) = \frac{-\pi^m}{m \log(1-\pi)} I_{\{1,2,\ldots,\infty\}}(m).$$

2.4 Are the following functions valid p.d.f.s? Explain why or why not.

(a) $f(x) = \frac{1}{x} I_{(1,\infty)}(x)$.

(b) $f(x) = \frac{1}{2}(x^3 - 1) I_{(0,2)}(x)$.

(c) For some positive constant $\alpha > 0$,

$$f(x) = \alpha(\alpha+1)x^{\alpha-1}(1-x) I_{(0,1)}(x).$$

(d) For two constants $\omega > \delta > 0$,

$$f(x) = \frac{2}{\delta\omega} x\, I_{(0,\delta]}(x) + \frac{2}{\omega(\omega-\delta)}(\omega - x) I_{(\delta,\omega)}(x).$$

(*Hint*: Start by plotting the function for some valid pairings of δ and ω.)

2.5 Verify the simple form of Bayes' rule in (2.1) by recognizing that $P[\mathcal{B} \text{ and } \mathcal{E}] = P[\mathcal{E} \text{ and } \mathcal{B}]$ and then applying the Multiplication Rule in (2c) to each side of this equality.

2.6 What happens to the conditional p.m.f. of $X|Y$ when X and Y are independent?

2.7 Suppose a and b are constants that exhibit no random variation and X is a random variable with p.d.f. $f_X(x)$. Verify the following indications from Section 2.1.3:

(a) $E[a] = a$.

(b) $E[bX] = bE[X]$.

(c) $E[a + bX] = a + bE[X]$.

2.8 Suppose X is a random variable with p.d.f. $f_X(x)$, finite population mean μ_X, and finite population variance σ_X^2.

(a) Verify the indication from Section 2.1.3 that the variance of X can be written as $\sigma_X^2 = E[X^2] - \mu_X^2$.

(b) Show that the second moment of X can be written as $E[X^2] = \mu_X^2 + \sigma_X^2$.

2.9 Suppose a and b are constants that exhibit no random variation and X is a random variable with finite mean μ_X and finite variance σ_X^2. Verify the following indications from Section 2.1.3:

(a) $\text{Var}[a] = 0$.

(b) $\text{Var}[bX] = b^2\text{Var}[X]$.

(c) Use these results to find $\text{Var}[a + bX]$.

2.10 Return to the six-sided die roll in Example 2.1.4 and calculate the variance of X directly. Compare this with that calculated from the expression for the variance of a discrete uniform random variable in Section 2.3.5.

2.11 Show that if X and Y are two independent random variables, then $\text{Cov}[X, Y] = 0$.

2.12 Show that the correlation coefficient, ρ, defined in Section 2.1.5 is contained in the interval $-1 \leq \rho \leq 1$ via the following steps (Casella and Berger 2002):

(a) Let X and Y be two random variables with means μ_X and μ_Y, variances σ_X^2 and σ_Y^2, respectively, and covariance σ_{XY}. Define the function

$$c(b) = E[\{b(X - \mu_X) + (Y - \mu_Y)\}^2]$$

and show that $c(b)$ can be expanded into $c(b) = b^2\sigma_X^2 + 2b\sigma_{XY} + \sigma_Y^2$.

(b) Show why $c(b) \geq 0$ for all real values of b.

(c) Recognize that $c(b)$ is a quadratic function in b. Show that it has at most one real root, that is, at most one real solution to the equation $c(b) = 0$.

(d) Recall that a quadratic equation with at most one real root must have a nonpositive discriminant. Find the discriminant for $c(b)$ and set this less than or equal to 0.

(e) Manipulate the expression in the preceding step into $-\sigma_X\sigma_Y \leq \sigma_{XY} \leq \sigma_X\sigma_Y$ and show that this is equivalent to $-1 \leq \rho \leq 1$.

2.13 Verify (2.14). (*Hint*: $\text{Var}[X_i + X_j] = E[\{(X_i + X_j) - (\mu_i + \mu_j)\}^2] = E[\{(X_i - \mu_i) + (X_j - \mu_j)\}^2]$. Now expand the square.)

2.14 Let $X \sim \text{Bin}(10, 0.2)$. Use direct calculation, published tables (where available), or a computer to find the following values:

(a) $P[X = 6]$

(b) $P[X \leq 2]$

(c) $P[X \geq 1]$

(d) $P[2 \leq X \leq 6]$

(e) $P[2 < X < 6]$

2.15 Let $X \sim \text{Poisson}(\lambda)$ for the values of λ given in the following. Use direct calculation, published tables (where available), or a computer to find the following values:

(a) Find $P[X = 3]$ for $\lambda = 4.95$

(b) Find $P[X > 0]$ for $\lambda = 4.95$

(c) Find $P[4 < X \leq 11]$ for $\lambda = 13.65$

(d) Find $P[X \geq 8.05]$ for $\lambda = 13.65$

(e) Find $P[X \leq 4]$ for $\lambda = 0.55$

2.16 Suppose $X \sim \text{Geom}(\pi)$ as in Section 2.3.3. Show that X possesses a similar 'memoryless property' as the exponential distribution in (2.31). That is, show that $P[X \geq t | X \geq u] = P[X \geq t - u]$ for any positive integers t and u such that $t > u$.

2.17 Let $Z \sim \text{N}(0, 1)$. Use published tables (if available) or a computer to find the following values:

(a) $P[Z \leq 2.63]$

(b) $P[Z > 2.63]$

(c) $P[\,|Z| \leq 2.63]$

(d) $P[\,|Z| \geq 2.63]$

(e) $z_{0.025}$

(f) $z_{0.05}$

(g) $z_{0.005}$

(h) $z_{0.01}$

2.18 Let $X \sim \text{N}(\mu, \sigma^2)$. Use published tables (if available) or a computer to find the following values:

(a) Find $P[X \leq 11.82]$ for $\mu = 1.3$ and $\sigma^2 = 16$

(b) Find $P[X > 5.39]$ for $\mu = -2.5$ and $\sigma^2 = 9$

(c) Find $P[\,|X| \leq 11.82]$ for $\mu = 1.3$ and $\sigma^2 = 16$

(d) Find $P[\,|X| \geq 5.39]$ for $\mu = -2.5$ and $\sigma^2 = 9$

2.19 The Poisson distribution's closure under addition allows for some useful probability calculations. Recall the microprocessor reliability problem in Example 2.3.3. In practice, it might be acceptable for as many as two chips to fail in every 100 000 h of use.

Suppose an i.i.d. sample of $n = 100$ chips is taken to study their actual failure occurrences and the sample mean $\overline{X}$ is calculated using (2.38).

(a) Determine the probability that the mean of the sample exceeds two failures, that is, find $P[\overline{X} > 2]$. (*Hint*: recognize that $P[\overline{X} > a] = P\left[\sum_{i=1}^{n} X_i > na\right]$, and use the closure of the Poisson under addition to find the distribution of $\sum_{i=1}^{n} X_i$.)

(b) Appeal to the CLT in Section 2.3.9 to approximate $P[\overline{X} > 2]$. (*Hint*: here, the population mean of X is $\lambda = 2.7$ and the population variance is also $\lambda = 2.7$.)

2.20 Find the entropy $H(f_X)$ from (2.7) under the following distributions.

(a) $X \sim \text{Bin}(1, \pi)$ (use $\log_2$ in place of the natural logarithm).

(b) $X \sim U(0,\theta)$ and in particular $X \sim U(0, 1)$.

(c) $X \sim \text{Exp}(\beta)$. Plot $H(f_X)$ as a function of $\beta > 0$.

(d) $X \sim \text{N}(0, \sigma^2)$. Plot $H(f_X)$ as a function of $\sigma > 0$.

2.21 Let $X \sim \chi^2(\nu)$ for the values of ν given in the following. Use published tables (if available) or a computer to find the following quantities:

(a) Find $P[X > 16.38]$ if $\nu = 8$

(b) Find $P[X \leq 1.8]$ if $\nu = 10$

(c) Find $P[X > 17.1]$ if $\nu = 10$

(d) Find $P[1.8 \leq X \leq 17.1]$ if $\nu = 10$

(e) Find $\chi^2_{0.01}(\nu)$ if $\nu = 5$

(f) Find $\chi^2_{0.05}(\nu)$ if $\nu = 5$

(g) Find $\chi^2_{0.05}(\nu)$ if $\nu = 15$

(h) Find $\chi^2_{0.05}(\nu)$ if $\nu = 25$

2.22 Let $T \sim t(\nu)$ for the values of ν given in the following. Use published tables (if available) or a computer to find the following quantities:

(a) Find $P[T \leq 2.63]$ for $\nu = 4$

(b) Find $P[T > 2.63]$ for $\nu = 4$

(c) Find $P[\ |T| \leq 2.63]$ for $\nu = 13$

(d) Find $P[\ |T| \geq 2.63]$ for $\nu = 13$

(e) Find $t_{0.025}(\nu)$ for $\nu = 4$

(f) Find $t_{0.05}(\nu)$ for $\nu = 4$

(g) Find $t_{0.05}(\nu)$ for $\nu = 11$

(h) Find $t_{0.05}(\nu)$ for $\nu = 33$

(i) Find $t_{0.05}(\nu)$ for $\nu = 88$

2.23 If you only had access to a table or computer program of F-distribution critical points, how could you use it to find $t_\alpha(\nu)$?

2.24 Let $F \sim F(\nu_1, \nu_2)$.

(a) Show that $1/F \sim F(\nu_2, \nu_1)$.

(b) Show that $F_\alpha(\nu_1, \nu_2) = 1/F_{1-\alpha}(\nu_2, \nu_1)$.

2.25 Let $F \sim F(\nu_1, \nu_2)$ for the values of ν given in the following. Use published tables (if available) or a computer to find the following quantities:

(a) Find $P[F \leq 1.9]$ if $\nu_1 = 13$, $\nu_2 = 28$

(b) Find $P[F > 3.4]$ if $\nu_1 = 21$, $\nu_2 = 9$

(c) Find $P[F \geq 6.2]$ if $\nu_1 = 1$, $\nu_2 = 4$

(d) Find $F_{0.02}(1, 4)$

(e) Find $F_{0.05}(1, 4)$

(f) Find $F_{0.05}(8, 7)$

(g) Find $F_{0.01}(3, 49)$

2.26 Show that the Poisson p.m.f. in (2.21) is a member of the exponential family in (2.42).

2.27 Return to the negative binomial p.m.f. in Section 2.3.4.

(a) For the standard parameterization in (2.24), set $r = 4$. Show that the corresponding p.m.f. is a member of the exponential family in (2.42).

(b) For the standard parameterization in (2.24), assume the parameter r is any known, positive integer. Show that the corresponding p.m.f. is a member of the exponential family in (2.42).

(c) What does the result in Exercise 2.27b tell you about the Geometric p.m.f. and how it relates to the exponential family in (2.42)?

(d) For the redefined parameterization in (2.25), set $\delta = 2$. Is the corresponding p.m.f. a member of the exponential family in (2.42)?

2.28 It is important to emphasize the difference between the univariate exponential distribution from Section 2.3.7 and the larger exponential family in Section 2.3.11. The former is a standalone model for a particular continuous p.d.f., while the latter is an entire class of distributions. Nonetheless, the exponential p.d.f. in (2.28) *is* a member of the exponential family in (2.42). Prove this.

2.29 Suppose a continuous random variable X has the following p.d.f.

$$f_X(x) = \frac{\beta \gamma^\beta}{x^{\beta+1}} I_{(\gamma,\infty)}(x)$$

for $\beta > 0$ and $\gamma > 0$. This is known as the *Pareto distribution*.

(a) Find $E[X]$.

(b) Find $\text{Var}[X]$.

(c) Assume γ is a known positive value. Show that the corresponding p.m.f. is a member of the exponential family in (2.42).

3

Data manipulation

The probability theory described in Chapter 2 lies at the core of any statistical calculation. It is, however, only a preliminary step in conducting a data-analytic exercise. In this chapter, a brief introduction is given to basic data manipulation. The goal is to provide (and/or review) the fundamental building blocks of statistical summarization. As previously, readers familiar with these concepts may wish to skip forward to Chapter 4 and its introduction to basics of data visualization, or on to Chapter 5 and its discussion of the more-advanced aspects of statistical inference.

3.1 Random sampling

As seen with the various statistical distributions in Section 2.3, a random variable X is typically characterized in terms of one or more *parameters*. At its basic level, a parameter is a quantity that describes a critical feature of a random variable. For example, the normal distribution in Section 2.3.9 has two parameters: the population mean μ and the population variance σ^2. These are sufficient to completely characterize the normal probability density function (p.d.f).

In most data-analytic settings, the parameters are unknown and must be estimated. To do so, we take a *random sample* of observations, $X_1, X_2, \ldots, X_n$, from the population. The *sample size*, n, is usually known and fixed in advance. In the simplest case, the observations are all taken from the same 'identical' distribution and are statistically independent of each other. Standard notation for this is $X_i \sim$ i.i.d. $f_X(x)$, where $f_X(x)$ is the probability mass function (p.m.f.) or p.d.f. of X. (As in Section 2.3.9, 'i.i.d.' is shorthand for 'independent, identically distributed'.) If the specific p.m.f. or p.d.f. is from one of the families in Section 2.3, then '$f_X(x)$' is replaced by the further shorthand notation for that distribution; for example, i.i.d. sampling from a normal distribution is indicated by $X_i \sim$ i.i.d. $\mathrm{N}(\mu, \sigma^2)$, $i = 1, \ldots, n$.

A fundamental requirement of any random sample is that it be representative of the population under study. For instance, if a data mining study on skin cancer randomly samples patient records in southern California, would that sample necessarily be representative of cancer patients throughout the United States? Perhaps not. Subjects in California often experience

Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery,
First Edition. Walter W. Piegorsch.

www.wiley.com/go/piegorsch/data_analytics

increased sun exposure; the geographically constrained sampling may provide data on skin cancer of little value to oncologists in, say, central Alaska.

Left unrecognized, distorted or haphazard sampling can introduce severe biases into the data, restricting the scope of the corresponding statistical inferences. In order to avoid systematic distortion in the sampling process, the population must be sampled in a random, unbiased manner. The primary focus in this chapter is on the i.i.d. case, often referred to as *simple random sampling* or a *simple random sample* (SRS). An SRS recruits subjects or units from a larger population such that each subject has an equal chance of being selected. Also, under the independence criterion, we require each sampled observation to have no impact or influence on any other sampled observation.

Wherever possible, the SRS must avoid (or find some way to account for) the impact of peripheral or superfluous factors and should include blinding to avoid potential investigator bias. Unfortunately, even with maximal care and effort, control of every possible extraneous factor cannot always be achieved. To balance out uncontrolled systematic effects in a sample, we employ *randomization*, the random assignment of subjects or units to the sample, before any distinguishing conditions or treatments are applied (Fisher 1926). The concept is straightforward: before imposition of some treatment or intervention, the subjects are allocated so that each subject has the same chance of being assigned to each treatment level. This process is standard in designed experiments, such as laboratory experiments or prospective clinical studies, but may be more difficult to implement fully if the study is observational in nature.

Example 3.1.1 Advertising study. Consumer interest in new-product advertising is often assessed by evaluating panels of subjects' interest in different advertising-campaign designs. In a controlled experiment, the subjects are asked to view one of $T > 1$ increasingly complex designs planned for the advertising program.

Assume that, due to an oversight, the subjects are *not* randomized in their assignment to the advertising designs. Suppose that the first 10 selected subjects are assigned to the current, existing design (a form of *control group*); the next 10 subjects are assigned to the simplest new design, the next 10 to the next more-complex design, and so on until the last 10 subjects are assigned to the most complex design. Now, suppose further that younger subjects are selected first and, unbeknownst to the investigator, younger subjects tend to favor the newer, more-complex advertisements. As a result of this naïve allocation, the youngest subjects will assess the least-complex designs, slightly older subjects will assess designs of intermediate complexity, and this will continue in a systematic manner until the older subjects assess the most complex designs. The likely result of this assignment is that older subjects will respond less favorably to the newer material and vice versa. This can grossly underestimate the advertising's impact on the population at large.

Clearly, such a nonrandom allocation can produce misleading inferences. A better scheme would employ complete randomization, where subjects are assigned randomly to the various designs, in order to vary the subjects' particular advertising exposures in a random, less-systematic manner. □

In general, to assign subjects or units randomly to T different treatment levels, one must use a random device such as a random number generator. From this, one assigns labels or identifiers to each subject and uses these labels to assign each subject randomly to a treatment group. Random number generators are common in most statistical packages and programs

(one can also find hard-copy random number tables in many older statistics textbooks). For example, in **R**, the `runif(n, min=a, max=b)` command produces `n` uniform random numbers in the interval from `a` to `b`. The default is `a=0` and `b=1`, producing random numbers from $U(0, 1)$. (Technically, these are *pseudorandom* numbers, because the proffered random variates are computed via a deterministic algorithm. Modern algorithms employ very clever generators, however; when properly applied, they can mimic true randomness quite effectively (Gentle 2003, Chapter 1).]

Many other strategies can be applied for forming a random sample. As mentioned in Chapter 1, some approaches can stratify or otherwise adapt the sampling to meet highly specific target needs; see Thompson (2012) or Lohr (2010) for greater detail.

3.2 Data types

The types of data available for mining and informatic study are as varied as the underlying distributions from which they are generated; perhaps even more so. The fundamental feature of a single data point distinguishes whether or not it is a number: a random observation is *quantitative* if it represents a true number, amount, or other quantitative characteristic. Data that are not quantitative are called *qualitative*; these often represent categories of some status, such as 'healthy versus 'diseased', or 'accepted'/'pending'/'declined,' and so on. Of course, one can always quantify a qualitative outcome when necessary, for example, use a coding such as 'accepted' = 1, 'pending' = 2, and 'declined' = 3. These will still be arbitrary labels, however, if the quantification does not represent a true numerical separation between the values. For example, if 'pending' = 2 and 'declined' = 3, does 'declined' actually represent one additional unit of measure (or, for that matter, 50% more) above 'pending'? Likely not; indeed, the numerical labels could be reversed here with no loss or disruption of information. It is important to keep this distinguishing feature in mind when operating with qualitative data: just because a datum is given as a number does not always mean it imparts quantitative information.

A data point is called *nominal* if it describes basic categories and essentially nothing more. Both qualitative examples in the previous paragraph represent nominal data. In fact, truly quantitative data are not typically nominal, because the quantitative feature imparts additional information (see the following text). A quantitative, nominal variate is usually a qualitative observation that has been coded with an arbitrary numbering scheme, such as the 'accepted'/'pending'/'declined' illustrated earlier.

A refinement to nominal data occurs when the outcomes occupy some sort of ordered scale. This is called *ordinal* data. Ordinal data are common when a complete quantification is not possible, but a natural ordering nonetheless exists among the outcomes. For instance, a cancer study may classify patients over progressing stages of a disease, from 'healthy' to 'mild,' though 'moderate,' to 'severe' (with possible substages in between). Or, questionnaires often employ the famous Likert Scale (Likert 1932) on respondent attitudes for a statement or product: for example, 1 = 'completely dissatisfied' to 5 = 'completely satisfied', and so on.

One way of distinguishing ordinal data from more-intricate numerical values is that ordinal outcomes cannot be referenced to any unambiguous 'absolute zero' point.

Qualitative categorical data on an ordered scale are often quantified into ordinal quantitative values. In some cases, however, the ordinal data can also originate as quantitative values with no underlying categories other than the actual numbers themselves. In either

case, however, assigning more than an ordinal interpretation to a quantitative ordinal variate is inappropriate.

When ordinal data exist on a scale where the degree of difference between them is meaningful, they are called *interval data*. A classic example is the Fahrenheit temperature scale: the one-unit difference between 10° and 11°F is interpreted identically as that between 90 and 91°F, so a unit difference has meaning. There remains no unambiguous 'absolute zero' point, however: interval data represent only a 'next step' in the numerical progression. (There is such a thing as 0°F, but this is just an arbitrary point on that particular scale. Note that this also holds true for the Celsius temperature scale. There too, 0°C is essentially arbitrary: while it is defined as the freezing point of water at standard atmospheric pressure, any other 'freezing point' could instead have been employed to define the scale.)

When data are observed as continuous, quantitative measurements on an interval scale where an unambiguous zero point exists, they are known as *ratio data*. Technically, a ratio datum indicates the degree of difference between the recorded value and a unit value of the same measure. So, for example, 4 g of mass is twice as much as 2 g of mass. Indeed, most physical measurements such as length, energy, and elapsed time are ratio data. One can even measure temperature on a ratio scale: use the Kelvin scale where absolute zero is really absolute zero. Here, for example, 20°K is truly twice as 'hot' as 10°K, in contrast to Fahrenheit or Celsius.

For quantitative data, an alternative and more-critical delineation distinguishes between discrete and continuous variables, mimicking the characterization with probability functions in Section 2.1.2. Indeed, sampling from a discrete p.m.f. produces a discrete observation, while sampling from a continuous p.d.f. produces a continuous observation. One can also mix the two features when sampling bivariate or multivariate data; for example, a subject in a pharmaceutical study can simultaneously provide the continuous measurements $U = \{\text{blood concentration of the drug}\}$ and $V = \{\text{age}\}$ along with the discrete indicators $W = \{\text{disease status}\}$ and $X = \{\text{sex}\}$. As might be imagined, instances of mixed data are not uncommon in informatic studies. In the following sections, focus will be on calculations and manipulations with quantitative data, although some operations with qualitative data will also be mentioned when pertinent.

3.3 Data summarization

Just as it is useful to construct measures that summarize features of a population's distribution, as in Section 2.1.3, it is also useful in data analytics to construct measures that summarize the data from a random sample. Indeed, many of these measures are sample analogs of the population quantities seen in Chapter 2. Throughout this section, assume that a random sample of data is taken as $X_i \sim \text{i.i.d. } f_X(x),\ i = 1, \ldots, n$, from some p.m.f. or p.d.f. $f_X(x)$.

3.3.1 Means, medians, and central tendency

Most readers will be familiar with the simple arithmetic average of a set of numbers,

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i. \tag{3.1}$$

Table 3.1 A selection (smallest and largest values) of wheat kernel length (in mm) from a larger data set of $n = 210$ observations.

4.899	4.902	4.981	···	6.581	6.666	6.675

Source: Charytanowicz et al. (2010).

and indeed, this was introduced as the *sample mean* in Section 2.3.9 when discussing the distribution of an average of normal random variables. The sample mean is useful for far more than normal random samples, however, because it is a sample analog of the population mean $\mu = E[X]$. It is a natural and intuitive way to measure the central value or *central tendency* of most any random sample.

Example 3.3.1 Wheat kernels. In an agricultural study of wheat grain characteristics (Charytanowicz et al. 2010), data were taken on a number of grain measurements, including $X = \{\text{kernel length in mm}\}$. A sample of $n = 210$ wheat kernels produced the data in Table 3.1. (Owing to the size of the data set, a selection of only the smallest and largest measurements is given in the table. The complete data are available at http://archive.ics.uci.edu/ml/datasets/seeds.) To find the sample mean of the $n = 210$ observations, add up the total and divide by n. Here, this is $\sum_{i=1}^{n} X_i/n = 1181.992/210 = 5.6285$ mm. For very large data sets, the calculation can become tedious, however, so we often turn to the computer. In the **R** statistical language, the sample mean is available via a number of functions. The simplest of these is `mean(x)`, where the single set of n observations is collected into the sample vector `x`. Applying this to the vector `klength` containing all 210 observations from this data set corroborates the calculation above:

```
> mean( klength )
 [1] 5.628533
```

One can also verify the individual components of the calculation: total/(sample size). In **R**, this is simply `sum(x)/length(x)`, where the `sum(x)` function gives $\sum_{i=1}^{n} X_i$ and the `length(x)` function gives the number of elements in `x`.

```
> sum( klength )
 [1] 1181.992

> length( klength )
 [1] 210

> sum( klength )/length( klength )
 [1] 5.628533
```

□

The mean $\overline{X}$ can be extended into a *weighted average* of the sample observations. For a prespecified set of weights $w_i \geq 0$, let

$$\overline{X}_w = \frac{\sum_{i=1}^{n} w_i X_i}{\sum_{i=1}^{n} w_i}. \tag{3.2}$$

If the weights are normed such that $\sum_{i=1}^{n} w_i = 1$, this is similar to the linear combination in Example 2.2.1. One recovers the sample mean in (3.1) when $w_i = 1/n$ for all $i = 1, \ldots, n$. Weighted averages are most useful when the observations have been sampled under some form of heterogeneity, so that certain X_is should receive more 'weight' than others when summarizing central tendency in X.

While it is a standard and accepted way to quantify the center of a data set, the sample mean $\overline{X}$ has a substantial weakness: it can be sensitive to excessively large (or excessively small) values in a random sample. A classic example is with data on household incomes: in the United States, for example, most households report annual income in a range of about \$25 000–\$85 000, but a few households report annual incomes in the millions and even billions of dollars. This is an example of a large right *skew* and is a data analog to the population concept of skew introduced in Section 2.1.2. In that section, a more-resilient (or *robust*) measure of central tendency for skewed distributions was given as the population median Q_2, that is, the point below which and above which at least half of the density or mass rests. An analog for random samples is the *sample median*, denoted as $\hat{Q}_2$. (Statisticians often add a circumflex accent (ˆ) above a parameter to indicate that it is an estimate based on sample observations.) If a sample median lies far to the left (right) of the corresponding sample mean, a large right (left) skew may be evident in the larger sample.

To find the sample median for a set of data, $\{X_1, X_2, \ldots, X_n\}$, first, order the observations from smallest to largest. The collection of all n-ordered observations is called the *order statistics* of the random sample. Specialized notation for this is $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$; that is, parentheses around the index on an observation, $X_{(i)}$, indicate that it is the ith-ordered observation in the sample. For example, the sample minimum is $X_{(1)}$, while the sample maximum is $X_{(n)}$. The sample median is then that order statistic resting between the lower half and the upper half of the sample. If n is odd, this is simply the $\left(\frac{n+1}{2}\right)$th-order statistic. If n is even, mimic the operation with discrete-population medians in Section 2.1.4 and take the sample median as the midpoint between the $\left(\frac{n}{2}\right)$th- and $\left(\frac{n}{2}+1\right)$th-order statistics. That is,

$$\hat{Q}_2 = \begin{cases} X_{([n+1]/2)} & \text{if } n \text{ is even} \\ \frac{1}{2}\{X_{(n/2)} + X_{([n/2]+1)}\} & \text{if } n \text{ is odd.} \end{cases} \tag{3.3}$$

Example 3.3.2 **Wheat kernels (Example 3.3.1, continued).** If an SRS contains only a handful of observations, ordering the values and finding their median via (3.3) is not difficult. The effort can grow in complexity as n gets large, however. Again, the computer facilitates the calculation.

For example, return to the wheat kernel length data in Table 3.1. Direct calculation via (3.3) requires one to order the data and, because $n = 210$ is even, find the $\frac{210}{2} = 105$th- and $\frac{210}{2} + 1 = 106$th-ordered values. To create the order statistics for the observations in the vector `klength` in **R**, use `sort(klength)`. From these, we find $X_{(105)} = 5.52$ and $X_{(106)} = 5.527$, with midpoint $(5.52 + 5.527)/2 = 11.047/2 = 5.5235$.

To compute $\hat{Q}_2$ with a single command, however, the pertinent **R** function is simply `median()`:

```
> median( klength )
 [1] 5.5235
```

Notice that with `median()`, there is no need to order the observations first – the function internally instructs to **R** do it for us!

In any case, we find $\hat{Q}_2 = 5.5235$ mm for these kernel lengths. This compares with the slightly larger value of $\overline{X} = 5.6285$ mm from Example 3.3.1. These features of the data are explored in further examples in the following. □

Another summary statistic, resilient to skew, for measuring central tendency is called a *trimmed mean*. The concept is simple: if the extreme values of the sample are felt to be unstable or unreliable for determining the central tendency of the data, 'trim' the smallest $\frac{k}{2}$ and largest $\frac{k}{2}$ observations from the sample and calculate the mean on the remaining $n - k$ observations:

$$\overline{X}_{T(k)} = \frac{1}{n-k} \sum_{i=\frac{k}{2}+1}^{n-\frac{k}{2}} X_{(i)}. \tag{3.4}$$

(The even trimming constant, k, is specified *a priori*.) Trimming need not be symmetric. If it is known in advance that the data possess greater instability in one tail of the sample, we may wish to trim away a larger fraction from that tail. That is, trim the smallest k_1 values and the largest $k_2 \neq k_1$ values from the sample before calculating the mean (Barnett and Lewis 1995, Section 3.2.1).

$\overline{X}_{T(k)}$ is often referred to in terms of its trimming fraction: the '$100\frac{k}{n}$% trimmed mean.' Preference for this fraction varies: common choices usually rest between 1% and 5% trimming, and up to 10% if the data are rich enough to support that much excision. Of course, when n is small, trimming can remove a nontrivial amount of data; the trimming fraction must be chosen purposefully.

Example 3.3.3 Wheat kernels (Example 3.3.1, continued). Trimming is accomplished in **R** by options in the `mean()` function: `mean(x, trim=)`, where `x` is the data vector and the `trim=` option accepts any proportion up to 50%. The trimming is symmetric.

For the wheat kernel data in Table 3.1, the sample size is large enough to accommodate comfortably a trim of up to 10%. For example, with $k = 10$ in (3.4), we take a 9.5% trimmed mean:

```
> mean( klength, trim=0.095 )
 [1] 5.599884
```

We see that $\overline{X}_{T(10)} = 5.5999$ mm, which is intermediate to both the standard sample mean of $\overline{X} = 5.6285$ mm and the sample median of $\hat{Q}_2 = 5.5235$ mm calculated earlier. Roughly speaking, we find that these wheat kernels average about 5.5–5.6 mm in length. □

A modification to trimming that retains a total of n observations in the sample is known as *Winsorizing*: rather than simply excising the lower and upper $\frac{k}{2}$ observations from the data, *replace* them with the values of the observation closest to those retained in the data. That is, if after k-fold trimming the smallest retained observation is $X_{(\{k/2\}+1)}$ and the largest is $X_{(n-\{k/2\})}$, the *Winsorized mean* is

$$\overline{X}_{W(\{k/2\})} = \frac{kX_{\left(\frac{k}{2}+1\right)} + \sum_{i=k/2+1}^{n-\frac{k}{2}} X_{(i)} + kX_{(n-\{k/2\})}}{n}.$$

The Winsorized mean typically decreases variability in the data and buffers the effects of untoward observations in the tails, while still providing a 'sample' of n observations for summarizing central tendency.

It is important to warn that the entire concept of removing or changing portions of the data should be approached with caution. It is from the information in the data that any conclusions will be drawn, and removing or modifying that information will in some way affect descriptive or inferential quality. Operations such as trimming can improve the stability of an unstable or substandard data and will in some instances improve the eventual information they provide. If performed indiscriminately, however, post-sampling data manipulations can just as often – and, perhaps, more often – detrimentally affect the larger analytic enterprize. One must always undertake such operations with a clear, conscious understanding of their effects and consequences.

3.3.2 Summarizing variation

As with measures of central tendency, measures of variability in a random sample mimic the corresponding population measures in Section 2.1. Thus the *sample variance* is a sort of average squared deviation from the mean, using sample information:

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2, \tag{3.5}$$

where $\overline{X}$ is given by (3.1). The division by $n-1$ in the denominator of S^2 occurs because there are, in effect, only $n-1$ components of quantifiable information in the sum. (See Exercise 3.7.) For operations on the same scale as the original data, use the *sample standard deviation*: the positive square root of the sample variance, $S = \sqrt{S^2}$.

The variance in (3.5) has convenient computing forms

$$S^2 = \frac{\left(\sum_{i=1}^{n} X_i^2\right) - n\overline{X}^2}{n-1} = \frac{\left(\sum_{i=1}^{n} X_i^2\right) - \frac{1}{n}\left(\sum_{i=1}^{n} X_i\right)^2}{n-1}, \tag{3.6}$$

either of which is usually faster to compute than (3.5). In **R**, the sample variance of a vector `x` is available via the `var(x)` function; the corresponding sample standard deviation is `sd(x)`.

Example 3.3.4 Wheat kernels (Example 3.3.1, continued). For the wheat kernel length data in Table 3.1, $\overline{X} = 5.6285$. To compute the sample variance, the calculations necessary for $S^2 = \sum_{i=1}^{210}(X_i - 5.6285)^2/209$ are not trivial. Using **R**, however, we quickly find

```
> var( klength )
 [1] 0.1963052
```

and

```
> sd( klength )
 [1] 0.4430635
```

that is, $S^2 = 0.1963$, with $S = 0.4431$. Direct calculation via (3.6) confirms these values. □

In some settings, it is useful to quantify an observation, X_i, in terms of how it relates to the sample mean, for example, is X_i very much larger than $\overline{X}$ or is it very near but slightly

smaller than $\overline{X}$, and so on. Distance is always relative, however: if variation – as measured by S^2 – is very small, then points very far from $\overline{X}$ represent real deviations, whereas if S^2 is large, then large deviations from $\overline{X}$ may not be as meaningful. To represent this quantitatively, data analysts use what is called a *z-score*. Given an observation X_i from a sample of size n with mean $\overline{X}$ and variance S_X^2, the z-score for X_i is the centered and scaled variate

$$Z_i = \frac{X_i - \overline{X}}{S_X}. \tag{3.7}$$

(Notice that this is a unitless quantity.) In effect, Z_i measures the standard-deviation-scaled distance of X_i from its sample mean.

A z-score near zero indicates that the original observation was very near to $\overline{X}$; a z-score near ± 1 indicates that the observation is one standard deviation away from $\overline{X}$; a z-score near ± 2 indicates that the observation is two standard deviations away; and so on. A general rule-of-thumb with symmetric data concentrated about their mean is that about 68% of a sample's z-scores will lie between ± 1, about 95% of the z-scores will lie between ± 2, and about 99.7% of the z-scores will lie between ± 3. (Correspondingly, about 68% of the sample data will lie between $\overline{X} \pm S$, 95% will lie between $\overline{X} \pm 2S$, and 99.7% will lie between $\overline{X} \pm 3S$.) This is known as the *empirical 68–95–99.7% rule*.

Another approach to estimate variation employs ordered measures similar to the sample median from Section 3.3.1. For example, the *sample quartiles* mimic their population counterparts from Section 2.1.4: the *first (or lower) sample quartile* is the point in the data below which one-quarter of the X_is lie and above which three-quarters of them lie. Denote this as $\hat{Q}_1$. Similarly, the *third (or upper) sample quartile* is the point in the data below which three-quarters of the X_is lie and above which one-quarter of them lie. Call this $\hat{Q}_3$. As might be expected, the median is also known as the *middle quartile*, $\hat{Q}_2$.

Given the sample quartiles, the *sample interquartile range* (*IQR*) is then $\widehat{\text{IQR}} = \hat{Q}_3 - \hat{Q}_1$. This is the sample analog to the population IQR in Section 2.1.4, with comparable interpretation as a measure of spread. Indeed, if $X_i \sim \text{i.i.d. } \text{N}(\mu, \sigma^2),\ i = 1, \ldots, n$, then $\widehat{\text{IQR}}/1.35$ is an alternative estimator for σ.

A useful way to assemble these various statistics together is known as the *five-number summary*. Colloquially, the five-number summary for a set of data, $\{X_1, \ldots, X_n\}$, is given as the collection $\{X_{(1)}, \hat{Q}_1, \hat{Q}_2, \hat{Q}_3, X_{(n)}\}$, that is, the minimum, three quartiles, and maximum of the data. (This has a useful graphical analog known as a 'boxplot,' as seen in Section 4.1.2) From the five-number summary, the sample IQR is easily calculated. A slight modification to this construction replaces the two outer quartiles with the so-called *hinges*: the lower hinge is the median of the lower half of the data, while the upper hinge is the median of the upper half. In some instances, the hinges will equal the outer quartiles, but this is not guaranteed. In **R**, the command `fivenum()` returns a five-number summary made up of the minimum, lower hinge, median, upper hinge, and maximum of the data.

For a more-general pth quantile, $\hat{q}_p$, one can define the *sample (or empirical) quantile function* using the order statistics, although this need not be unique. Perhaps the simplest definition is

$$\hat{q}_p = X_{(i)} \quad \text{if} \quad \frac{i}{n} \leq p < \frac{i+1}{n}\ . \tag{3.8}$$

Ostensibly, the lower and upper sample quartiles are then $\hat{Q}_1 = \hat{q}_{0.25}$ and $\hat{Q}_3 = \hat{q}_{0.75}$, respectively. The definition in (3.8) produces a discontinuous function of p, however, and can be

difficult to work with in practice. Other competitors can be constructed for defining the sample quantiles, most of which use a weighted average of adjacent order statistics. Some even build continuous empirical quantile functions. Thus for the same set of data, one computer program's lower quartile may not match exactly with another's, although they will generally be very close. (Users should read the help or manual pages for any software to be sure that they are calculating the desired quantiles.) **R** includes up to nine different definitions in its `quantile(x, type=)` function, each a slight variant of the others. (Use the `type=` option to change them; `type=7` is the default, giving a continuous quantile function based on an intricate weighted average of consecutive order statistics. `type=1` corresponds to (3.8). See `help(quantile)` in **R** for more information.) A complete survey of quantile estimation exceeds the scope here; interested readers may refer to, for example, Davis and Steinberg (2006).

Example 3.3.5 Wheat kernels (Example 3.3.1, continued). All these summary quantities can be produced by a series of commands in **R**. For instance, the `summary()` command gives the (colloquial) five-number summary along with the mean. For the wheat kernel data in Table 3.1, this is

```
> summary( klength )
   Min. 1st Qu.  Median    Mean  3rd Qu.    Max.
  4.899   5.262   5.524   5.629   5.980   6.675
```

As noted earlier, the `fivenum()` command replaces the outer quartiles with the hinges (and drops the sample mean):

```
> fivenum( klength )
 [1] 4.8990  5.2620  5.5235  5.9800  6.6750
```

The `quantile()` command reproduces the `summary()` output but drops the sample mean:

```
> quantile( klength )   #default is quartiles
      0%     25%     50%     75%    100%
  4.89900 5.26225 5.52350 5.97975 6.67500
```

In its more-general form, `quantile(x, probs=)` gives any pth quantile, via the `probs=` option (the argument of which can even be a vector of probabilities). For instance, the four *sample quintiles* of the data are

```
> quantile( klength, probs=c(0.2, 0.4, 0.6, 0.8) )  #quintiles
     20%     40%     60%     80%
  5.2198 5.3962 5.7014 6.0680
```

Lastly, for the IQR, use the eponymous `IQR()` function or calculate directly from any of these other outputs:

```
> IQR( klength )
 [1] 0.7175

> quantile(klength)[4] - quantile(klength)[2]
 0.7175
```

□

3.3.3 Summarizing (bivariate) correlation

One often collects random samples where bivariate *pairs* of observations are recorded on each subject, say, (X_i, Y_i), $i = 1, \ldots, n$. In this case, interest may center on summarizing the association in evidence between the two outcome variables. A measure of bivariate association useful for such settings is the population correlation coefficient from Equation (2.10). Its sample analog is known as the *Pearson product-moment correlation coefficient*

$$r_{XY} = \frac{S_{XY}}{S_X S_Y} = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2 \sum_{i=1}^{n}(Y_i - \overline{Y})^2}}, \tag{3.9}$$

where the *sample covariance* is

$$S_{XY} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y}).$$

Notice that both r_{XY} and S_{XY} are symmetric measures; for example, $r_{XY} = r_{YX}$.

Numerous computing forms exist for (3.9); these include

$$r_{XY} = \frac{\sum_{i=1}^{n} X_i Y_i - n\overline{X}\,\overline{Y}}{\sqrt{\left(\sum_{i=1}^{n} X_i^2 - n\overline{X}^2\right)\left(\sum_{i=1}^{n} Y_i^2 - n\overline{Y}^2\right)}}$$

and

$$r_{XY} = \frac{1}{n-1}\sum_{i=1}^{n} z_{xi} z_{yi},$$

where $z_{xi} = (X_i - \overline{X})/S_X$ and $z_{yi} = (Y_i - \overline{Y})/S_Y$ are the individual z-scores for each X_i and Y_i in the paired sample ($i = 1, \ldots, n$).

The sample correlation coefficient r_{XY} was developed by Karl Pearson (1896). It is an effective measure for quantifying association between paired observations and has become one of the most widely used (and, not infrequently, misused) summary statistics in data analytics.

As with the population correlation coefficient, the sample version in (3.9) satisfies $-1 \leq r_{XY} \leq 1$, and it measures *linear* association between the paired variables. When $r_{XY} = 0$, the data suggest no apparent relationship between X_i and Y_i, while when $|r_{XY}| = 1$, a perfect linear relationship is in place between them. In the latter case, a plot of Y_i versus X_i would produce a perfectly straight line, with positive slope if $r_{XY} = 1$ and negative slope if $r_{XY} = -1$. (Summary graphics for bivariate relationships are discussed in Section 4.2.)

Example 3.3.6 College admissions. The admissions department of a large US public university studied the association between different measures used to evaluate undergraduate applicants. Among other variables, the admissions officers collected $X = \{$High school class rank (as a percentile; higher percentiles indicate higher rank)$\}$ and $Y = \{$ACT score$\}$ in a

Table 3.2 Selected data pairs (X, Y) with $X = \{\text{High school class rank (\%)}\}$ and $Y = \{\text{ACT score}\}$, from a larger set of $n = 705$ observations.

X = Class rank (%)	61	84	74	95	47	···	97	97	99
Y = ACT score	20	20	19	23	23	···	29	29	32

sample of $n = 705$ students from its incoming freshman class. (The ACT is a standardized test provided by the American College Testing program and used by many US colleges and universities for quantifying academic achievement. The overall score studied here ranges from a minimum of 1 to a maximum of 36. See http://www.act.org.)

The paired data, from Kutner et al. (2005, Appendix C), appear in Table 3.2. (As above, a selection of only the smallest and largest measurements is given in the table. The complete data are available at http://www.wiley.com/go/piegorsch/data_analytics.)

For univariate summary statistics, we find that the average student's class rank percentile is $\overline{X} = 76.9532\%$, while the average ACT score is $\overline{Y} = 24.5433$. Also, $S_X = 18.6339$ and $S_Y = 4.0136$.

Of interest with these data is whether and to what degree the two admissions criteria associate. (If there were very strong association between the two – negative or positive – then perhaps only one of the two criteria would be needed for admissions decisions. On the other hand, if the association is weak, the two measures may provide complementary information.)

For such a large data set, computer calculation is indicated: in **R**, the correlation coefficient from (3.9) is available via the `cor()` function. This takes two required inputs, the names of the two paired variables (order is unimportant). Here, **R** gives

```
> cor( rank, ACT )
 [1] 0.4425075
```

Thus for these 705 students, the sample correlation is $r_{XY} = 0.4425$. This is a moderate level of positive correlation, suggesting that while the two measures do associate – as might be expected – the differences between them are substantial enough to warrant continued, joint consideration. □

As the sample correlation is a measure of linear association, it can provide misleading inferences if the underlying relationship between X and Y is more complex. For example, if the two variables relate in a tight quadratic or other curvilinear manner, r_{XY} may calculate close to zero, but this does not mean they are unrelated. Issues of how to explore and understand bivariate relationships more fully are discussed in Chapter 6.

3.4 Data diagnostics and data transformation

3.4.1 Outlier analysis

As mentioned towards the end of Section 3.3.1, there are occasions when one or more data points in an SRS may not be truly representative of the population from which the sample was drawn. This could be due to some sort of contamination in the sample, an unrecognized bias in the sampling scheme, a recording error, or just an unusual outlier in the data. Indeed, the term 'outlier' is often used for situations such as this, usually to indicate an observation that appears inconsistent with the pattern(s) in the remainder of the data (Barnett and Lewis 1995,

Table 3.3 A selection (smallest and largest values) of average daily net carbohydrate consumption (in grams) from a larger data set of $n = 778$ observations.

43	64	75	75	···	402	407	437	738

Chapter 1). When associated with detection of anomalies in a data stream – for instance, indication of false insurance claims or of fraudulent credit card purchases – outlier identification becomes an important data mining tool (Hand et al. 2001, Section 2.7).

The specification of what constitutes an 'outlier' can be made more formal. We say a potential *outlier* in a random sample $\{X_1, X_2, \ldots, X_n\}$ is any point X_i satisfying

$$X_i < \mathcal{F}_1 \quad \text{or} \quad X_i > \mathcal{F}_3,$$

where

$$\mathcal{F}_1 = \hat{Q}_1 - (1.5)(\widehat{\text{IQR}}) \qquad \text{and} \qquad \mathcal{F}_3 = \hat{Q}_3 + (1.5)(\widehat{\text{IQR}}), \tag{3.10}$$

are called the *inner fences* of the data set. (For 'outer fences,' replace 1.5 with 3.0 in the definition.) Essentially, any data point lying outside the 'fences' defined by extending the outer quartiles by 150% of the IQR distance is a potential outlier (Hoaglin et al. 1986). When so identified, the datum is worthy of closer inspection to determine if it represents an abrogation in the sampling process or other form of discordant observation.

In passing, note that many alternative conditions can be used to define an 'outlier.' Some authors employ the 'outer' fences to define an outlier (then often distinguished as an 'excessive outlier'). Others say an outlier is any point whose z-score from (3.7) exceeds 3.0 (or 4.0, or 6.0, etc.) in absolute value. See Barnett and Lewis (1995) for a comprehensive discussion.

Example 3.4.1 **Carbohydrate intake.** A biomedical research team studied carbohydrate intake in a population of Caucasian males between the ages of 45 and 55. The subjects were participating in a weight reduction/maintainence program. Recorded was the average daily intake of net carbohydrates (total carbohydrates − dietary fiber) in grams, from a simple random sample of $n = 778$ individuals. The data (sanitized to remove any identifying information) appear in Table 3.3. (As above, a selection of only the smallest and largest measurements is given in the table. The complete data are available at http://www.wiley.com/go/piegorsch/data_analytics.)

Assume that the data are collected into the **R** vector `carbs`. For summary statistics, we find

```
> mean( carbs )
 [1] 221.4357

> quantile( carbs )
    0%    25%    50%    75%   100%
 43.00 192.00 215.00 245.75 738.00
```

Clearly, $\overline{X}$ is larger than $\hat{Q}_2$ here, but only by about 3%. So, no gross skew in the data is indicated. Next,

```
> IQR( carbs )
 [1] 53.75
```

```
> quantile(carbs)[2] - 1.5*IQR(carbs)     #lower fence
      25%
  111.375

> quantile(carbs)[4] + 1.5*IQR(carbs)     #upper fence
      75%
  326.375
```

We find that the outer quartiles are $\hat{Q}_1 = 192$ and $\hat{Q}_3 = 245.75$, producing $\widehat{\text{IQR}} = 53.75$ and (inner) fences of $\mathcal{F}_1 = 192 - (1.5)(53.75) = 111.375$ and $\mathcal{F}_3 = 245.75 + (1.5)(53.75) = 326.375$. Current recommendations for average daily carbohydrate intake vary between 180 and 300 g, so any observations lying outside these fences could be considered potential outliers in this data set and deserving of further attention.

Indeed, a number of observations exceed these bounds. (A fast way to proceed in **R** is to use the language's subsetting feature, via `carbs[ carbs<f1 ]` and `carbs[ carbs>f3 ]`.) Ten observations lie below $\mathcal{F}_1$ and 24 lie above $\mathcal{F}_3$. Of these, the most excessive outlier is the largest observation at $X_{(778)} = 738$ g. This is far above the next-largest observation at $X_{(777)} = 437$ g, which itself is of some interest: these are the only two data points lying outside of the 'outer' fences for these data and, hence, may be worthy of closer inspection. □

Once an outlier has been identified, the more-challenging question is, how should it be treated? Automatized removal of any datum simply because it exceeds a numerical outlier standard is poor statistical practice and poor analytics. Careful authors use the term 'potential outlier' for good purpose: a point that seems unusual, for whatever reason, deserves additional study to determine whether and why it disturbs larger patterns in the data. It is this study, however, that should guide further operations with (or without) a questionable data point. For instance, in the carbohydrate intake data from Example 3.4.1, the clearly anomalous observation of 738 g may indicate an individual not suited for the larger weight-loss study, or someone who is younger than the target age group (and thus consumes more in terms of average daily carbohydrates), or even just a data entry error. Or, it may be a perfectly reasonable, if extreme, realization of the process under study and should be retained. Only closer inspection of the actual datum can clarify the matter. (After further inspection of the potential carbohydrate intake outliers in Example 3.4.1, no disqualifying features were identified for those subjects: as far as could be determined they simply consumed larger-than-recommended amounts of carbohydrates, on average, during the data acquisition period.)

3.4.2 Entropy*

If a particular data point (or set of points) has been identified for potential removal from the data set, it may be of interest from an exploratory data analysis (EDA; cf. Section 1.3) perspective to quantify how much or even whether homogeneity in the sample improves after excision. To do so, one can turn to sample diagnostics. For instance, the *sample entropy* is analogous to the population entropy $H(f_X)$ described in Section 2.1.3. Just as $H(f_X)$ quantifies the 'disorder,' or more properly, the dispersion or heterogeneity in a distribution, its sample estimator can be used to quantify the heterogeneity in a set of data. Many different estimators have been proposed for this process (Beirlant et al. 1997); here the focus is on a nonparametric estimator based on lagged separations between the sample order statistics. (The estimator

is 'nonparametric' in that, to construct it, no specific parametric family from Section 2.3 is assumed for the distribution of X.)

Given the order statistics $X_{(1)} \le X_{(2)} \le \cdots \le X_{(n)}$, let $X_{(i+m)} - X_{(i-m)}$ be the differences between the $(i+m)$th- and $(i-m)$th-ordered values for some positive lag parameter $m < \frac{n}{2}$. With these, Vasicek (1976) described the preliminary estimator

$$\hat{H}_V = \frac{1}{n} \sum_{i=1}^{n} \log \left\{ \frac{n}{2m} \left[X_{(i+m)} - X_{(i-m)} \right] \right\}, \tag{3.11}$$

where one truncates to $X_{(j)} = X_{(1)}$ in (3.11) for any index such that $j < 1$ and to $X_{(j)} = X_{(n)}$ for any $j > n$. Vasicek identified a systematic bias with this simple estimator, however, and proposed a corrected version for practical use. Take

$$\hat{H}'_V = \hat{H}_V - \log\left(\frac{n}{2m}\right) - \frac{(n-2m)\psi(2m)}{n} + \psi(n+1) - \frac{2}{n} \sum_{j=1}^{m} \psi(j+m-1), \tag{3.12}$$

where $\psi(t)$ is the *digamma function*

$$\psi(t) = \frac{d \log\{\Gamma(t)\}}{dt} = \frac{\Gamma'(t)}{\Gamma(t)},$$

that is, the derivative of the log of the gamma function in Equation (2.26) (Spouge 1994). For specifying m, Hampel (2008) recommended a default lag of $m = 4$. If many ties exist among the data points, set m to the smallest integer larger than 4 such that $X_{(i+m)} - X_{(i-m)} > 0$ for all i.

Example 3.4.2 Carbohydrate intake (Example 3.4.1, continued). The $\hat{H}'_V$ statistic in (3.12) is complex enough to demand calculation by computer. In **R**, direct programming is necessary, although the internal `digamma(t)` function facilitates the effort. The `sort()` command is also useful, with its ability to order the data and produce the order statistics. The following is a sample **R** function.

```
>  HVprime = function( x , m=4 ) {
                          xo <- sort( x )
                          n <- length( xo )
                          n2m <- n/(2*m)
                  xup <- rep( xo[n],n )
                  xlo <- rep( xo[1],n )
          for ( j in 1:(n-m) ) xup[j] <- xo[j+m]
          for ( j in (1+m):n ) xlo[j] <- xo[j-m]

                  HV <- mean( log(n2m*(xup-xlo)) )
   bias <- log(n2m) + (1-(1/n2m))*digamma( 2*m )
            - digamma( n+1 ) + (2/n)*sum(digamma( (1:m)+m-1) )

       return( HV - bias )  #return HVprime statistic
     }  #end function
```

For the carbohydrate intake data in Table 3.3, numerous ties exist among the interior data points. For any $m \le 6$, this produces cases where $X_{(i+m)} - X_{(i-m)} = 0$ for some values of i. Thus, take $m = 7$. The estimated entropy for all $n = 778$ entries is then found as $\hat{H}'_V = 5.1775$.

Removing the extreme observation at $X_{(778)} = 738$ g drops the sample entropy to $\hat{H}'_V = 5.1565$ (at $m = 7$). This is a small decrease in entropy – less than 1% – suggesting that this single large observation may not substantially affect heterogeneity in the larger sample.

Going further and removing all observations greater than 400 g drops the entropy down to $\hat{H}'_V = 5.1365$ (at $m = 7$). Again, not a large improvement for decreasing sample heterogeneity.

□

Given its information-theoretic origins (cf. Section 2.1.3), the sample entropy can be viewed as an index of the *information* in a sample: as heterogeneity or variation increases, information decreases. Thus in a certain sense, small values of $\hat{H}'_V$ indicate greater information. Another measure of sample information known as the Fisher information number will be discussed in Section 5.1.

3.4.3 Data transformation

In some instances, a data set may not contain any extreme outliers or may have potential outliers that do not detrimentally affect homogeneity of the sample, but it may still exhibit a large skew that hinders summary calculations. If so, another approach for managing the data is to transform the observations into a more operable form. For example, as mentioned in Section 2.1.2, the (natural) logarithmic transform $Y_i = \log(X_i)$ often is used to reduce heavy right skew in positive-valued observations.

In general, a *data transformation* is a mathematical function, $Y_i = g(X_i)$, applied to the original observations in order to achieve a more stable, target characteristic in the transformed data. The log transform is clearly $Y_i = g(X_i) = \log(X_i)$. Other common transformations include

- the square root transform, $g(X_i) = \sqrt{X_i}$ (for $X_i \geq 0$)
- the reciprocal transform, $g(X_i) = 1/X_i$ (for $X_i \neq 0$)
- the *logit* transform, $g(X_i) = \log\left\{\frac{X_i}{100-X_i}\right\}$ (for percentages between 0 and 100)
- the simple arc-sine/square-root transform, $g(X_i, N_i) = \arcsin\sqrt{X_i/N_i}$ (for proportions, X_i/N_i).

Note that an alternative form of the arc-sine/square-root transform for proportions is

$$g(X_i, N_i) = \arcsin\sqrt{\frac{X_i + \frac{3}{8}}{N_i + \frac{3}{4}}}\,.$$

An entire class of transforms that contains many of these individual forms is known as the Box–Cox *power transformation* (Box and Cox 1964)

$$g_\lambda(X_i) = \frac{X_i^\lambda - 1}{\lambda}\,. \tag{3.13}$$

Ignoring constants, the square root and reciprocal transforms obtain at $\lambda = \frac{1}{2}$ and $\lambda = -1$, respectively; the log transform corresponds to the limiting form as $\lambda \to 0$ (Exercise 3.18). Also useful in certain cases are integer powers such as the quadratic and cubic forms: $\lambda = 2$

and $\lambda = 3$ in (3.13). If desired, one can estimate λ directly from the data, although this can produce unintended instabilities in the transformed variates; users should apply caution when estimating a power transformation parameter. See Carroll and Ruppert (1988) for more on this and other issues regarding data transformations.

Example 3.4.3 **Carbohydrate intake (Example 3.4.1, continued).** For the carbohydrate intake data in Table 3.3, a transformation could be useful for attenuating the extremely large observations in the upper tail of the sample. As the data are strictly positive, either the square root or logarithm are likely candidates. Consider the former: take $Y_i = \sqrt{X_i}$. The transformed data corresponding to the displayed values in Table 3.3 become

$$6.56,\ 8.00,\ 8.66,\ 8.66,\ \ldots\ ,\ 20.05,\ 20.17,\ 20.90,\ 27.17.$$

Deceases in the large separation among the upper (and lower) observations is clearly evident. Indeed, the sample entropy is also much smaller with the square-root-transformed Y_is: $\hat{H}'_V = 1.7958$ at $m = 7$.

The logarithmic transform with these data is explored in Exercise 3.16. □

Example 3.4.4 **College admissions (Example 3.3.6, continued).** For the paired College Admissions data in Table 3.2, $X = \{\text{Class rank}\}$ is recorded as a percentage and, therefore, is naturally bounded between 0% and 100%. This can at times lead to certain instabilities in the measure; for example, $\overline{X} = 76.9532$ is rather smaller than the sample median $\hat{Q}_2 = 81$. This suggests a left skew with those data. As noted above, a popular transform with percentage data is the *logit*: $g(X) = \text{logit}(X) = \log\{X/(100 - X)\}$. Applying the logit transform to the class ranks provides substantial attenuation in the skew (try it!). The sample entropy is also much smaller with the logit-transformed X_is than with the original class ranks in X_i: $\hat{H}'_V = 4.087$ for the original values versus $\hat{H}'_V = 1.693$ for the logits, both at $m = 38$ due to the large number of ties.

Recomputing the correlation coefficient between $g(X) = \text{logit}\{\text{Rank}\}$ and $Y = \text{ACT score}$ yields a slightly higher sample correlation than that seen in Example 3.3.6: in **R**, we find

```
> cor( log(class.rank/(100-class.rank)), ACT )
 [1] 0.4596824
```

□

3.5 Simple smoothing techniques

Summarization efforts with very large data sets can at times fall victim to the 'signal-versus-noise' problem: high variability ('noise') can swamp an underlying trend ('signal'), making it difficult to describe or identify pertinent features in the data. Indeed, in the modern age of ever-increasing, 'big' data, noise-to-signal ratios often grow *with* increasing information, as the breadth of data can overwhelm our ability to process underlying connections contained within (Silver 2012). While outlier detection or data transformation may alleviate this concern in select instances, such adjustments may still prove ineffective with very high noise-to-signal ratios. A possible solution then involves so-called *smoothing* of the data, that is, aggregating or averaging locally similar observations to de-noise ('smooth') the data and expose or extract their core features.

3.5.1 Binning

Perhaps the simplest way to smooth a set of data $\{X_1, \ldots, X_n\}$ is to assemble the X_is into a collection of $G > 1$ adjacent groups or 'bins.' By compressing highly noisy data into G bin means, $\overline{X}_g$, bin medians, $\hat{Q}_{2g}$, or frequencies (counts) of observations in each bin, say $\hat{f}_g$, one can often obtain smoother summary information on the phenomenon under study. (Here, $g = 1, \ldots, G$ indexes the individual bins.) Also, some data-analytic methods may only operate on discretized or categorized data, whereby some form of discrete *binning* may be necessary (Myatt 2007, Section 3.4.4). One might also view the binning as an optimization problem, where the data are distributed among the G bins to minimize the average squared (or absolute, etc.) distance between each X_i and its bin mean or median. The algorithm for doing so would be heavily computational, however (Kantardzic 2003, Section 3.5).

A natural concern when aggregating into collective bins is whether a loss of information occurs, due to apparent discretization of the original data. This is possible in some cases, and care must be taken not to 'oversmooth' or otherwise mask the target features of interest (Kuss 2013). At a certain level, smoothing is as much art as it is science.

If the break points between each bin are clearly indicated from the subject matter, then the binning is straightforward. (For example, person-age is often separated into 10-year bins.) If the break points are not prespecified, however, then the simplest and often most-propitious approach allocates the observations equally among the G bins. Thus, for example, if $G = 4$, select breaks at the three sample quartiles $\hat{Q}_j$ ($j = 1,2,3$).

Binning is perhaps most useful when applied to produce frequency counts, $\hat{f}_g$, from univariate data. For example, suppose interest centers on estimating distributional features of the X_is such as their p.d.f. or p.m.f. A familiar graphical device for visualizing a p.d.f. or p.m.f. is known as a *histogram* (described more fully in Section 4.1.4), where the $\hat{f}_g$ values are plotted as a vertical bar chart against the binned categorizations of the X_is. The number of bins, G, and the break points can be selected in many different ways; however, careful construction is required, in order to properly visualize the shape of the distribution. A common default is known as Sturges' rule (Sturges 1926): $G = \lceil \log_2(n) + 1 \rceil$, where $\lceil x \rceil$ is the ceiling operator, that is, the smallest integer greater than or equal to x. The break points are then constructed to give G bins of equal width between the smallest and largest observations, usually using round numbers for simple bin separators.

Sturges' rule works well with data that center near their mean and vary symmetrically around it, that is, unimodal with little skew. An alternative for more general use is given by Scott (1979) and is known as Scott's *normal reference rule*: $G = \lceil n^{1/3} \{X_{(n)} - X_{(1)}\}/(3.49 S_X) \rceil$, where S_X is the sample standard deviation. Note that this assumes a desired range of $X_{(1)} \le X \le X_{(n)}$ for the spread of the bins. If round numbers are used for cleaner bin separators, use $G = \lceil n^{1/3} \{B_U - B_L\}/(3.49 S_X) \rceil$, where B_L is the desired lower boundary of the first bin and B_U is the desired upper boundary of the final bin.

Example 3.5.1 **Disease mortality.** Among its indicators of national health status, the United Kingdom collects data on mortality due to coronary heart disease, stroke, and related circulatory conditions for persons under 75 years of age. The British government releases their various data sets online at http://data.gov.uk/dataset.

For example, standardized circulatory-disease mortality rates per 100 000 population for $n = 397$ locations throughout the United Kingdom in 2008 appear in Table 3.4. (As above, a selection of only the smallest and largest measurements is given in the table. The complete data are available at http://www.wiley.com/go/piegorsch/data_analytics.)

Table 3.4 A selection (smallest and largest values) of standardized year-2008 circulatory-disease mortality rates per 100 000 population from a larger data set of $n = 397$ localities' rates throughout the United Kingdom.

33.97	36.99	37.42	37.95	···	118.37	122.99	126.47	126.86

Source: http://data.gov.uk/.

To explore these data for possible smoothing and other graphical summarization, begin with a simple binning operation: application of Sturges' rule produces a recommended $G = \lceil \log_2(397) + 1 \rceil = \lceil 9.63 \rceil = 10$ bins. Appeal to **R** facilitates calculation of the corresponding break points and bin frequencies, since Sturges' rule is the default binning rule in the program's `hist()` function for plotting histograms. If the data are collected into, say, the vector `smr08`, then use

```
> hist(smr08)$breaks
 [1]  30  40  50  60  70  80  90 100 110 120 130
> hist(smr08)$counts
 [1]   7  39 112  83  62  56  16  15   4   3
```

Notice that **R** has selected rounded bins, with $B_L = 30$ and $B_U = 130$, slightly below $X_{(1)} = 33.97$ and slightly above $X_{(n)} = 126.86$. (The program employs its internal `pretty()` function for selecting equally spaced, 'round' values; see `help(pretty)`.) The consequent $G = 10$ bins are represented by the values between the `breaks`; the `counts` are the binned frequencies $\hat{f}_g$ ($g = 1, \ldots, 10$). That is,

Bin	30-40	40-50	50-60	60-70	70-80	80-90	90-100	100-110	110-120	120-130
$\hat{f}_g$	7	39	112	83	62	56	16	15	4	3

Binning here shows that the mortality rates concentrate at about 50–70 deaths per 100 000 population, clearly below the center of the range. A longer tail runs out to a few rates near 110–130 deaths per 100 000. (The sample mean is about 3.5 units larger than the sample median for these data, corroborating the potential right skew; see Exercise 3.11.) Example 4.1.5 explores how to use these values for graphing a histogram of the data.

Given the slight right skew, application of Scott's normal reference rule for finding G is a viable alternative. Exercise 3.19 applies Scott's rule and builds an alternate set of bins for these data. □

3.5.2 Moving averages*

A less-coarse approach for clearing noise from a large data set appeals to the inherent summary/smoothing features of the arithmetic mean. Suppose the data are structured so that values adjacent to each X_i are felt to be related. For example, the data may be recordings of a quantitative outcome over consecutive time periods, called a *time series* (Box et al. 2008).

Then, taking local averages of related values can act to smooth out small disturbances in the larger trend. The simplest kind of local smoother averages the m values above and below

(and including) each X_i ($i = 1, \ldots, n$). The consequent subset of $2m + 1$ local elements is called the averaging *window* or *span*. As the averaging moves across the data index i, the result is known as a *moving average*:

$$\hat{X}_i = \frac{1}{2m+1} \sum_{j=-m}^{m} X_{i+j} \, . \tag{3.14}$$

Near to the lower data boundary at $i = 1$ and the upper boundary at $i = n$, the indexing may be ill-defined; if so, truncate the averaging window at each boundary. If the data are extensive enough, one can also simply ignore any $\hat{X}_i$ for which fewer than $2m + 1$ elements are included in the window. (Another possibility is to 'reflect' the smoother at $i = 1$ and $i = n$ and reuse data values near each boundary, although this can lead to instabilities in certain cases.)

An extension of (3.14) that allows for heterogeneous weighting within each window is

$$\hat{X}_i = \frac{\sum_{j=-m}^{m} w_j X_{i+j}}{\sum_{j=-m}^{m} w_j} \, , \tag{3.15}$$

similar to (3.2). As there, the weights satisfy $w_j \geq 0$. The principal weight w_0 is usually the largest; a popular choice for the remaining weights is $w_{-j} = w_j$, creating a *symmetric (weighted) moving average*. For instance, a symmetric, triangular weighted average employs the linearly decreasing weights $w_0 = m + 1, w_1 = w_{-1} = m, w_2 = w_{-2} = m - 1, \ldots, w_{m-1} = w_{m-1} = 2, w_m = w_{-m} = 1$.

A special, *asymmetric* variant for (3.15) takes $w_j = 0$ for all $j = 1, \ldots, m$. This is a *retrospective moving average*, the simplest form of which sets $w_j = 1$ for all $j = -m, \ldots, 0$. That is, the previous m observations are averaged equally with the current observation.

Example 3.5.2 Financial moving average. Retrospective moving averages are often seen with financial time series such as stock market indices, where multiday moving averages are used to study longer-term features in the index. A typical data set of this sort is the daily closing prices of the US Dow Jones Industrial (DJI) stock index. DJI data can show highly irregular patterns in response to the financial market's day-to-day activities. Applying, say, a retrospective m-day moving average

$$\hat{X}_i = \frac{1}{m+1} \sum_{j=-m}^{0} X_{i+j}$$

smooths out the irregularities and gives a clearer indication of the index's pattern of movement. The value taken for m can be 50 days, 100 days, 200 days, and so on, depending on the length of the larger series under study.

Ley (1996) gave daily closing prices for the DJI, a 20-year selection of which (June 1974–June 1993) is represented in Table 3.5. (Only the earliest and latest measurements are given in the table. The complete data are available at http://www.wiley.com/go/piegorsch/data_analytics.)

Figure 3.1 plots the full 20-year data set of $n = 4802$ DJI closing values, with a retrospective 200-day moving average superimposed. (The basic graphic is eponymously known as a *times series plot*; see Section 4.2.5.) As expected, the jagged pattern of daily index prices is ameliorated by the longer-term moving average. □

Table 3.5 Closing US Dow Jones Industrial (DJI) stock market average from 17 June 1974 to 14 June 1993; selection from larger collection of $n = 4802$ index values.

833.23	830.26	826.11	820.79	$\cdots$	3511.94	3491.72	3505.02	3514.7

Source: Ley (1996).

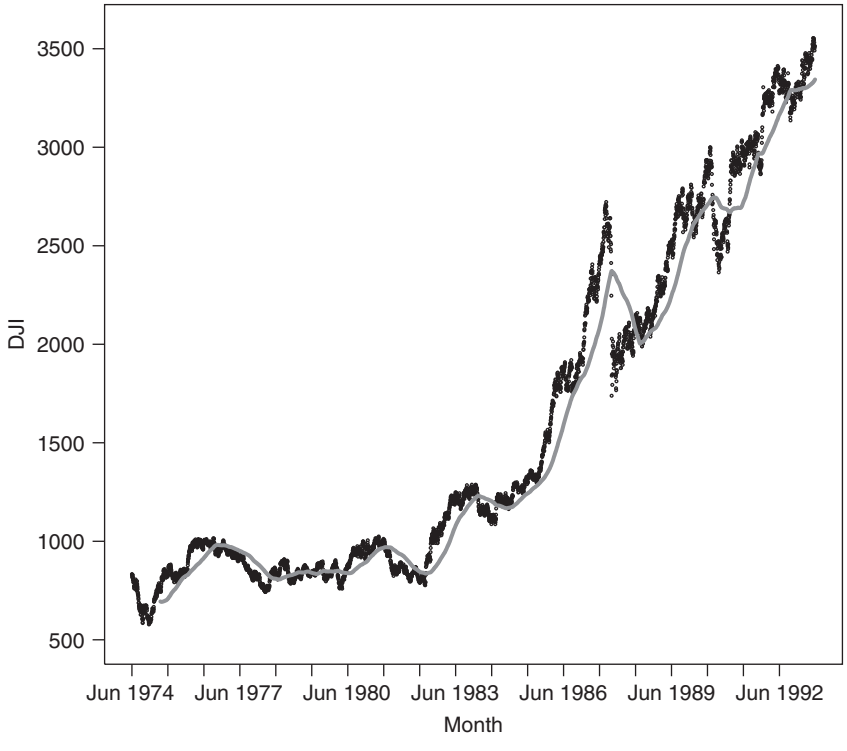

Figure 3.1 Dow Jones Industrial (DJI) stock market average (dots, $\cdots$) over 20-year period (17 June 1974–14 June 1993) and 200-day moving average (gray curve, —). Source: Data from Ley (1996).

3.5.3 Exponential smoothing*

If, in a retrospective moving average, it is felt that data farther from X_i are not as representative as those closer, the weights can be constructed to dampen exponentially as $j \to -m$ (again, with $w_j = 0$ for all $j = 1, \ldots, m$). This is known as *exponential smoothing*. In this case, rather than progress a moving window of m previous observations, one includes all the observations up to and including X_i but dampens earlier observations more severely. This is accomplished, for example, via

$$\hat{X}_i \propto \lambda\{X_i + (1-\lambda)X_{i-1} + (1-\lambda)^2 X_{i-2} + \cdots + (1-\lambda)^{i-1} X_1\}$$

for some $\lambda \in (0, 1)$. Translated into a weighted, reciprocal, moving average, the weights become $w_{hi} = \lambda(1-\lambda)^{i-h}$ and the exponential smoother is

$$\hat{X}_i = \frac{\sum_{h=1}^{i} \lambda(1-\lambda)^{i-h} X_h}{\sum_{h=1}^{i} \lambda(1-\lambda)^{i-h}} \,. \tag{3.16}$$

Notice that the weights w_{hi} are now allowed to evolve over differing values of i. Appeal to the finite geometric series in (2.23) simplifies (3.16) to

$$\hat{X}_i = \frac{\lambda \sum_{h=1}^{i} (1-\lambda)^{i-h} X_h}{1-(1-\lambda)^i} \,. \tag{3.17}$$

Now, as $i \to \infty$, the denominator in (3.17) approaches 1 for any $\lambda \in (0, 1)$. This produces an asymptotic approximation to (3.17) in the form of a recursive relationship: for very large i, $\hat{X}_i \approx \lambda X_i + (1-\lambda)\hat{X}_{i-1}$. Notice that this is a simple weighted average of the current observation X_i and the past estimate $\hat{X}_{i-1}$. (Also see Exercise 3.25.)

Exponential smoothing is also useful for *forecasting* the value of a future data point from the ensemble of current observations. Developed in a slightly different form from (3.17), the process begins with the recursive relationship

$$\begin{aligned} \hat{X}_2 &= X_1 \\ \hat{X}_i &= \alpha X_{i-1} + (1-\alpha)\hat{X}_{i-1} \qquad (i > 2), \end{aligned}$$

where $\alpha \in (0, 1)$. Manipulating this recursive definition for $i > 2$ produces

$$\hat{X}_i = (1-\alpha)^{i-2} X_1 + \alpha \sum_{j=1}^{i-2} (1-\alpha)^{j-1} X_{i-j} \,, \tag{3.18}$$

or simply $\hat{X}_i = \sum_{h=1}^{i-1} w_{hi} X_i$ for

$$w_{hi} = \begin{cases} (1-\alpha)^{i-2} & \text{if } h = 1 \\ \alpha(1-\alpha)^{(i-1)-h} & \text{if } h = 2, \ldots, i-1 \,. \end{cases} \tag{3.19}$$

It can be shown (Exercise 3.26) that $\sum_{h=1}^{i-1} w_{hi} = 1$ for the weights in (3.19). Thus (3.18) is itself a weighted moving average. In the forecasting literature, this simple exponential smoother is often called an exponential weighted moving average (or EWMA) and is available in **R** via the external *qcc* or *forecast* packages.

The simple exponential smoother/EWMA is sensitive to underlying trends in the observations. A modification that adjusts for linear trends is known as 'double exponential smoothing.' In fact, a variety of methods that enhance the simple exponential smoother for use with forecasting applications, many of which connect to autoregressive, integrated moving-average (ARIMA) models in time series analysis, are available. For more on this, see Hyndman et al. (2008).

The general concept of a moving average can be extended to locally averaged estimators for a p.d.f., generalizing the discrete binning approach that underlies construction of histograms. These are known as *density estimators* and are discussed in Section 4.1.4.

Exercises

3.1 Use a computer (or any other appropriate vehicle) to generate random samples for the following scenarios. (*Hint*: in **R**, the use the `sample()` function or any of the general class of `r*` random number generator functions, e.g., `rnorm()` for $N(\mu, \sigma^2)$.)

(a) A random sample of size $n = 50$ from the first 1000 positive integers, without replacement.

(b) A random sample of size $n = 100$ from the first 1000 positive integers, with replacement.

(c) A random sample of size $n = 50$ from the first 100 positive integers, without replacement.

(d) A random sample of size $n = 100$ from the first 50 positive integers, with replacement.

(e) A random sample of size $n = 50$ from Bin(20, 0.5).

(f) A random sample of size $n = 100$ from Bin(100, 0.25).

(g) A random sample of size $n = 100$ from N(63, 70).

(h) A random sample of size $n = 250$ from N(97.5, 122).

(i) In Exercise 3.1d, what happens if you sample without replacement? Why?

3.2 Hand et al. (1994, Section 231) listed a random sample of heights for married adults from the UK Office of Population Censuses and Surveys (OPCS). Data on husbands' heights (in mm) comprise $n = 199$ observations, a selection of which follow (download the full data set at http://www.wiley.com/go/piegorsch/data_analytics):

$$1809, 1841, 1659, 1779, 1616, \ldots, 1675, 1641, 1743, 1823, 1720$$

Calculate the sample mean, the sample variance, and the standard deviation for these 199 men's heights.

3.3 A study examined factors that affect vulnerability to hazardous events among $n = 132$ of the largest cities in the United States. An index was produced that quantified the frequency and diversity of natural hazards such as tornados, hurricanes, and floods on each city; higher values indicated greater urban vulnerability to hazardous events (Piegorsch et al. 2007). Taken on a natural logarithmic scale, a selection of the values follows:

$$0.6768, 0.9712, 1.0202, 1.0460, \ldots, 2.2793, 2.2801, 2.3996, 2.4946$$

(Download the complete data set at http://www.wiley.com/go/piegorsch/data_analytics.) Calculate the sample mean, the sample variance, and the standard deviation for these hazard index values.

3.4 The US National Aeronautics and Space Administration (NASA) via its Solar Radiation and Climate Experiment (SORCE) collects data on extreme ultraviolet (XUV) solar irradiance. Recordings from one of the experiment's photometers targeted in

the 0.1–7.0 nm range produced $n = 3510$ observations of median irradiance values (in W/m^2), a selection of which follows:

$$0.000004, 0.000004, 0.000497, \ldots, 0.001020, 0.001020, 0.001220$$

(Download the complete data set at http://www.wiley.com/go/piegorsch/data_analytics.) Calculate the sample mean, the sample variance, and the standard deviation for these values.

3.5 To explore the resilience/robustness of the sample median, $\hat{Q}_2$, to extreme observations, return to the carbohydrate intake data from Table 3.3. For simplicity, create a randomly selected subset of seven values from the larger data set using, for example, `sample(carbs, size=7)` in **R**. Find the sample mean and the median of this subset. Comment on the proximity of the two values. Now, include as an additional value the largest observation in the data set, $X_{(778)} = 738$, and calculate the sample mean and sample median of the new subset. By how much have the mean and median changed? (Use both absolute and relative differences.)

3.6 For the following data sets, calculate the 10% trimmed mean. Comment on how it compares to the corresponding sample mean.

(a) The carbohydrate intake data in Example 3.4.1.

(b) The hazard vulnerability data in Exercise 3.3.

3.7 To see how the sample variance in (3.5) is based on only $n - 1$ components of information, execute the following steps (Casella and Berger 2002, Section 5.3):

(a) Focus on the numerator and write the sum as $(X_1 - \overline{X})^2$ plus another sum made up of only $n - 1$ terms. What are these $n - 1$ terms?

(b) Show that $\sum_{i=1}^{n}(X_i - \overline{X}) = 0$.

(c) Write $(X_1 - \overline{X})^2$ in terms of the same $n - 1$ terms in Exercise 3.7a. (*Hint*: From the result in Exercise 3.7b, to what is $(X_1 - \overline{X})$ equal?)

(d) Argue that the sum $\sum_{i=1}^{n}(X_i - \overline{X})^2$ is, therefore, made up of only $n - 1$ different terms. What are these terms?

3.8 Show that the expressions for the sample variance in (3.5) and (3.6) are algebraically equivalent.

3.9 Return to the husbands' heights data in Exercise 3.2.

(a) Calculate the five-number summary and IQR. Do the data appear symmetric, or are they skewed in any way?

(b) Calculate the 10% trimmed mean for these data. How does it compare to the sample mean in Exercise 3.2? Does this give you further guidance on possible skew in the data?

(c) Calculate the z-scores from (3.7) for these 199 observations. How many lie between ± 1, ± 2, and ± 3? How does this compare with the 68–95–99.7% rule?

3.10 Return to the solar observation data in Exercise 3.4.

(a) Calculate the five-number summary and IQR. Do the data appear symmetric, or are they skewed in any way?

(b) Calculate the 10% trimmed mean for these data. How does it compare to the sample mean in Exercise 3.4? Does this give you further guidance on possible skew in the data?

3.11 Return to the circulatory-disease mortality data in Table 3.4 and calculate the following summary statistics.

(a) The sample mean, sample variance, and sample standard deviation.

(b) The sample median and rest of the five-number summary, along with the IQR.

(c) Compare the mean in Exercise 3.11a to the median in Exercise 3.11b. Use this, along with the five-number summary, to comment on possible skew in the data.

(d) Calculate the 10% trimmed mean for these data. How does it compare to the sample mean in Exercise 3.11a? Does this give you further guidance on possible skew in the data?

3.12 The data set in Exercise 3.2 also contains heights (in mm) for the wives from each married couple. As each couple constitutes a natural pairing, one can report the data as bivariate pairs; for example, the pairs (X = {Wife's height}, Y = {Husbands's height}) corresponding to the selected values displayed in Exercise 3.2 are

X = Wife's height:	1590	1560	1620	$\cdots$	1560	1630	1530
Y = Husband's height:	1809	1841	1659	$\cdots$	1743	1823	1720

(Download the full data at http://www.wiley.com/go/piegorsch/data_analytics.) Calculate the Pearson correlation coefficient in (3.9) to quantify any association between these husbands' and wives' heights. What do you find? Is this surprising?

3.13 Show that the various expressions for the Pearson correlation coefficient given in Section 3.3.3 are all algebraically equivalent.

3.14 For the following data sets, conduct a preliminary outlier analysis. Find the outer sample quartiles and the sample IQR, and from these, find the fences from (3.10). Determine if any data points exceed the upper fence or drop below the lower fence and could be potential outliers. Also find the z-scores from (3.7) for the potential outliers to study how that metric compares.

(a) The wheat kernel data in Table 3.1.

(b) The circulatory-disease mortality data in Table 3.4.

(c) The husbands' heights data in Exercise 3.2.

(d) The hazard vulnerability data in Exercise 3.3.

3.15 Return to the wheat kernel data in Table 3.1.

(a) Calculate the sample entropy, $\hat{H}'_V$, from (3.12) for the full data set.

(b) Apply a square root transformation to the data: take $Y_i = \sqrt{X_i}$. Examine the tails of the data and comment on any attenuation in the spread among the transformed values.

(c) Calculate the sample entropy, $\hat{H}'_V$, for the square-root-transformed data. Comment on any changes versus the untransformed data set.

(d) Repeat the operation in Exercise 3.15b, but now apply a logarithmic transformation. Also calculate the sample entropy, $\hat{H}'_V$, for the log-transformed data and comment on any changes.

3.16 For the carbohydrate intake data in Example 3.4.1, apply a logarithmic transformation to the original observations in Table 3.3. Has a further reduction in spread occurred among observations in the tails? What happens to the sample entropy, $\hat{H}'_V$, after applying the log transform?

3.17 Recall that the circulatory-disease mortality data in Table 3.4 exhibit a possible skew. Apply a logarithmic transformation to the X_is. Calculate the five-number summary and compare it to that found in Exercise 3.11b for the original data. What conclusions might you draw from the comparison?

3.18 Take the limit of the power transformation in (3.13) as $\lambda \to 0$, and show this produces a logarithmic transform. (*Hint*: recall l'Hôpital's rule from univariate calculus.)

3.19 Return to the circulatory-disease mortality data in Table 3.4 and rebin the values via Scott's normal reference rule: calculate the number, G, of bins using Scott's rule, construct the bins, and calculate the bin frequencies, $\hat{f}_g$, $g = 1, \ldots, G$. In **R**, the `nclass.scott()` function will give Scott's G if binning is restricted to $X_{(1)} \leq X \leq X_{(n)}$. To mimic the 'pretty' boundaries used in Example 3.5.1, however, calculate the value directly using $B_L = 30$ and $B_U = 130$. Do the two values for G differ?

3.20 Smooth the following data sets by binning the observations: (i) select the number, G, of bins via a reasonable determinator such as Scott's rule or Sturges' rule (clearly indicate which you choose), (ii) construct the bins, and (iii) calculate the bin frequencies, $\hat{f}_g$, $g = 1, \ldots, G$.

(a) The wheat kernel data in Table 3.1.

(b) The square-root-transformed wheat kernel data in Exercise 3.15b.

(c) The log-transformed circulatory-disease mortality data in Exercise 3.17.

(d) The husbands' heights data in Exercise 3.2.

(e) The hazard vulnerability data in Exercise 3.3.

(f) The solar observation data in Exercise 3.4.

3.21 For the weighted moving average in (3.15), assume the X_is are independent with constant variance $\text{Var}[X_i] = \sigma^2$. Find $\text{Var}[\hat{X}_i]$.

3.22 Similar to the data in Example 3.5.2, Ley (1996) also studied the Standard and Poor's 500-stock index (the 'S&P500'). Closing prices of the S&P500 over the same 20-year span as pictured in Figure 3.1 are available at http://www.wiley.com/go/piegorsch/data_analytics. Download these data and plot them over time. Superimpose a 200-day moving average and compare the pattern(s) with those seen in the figure.

3.23 Piegorsch and Bailer (2005, Exercise 5.11) examined data on yearly snow water equivalent (SWE, a measure of the amount of water in snow) at a watershed in the US Yellowstone Park between 1935 and 1996. The data are available online at http://www.wiley.com/go/piegorsch/data_analytics; a sample follows:

Year:	1935	1936	1937	···	1994	1995	1996
SWE:	8.2	11.7	9.30	···	12.92	14.28	18.13

Examine these data as follows:

(a) Show that an upward trend is evident by plotting SWE versus year. Note the ragged appearance of the plot.

(b) Smooth the data by calculating a 10-year moving average.

(c) Superimpose the 10-year moving average on the original data plot. Does the moving average show a smoother trend? What further investigations might be considered from this analysis?

3.24 Verify that the exponential smoother in (3.16) with weights $w_{ji} = \lambda(1-\lambda)^{i-j}$ simplifies to the expression in (3.17).

3.25 Show that the exponential smoother in (3.17) satisfies the recursive relationship

$$\hat{X}_i = \frac{\lambda}{w_{+[i]}} X_i + \frac{w_{+[i-1]}}{w_{+[i]}}(1-\lambda)\hat{X}_{i-1},$$

where $w_{+[i]} = 1 - (1-\lambda)^i$.

3.26 Show that the weights in (3.19) sum to 1 for any $\alpha \in (0, 1)$.

4

Data visualization and statistical graphics

Readers are likely aware of the famous aphorism that 'a picture is worth a thousand words' (ascribed to various sources throughout history, but cementing itself in the lexicon after appearing in US newspapers and advertisements in the early 1900s). Taken literally, it is often an exaggeration – a thousand well-placed words can tell a good story! – but it is still a potent reminder of the power of graphical visualization. (Gelman and Unwin (2013, p. 7) taken it a step further: "… a picture may be worth a thousand words, but a picture plus 1000 words is more valuable than two pictures or 2000 words.") This is especially true in data analytics: if properly constructed and interpreted, a statistical graphic can summarize features of a large data set as well as, and often better than, many summary statistics. It can communicate unseen or blatant aspects of a data stream and lead to effective knowledge discovery, especially when exploring very large amounts of data. These two goals, communication and discovery, often overlap between exploratory data-analytic studies and graphical data investigations; see Gelman and Unwin (2013) and in particular the thought-provoking discussion therein.

Construction of graphical summaries, like much of data analytics, benefits from transdisciplinary input. Recent terms describing this include information visualization (or 'infovis') and infographics. At first blush, the two sound similar, but in fact, careful distinction is made between them (Wickham 2013): the former has anchors in the field of computer science and is an ongoing area of scholarly development, where interdisciplinary inclusion of statistical input can enhance the broader technological endeavor. The latter exhibits greater focus on design and creative features of presenting information graphics. Here, data analytics plays a lesser role. Both infovis and infographics contribute, however, to contemporary advancement in data visualization.

Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery,
First Edition. Walter W. Piegorsch.

www.wiley.com/go/piegorsch/data_analytics

In this chapter, a brief introduction is given to the basic building blocks of statistical data visualization. As previously, readers familiar with these concepts may wish to skip forward to Chapter 5 and its presentation on more-advanced aspects of statistical inference.

4.1 Univariate visualization

Throughout this section, assume the data points, X_i, are taken from a simple random sample (SRS), $i = 1, \ldots, n$. These can be discrete or continuous, with underlying probability mass function (p.m.f.) or probability density function (p.d.f.) $f_X(x)$, respectively.

4.1.1 Strip charts and dot plots

One of the simplest ways to graph information in the data is to plot a point along a number line (or 'strip') at each observed value. This is known as a *strip plot* or *strip chart* and is, in effect, a one-dimensional plot of the scatter in the data. The strip chart is a very elemental approach for graphical display, but it can be useful for visualizing univariate data if the number of observations is not too large.

Example 4.1.1 **Disease mortality (Example 3.5.1, continued).** Figure 4.1 presents a strip chart for the circulatory-disease mortality data from Table 3.4. The plot was created in **R** from the command `stripchart(SMR,pch=1,xlab='SMR')`, where `SMR` is the data vector of mortality rates.

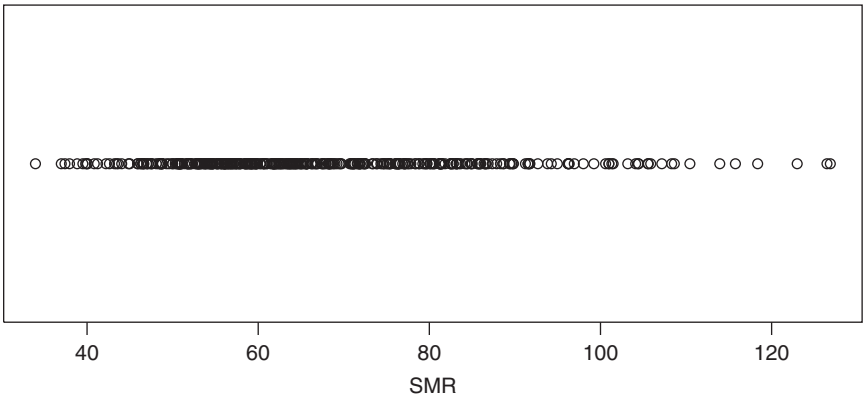

Figure 4.1 Simple strip chart in Example 4.1.1 for circulatory-disease mortality data from Table 3.4. Source: Data from http://data.gov.uk/dataset/ni_121_-_mortality_from_all_circulatory_diseases_at_ages_under_75.

The main feature in the figure is the long 'strip' of data points, which visualizes the scatter. One sees a greater density of values between about 55 and 80 deaths per 100 000 population, which tails off slowly as X grows. This corroborates results from the earlier binning exercise in Example 3.5.1, indicating that the data possess a right skew. □

Some variants of the strip chart offset or stack dots on the graphic when multiple observations have the same (or nearly the same) value. The result is then known as a 'Wilkinson'

Table 4.1 Selection of oldest ages for people above 65 years recalled by a sample of $n = 755$ US Internet users.

65	65	65	66	66	66	66	···	117	118	119	119	119	120	120

Source: http://stickers.prudential.com/.

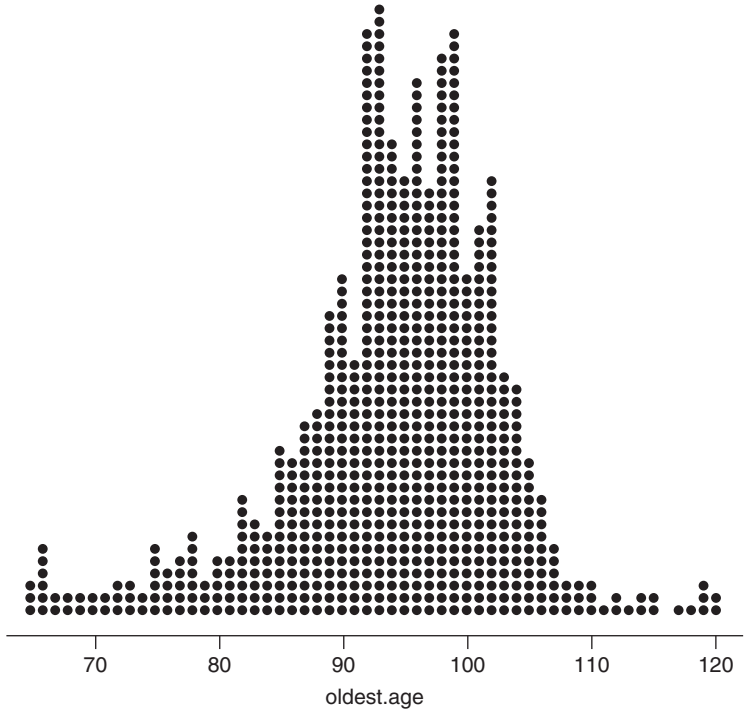

Figure 4.2 Wilkinson dot plot in Example 4.1.2 for oldest-age data from Table 4.1. Source: Adapted from http://stickers.prudential.com/. Data downloaded on 17 April 2013.

dot plot (Wilkinson 1999). (This differs from an alternative form known as the 'Cleveland dot plot' (Cleveland 1993, Section 2.5), which is not discussed here.) The next example illustrates the graphic device.

Example 4.1.2 Oldest ages. In a popular US television commercial, the Prudential Insurance Company asked a sample of individuals to give the age of the oldest person they knew above 65 years (living or dead). They then plotted these data by placing colored dots one above another on a large wall at the given age. What was produced, in effect, was a dot plot of maximum ages above 65 among a selected sample of the US population.

The company continued this experiment online at http://stickers.prudential.com/, asking Internet users to enter the age of the oldest person they knew above 65. The data from $n = 755$ respondents (as of 17 April 2013; selected values only) appear in Table 4.1.

Figure 4.2 presents a Wilkinson dot plot for these oldest ages. The plot was created in **R** via the `DOTplot()` function from the external *UsingR* package. One could create a roughly similar display in **R** via the command

```
> stripchart( oldest.age, method='stack', pch=16, cex=.8,
            offset=.3 )
```

The figure presents an interesting picture of how oldest ages are recalled by individuals, with ages in the mid-to-upper 90s most common among the respondents.

Some qualifications are in order: (i) How the company chose the original participants is unclear, and the Internet users who participated further are by definition self-selected. Thus the sample here is likely biased and should be viewed as *non*random. (ii) Participants are responding with their memories of 'oldest' ages, which themselves may be biased by recall distortions. (iii) The outcome variable is censored below so that 65 is the lowest possible value. In addition, it is the *oldest* age that can be recalled, so these are a form of data known as *extreme values* (Coles 2001). Censored extremes such as this often vary differently from central measures such as a mean.

These caveats warn that the graphic's value is more illustrative than inferential. Nonetheless, it is an interesting and even amusing use of the dot plot as statistical information employed by the commercial media. □

4.1.2 Boxplots

One of the most useful graphical displays for visualizing univariate data is known as the *boxplot*. Although it is similar to the strip chart, in that it references along a one-dimensional number line, the boxplot is designed to provide a much larger amount of information. In effect, it is a graphical representation of the five-number summary from Section 3.3.2. Given the sample minimum $X_{(1)}$, the lower quartile $\hat{Q}_1$, the median $\hat{Q}_2$, the upper quartile $\hat{Q}_3$, and the sample maximum $X_{(n)}$, a boxplot draws bars at each of these five values along their respective locations on a number line. It then connects (i) the bar at the minimum with the bar at the lower quartile and (ii) the bar at the maximum with the bar at the upper quartile, each via a dashed line (the 'whiskers'). In between the two whiskers, it connects the bars at the lower and upper quartiles with a box (which by construction will also contain the bar at the median). Notice that the IQR for the data is then the distance across the 'box' portion of the plot.

A number of variations have evolved from this basic design:

(a) Many graphic programs replace the lower quartile with the lower hinge (the median of the lower half of the data – see Section 3.3.2) and the upper quartile with the upper hinge (the median of the upper half of the data). This is the default in **R**'s `boxplot()` function.

(b) In order to identify potential outliers, the whiskers can be abridged to end at the lower and upper fences, $\mathcal{F}_1$ and $\mathcal{F}_3$ from (3.10), instead of the minimum and maximum, respectively. A further variant ends the upper (lower) whisker at the largest (smallest) datum no farther away from the box than the corresponding fence. Any observations outside of the fences are then plotted as individual points (say, with a dot or circle). Such points are, by definition, potential outliers within the sample; this device helps to visualize them quickly. Here again, this is the default setting in `boxplot()`. (For the simpler version with whiskers stretched to the data extremes, use the `range=0` option in `boxplot()`.)

(c) Another variant 'notches' the box at the median. The notches pinch the box to give it a 'waist' at $\hat{Q}_2$ and slowly draw up to the original box's edge below and above the

median. In **R**, the spread of the notch is approximately $\pm 1.58\ \widehat{\text{IQR}}/\sqrt{n}$ from $\hat{Q}_2$; it gives a rough location for where the true median may lie, such that any other boxplot's notch that overlaps likely has a similarly valued median. (The concept ties in with what is known as an 'hypothesis test' between two medians – statistical testing is introduced in Section 5.4.) If the notch distance extends past the corresponding hinge, the notch is drawn back on itself, producing sharply pointed edges on the box (McGill et al. 1978, Fig. F). In **R**, notched boxplots are available via the `notch=TRUE` option in `boxplot()`.

Many other enhancements have appeared for boxplot graphics; see, for example, McGill et al. (1978) or Benjamini (1988). Figure 4.3 dissects the anatomy of a standard boxplot, given in a horizontal perspective.

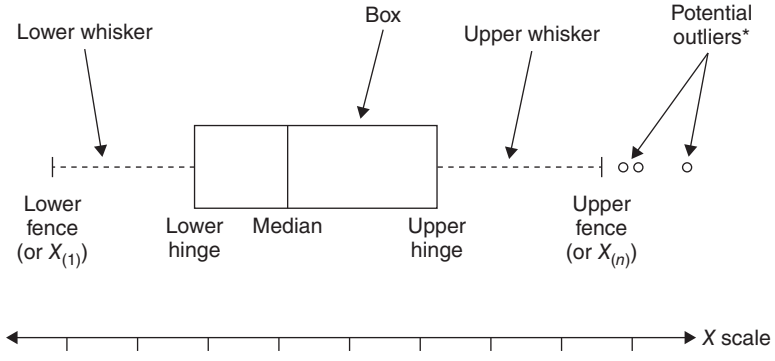

Figure 4.3 Anatomy of a box plot. * If whiskers end at fences.

The boxplot was originated by Tukey (1972, 1977) who actually developed two variants, called the *box-and-whisker plot* and the *schematic plot*. (The latter moniker has fallen out of use, while the former term is often used as a synonym for the contemporary version.) The modern boxplot as described here has evolved as a hybrid of these progenitors. Its primary value is as a compact-but-efficient graphic for examining the distributional features of the X_is. Symmetric data exhibit whiskers that extend from the box at roughly equal distances and with a median bar roughly centered within the box. Asymmetry and skew are evidenced by asymmetric whiskers, where one whisker is much longer (in the direction of the skew) than the other and/or the median locates closer to one end of the central box (away from the skew). Large occurrences of potential outliers in the direction of the skew may also be evident if using abridged whiskers.

The boxplot also has value for data analytics in that its structure is not adversely hindered by very large data sets: its design produces essentially the same output for any n. (The number of outlier points may grow busy with very large n, but this can be adjusted, e.g., by moving to outer fences in the display – simply use the `range=3` option in the `boxplot` command.)

Example 4.1.3 Disease mortality (Example 4.1.1, continued). Continuing with the circulatory-disease mortality data from Table 3.4, Figure 4.4 displays a boxplot for the observations using the **R** command

```
> boxplot( x=SMR, horizontal=T, boxwex=.33, xlab='SMR' )
```

where the data are available in the **R** vector `SMR`. (The `horizontal=T` option produces a horizontal orientation – **R**'s default is vertical boxplots – while `boxwex=.33` tightens the width of the box for a cleaner horizontal display.) The `range=` option in `boxplot()` was left unspecified, enforcing the default `range=1.5`. This generated abridged whiskers based on the fences $\mathcal{F}_1$ and $\mathcal{F}_3$ from (3.10). A *rug plot* of vertical marks is included at the bottom of the display, indicating the location of each individual observation. This is generated via the separate command `rug(x=SMR)`.

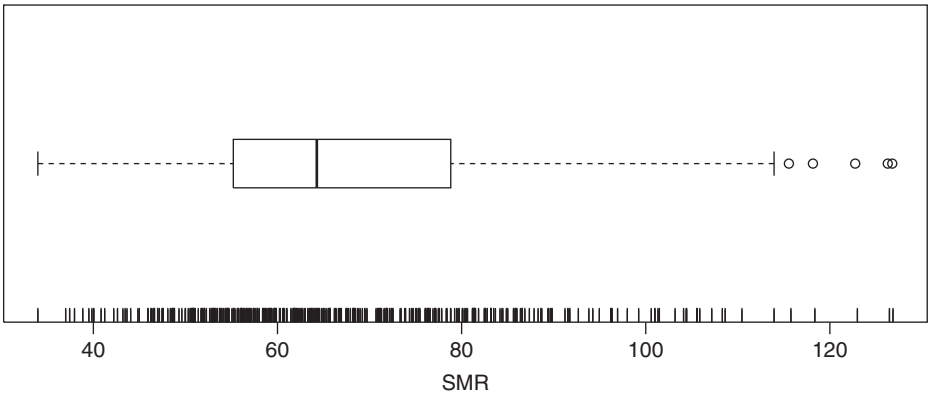

Figure 4.4 Box plot in Example 4.1.3 for circulatory-disease mortality data from Table 3.4; 'rug' at bottom marks individual data values. Source: Data from http://data.gov.uk/dataset/ni_121_-_mortality_from_all_circulatory_diseases_at_ages_under_75.

From the figure, we observe that the bulk of the data (the 50% inside the box) rests between about 55 and 80 deaths per 100 000 population. The data also exhibit a right skew: the right whisker extends farther out than its left counterpart, and the median bar is slightly left of center within the box. These results corroborate past indications from the binning in Example 3.5.1 and the strip chart in Example 4.1.1.

Commensurate with its design, Figure 4.4 also provides a quick outlier analysis: no observations appear below the lower fence; however, five observations are marked as potential outliers above the upper fence at 113.94 deaths per 100 000 population. (As per the defaults in **R**, this is the largest X_i that does not exceed $\mathcal{F}_3$. Find the actual statistic via `boxplot.stats(SMR)[[1]][5]`.) The corresponding UK communities would be strong candidates for further examination as to their high, potentially outlying circulatory-disease mortality rates. Also see Exercise 3.14b. □

4.1.3 Stem-and-leaf plots

Another graphical display – also developed by Tukey (1972) – is known as the *stem-and-leaf plot* or just *stemplot* for short. Different in style and structure from the boxplot, a stemplot is built from the individual numerals of each datum. This assumes, of course, that the data are quantitative. The 'stem' of each X_i is taken as the collection of its digits up to but not including its final digit (reading left-to-right). The 'leaf' is that final digit. For example, if $X = 118.37$,

then its stem is '118.3' and leaf is '7.' (For simplicity, one usually rounds this to 118.4 and sets the stem to '118.' and the leaf to '4.' Stemplots are used more for fast graphical display than highly precise data presentation.)

A stemplot takes this stem–leaf bifurcation and builds the stems along a vertical axis, smallest at the top. It then displays the leaves for each datum horizontally at each stem. Most programs also order the leaves at each stem from smallest to largest, left to right, a preferred option. If a nonproportional-width font (such as 'courier') is used to display the leaves, then the more leaves at each stem, the longer the plot extends. Taken as a whole, this gives an informative visual of how the data distribute over the sample. (As the stems run vertically, the visual is rotated 90° to the right, which takes some getting used to.)

Example 4.1.4 **Myocardial infarction.** Salzberg (1988) reported on a study of survival for 132 cardiac patients who suffered an acute myocardial infarction or 'heart attack'. Among the variables measured was the age (in years) at which patients suffered their (first) attack. Six patients did not report the age for their attack and so are not included here. Thus the final sample size is $n = 126$. Table 4.2 presents the data.

Table 4.2 Ages of (first) acute myocardial infarction (in years) among $n = 126$ cardiac patients.

35 46 46 47 48 50 50 51 52 52 53 54 54 54 54 54 54 55 55 55 55 55 56 56 56
57 57 57 57 57 57 57 57 58 58 59 59 59 59 59 59 59 60 60 60 60 60 60 61 61
61 61 61 61 61 61 62 62 62 62 62 62 62 62 62 62 62 63 63 63 63 63 63 63 64
64 64 64 64 64 64 65 65 65 65 65 66 66 66 66 66 67 67 67 68 68 68 68 69 69
69 70 70 70 70 70 71 71 72 72 73 73 73 73 74 75 77 78 78 78 79 79 80 81 85 86

Source: Salzberg (1988).

To gain an understanding of the age-at-attack distribution for these subjects, and possibly uncover new knowledge in the processes underlying myocardial infarction, we can construct a stemplot in **R** using the command `stem(age)`, where `age` is the data vector of ages. This produces the graphic in Figure 4.5 (with, as recommended, a nonproportional-width font). Notice how **R** formats the stem column for these data: each stem is listed twice. The first placement (closer to the top) of each stem is limited to only those leaves between `0` and `4`; the second (lower) is limited to leaves between `5` and `9`. This retains the basic ordering of the data while expanding the scale of the plot to better visualize the distributional features. (To reduce the scale into only single tens-digits for the stem here, use `stem(age,scale=0.5)`. The resulting graphic is visually less informative, however.)

The stemplot in Figure 4.5 reveals that ages in this sample of patients distribute in a generally symmetric, unimodal manner around a central value of 60–64 years. From `summary(age)`, we indeed find $\hat{Q}_2 = 62$ and $\overline{X} = 62.81$ years. For purposes of specifying the statistical distribution of these data, a normal (Gaussian) model from Section 2.3.9 might be appropriate. (Also see Example 4.1.7.) □

Since each datum's value in a stemplot is also presented via the stem-and-leaf structure, one can quickly see where values are distributing themselves in the sample – unusually large or small values will be evident, as will the approximate central tendency of the data.

Stemplots can be extended for comparing two random samples. First find the samples' common stems, then plot the leaves of the first sample to the right of the stems, and plot

```
The decimal point is 1 digit(s) to the right of the |
Stem | Leaf
   3 | 5
   4 |
   4 | 6678
   5 | 001223444444
   5 | 55555666777777889999999
   6 | 000000111111122222222233333334444444
   6 | 55555666667778888999
   7 | 00000112233334
   7 | 5788899
   8 | 01
   8 | 56
```

Figure 4.5 Stemplot in Example 4.1.4 for myocardial infarction data from Table 4.2. Source: Data from Salzberg (1988).

the leaves of the second sample to the left. This is known as a *side-by-side* or *back-to-back stemplot*.

While the stemplot is an informative summarizing graphic with small-to-moderate-sized data sets, for very large data sets, the plot can grow very busy. This limits its visual impact. One solution is to decrease the precision of the data values – for example, work with 118 instead of 118.37 – and many programs automatically apply this sort of strategy, including **R**. With very large sample sizes, however, other methods for graphic summarization may be more useful, as discussed in the next subsection.

4.1.4 Histograms and density estimators

A well-known graphic used to visualize the shape or distribution in a data set is the *histogram*, and most readers have likely seen one (or more). The graphic is essentially a simple chart of $G > 0$ nonoverlapping bars extending up in proportion to the frequency of occurrence for different values of the data. With categorical data, this is essentially a *bar chart* – see Section 4.2.1 – but for quantitative outcomes, the bar locations must first be selected based on the data values. The effort is essentially a smoothing operation and can be performed by binning the data as in Section 3.5.1. The result is a set of disjoint bins – also called *class intervals* – and a set of corresponding frequencies, $\hat{f}_g$, counting the number of X_is in each gth bin ($g = 1, \ldots, G$).

Choice for the number of bins, G, can be made using any pertinent binning rule. As discussed in Section 3.5.1, Scott's (1979) normal reference rule is usually preferred because it is less sensitive to asymmetries in the data than the earlier rule due to Sturges (1926). Note that a binning rule may be reexpressed in terms of the bin width, that is, the (constant) separation distance between bin boundaries. If the lower limit on the first bin is set to $B_L \leq X_{(1)}$ and the upper limit on the final bin is set to $B_U \geq X_{(n)}$, then the width of each bin is $h = (B_U - B_L)/G$. For example, Scott's rule is often written in terms of the bin width

$$h = 3.49 \, \frac{S_X}{n^{1/3}} \, ,$$

where S_X is the sample standard deviation.

A preferred variant for the histogram, because it allows for comparison with density estimates of $f_X(x)$ (see the following text), replaces the frequencies with the relative frequencies $\hat{f}_g/n$. In **R**, this is constructed via the `hist(x,freq=FALSE)` command, where `x` is the vector of data values. Or, one can use `truehist(x)` from the *MASS* package. Note that Sturges' rule is the default in `hist()`, while Scott's rule is the default in `truehist()`. In both cases, however, **R** will apply its `pretty()` function to give cleaner, rounded bin limits. These may differ from the actual numerical specification for G or h under either rule, unless overridden by direct specification of the break points via the `breaks=` option. For more on histograms in **R**, see Rizzo (2008, Section 10.1).

Example 4.1.5 Disease mortality (Example 3.5.1, continued). Continuing with the circulatory-disease mortality data from Table 3.4, the binning operation using Sturges' rule in Example 3.5.1 produced 10 nonoverlapping bins and a set of corresponding bin frequencies, $\hat{f}_g$. Converting these to relative frequencies after dividing by n and plotting gives the histogram in Figure 4.6. (The shading was created via the `col='gray'` option in `hist()`.) Notice the inclusion of a rug plot at bottom marking the individual data values, similar to that in Figure 4.4.

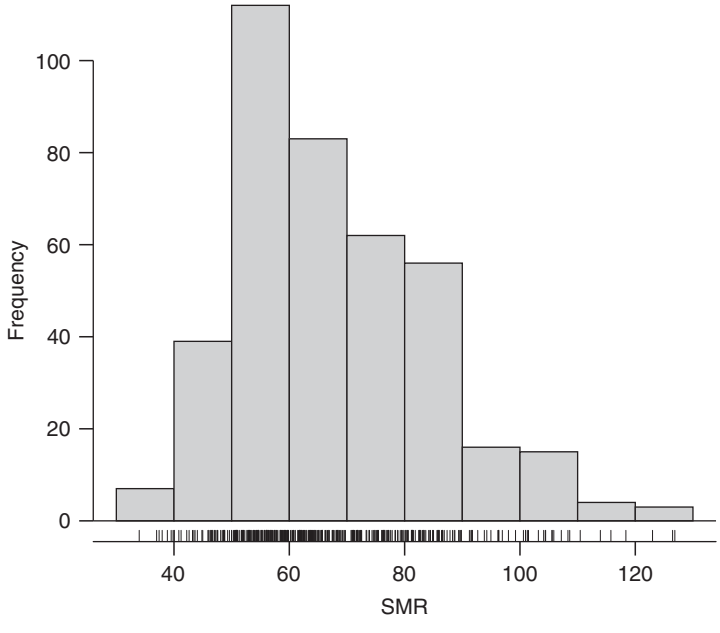

Figure 4.6 Histogram for circulatory-disease mortality data from Table 3.4; 'rug' at bottom marks individual data values. Bins produced via application of Sturges' rule (Sturges 1926). Source: Data from http://data.gov.uk/dataset/ni_121_-_mortality_from_all_circulatory_diseases_at_ages_under_75.

The graphical representation of mortality rates in the figure shows a unimodal distribution with mode near 55 deaths per 100 000 population. There is also a clear skew to the right: some localities have mortality rates as high 110–130 deaths per 100 000 population, as much as twice that in the central portion of the distribution.

As in Exercise 3.19, the skew could motivate the use of Scott's rule in place of Sturges' rule for specifying the number of bins. Doing so (Exercise 4.7a) gives a slightly different display but a qualitatively similar graphic.

These results corroborate all previous indications seen with these data and help to visualize the nature of circulatory disease mortality in these British communities. □

The histogram is an effective graphical device for visualizing the distribution of a set of observations. A common concern often raised with it, however, is that the binning operation central to its production is highly subjective. That is, the final graphic can depend heavily on the choice and location of the bins: too many and the bars may become ragged and sparse in certain regions; too few and the histogram is too coarse. Indeed, the plot can be manipulated, possibly unethically, to show different distributional shapes for the same set of data. Use of established binning procedures such as Scott's rule or Sturges' rule can avoid this concern, but then the data analyst should indicate how the bins were produced (as in Figure 4.6).

To alleviate some of these issues, more-advanced methods exist for estimating the underlying p.d.f, $f_X(x)$, from a set of (continuous) outcomes X_i. Known as *kernel density estimators*, the methods are again a form of smoothing that estimates the relative frequency of occurrence for any value of x in the support space $\mathcal{S}$.

The basic formula for a kernel density estimator based on data $X_1, \ldots, X_n$ is

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right), \tag{4.1}$$

where $K(\cdot)$ is a *kernel function* satisfying $\int_{-\infty}^{\infty} K(t)dt = 1$ and $h > 0$ is a smoothing parameter called the *bandwidth* or *window width*. Equation (4.1) is also known in some circles as the Parzen–Rosenblatt–Whittle window estimator, after three early pioneers of the technology (Parzen 1962; Rosenblatt 1956; Whittle 1958).

The kernel in (4.1) can be thought of as a weighting function for the smoothing operation. In most cases, it is chosen as a symmetric p.d.f., although it need have nothing to do with the p.d.f. estimated in (4.1). Common examples include the standard normal p.d.f. from Section 2.3.9, referred to as the *Gaussian kernel*

$$K_{\text{Gauss}}(t) = \frac{1}{\sqrt{2\pi}} \, \mathrm{e}^{-\frac{1}{2}t^2} \, ,$$

the *triangular kernel*

$$K_{\text{Tr}}(t) = (1 - |t|) \, I_{(-1,1)}(t) \, ,$$

or the *Epanechnikov kernel*

$$K_{\text{Ep}}(t) = \frac{3}{4}(1 - t^2) \, I_{(-1,1)}(t) \, ,$$

among many others. The Gaussian form appears as the default in many computer programs, including the `density()` function in **R**.

The bandwidth h in (4.1) controls the smoothness of $\hat{f}(x)$ and is analogous to the bin width of a histogram: small values intensify the amount of local smoothing around each x and produce 'bumpier' estimates of f, while larger values widen the inclusion window and

smooth out local irregularities. For example, under the Gaussian form $K_{\text{Gauss}}(t)$, the bandwidth acts as the standard deviation of the underlying kernel (Exercise 4.8): as h grows, the normal p.d.f. used for the kernel flattens, increasing the impact on $\hat{f}(x)$ of data points farther from x.

Choice of h can have a substantial impact on the final density estimator. Theoretical considerations (Venables and Ripley 2002, Section 5.6) show that the optimal choice must be proportional to $n^{-1/5}$. A popular rule-of-thumb with the Gaussian kernel sets

$$h = \tau \, \frac{\min\{S_X \, , \, \widehat{\text{IQR}}/1.34\}}{n^{1/5}} \, , \tag{4.2}$$

where $\tau > 0$ is a tuning parameter. Standard choices for τ include $\tau = 0.90$ (Silverman 1986) or $\tau = 1.06$ (Scott 1992). The former is the default for Gaussian kernels in **R**'s `density()` function, although the latter is also very popular. Data-based bandwidth selection is an active area of research (Ahmad and Ran 2004; Liao et al. 2010), and a variety of more-complex methods also exist (Jones et al. 1996).

In practice, determining how many values of x at which to calculate the density estimator is driven by computer resource constraints and/or how smooth a final output the analyst requires. (Increasing the number of points will typically produce a smoother graphic.) **R** requires at least 512 points in `density()` and recommends more – in powers of 2, so next would be $2^{10} = 1024$ – if resources permit. Note that this differs from the sample size, n (of course, larger values of n will produce density estimates more representative of the true $f_X(x)$).

Example 4.1.6 Disease mortality (Example 4.1.5, continued). Return to the circulatory-disease mortality data from Table 3.4. Recall that Figure 4.6 gave a histogram of the data, where a unimodal shape with a right skew was evidenced.

Consider enhancing the visual representation here via a kernel density estimator from (4.1). To explore how the choice of bandwidth, h, can affect the graphic, Figure 4.7 presents two different density estimators for these data, the first with a relatively small bandwidth of $h = 1$ and the second with a much larger bandwidth of $h = 10$. (In both the cases, the specific kernel was the Gaussian default in **R**'s `density()`, thus h acts as the standard deviation of the smoothing kernel. **R** scales all its smoothing kernels in `density()` to achieve this equivalence. The original histogram from Figure 4.6 is underlaid for reference.) The bandwidth effect is clear: setting h too low produces a ragged graphic, while driving it too high oversmooths the estimator (notice how the lower tail at $h = 10$ extends beyond the lower range of the data plot).

Figure 4.8 overlays the final kernel density estimator from (4.1) on the histogram, again employing the Gaussian kernel $K_{\text{Gauss}}(t)$. Choice of the bandwidth was taken as the **R** default using $\tau = 0.90$ in (4.2). With these data, it is $h = 4.62$ calculated in

```
> density(SMR,n=1024)$bw
```

The plot was produced in **R** via the following code:

```
> hist( SMR, freq=F, main='' )          #orig. histogram
> lines( density(SMR,n=1024) )          #overlay density() output
> rug( SMR )                            #add rug of data values
```

The figure's graphical representation of right-skewed mortality rates corroborates indications seen previously for these data. □

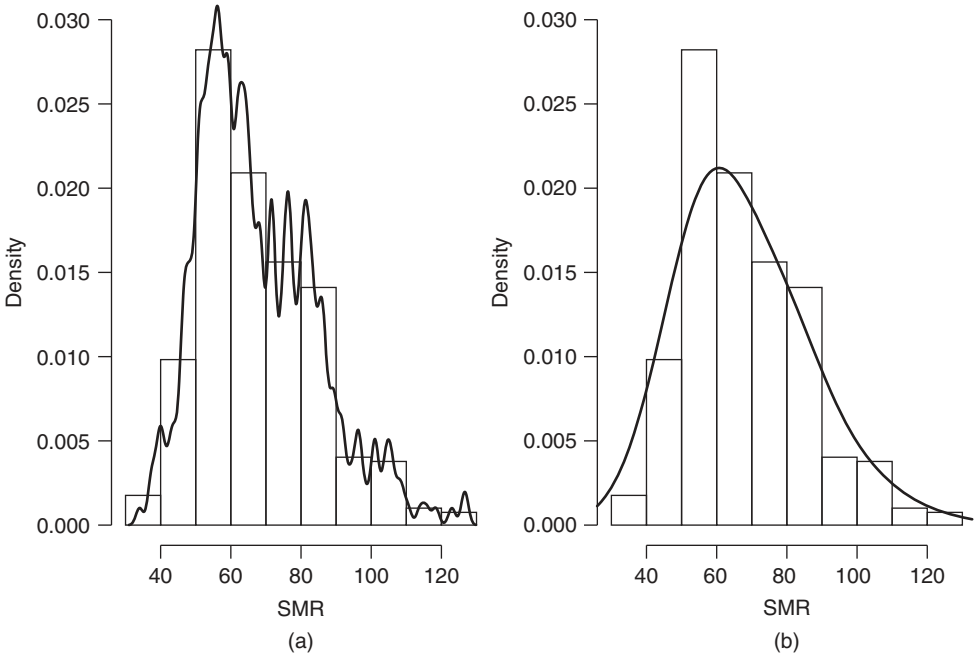

Figure 4.7 Kernel density estimates (solid curves) in Example 4.1.6, overlaid on histogram of circulatory-disease mortality data from Table 3.4. (a) Density estimate based on Gaussian kernel with bandwidth set to $h = 1$. (b) Density estimate based on Gaussian kernel with bandwidth set to $h = 10$. Histogram bins produced via application of Sturges' rule (Sturges 1926). Source: Data from http://data.gov.uk/dataset/ni_121_-_mortality_from_all_circulatory_diseases_at_ages_under_75.

In similar manner to stemplots, histograms can be extended for comparing two random samples. First, determine a binning structure and then apply (the same) bins to each samples' data. Next, score a vertical line according to the chosen bins. Plot the resulting histogram bars of the first sample to the right of the line, and plot the bars of the second sample to the left. This is known as a *side-by-side* or *back-to-back histogram*. **R** can produce back-to-back histograms via, for example, the `bi.bars()` function in the external *psych* package.

4.1.5 Quantile plots

The histogram and density estimator in Section 4.1.4 are useful visualizations, but they can only go as far as suggesting the features of an unknown, true p.d.f. A graphic device that can assess the validity of a formal specification for $f_X(x)$ takes advantage of the information available in the sample quantiles from (3.8). The concept is simple: start with the cumulative distribution function (c.d.f.) from the proposed distribution,

$$F_X(x) = \int_{-\infty}^{x} f_X(t)dt \ .$$

From this, determine the population quantiles, $q_p = F_X^{-1}(p)$ from Section 2.1.4 over a selection of values for $p \in (0, 1)$. Then, given data $X_i \sim$ i.i.d. $f_X(x)$ $(i = 1, \ \ldots \ , n)$, find the sample quantiles $\hat{q}_p$ via (3.8) and compare them to the corresponding population values, q_p.

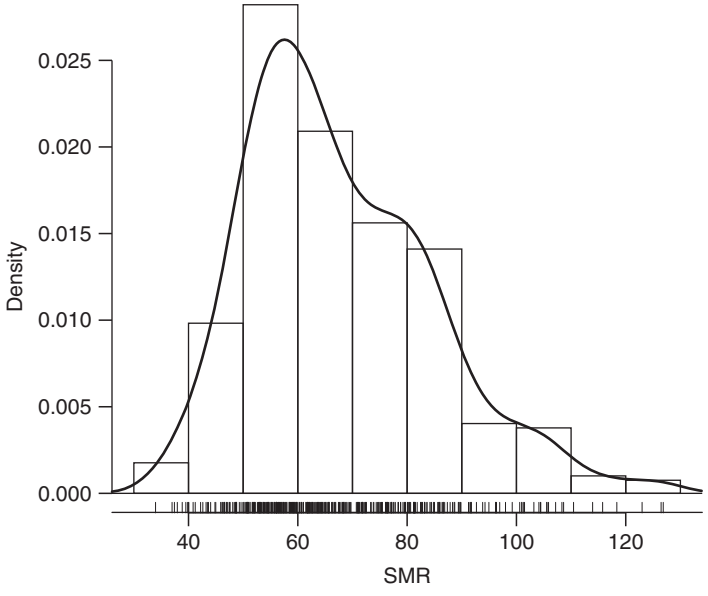

Figure 4.8 Kernel density estimate (solid curve) based on Gaussian kernel with data-determined bandwidth ($h = 4.62$) in Example 4.1.6, overlaid on histogram of circulatory-disease mortality data from Table 3.4. Rug plot at bottom marks individual data values. Histogram bins produced via application of Sturges' rule (Sturges 1926). Source: Data from http://data.gov.uk/dataset/ni_121_-_mortality_from_all_circulatory_diseases_at_ages_under_75.

Plotting $\hat{q}_p$ versus q_p creates what is known as a *quantile plot*. (By convention, start with the smallest p near 0 and plot the points by increasing p towards 1.) If the sample quantiles approximate the true quantiles from $F_X(x)$, the plotted points will collect along a (roughly) straight line. If the plot shows strong curvature, however, poor correspondence is evidenced between the quantile information posited under $F_X(x)$ and that in the sample. Verzani 2005, Fig. 3.4) gives an instructive graphic.

Perhaps the most common use of a quantile plot occurs when comparing against a normal distribution, $X_i \sim$ i.i.d. $\mathrm{N}(\mu, \sigma^2)$. The plot is then called a *normal quantile plot* or sometimes a *normal probability plot*. In this case, the sample quantiles are plotted against the corresponding normal quantiles. Given its popularity, the normal quantile plot is available as the dedicated function `qqnorm()` in **R**.

Example 4.1.7 Myocardial infarction (Example 4.1.4, continued). Return to the data in Table 4.2 on ages at myocardial infarction ('heart attack') of $n = 126$ cardiac patients. Recall from the stemplot of those data in Example 4.1.4 that the age-at-attack distribution appeared unimodal, symmetric, and possibly normal. To assess this more fully, Figure 4.9 displays a normal quantile plot for these data using the **R** command `qqnorm(age)` for the data in `age`. A straight line is overlayed for reference via the **R** command `qqline(age)`.

The plot shows a typical pattern of fit for normally distributed data: the majority of points in the middle of the plot coincide with the reference line. While some deviation is evidenced in the tails – that is, at the lower and upper limits of the plot – they do not bend away so appreciably as to suggest drastic departure from normality. □

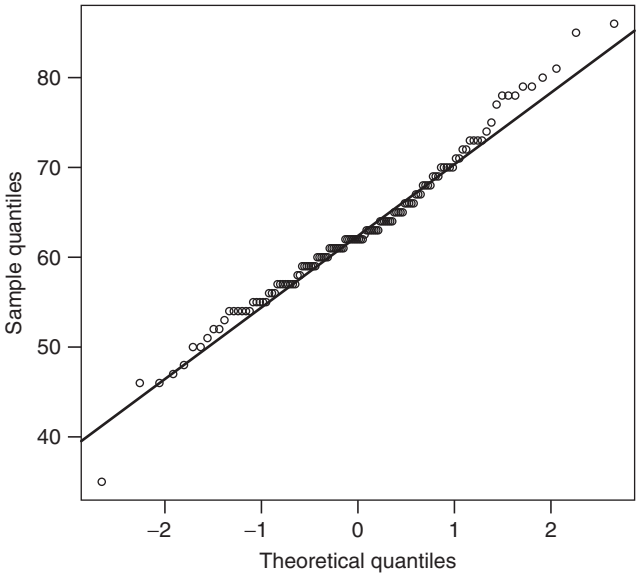

Figure 4.9 Normal quantile plot and reference line (solid line) in Example 4.1.7, for myocardial infarction data from Table 4.2. Source: Data from Salzberg (1988).

Example 4.1.8 Disease mortality (Example 4.1.5, continued). In contrast to Example 4.1.7, Figure 4.10 plots a normal quantile plot (and overlaid reference line) for the circulatory-disease mortality data from Table 3.4. Recall that our previous analyses with these data consistently showed a clear right skew, questioning whether a symmetric distribution such as the normal would provide an appropriate fit; see, for example, Figure 4.8.

The graphic in Figure 4.10 shows clear departure from the reference line and obvious curvilinearity in the quantile plot. Right skew in the data is evidenced by the plot's convexity: the sample quantiles are more concentrated than the normal quantiles for $p < 0.50$ and less concentrated for $p > 0.50$, a consequence of the right skew. This forces the plot to bow downwards in the middle, indicating that these data do not appear normally distributed. □

One can extend the quantile plot from a single-sample comparison against a specific distribution to a pairwise comparison between the quantiles from two random samples. See Section 4.2.2.

4.2 Bivariate and multivariate visualization

Generalizing the univariate setting described in Section 4.1, data analytics often involve summarization of data taken on multiple samples or over multiple outcomes. For continuous responses, the natural progression involves two random samples: $X_i \sim$ i.i.d. $f_X(x)$, $i = 1, \ldots, n$, independent of $Y_j \sim$ i.i.d.$_Y(y)$, $j = 1, \ldots, m$, where $f_X(x)$ and $f_Y(y)$ are the two samples' p.d.f.s. (Extensions to more than two samples are similar.)

In addition, categorical data can occur in multivariate form. Suppose observations are recorded and classified into $K > 1$ disjoint outcome categories. The counts of how many

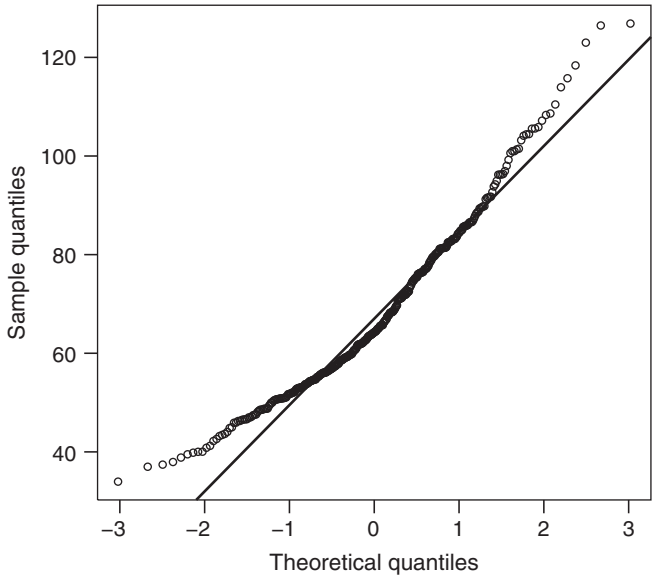

Figure 4.10 Normal quantile plot and reference line (solid line) in Example 4.1.8, for circulatory-disease mortality data from Table 3.4. Source: Data from http://data.gov.uk/dataset/ni_121_-_mortality_from_all_circulatory_diseases_at_ages_under_75.

observation fall into each category, say, $Y_1, \ldots, Y_k$ $(k = 1, \ldots, K)$, then represent a set of K multivariate response variables.

In either case – continuous or categorical – graphic summarization of bivariate and multivariate data becomes correspondingly more complex than visualization methods for the simple univariate setting. A number of popular methods for such are described in this section.

4.2.1 Pie charts and bar charts

Of all the statistical graphics seen in practice, the *pie chart* is possibly the most ubiquitous and also the most derided. It achieves both feats based on the same feature: its simplicity. Into a circular 'pie', place 'slices' of area proportional to the percentage of occurrence for a given category in a set of data. Thus, for example, if 40% of a company's customer base come from the north quadrant of the city, 30% from the south quadrant, 20% from the east quadrant, and the remaining 10% from the center and west quadrant, the pie has four slices in the proportions 4:3:2:1, respectively. See Figure 4.11.

Pie charts are most appropriate when there are only a handful – literally – of possible categories being studied and when the proportions they exhibit are not too small. Change the proportions in the customer base illustration above to 96%, 2%, 1%, and 1% and the graphic's message becomes more muddy (other than 'most customers come from the north quadrant,' for which no graphic is necessary). Or, delineate the base into 40 regions instead of four, and the graphic information becomes too busy to interpret at all!

The pie chart's derision stems from its overuse: being so simple, it is often the first graphic a novice analyst will employ. It is not always the best choice for a summary graphic, however, because it forces the viewer to visualize the information in terms of relative area.

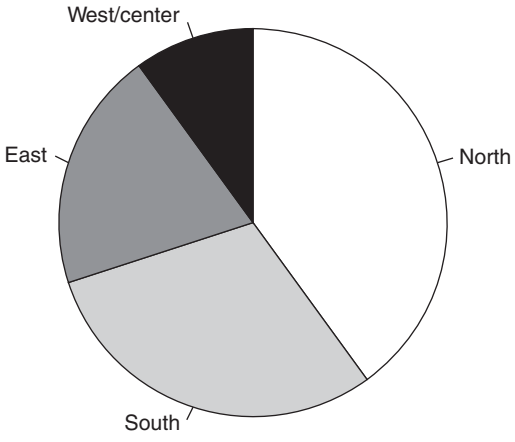

Figure 4.11 Sample pie chart for artificial customer-base illustration; labels are customer quadrants within a city.

In Figure 4.11, the difference between north quadrant and west/center customer bases is clear, but south quadrant and east quadrant seem much more similar. Is the 50% (relative) difference between them easy to discern?

Perhaps **R** makes it clearest in `help(pie)`: "Pie charts are a very bad way of displaying information. The eye is good at judging linear measures and bad at judging relative areas. A bar chart or dot chart is a preferable way of displaying this type of data."

The dot chart to which **R** refers can be either the Wilkinson dot pot (suitably modified for categorical data input) or the Cleveland dot plot, both mentioned in Section 4.1.1. Also given as an alternative, and favored here, is the *bar chart*. Generally ascribed to Playfair (1786) as one of the earliest forms of statistical graphic, a bar chart is a simple chart of $K \geq 1$ nonoverlapping bars extending up in proportion to the frequency of occurrence for different values of the data. When the data are continuous and from a single sample, this essentially produces the histogram from Section 4.1.4. For a set of K categorical outcomes, however, the bars are simply the frequencies of occurrence for each category/label. If the categories are ordered in some manner, then the bars are displayed across the increasing (or decreasing, as desired) category levels. If the categories possess no implicit ordering, then the bars are displayed in any desired order. Any number of categories may be used, and the frequencies can take on any values. Properly constructed such that the bar widths are equal, the areas of the bars in a bar chart will convey information about each category's relative impact on the phenomenon under study, as will their heights.

In **R**, bar charts are implemented via the `barplot()` function. A `barchart()` function is also available, with advanced features for applying trellis graphics.

Example 4.2.1 Driving speeds. From a study of driving speeds, Kadane and Lamberth (2009) provided data on speeds of motorists when they exceed the speed limit on the US New Jersey (NJ) Turnpike. The observations comprise $n = 6536$ recordings by radar of driving speed, posted speed limit, and vehicles state of origin on the license plate.

The NJ Turnpike is a major artery linking communities along the US East Coast in what is known as the 'I-95 corridor.' I-95 stands for the major interstate highway (Interstate route #95) creating this transit corridor, from the US state of Florida (FL) to the US state of Maine (ME).

Although the larger study collected state-of-origin data from all US states and the District of Columbia (DC), for simplicity, here the analysis is restricted to only those 15 states and DC comprising the I-95 corridor: CT, DC, DE, FL, GA, MA, MD, ME, NC, NH, NJ, NY, PA, RI, SC, and VA. This reduces the data set to $n = 6252$ individual records.

Speed limits on the NJ Turnpike vary, depending on road conditions and other factors. Thus for better comparison across drivers, the original data were modified into recorded speed (in mph) *above* the posted limit. These appear in Table 4.3. (As previously, only a selection of the measurements is given in the table. The complete data are available at http://www.wiley.com/go/piegorsch/data_analytics.)

Table 4.3 Selected driving speeds on the NJ Turnpike from a larger set of $n = 6252$ observations.

License state of origin	Recorded speed (mph) above limit
CT	2, 2, 2, 3, … , 19, 19, 19, 19
DC	4, 4, 5, 5, … , 19, 21, 21, 22
⋮	⋮
NH	6, 7, 7, 7, … , 15, 15, 16, 18
NJ	1, 1, 2, 2, … , 30, 32, 37, 38
NY	1, 2, 2, 2, … , 29, 29, 31, 35
⋮	⋮
VA	1, 1, 2, 2, … , 24, 26, 26, 31

Source: Kadane and Lamberth (2009).

For an initial analysis of these data, consider a simple breakdown of how (speeding) drivers distribute across the 16 different states (and DC). If `licenseI95` is the character vector of states-of-origin for the $n = 6252$ observations, then via the `table()` function in **R** one finds

```
> table( licenseI95 )
    licenseI95
     CT   DC   DE   FL   GA   MA   MD   ME
    214   53  243  164   40  185  702   12

     NC   NH   NJ   NY   PA   RI   SC   VA
    137   16 2267 1106  539   24   45  505
```

A bar chart of these summarized data appears in Figure 4.12 from the **R** command

```
> barplot( sort(table(licenseI95)), ylab='Frequency', xlab='State' )
```

The graphic here illustrates that, as might be expected, NJ license plates greatly outnumber the rest, with the highly populous, adjoining, or nearby states of NY, MD, PA, and VA following thereafter. States farther away – such as ME or GA – and less-populous states – such as RI and NH – show smaller bars in the plot. These data will be analyzed further in examples to follow. □

Bar charts can also be used to visualize features between two or more random samples: the length of the bars is taken as some quantitative measure of interest such as the mean or median of each sample. Whiskers and/or ticks, known as *error bars*, can be added above and

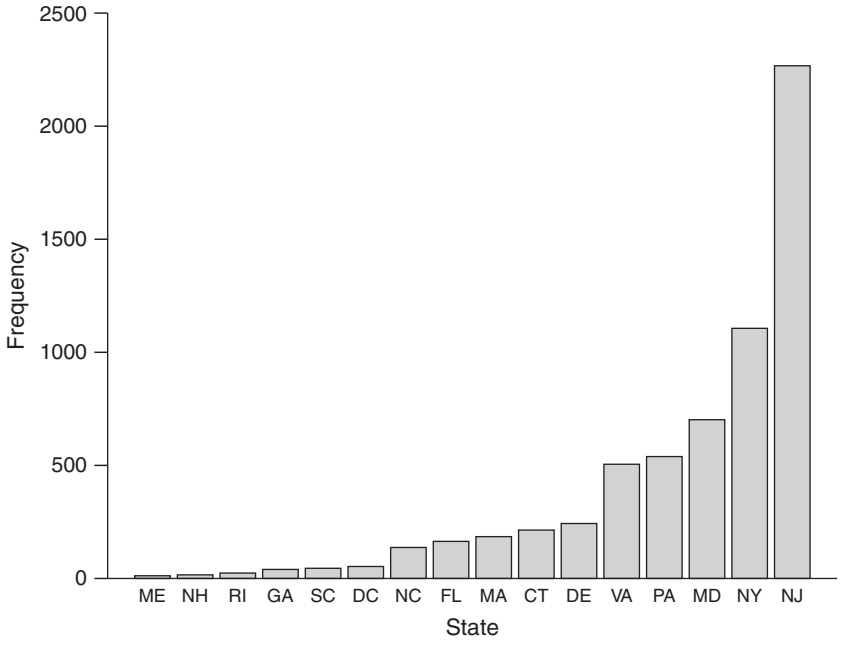

Figure 4.12 Bar chart for state-of-origin frequencies in Example 4.2.1, with NJ driving speeds data from Table 4.3. Source: Data from Kadane and Lamberth (2009).

below the mean to indicate variation in the sample. These are usually displayed as $\overline{X} \pm S$ at each bar.

Example 4.2.2 **Driving speeds (Example 4.2.1, continued).** Continuing with the NJ driving speeds data in Table 4.3, Figure 4.13 plots a bar chart of the mean speeds above limit (in mph) for the 15 states and DC along the I-95 corridor. Error bars give $\pm S$ above and below the bars at the means for each state of origin. Sample **R** code for the core graphic follows. The code can be enhanced in a number of ways, and other programming steps to achieve the plot are also possible.

```
> mean <- tapply( overlimI95, factor(licenseI95), mean )
> sdev <- tapply( overlimI95, factor(licenseI95), sd )
> summary.df <- data.frame(overlimI95,table(licenseI95),mean,sdev)
> ord.df <- summary.df[ order(summary.df$Freq), ]
> barplot( ord.df$mean, ylim=c(0,max(ord.df$mean+ord.df$sdev)) )
> xloc <- barplot( ord.df$mean,
                ylim=c(0,max(ord.df$mean+ord.df$sdev)), plot=F )
> arrows( xloc, ord.df$mean-ord.df$sdev, xloc,
                     ord.df$mean+ord.df$sdev, angle=90, code=3 )
```

Here the graphic highlights the differences among states' average over-limit speeds – ME drivers appear to exceed the limit less than the others – although the wide error bars also show that substantial variability exists in the data. □

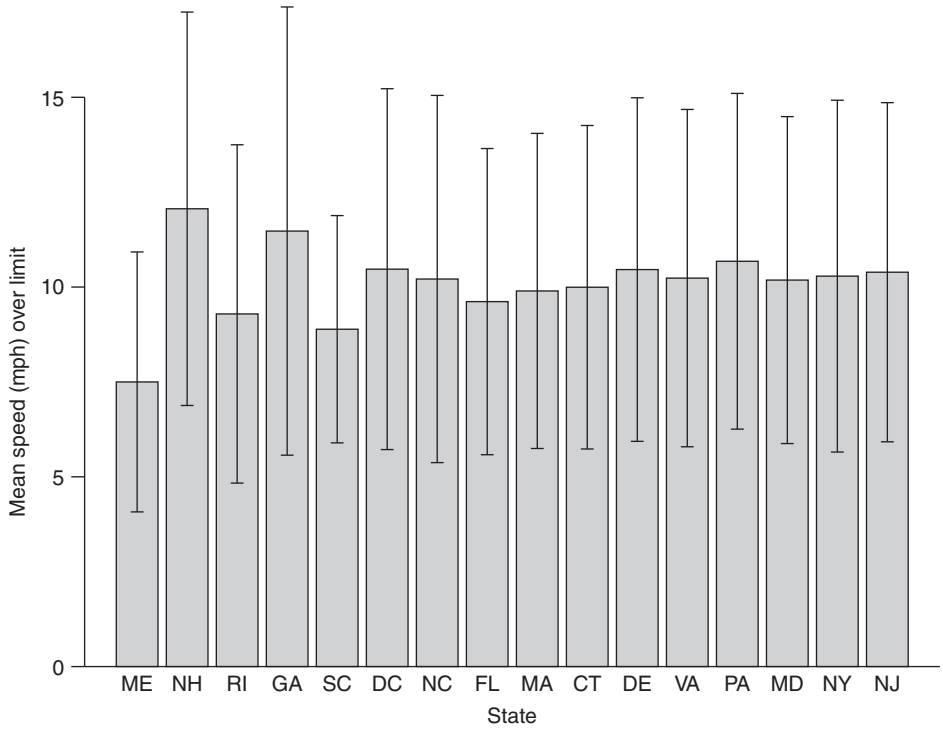

Figure 4.13 Bar chart of mean speeds (in mph) above posted limit and error bars (as mean ± standard deviation), for NJ driving speeds data in Example 4.2.2. State-of-origin order displayed as per Figure 4.12. Source: Data from Kadane and Lamberth (2009).

As Figure 4.13 illustrates, bar charts can graphically display sample means (along with standard deviations, error bars, etc.) and help to visualize similarities and differences among different samples. The graphic is still very basic, however, and for comparing quantitative information across multiple samples, other formats can do better; see Section 4.2.2.

An interesting extension of the bar chart places the bars on a polar coordinate grid so that they circle and extend from a central point. This makes for a visually intriguing display. Credited to famous nineteenth-century nurse and public health advocate Florence Nightingale, the graphic is known as a *polar area plot*, or sometimes a *coxcomb plot* or a *rose plot*. Nightingale used polar area graphics in her 'Diagram of the Causes of Mortality in the Army in the East' to identify mortality in the Crimean War from various causes, including preventable diseases (Nightingale 1858). The plot's striking visual effect adds to its popularity, especially when contrasting colors are employed; see http://infographicsnews.blogspot.com/2008/01/worth-thousand-words.html or http://www.sciencenews.org/view/generic/id/38937/description/. Polar area plots can be constructed in **R** via the `geom_coxcomb()` function in the external *ggsubplot* package.

As an infographic device, the polar area plot is popular; however, its ability to visualize statistical information is generally not better than a bar chart or other simpler graphic displays (Gelman and Unwin 2013). For example, a *line graph* replaces the bars in a bar chart with lines connecting the tops of the bars. If the heights represent frequencies, this is also known

as a *frequency polygon*. Such replacement has value if the categories along the horizontal axis are naturally ordered from left to right. This also makes for easier comparison between two or more different sets of frequencies over the same set of (horizontal axis) categories, because overlaying multiple bar charts on top of each other sometimes creates more graphic confusion than it prevents. (With Nightingale's mortality data, which occur over time, a better alternative might also be a *time series plot*; see Section 4.2.5).

4.2.2 Multiple boxplots and QQ plots

As seen in Section 4.1.2, the boxplot is an effective graphic for summarizing features of a random sample. Given two independent samples, it is a simple effort to extend the graphic and include boxplots for both. In fact, this can be applied to any number of $K \geq 1$ independent random samples, producing a *multiple boxplot*. Consider first the two-sample case with data $X_i \sim$ i.i.d. $f_X(x)$, $i = 1, \ldots, n$, independent of $Y_j \sim$ i.i.d. $f_Y(y)$, $j = 1, \ldots, m$. From each random sample, compute the five-number summaries $\{X_{(1)}, \hat{Q}_{X1}, \hat{Q}_{X2}, \hat{Q}_{X3}, X_{(n)}\}$ and $\{Y_{(1)}, \hat{Q}_{Y1}, \hat{Q}_{Y2}, \hat{Q}_{Y3}, Y_{(m)}\}$. Include variants such as replacing the outer quartiles with hinges as in **R**'s `boxplot()` – cf. Section 3.3.2 – or abridging the whiskers at the fences and plotting potential outliers, as desired. Then, *on the same scale*, plot the corresponding boxplots side by side. These are usually displayed as vertical boxplots, mimicking the format of the quantitative bar chart, above. Differences between the samples highlighted by the boxplots' features will become immediately clear.

For $K > 2$ samples or groups, simply construct the five-number summaries for each sample and plot all K boxplots on the same scale. (Again, vertical boxplots are common.) For very large values of K, it can become difficult to compare all groups easily, so use of this graphic device is best reserved for K in, say, the range $2 \leq K \leq 20$.

Example 4.2.3 **Driving speeds (Example 4.2.2, continued).** Continuing with the NJ driving speeds data in Table 4.3, Figure 4.14 plots a multiple boxplot of the speeds (in mph) over the posted limit, stratified by licenses state of origin.

The plot was produced in **R** via the command

```
> boxplot( overlimI95 ~ factor(licenseI95, levels=freqorder) )
```

where `overlimI95` is the vector of over-limit speeds and `licenseI95` is the corresponding vector of states-of-origin. The tilde `~` instructs **R** to 'model' the response variable as `overlimI95` and the explanatory variable as `licenseI95`. The `factor()` function builds the explanatory levels from the different states of origin in the character variable `licenseI95` and orders them using the `levels=` option by the frequencies in Figure 4.12. The separate character variable `freqorder` (not shown) contains the pertinent ordering. Employed within `boxplot()`, this creates multiple boxplots for `overlimI95` over the different levels of `licenseI95`. As in Example 4.2.2, consideration is limited to states along the I-95 corridor.

Figure 4.14 enhances the descriptive information provided on state origin-specific over-limit speeds, compared to the simpler bar chart in Figure 4.13. The similarities in central tendencies among states remain, as seen by the roughly similar medians in each box's center bar. Clearer indications are given, however, as to possible skew in over-limit speed among many of the states: an (upper) right skew is evidenced by the numerous possible upper outliers, and *no* lower outliers, in the plot. Further, there are clear differences in how

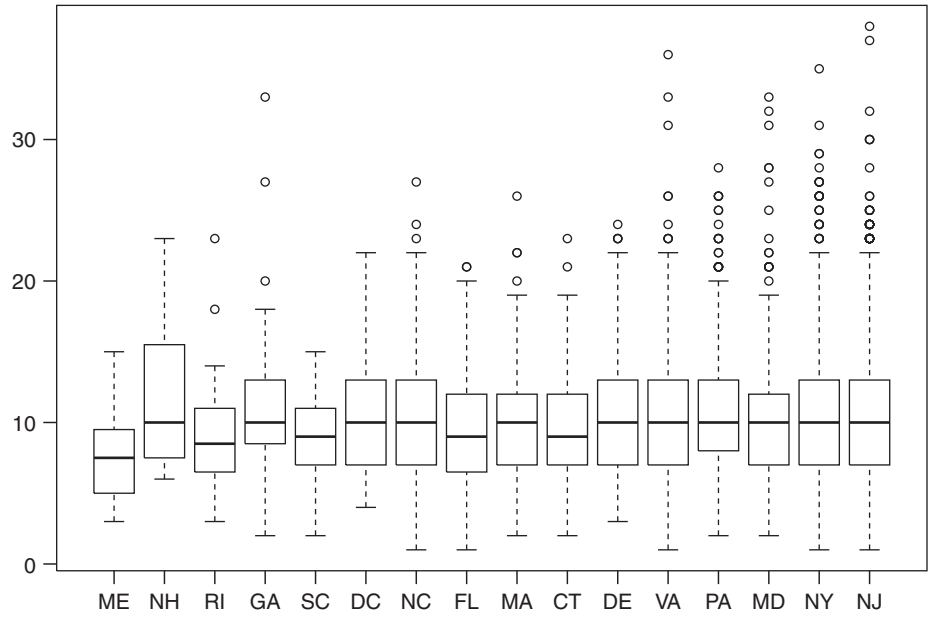

Figure 4.14 Multiple boxplot of mean speeds (in mph) above posted limit in Example 4.2.3, for NJ driving speeds data from Table 4.3. State-of-origin order displayed as per Figure 4.12. Source: Data from Kadane and Lamberth (2009).

the skews distribute themselves among the different states of origin. Some states exhibit multiple outliers, and a few – such as ME and SC – exhibit none. Indeed, the SC plot is essentially symmetric. The multiple boxplot graphic exposes a number of interesting features in these data. □

At the core of a boxplot's construction is its use of quantile information, specifically the three sample quartiles. Extending the operation to include additional quantiles and then plotting them, as in the quantile plot of Section 4.1.5, produces a further comparison between two samples or groups. To wit, suppose again that data are generated from two random samples: $X_i \sim$ i.i.d. $f_X(x),\ i = 1, \ldots, n$, independent of $Y_j \sim$ i.i.d. $f_Y(y),\ j = 1, \ldots, m$. From each sample, compute the respective sample quantiles, say, $\hat{q}_{xp}$ and $\hat{q}_{yp}$ over a range of $p \in (0, 1)$. If $f_X(\cdot)$ and $f_Y(\cdot)$ are the same, one expects the sample quantiles to be roughly equal at each p. To visualize this, plot $\hat{q}_{yp}$ versus $\hat{q}_{xp}$. This is known as a *quantile–quantile (Q–Q) plot*.

The Q–Q plot is a useful device for exploring similarities or differences between two samples' distributions, although it does take some practice to interpret it correctly. If $f_X(\cdot)$ and $f_Y(\cdot)$ are the same, the plotted points will collect along a (roughly) straight line. In fact, this will be a 45° line. Deviations between the two distributions will appear as deviations from the 45° line, either as a straight line with a slope different from 1 (X is linearly related to Y) or as a curvilinear pattern (X and Y have different patterns of skew).

Example 4.2.4 Residential energy loads. Tsanas and Xifara (2012) presented a study of factors that affect energy performance in residential buildings. They derived two response variables, $X = \{\text{Heating load}\}$ and $Y = \{\text{Cooling load}\}$, over a variety of 768

different building conditions. The paired data appear in Table 4.4. (As above, only a selection of measurements is given in the table. The complete data are available at http://www.wiley.com/go/piegorsch/data_analytics.)

The distributional features of each variable are explored in Exercise 4.9. Here, consider comparison of the two distributions via a Q–Q plot. Figure 4.15 displays the plot, produced using the **R** command `qqplot(HeatLoad,CoolLoad)`. A 45° line is superimposed via the command `abline(0,1)`.

Table 4.4 Selected data pairs (X, Y) with $X = \{\text{Heating load}\}$ and $Y = \{\text{Cooling load}\}$, from a larger set of 768 paired observations.

(6.01, 10.94)	(6.04, 11.17)	(6.05, 11.19)	$\cdots$	(42.96, 39.56)	(43.10, 39.41)

Source: Tsanas and Xifara (2012).

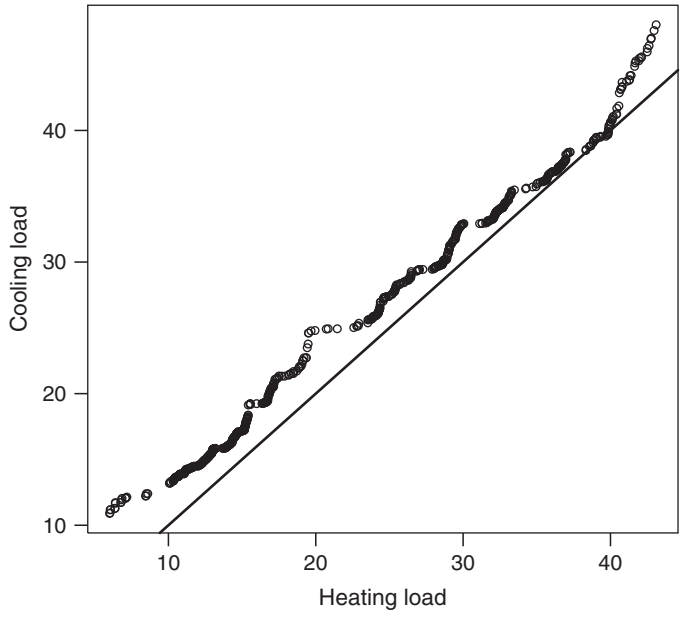

Figure 4.15 Q–Q plot for $X = \{\text{Heating load}\}$ and $Y = \{\text{Cooling load}\}$ for energy load data in Example 4.2.4. Source: Data from Tsanas and Xifara (2012).

The points in the Q–Q plot follow a roughly linear pattern, although slight convexity is seen at the upper quantiles. This suggests that the two distributions' shapes may be roughly similar, with possible deviation in the upper tails. As the Q–Q plot is shifted away from the 45° line, however, there is a likely difference in their means or variances (or in both). □

Notice that Example 4.2.4 illustrates the use of the Q–Q plot with paired data; however, this is not a requirement of the method. Q–Q plots can be constructed for any two random variables, whether their observations are paired or not. Indeed, the numbers of observations in each sample, n and m, can be quite different. (The actual plot itself *does* involve paired quantiles, but these need not be constructed from paired observations.)

4.2.3 Scatterplots and bubble plots

When bivariate data appear as *pairs* of observations (X_i, Y_i), it is natural to plot them on an (x, y) Euclidean grid. The resulting scatter of points is called a *scatterplot*, and it is one of the most effective ways to visualize two-dimensional data. Strong linear or curvilinear relationships between Y and X become evident, wide dispersion or variation in one or both variables is usually easy to see, and even the lack of any relationship between the variables can be quickly recognized (as, e.g., a cloud of points with no apparent slope or pattern).

The scatterplot is a mainstay instrument in the data-visualization toolkit. In **R**, there are a number of different functions or commands that can produce it in one form or another. The workhorse is `plot(x,y)` or `plot(y ~ x)`. Both forms produce a plot of `y` (vertical axis) against `x` (horizontal axis). The latter is consistent with other uses of the tilde operator in **R**, as an indicator of a model relationship of the form *response variable* ∼ *explanatory variable(s)*.

Example 4.2.5 Driving speeds (Example 4.2.1, continued). Return to the NJ driving speeds data in Table 4.3. Recall from the state-of-origin frequency analysis in Example 4.2.1 that the majority of motorists in these data were identified with NJ license plates. These were followed by plates from populous states such as NY, PA, MD, and VA. While many of these states are close to NJ, it is worth asking whether state population could also play a role, because the observed frequencies were not adjusted for home-state population.

Table 4.5 lists populations in 2009 of the 15 US states, and DC, along the I-95 transit corridor, corresponding to the license plates recorded in this speeding study. As is well known, NY and FL are the largest in population, followed by PA, GA, NC, and then NJ.

Table 4.5 Populations in 2009 for US states (and DC) along I-95 transit corridor.

State	Population	State	Population	State	Population
CT	3 518 288	MA	6 593 587	NY	19 541 453
DC	599 657	MD	5 699 478	PA	12 604 767
DE	885 122	ME	1 318 301	RI	1 053 209
FL	18 537 969	NC	9 380 884	SC	4 561 242
GA	9 829 211	NH	1 324 575	VA	7 882 590
		NJ	8 707 739		

Source: http://www.census.gov/popest/data/historical/2000s/vintage_2009/state.html.

Figure 4.16 plots the state-of-origin frequencies from Figure 4.12 against the populations in Table 4.5, using the `plot()` command. The scatterplot shows an increasing band of points from left to right: the state-of-origin frequencies from Figure 4.12 do appear to change in proportion to increasing population size, validating the supposition that larger states tend to supply more license plates for possible identification in this study. The pattern is not very concentrated, however, suggesting that other factors could be at play as well, at least for the motorists observed here.

Notice the clear outlier in the plot midway along the population scale but very high on the frequency scale. As expected, this is NJ, where a 'home-state' effect likely is driving the large number of NJ license plates observed with these data. □

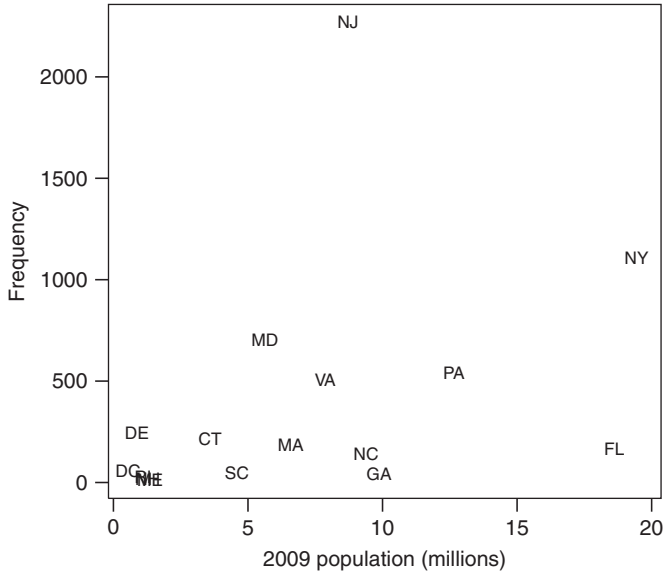

Figure 4.16 Scatterplot in Example 4.2.5 of $Y = \{\text{State-of-origin frequency}\}$ from Figure 4.12 versus $X = \{\text{State population}\}$ from Table 4.5. Points are labeled by state of origin; large outlier at top-middle is NJ. Source: Data from Kadane and Lamberth (2009) and http://www.census.gov/popest/data/historical/2000s/vintage_2009/index.html.

Example 4.2.5 gives only a brief indication of the power of a scatterplot and how it can be used for visualizing information in two dimensions. Indeed, the `plot()` function in **R** has a broad array of capabilities for creating two-dimensional graphical displays. Readers are encouraged to explore its various features in greater depth.

Extension of the scatterplot when $K = 3$ triplet variables are under study is a more enigmatic endeavor. Interactive three-dimensional (3D) holographic projections are perhaps in our future; until then, programs do exist that rotate two-dimensional representations of the 3D relationship for useful visualization, as in the external *rgl* package. Simpler still are methods that distill the information down to a standard two-dimensional graphic; see, for example, the external *scatterplot3d* package. Suppose the data triples are (W_i, X_i, Y_i), $i = 1, \ldots, n$. One obvious way to visualize relationships among the three variables is to build all (three) possible pairwise scatterplots: W versus X, W versus Y, and X versus Y. This strategy can be extended to any $K > 2$, however, and so is explored further in the presentation of scatterplot matrices, later in this section.

Another approach suitable for the $K = 3$ case is known as a *bubble plot* (Everitt 2005, Section 2.4). The concept is fairly simple: begin with a standard scatterplot of Y_i versus X_i but then enhance the plotted points by drawing circles around each point. The circle's areas extend in proportion to the value of W_i at that i. (That is, the radius of each circle is proportional to $\sqrt{W_i/\pi}$. Some authors alternatively size the circles' radii directly in proportion to W_i, although the effect has greater visual perspicuity if the circles are sized by area.) The result is a scatterplot with varying 'bubbles' showing the impact of W on the X, Y pattern. To explore

other aspects of the trivariate relationship, switch the combination(s) of scatterplot pairings and bubble variable.

Numerous variations exist for creating bubble plots: one can change the circles to squares, or to diamonds, add labels or colors, and so on. If using **R**, the program's powerful variety of graphic capabilities can enhance the visual effect.

A popular infographic offshoot that quantifies text/word frequencies by mimicking the bubble plot's approach for relative sizing is in http://www.wordle.net/.

Example 4.2.6 **Automobile fuel economy.** The US Department of Energy, in concert with the US Environmental Protection Agency (EPA), reports on automobile fuel economy in terms of miles driven per gallon of gasoline (MPG). For the 2011 automobile model year, these agencies gave MPG data on $n = 652$ automatic-transmission vehicles and included a number of other possible variables associated with each vehicle's energy efficiency. Table 4.6 presents $Y = \{\text{MPG}\}$ for these vehicles (combined highway/city, conventional fuel), along with two additional variables: $X = \{\text{Engine displacement (in liters)}\}$ and $W = \{\text{Number of cylinders}\}$. (As above, only a selection of measurements is given in the table. The complete data are available at http://www.wiley.com/go/piegorsch/data_analytics.)

Table 4.6 Selected data triplets on automobile fuel economy: (W, X, Y) with $W = \{\text{No. cylinders}\}$, $X = \{\text{Engine displacement (L)}\}$, and $Y = \{\text{MPG}\}$, from a larger set of 652 triplicate observations.

(2, 1.3, 24.2445)	(4, 2.0, 37.5530)	(4, 1.6, 37.4670)	$\cdots$	(16, 8.0, 12.4782)

Source: http://www.fueleconomy.gov/feg/download.shtml.

A simple scatterplot of $Y = \{\text{MPG}\}$ versus $X = \{\text{Engine displacement}\}$ (Exercise 4.16) shows a curvilinear, inverse relationship between the two variables, as might be expected. To visualize how $W = \{\text{Number of cylinders}\}$ also relates, Figure 4.17 presents a bubble plot where the bubbles are proportional in area to W. The figure is actually presented to exemplify two slight alternatives for the plot: (i) Figure 4.17a is a basic bubble plot with the X, Y points plotted as small dots, and with transparent bubbles overlayed; this helps emphasize the precise locations of the X, Y points. (ii) Figure 4.17b is a simpler bubble plot only showing the bubbles but shading them (here, in grayscale) and adding contrasting borders. This loses location specificity for the plotted points but counters by emphasizing the bubble effect. (Readers can decide which gives a more instructive graphic for delivering the visual information.)

R code to produce the plots (not including label enhancement) takes advantage of the `symbols()` function:

```
> #left panel: scatterplot with bubble overlay
> plot( Y ~ X , pch='.' , xlim=c(1,8.5), ylim=c(10,60) )
> radius <- sqrt( W/pi )          #for area proport'l to W
> symbols( X , Y, circles=radius, inches=.21, add=T )
>
> #right panel: shaded bubble plot
> symbols( X, Y, circles=radius, inches=.21, bg='gray',
               fg='white', xlim=c(1,8.5), ylim=c(10,60) )
```

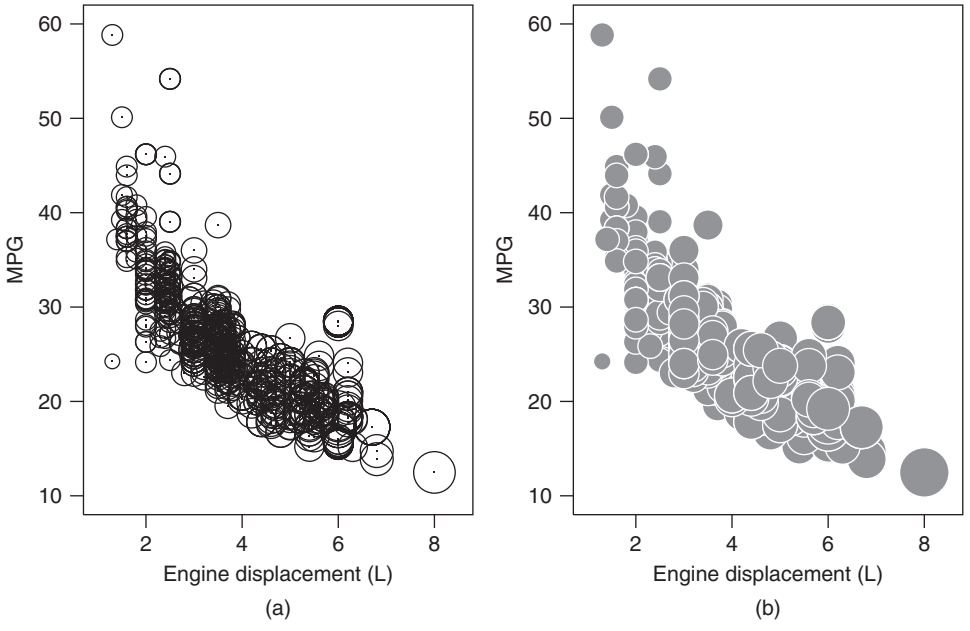

Figure 4.17 Two forms of bubble plot from Example 4.2.6 with $Y = \{\text{MPG}\}$ plotted against $X = \{\text{Engine displacement (L)}\}$, and with overlayed bubble areas proportional to $W = \{\text{No. cylinders}\}$ for fuel economy data in Table 4.6. Panel (a) gives specific locations of X, Y points (dots); panel (b) employs shaded bubbles for contrast. Source: Data from http://www.fueleconomy.gov/feg/download.shtml.

In the end, both versions of the bubble plot meet the graphical objective: we see that engine displacement detrimentally affects MPG in an apparently curvilinear manner. There is also a concomitant, recognizable, detrimental effect of number of cylinders, in that lower MPG appears related to higher numbers (i.e., bigger bubbles). □

When the number of variables grows to $K \geq 3$, the visualization process increases in complexity. Suppose the jth K-tuplet is $(X_{1j}, X_{2j}, \ldots, X_{Kj})$, $j = 1, \ldots, n$. As noted above, one approach for graphing the variable relationships is to construct two-dimensional scatterplots over all $\binom{K}{2} = \frac{1}{2}K(K-1)$ possible variable pairings. A popular device for this is known as a *scatterplot matrix*: each plot of the pairwise scatter between X_{ij} and X_{kj} $(i \neq k)$ is drawn as a small square scatterplot, then all the smaller plots are trellised into a larger $K \times K$ square 'matrix' of scatterplots. Unique information is only provided in the upper (or lower) triangle of smaller scatterplot cells, because the (i, k)th cell plots the same variables as the (k, i)th cell. (The reflected perspective can often yield some useful visualizations, however.)

Notice that the main diagonal of a scatterplot matrix will trivially contain scatterplots of each X_{ij} against itself that is, straight 45° patterns – and is, therefore, filled only with the ith variable's name. Alternatively, other clever graphics can be inserted into the diagonal cells, such as univariate histograms of X_{ij}; see (Everitt 2005, Section 2.5).

In **R**, a scatterplot matrix is produced via the `pairs()` function, although other alternative functions in various specialized packages can generate it as well.

Example 4.2.7 Wheat kernels (Example 3.3.1, continued). Return to the agricultural study of wheat grain characteristics (Charytanowicz et al. 2010) from Example 3.3.1. Along with $n = 210$ measurements of $X_1 = \{\text{Kernel length}\}$ in Table 3.1, the following additional variables were recorded:

$X_2 =$ Kernel width
$X_3 =$ Kernel asymmetry
$X_4 =$ Kernel groove length
$X_5 =$ Kernel area
$X_6 =$ Kernel perimeter, and
$X_7 =$ Kernel compactness $= 4\pi X_5 / X_6^2$.

Table 4.7 lists a selection of measurements from this larger data set. (The complete data are available at http://www.wiley.com/go/piegorsch/data_analytics.)

Table 4.7 A selection of wheat kernel data from a larger set of $n = 210$ observations.

	Variables						
Index, i	X_{1i}	X_{2i}	X_{3i}	X_{4i}	X_{5i}	X_{6i}	X_{7i}
1	4.899	2.787	4.975	4.794	10.59	12.41	0.8648
2	4.902	2.879	2.269	4.703	11.23	12.63	0.8840
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
209	6.666	3.485	4.933	6.448	18.36	16.52	0.8452
210	6.675	3.763	3.252	6.550	19.94	16.92	0.8752

Variables are defined in Example 4.2.7.
Source: Charytanowicz et al. (2010).

To visualize possible interrelationships among these variables, construct a scatterplot matrix. For simplicity, limit attention to only the $K = 5$ variables X_1, X_2, X_3, X_4, andX_7. (One might expect a priori that X_5, X_6, and X_7 will be highly interrelated; see Exercise 4.17.) To produce the matrix in **R**, the data vectors are collected together into an **R** *data frame* (see Appendix B), say `wheat.df`, which becomes the single argument of the `pairs()` function: `pairs(wheat.df)`. The result appears in Figure 4.18.

The scatterplot matrix here identifies strong relationships between (i) kernel length versus width; (ii) length versus groove length; and, to a lesser degree, (iii) width versus groove length. None of these are terribly surprising. A strong, possibly curvilinear relationship appears between (iv) kernel width versus compactness, while the connections between (v) kernel length versus compactness and (vi) groove length versus compactness are less certain; the scatters are broadly dispersed, but possible patterns may be buried therein. These interrelationships may be worthy of further investigation. Lastly, all four scatterplots based on the kernel asymmetry measure appear dispersed, with disorderly scatter. As a consistent pattern, this may represent its own kind of informative feature. The scatterplot matrix here gives the analyst much to consider. □

4.2.4 Heatmaps

Suppose multidimensional data are available in the form of a rectangular matrix, **M**, where the rows are categorized by a row variable R_i and the columns are categorized by a

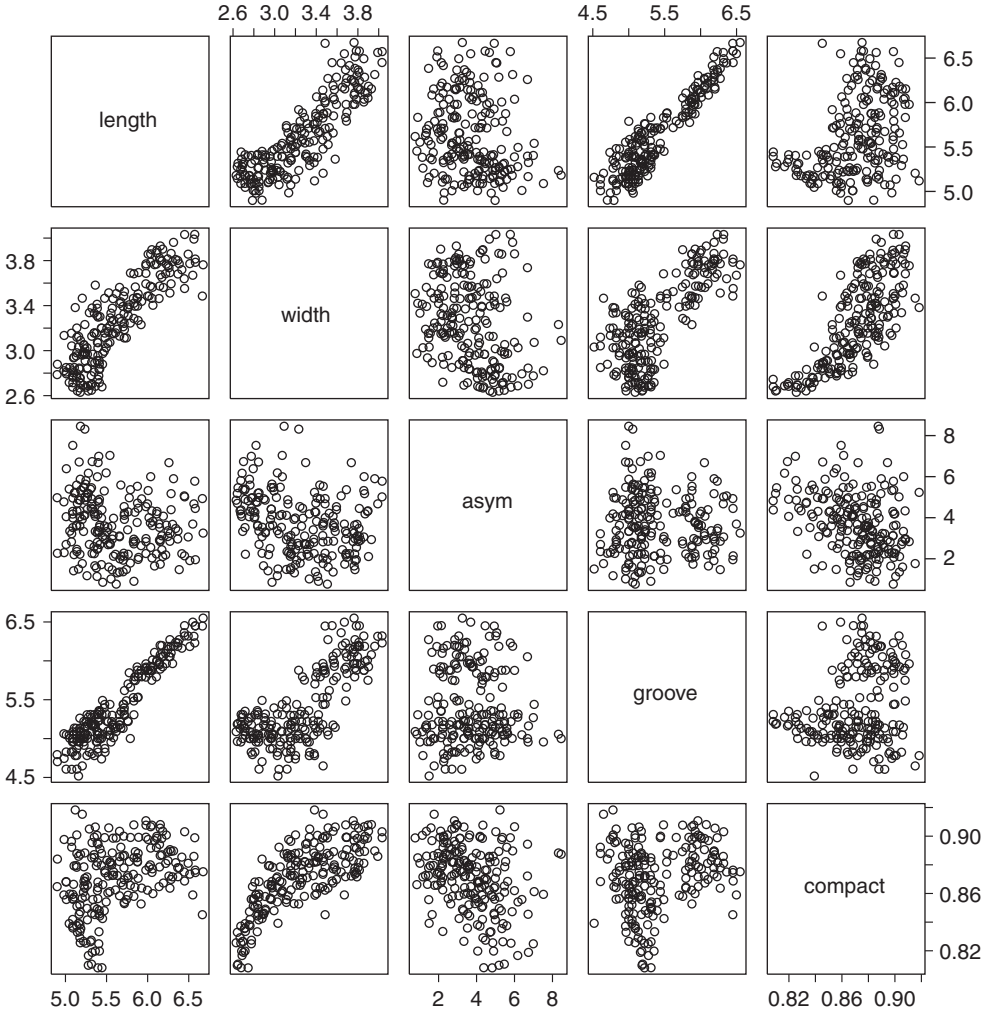

Figure 4.18 Scatterplot matrix in Example 4.2.7 of $K = 5$ wheat kernel variables from Table 4.7: X_1 = length, X_2 = width, X_3 = kernel asymmetry, X_4 = groove length, and X_7 = compactness. Source: Data from Charytanowicz et al. (2010).

column variable C_j, $i = 1, \ldots, I; j = 1, \ldots, J$. The categorization variables can be qualitative or quantitative, although the individual elements, m_{ij}, of $\mathbf{M}$ are assumed quantitative. Another enhanced graphic useful for visualizing the relationships among the m_{ij}s using the row–column ordering is known as a *heatmap*.

The heatmap is a simple display that takes advantage of modern graphical plotting hardware and software. First, from a user-designated color spectrum, it assigns a specific color to each possible value of m_{ij}. Then, it plots these colors in place of the actual m-values in the same location/order they appear in $\mathbf{M}$. (If true color is not available, the scheme can involve simple shades of gray.) The effect is a color-based graphic from which it is often easier to visualize patterns in the data than if one simply examined the raw values of m_{ij}.

If the R and C variables have a natural order or quantification, the graphic also doubles as a form of surface plot. Indeed, heatmaps are a common vehicle for displaying topographic or terrain-based information. More generally, however, they can serve as useful visualization for exploring patterns in very large matrices. The next example gives a simple illustration.

Example 4.2.8 Heatmap for correlation matrix. Heatmaps are often applied for visualizing patterns in correlation matrices, that is, matrices that contain the sample correlations, r_{ij}, over multiple pairs of variables X_i and X_j. The resulting matrix **R** is then square and symmetric of order J, where J is the number of variables under study. When J is small, visualization of the correlation structure is usually straightforward. When J grows past about 8–10, however, teasing out potential patterns among the correlations can become problematic. A heatmap of **R** provides a useful visualization tool.

For instance, Lin and Bhattacherjee (2010) presented correlations among $J = 19$ variables from a psychometric study of computer-game users. Of interest was the users' acceptance of evolving online and interactive gaming/entertainment technologies. The 19 variables covered a range of measurement constructs, involving Usage Intention ('UI' – 3 variables), Attitude ('AT' – 2 variables), Perceived Enjoyment ('PE' – 3 variables), Social Image ('SI' – 3 variables), Technical Quality ('TQ' – 3 variables), and Interaction Quality ('IQ' – 4 variables); see Lin and Bhattacherjee (2010, Appx. A). The resulting correlations, shown in Table 4.8, are used to examine how the variables interact with each other. (The ordering in the table is arbitrary; for optimal use of the heatmap, the different variables are listed together by pertinent construct groups.)

The heatmap corresponding to the correlation matrix in Table 4.8 appears in Figure 4.19. The basic heatmap can be constructed in **R** using the command

```
> heatmap( R, Rowv=NA, Colv=NA, scale='none', revC=T,
                      col=heat.colors(256), margins=c(4,4) )
```

where the input argument `R` is the correlation matrix of values from Table 4.8. The `Rowv=` and `Colv=` options provide tree-structured diagrams (called *dendrograms*; see Section 9.4.1 or 11.1.1) that add classification capabilities to the basic heatmap; these are suppressed here. The `revC=T` option reverses the column order for plotting, while `col=` calls for the 'hot' red-to-white color spectrum seen in the figure. (Users are encouraged to experiment with the wide capabilities of the `heatmap` command.)

The heatmap in Figure 4.19 quickly highlights a number of patterns, including a grouping of high correlations among the IQ variables and among the TQ variables (but not between them). Relatively large correlations are also seen among and between the UI and AT variables. The AT variables – particularly AT3 – also correlate marginally with the TQ variables. Other patterns are more muted, with correlations closer to 0 (darker color) appearing throughout the map. Overall, the heatmap gives a rapid and effective visualization of the correlation structure seen with these variables. □

Heatmaps have surprisingly inherent flexibility and come in a number of versions; see Exercises 4.20 and 11.4c. A closely related graphic is the *choropleth map* and its many variants, popular in cartographic displays of statistical data (Stewart and Kennelly 2010). (For choropleth maps in **R**, one can use, e.g., the external *ggplot2* package.) With increasing availability of powerful computer graphic and visualization hardware, use of this statistical display is limited only by the analyst's imagination.

Table 4.8 A 19 × 19 Correlation matrix among $J = 19$ variables in Example 4.2.8 reporting gamer acceptance of online/interactive entertainment technologies (upper triangular portion only; lower triangle is symmetric).

	UI1	UI2	UI3	AT1	AT2	AT3	PE1	PE2	PE3	
UI1	1	0.73	0.40	0.42	0.43	0.31	0.24	0.28	0.27	
UI2		1	0.51	0.48	0.50	0.36	0.27	0.30	0.24	
UI3			1	0.38	0.49	0.36	0.25	0.19	0.22	
AT1				1	0.64	0.45	0.33	0.38	0.29	
AT2					1	0.50	0.32	0.36	0.32	
AT3						1	0.25	0.36	0.31	
PE1							1	0.56	0.33	
PE2								1	0.53	
PE3									1	
	SI1	SI2	SI3	TQ1	TQ2	TQ3	IQ1	IQ2	IQ3	IQ4
UI1	0.22	0.28	0.20	0.18	0.17	0.19	0.17	0.13	0.23	0.18
UI2	0.23	0.37	0.24	0.23	0.23	0.26	0.21	0.22	0.27	0.22
UI3	0.20	0.31	0.30	0.16	0.18	0.17	0.24	0.26	0.30	0.29
AT1	0.41	0.38	0.45	0.36	0.31	0.36	0.17	0.22	0.21	0.28
AT2	0.30	0.34	0.32	0.34	0.33	0.36	0.19	0.23	0.30	0.31
AT3	0.28	0.23	0.25	0.50	0.49	0.49	0.27	0.31	0.32	0.35
PE1	0.39	0.26	0.53	0.15	0.15	0.24	0.19	0.15	0.19	0.27
PE2	0.35	0.20	0.36	0.23	0.24	0.25	0.27	0.28	0.32	0.34
PE3	0.32	0.25	0.25	0.21	0.22	0.19	0.20	0.19	0.19	0.24
SI1	1	0.55	0.52	0.28	0.22	0.27	0.25	0.26	0.18	0.26
SI2		1	0.46	0.21	0.21	0.20	0.16	0.14	0.19	0.20
SI3			1	0.15	0.17	0.17	0.23	0.20	0.25	0.24
TQ1				1	0.74	0.67	0.23	0.26	0.21	0.30
TQ2					1	0.71	0.21	0.24	0.22	0.23
TQ3						1	0.24	0.26	0.19	0.28
IQ1							1	0.74	0.66	0.58
IQ2								1	0.65	0.59
IQ3									1	0.60
IQ4										1

Source: Lin and Bhattacherjee (2010).

4.2.5 Time series plots*

When observations are taken as a sequence of recordings over consecutive time periods, they are known as a *time series* (Box et al. 2008). An illustration of this was seen in Example 3.5.2, with daily closing prices of the US Dow Jones Industrial stock index. Times series data are quite common in financial applications; however, they also occur in many other domains, such as medicine, environmental/ecological science, and sociology.

Time series are distinguished by their connection with a consecutive, increasing time unit, t. The recordings are usually equally spaced so that the observation X_t may be indexed against

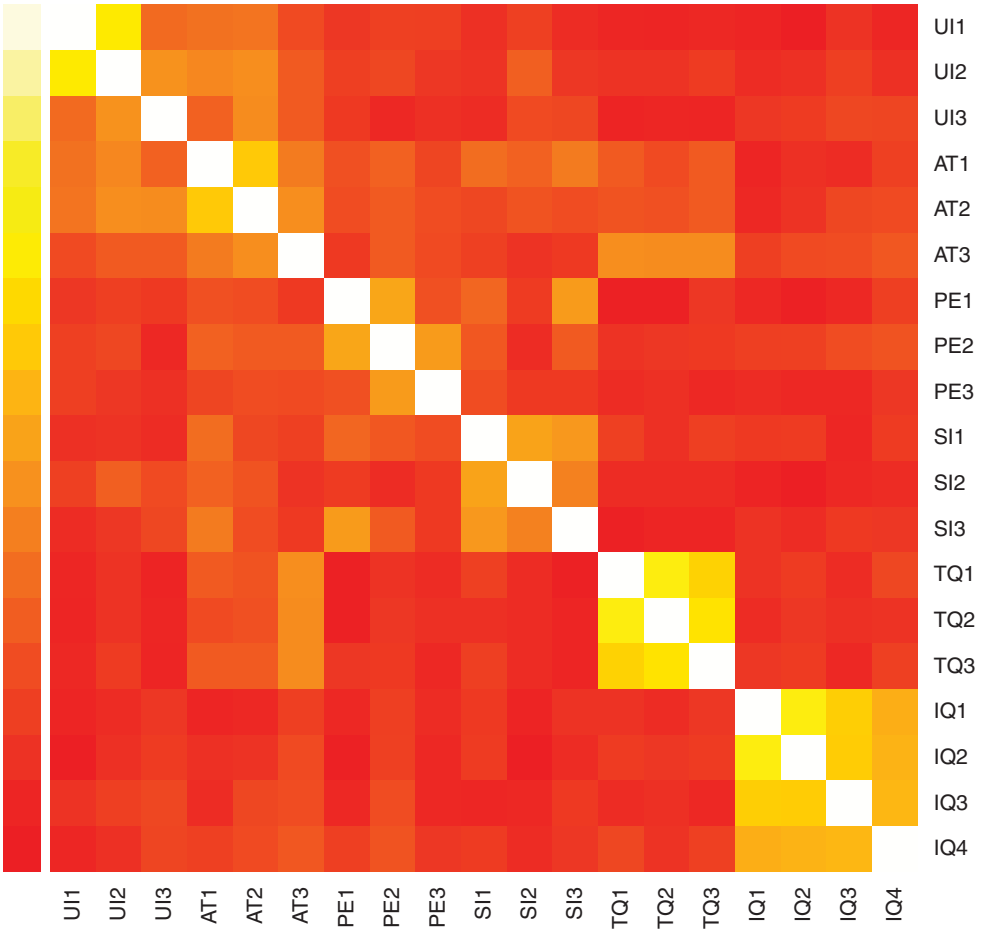

Figure 4.19 Heatmap of gaming user correlation matrix from Example 4.2.8. Lighter colors indicate correlations near 1, darker colors near 0. Scale at left gives the correlation spectrum from 1 (white, top) to 0 (dark, bottom). Source: Data from Lin and Bhattacherjee (2010).

$t = 1, \ldots, n$. If so, a graphic device to visualize any temporal patterns in the data is known as a *time series plot*, or sometimes also a *trace plot*, because it traces the progression of the process for each successive X_t. The concept is essentially a scatterplot of X_t versus t, although the consecutive points are usually connected with lines to better explicate the temporal progression. (If the points are dense enough, as in Figure 3.1, connecting the points may not be necessary. Figure 3.1 also illustrated overlay of a moving average to visualize smoothed patterns in the times series.)

Example 4.2.9 Nightingale's Mortality Data. A famous data set in statistical graphic design involves deaths due to differing causes in the British Army during the Crimean War, as reported by Florence Nightingale (1858). Entitled 'Diagram of the Causes of Mortality in

the Army in the East,' Nightingale used a novel infographic (now) known as a polar area plot (see http://www.sciencenews.org/view/generic/id/38937/description/Florence_Nightingale_The_passionate_statistician) to argue for improved sanitary conditions in hospitals and medical facilities (not just in war zones); see McDonald (2014).

As mentioned in Section 4.2.1, however, the visualization value of a polar area plot can often be improved. As the British mortality data were taken over time (months between April 1854 and March 1856), a time series plot provides an effective way to visualize the rates. It can also compare and highlight differences across causes of death over the same time scale, as was Nightingale's intent.

The data in Table 4.9 present the British Army's annualized mortality rates (per 1000 soldiers) during the Crimean War. The three separate causes of death are wounds and injury, preventable/mitigable disease, and 'all other,' where primary comparison is focused on the former two. (As above, only a selection of measurements is given in the table. The complete data are available at http://www.wiley.com/go/piegorsch/data_analytics.) To compare the mortality rates in Table 4.9 due to wounds and injury with those due to preventable diseases, Figure 4.20 graphs the two rates over time in the same time series plot. (The different series are indicated by different line styles, but for greater effect, contrasting colors could also be used; cf. Gelman and Unwin (2013, Fig. 6).] The difference between the two series is clear. The complexity other graphic devices have used to illustrate this feature belie their value, however; the simple time series plot does the job quite effectively. Also see Exercise 4.18.

Table 4.9 Selected data on annualized mortality (per 1000) of British casualties during the Crimean War from a larger set of 24 monthly observations, made famous by Florence Nightingale (1858).

Month	Wounds & injury	Preventable disease	All other causes
April 1854	0.0	1.4	7.0
May 1854	0.0	6.2	4.6
⋮	⋮	⋮	⋮
January 1855	30.7	1022.8	120.0
February 1855	16.3	822.8	140.1
⋮	⋮	⋮	⋮
March 1856	0.0	3.9	9.1

Source: http://understandinguncertainty.org/node/214

□

The basic time series plot can be enhanced if additional, multivariate information is available at each time t. For instance, suppose multiple sites or subjects provide data at each t. If the number of these replications is large, a simple plot of all the points may hide or distract from the larger trend(s). Conversely, only plotting the summary location values such as the median or mean of the replicates loses information about dispersion in the data. Better in this case is to replace the points with a summary graphic such as a (vertical) boxplot at each t. The bars at the medians can be replaced with dots or other easy-to-see symbols to help visualize trend(s), while the boxes and whiskers provide visual information on other facets of the time series process.

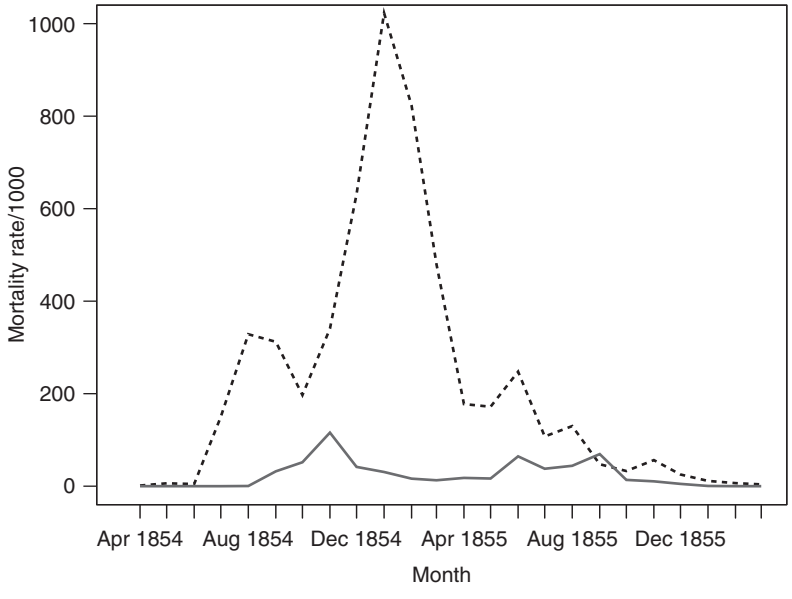

Figure 4.20 Overlayed time series plots in Example 4.2.9 to contrast mortality rates during the Crimean War; data from Table 4.9. Dashed line (– – –) is mortality due to preventable disease; solid gray line (—) is mortality due to wounds and injury. Source: Data from http://understandinguncertainty.org/node/214. Graphic adapted from Gelman and Unwin (2013).

Example 4.2.10 Rocky Mountain Front Range rainfall. The US National Center for Atmospheric Research (NCAR) collects data on various meteorologic phenomena for climatological modeling. For example, in a precipitation study in the US Rocky Mountains' Colorado Front Range, data were collected on $X = \{\text{Total summer (April–October) rainfall}\}$ over the period 1949–2001. The data were taken at a series of 56 separate recording stations, allowing for study of rainfall totals both within and across time points. Selected values appear in Table 4.10. (The complete data are available at http://www.wiley.com/go/piegorsch/data_analytics.)

Table 4.10 A selection of total summer rainfall totals (inches) from a larger study of US Rocky Mountain Front Range precipitation.

Year	Station 1	Station 2	Station 3	···	Station 54	Station 55	Station 56
1949	17.74	–	12.27	···	10.17	11.27	–
1950	11.99	–	10.37	···	7.49	9.10	–
⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮
2000	11.50	4.80	–	···	7.40	11.80	8.70
2001	9.90	9.90	–	···	6.40	7.50	10.80

Dashes (–) indicate no (complete) summer data were available at that station.
Source: http://www.image.ucar.edu/~nychka/FrontrangePrecip/.

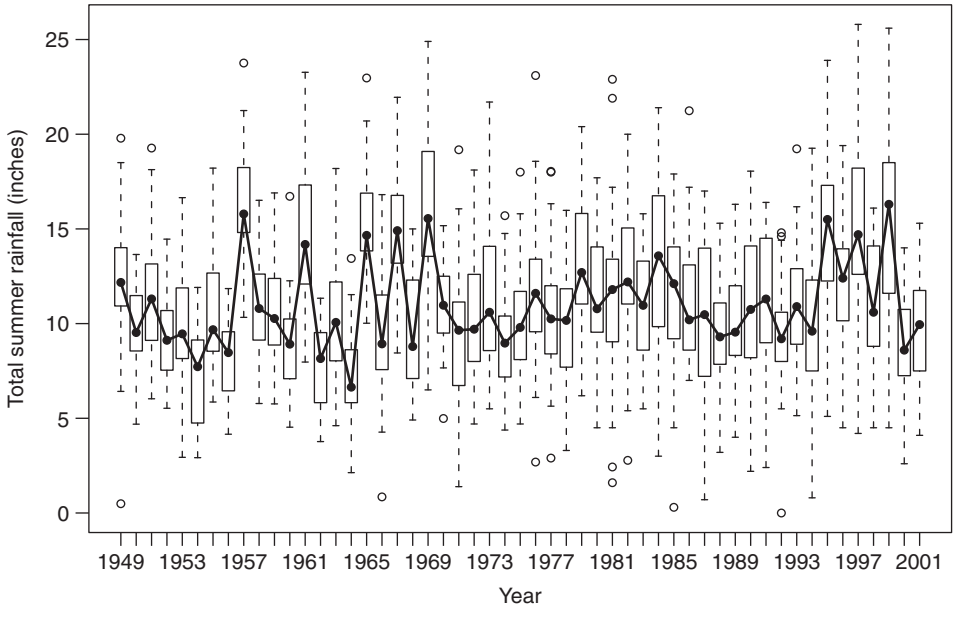

Figure 4.21 Time series plot with boxplot graphics in Example 4.2.10 of total summer rainfall (in inches) for US Rocky Mountain Rainfall data from Table 4.10. Source: Data from http://www.image.ucar.edu/~nychka/FrontrangePrecip/.

To visualize the rainfall totals over time, Figure 4.21 displays a time series plot, with embedded boxplots to visualize across-station variation. The graphic was created in **R** via

```
> boxplot( X, use.cols=FALSE, names=seq(1949,2001) )
```

where `X` is a 53×56 matrix whose rows are the 56 summer station totals. The default horizontal bars at the medians were replaced with dots using the options `medlty='blank'` and `medpch=19`. The median dots were connected with repeated use of the `lines()` function.

The time series plot shows wide variation in rainfall totals, which may prompt the NCAR's study of its extremes. The central tendency of the totals is between 7 and 15 inches per summer, but they can reach as high as 25 inches. The graphic also identifies a few very low points: see, for example, the possible lower outliers in 1949, 1966, 1985, or 1992. The latter is a record of 0 inches – that is, *no* rainfall – at Station #25 that summer. This may be a naturally occurring drought event, a mechanical/system recording malfunction, or just data entry error. Exploring which of these possibilities could account for the unusual data point(s) is worth deeper investigation. □

R has a number of ways to construct times series plots, including the powerful, omnibus `plot()` function and also `ts.plot()` and `plot.ts()`. Users should experiment with the various options to chose the best routine for their individual needs.

The material in this chapter only begins to illustrate the value statistical visualization can bring to the data analytic enterprise. Indeed, the static graphics presented here are only a first

step: dynamic and interactive data visualization is an evolving area, populated by members of both the statistical and infovis communities (Henderson 2004; Huang et al. 2012; Young et al. 2006); also see the web site http://www.gapminder.org (and Exercise 4.15).

To learn more about statistical visualization, see the advanced treatments in, for example, Murrell (2011) or Wickham (2012) (and the references therein), the seminal works by Cleveland (1993) and Wilkinson (2005), or the popular expositions by Tufte (2001) and Yau (2011). The popularity of graphic design also allows for some humorous, tongue-in-cheek perspectives; see http://xkcd.com/688/ or http://www.theonion.com/articles/americas-most-popular-charts,7492/ and the associated web sites.

Exercises

4.1 Return to the following data sets and construct a strip chart for each. Comment on the patterns the graphic exposes, if any.

(a) The wheat kernel data in Table 3.1.

(b) The square-root-transformed wheat kernel data in Exercise 3.15b.

(c) The log-transformed circulatory-disease mortality data in Exercise 3.17.

(d) The husbands' heights data in Exercise 3.2.

4.2 Recall the circulatory-disease mortality data in Example 4.1.1. Construct a dot plot of the data and comment on the patterns that emerge. Compare this to the strip chart in Figure 4.1.

4.3 Return to the following data sets and construct a boxplot for each. Include a rug at bottom with the data values. What does the graphic indicate about the distribution of the data?

(a) The hazard vulnerability data from Exercise 3.3.

(b) The solar radiation data from Exercise 3.4.

4.4 Return to the circulatory-disease mortality data in Table 3.4 and construct a stemplot. (If using **R**, apply both `scale=1` and `scale=2` in `stem()` to see the effect of expanding the scale.) How does this compare to the histogram for these data plotted in Figure 4.6?

4.5 For the following data, construct a histogram (indicate the bin selection algorithm you use). Overlay a kernel density estimator and include a rug at bottom with the data values. Comment on how these compare to the corresponding boxplot in Exercise 4.3 in terms of summarizing features of the data.

(a) The hazard vulnerability data from Exercise 3.3.

(b) The solar radiation data from Exercise 3.4.

4.6 Construct a histogram for the myocardial infarction data in Table 4.2. Include a rug at bottom with the data values. How does the graphic compare to the stemplot (if viewed at 90° rotation) plotted in Figure 4.5?

4.7 Recall that in Examples 4.1.3 and 4.1.5, a right skew was evidenced with the circulatory-disease mortality data from Table 3.4. With this in mind, explore the following alternatives:

(a) Apply Scott's normal reference rule for selecting the number of bins, using the results in Exercise 3.19. Plot the corresponding histogram (include a rug at bottom with the data values). Does it differ substantively from the plot in Figure 4.6?

(b) Apply a (natural) logarithmic transformation to the original mortality rates as in Example 3.17 and construct (i) a box plot, (ii) a stem plot, and (iii) a histogram for the transformed rates (overlay a kernel density estimator on the histogram and include a rug at bottom with the data values). Did the transformation act to alleviate the skew? Also graph (iv) a normal quantile plot. Do the transformed data appear normal?

4.8 Insert the Gaussian kernel $K_{\text{Gauss}}(t) = (2\pi)^{-1/2}\ \mathrm{e}^{-(1/2)t^2}$ into the general expression for a kernel density estimator in Equation (4.1). Show that the result can be written as a weighted average of normal p.d.f.s, each with constant variance h^2. What are the weights in this 'weighted average?' Can you imagine changing these weights to produce a different, specialized density estimator? How?

4.9 Return to the energy load data from Table 4.4. To explore distributional features of the individual variables $X = \{\text{Heating load}\}$ and $Y = \{\text{Cooling load}\}$, construct the following summary graphics. In all cases, comment on the visual similarities or differences between the variables' distributions. Compare this with the conclusions reached using the Q–Q plot in Example 4.2.4.

(a) Construct stemplots for each variable. If you have the programming capability, display this as a side-by-side stemplot.

(b) Bin each variable using Scott's normal reference rule and plot the corresponding histograms. Overlay kernel density estimators and include a rug at bottom for each.

(c) Construct multiple boxplots (on the same plot) for each variable.

4.10 For the NJ driving speeds data in Table 4.3, construct a pie chart for the state-of-origin frequencies in Example 4.2.1. If in **R**, use the `pie()` function. Comment on the quality of information the plot presents.

4.11 Return to the husbands' and wives' heights data in Exercise 3.12.

(a) Construct a multiple boxplot graphic for both variables. How do the two individual boxplots compare with one another?

(b) Graph a Q–Q plot to compare the two variables. Does the information in the Q–Q plot corroborate that seen in the multiple boxplot?

(c) Graph a scatterplot for the paired data. Comment on any patterns.

4.12 The solar radiation data in Exercise 3.4 are actually part of a larger collection, where $K = 4$ different photometers recorded median extreme ultraviolet (XUV) irradiance over the 0.1–7.0 nm range. Selected data are

Device	XUV irradiance				
D1	0.000004	0.000004	···	0.001020	0.001220
D2	0.000123	0.000124	···	0.001140	0.001210
D7	0.000041	0.000044	···	0.001540	0.001560
D9	0.000241	0.000242	···	0.000788	0.000838

(Download the full data set at http://www.wiley.com/go/piegorsch/data_analytics.) Graph a multiple boxplot for XUV irradiance across all four devices. What patterns do you discover?

4.13 A novel combination of a boxplot and a kernel density estimate is known as a *violin plot*. This essentially overlays the density estimate on the boxplot and then reflects it around the boxplot to give a stylize graphic that looks somewhat like a violin. Violin plots are available in **R** via the external *vioplot* package. Download this or any other software that can produce violin plots and use it to build a multiple violin plot for the NJ driving speed data in Example 4.2.3. Compare the graphic to the multiple boxplot in Figure 4.14. Does the new plot provide a more descriptive perspective on these data?

4.14 Return to the college admissions data in Example 3.3.6.

(a) Construct a Q–Q plot to compare the paired variables $X = \{\text{ACT score}\}$ and $Y = \{\text{Class rank}\}$. How do the patterns of variation compare?

(b) Graph a scatterplot for the paired data. Comment on any patterns.

4.15 The bubble plot gained in popularity after its adroit use by Swedish statistician/physician Hans Rosling to visualize worldwide life expectancy as a function of national gross domestic product (GDP); see http://www.gapminder.org/world. To imitate his famous graphic, consider the variables

$W = 2011$ Total population,
$X = 2011$ GDP per capita (in 2000 dollars, inflation-adjusted), and
$Y = 2011$ Life expectancy at birth

for a given nation. There are $n = 149$ nations in the archives at http://www.gapminder.org for which all three values are available. From them, we can form the triplets (W_i, X_i, Y_i):

$$\begin{array}{cc} (86\,165, 629.955, 51.093) & (3\,215\,988, 2255.225, 73.131) \\ (89\,612, 11601.630, 75.901) & (245\,619, 5671.912, 74.402) \\ \vdots & \vdots \\ (548\,377, 757.401, 75.181) & (13\,474\,959, 347.746, 51.384) \end{array}$$

(Download the full data set at http://www.wiley.com/go/piegorsch/data_analytics.)

(a) Construct a bubble plot: plot $Y = \{\text{Life expectancy}\}$ against $X = \{\text{GDP per capita}\}$ and build bubbles whose areas are proportional to $W = \{\text{Total population}\}$. What pattern(s) emerge?

(b) While the number of nations providing data triplets here is $n = 149$, some might argue that the number of 'subjects' is much larger: each W_i is (an estimate of) that nation's population, as contributing to the bubble sizes in the plot. What is the total, $W_+ = \sum_{i=1}^{149} W_i$? What does this number represent?

4.16 For the automobile fuel economy data in Table 4.6, build a scatterplot matrix for the three variables. Does the MPG versus engine displacement subplot reproduce features of the bubble plot in Figure 4.17? What other patterns become evident in the scatterplot matrix?

4.17 Recall the full wheat kernel data in Table 4.7. Construct a scatterplot matrix for all $K = 7$ variables. Do any new or unexpected patterns emerge? Also calculate all pairwise correlations between the variables and comment on any correspondences.

4.18 For Nightingale's mortality data in Table 4.9, expand the visual in Figure 4.20 to include all three causes of mortality. How does this new graphic add to the story?

4.19 A classic set of time series data that led to knowledge discovery involved the 11-year solar sunspot cycle. In 1849, J.R. Wolf of the Berne Observatory proposed and later published (Wolf 1856) an index, with which he intended to quantify the relative pattern of dark prominences ('spots') seen on the surface of Earth's sun. From this, Wolf amassed a database of relative sunspot indices, which has since been continuously updated. Different versions of the series exist, depending on which data source one accesses; here, consider a set studied by Piegorsch and Bailer (2005) with the monthly indices from 1900 to 1977. The data are available online at http://www.wiley.com/go/piegorsch/data_analytics; a sample follows:

Month/year	Spot index	Month/year	Spot index
Jan 1900	9.4	Feb 1900	13.6
Mar 1900	8.6	Apr 1900	16.0
⋮	⋮	⋮	⋮
Jan 1977	23.1	Feb 1977	8.7

(a) Build a time series plot of the sunspot activity as a function of time. Is the famous 11-year cycle evident?

(b) To visualize the 11-year cycle better, apply an 11-year moving average smoother (Section 3.5.2) to the raw sunspot data and overlay the smoothed response on the raw plot.

4.20 An interesting combination of a time series plot with a heatmap occurs when one of the dimensions of a heatmap's matrix $\mathbf{M}$ is time, producing a *temporal heatmap*. For example, the US Energy Information Administration (EIA) provides data on US energy production and consumption, available at http://www.eia.gov/electricity/data/browser/. From this source, monthly rates are available for each US state's (and the District of Columbia's) electric power consumption (in 10^7 MMBtu, by all fuel types) from January 2001 to May 2013. A sample is given as follows. (The full data are given online at http://www.wiley.com/go/piegorsch/data_analytics.)

Selected monthly energy consumption by US state, in 10^7 MMBtu, all fuel types

Region	State	Jan 2001	Feb 2001	···	May 2013
New England	Connecticut	14	11	···	10
	Maine	9	8	···	1
	⋮	⋮	⋮	⋱	⋮
West South Central	Oklahoma	44	38	···	46
	Texas	240	202	···	242
	⋮	⋮	⋮	⋱	⋮
Pacific Contiguous	California	100	87	···	57
	Oregon	11	11	···	4
	Washington	18	17	···	2
Pacific Noncontiguous	Alaska	5	5	···	4
	Hawaii	8	7	···	7

View the data as a 51 × 149 matrix with time in months as the column variable and construct from this a temporal heatmap. (If using **R**'s `heatmap` command, do not standardize the observations across rows, as is the default with that function.) Consider experimenting with color schemes to improve the visualization. Comment on the patterns that appear.

5

Statistical inference

The power of data analysis is best exemplified when its inferential engine is engaged, extending the summarizations available from simple statistical description. Statistical inference is designed to derive conclusions about a population, using information in random samples from that population. This chapter reviews the basic components of the inferential paradigm, building on the probability models described in Chapter 2 and on the descriptive methods reviewed in Chapters 3 and 4. Some readers will find much of this to be a review; others may benefit from a careful reading. The material begins with basic theoretical definitions and concepts and then segues to an introduction to some standard inferential methods. The material stands as a final foundation and gateway of the larger aspects of statistical data analytics that begin with supervised learning and regression modeling in Chapter 6 and beyond.

5.1 Parameters and likelihood

Suppose that a random sample $X_i \sim \text{i.i.d. } f_X(x)$, $i = 1, \ldots, n$ has been observed as in Section 3.1 and that a formal model is taken for the probability mass function (p.m.f.) or probability density function (p.d.f.) $f_X(x)$. The basic tenet of statistical inference is that the available information about $f_X(x)$ is contained within the sample's data. Although the various models for $f_X(x)$ can vary widely – cf. Section 2.3 – they usually exhibit one common feature: they depend on one or more *parameters* to describe variation in X. For example, the normal distribution in Section 2.3.9 has two parameters: the population mean μ and the population variance σ^2. These are sufficient to completely characterize the normal p.d.f. In most random samples, however, these values are not known in advance, and so they are referred to as *unknown* parameters. The process of statistical description is used to estimate these unknown quantities, but it is statistical inference that connects the estimates with statements about the population they describe.

The $p \geq 1$ unknown parameters from a p.m.f. or p.d.f. are assembled together into a *parameter vector* $\boldsymbol{\vartheta} = [\theta_1 \cdots \theta_p]^{\mathrm{T}}$, where θ_j is used as generic notation for the jth parameter in the

Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery,
First Edition. Walter W. Piegorsch.

www.wiley.com/go/piegorsch/data_analytics

model, and superscript T denotes the transpose of a vector. (See Appendix A for a review of vector and matrix terminology.) Similarly, the random sample can be collected together into a random vector $\mathbf{X} = [X_1 \cdots X_n]^{\mathrm{T}}$.

To explicate the dependence on $\boldsymbol{\vartheta}$ in the model, the notation for the p.m.f. or p.d.f. is extended into $f_X(x|\boldsymbol{\vartheta})$. The bar '|' is borrowed from the conditional probability notation in Section 2.1.1 – that is, probability mass or density for X is 'conditional' on the particular value $\boldsymbol{\vartheta}$ takes on – although no assumption is made that $\boldsymbol{\vartheta}$ is itself random. Similar to (2.11), under simple independent, identically distributed (i.i.d.) sampling the corresponding joint p.m.f. or p.d.f. for the entire random sample can be constructed by appeal to the Multiplication Rule (2c) from Section 2.1.1. The result is the product of the individual p.m.f.s or p.d.f.s:

$$f_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n|\boldsymbol{\vartheta}) = \prod_{i=1}^{n} f_X(x_i|\boldsymbol{\vartheta}) \,. \tag{5.1}$$

If, as introduced above, the available information about $\boldsymbol{\vartheta}$ is contained within the observations, then the joint probability function represents a statistical model for relating the data to $\boldsymbol{\vartheta}$, and vice versa. This fundamental concept was formalized by Fisher (1912, 1922) who essentially reversed the perspective in (5.1) and viewed the construct as a function of $\boldsymbol{\vartheta}$ given the information in the observed data $x_1, x_2, \ldots, x_n$. Fisher called this a *likelihood function*:

$$L(\boldsymbol{\vartheta}; x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} f_X(x_i|\boldsymbol{\vartheta}) \,.$$

As logarithms of products are sums of logarithms, it is often easier to work with the natural logarithm of the likelihood, known as the *log-likelihood function*:

$$\ell(\boldsymbol{\vartheta}) = \sum_{i=1}^{n} \log\{f_X(x_i|\boldsymbol{\vartheta})\}. \tag{5.2}$$

The log-likelihood in (5.2) quantifies model components represented in the data by $\boldsymbol{\vartheta}$. This leads naturally to its use as a measure of *information*. With only a single unknown parameter θ (so $p = 1$), we say the *Fisher information number* is the expected value of the negative second derivative of $\ell(\theta)$:

$$\mathcal{F}(\theta) = E[-\ell''(\theta)]. \tag{5.3}$$

Although the notation camouflages it, $\ell''(\theta)$ here is a function of the data, $X_1, \ldots, X_n$. Therefore, one can calculate its negative expectation with respect to the joint p.m.f. or p.d.f. in (5.1).

If there are $p > 1$ unknown parameters, the information numbers for each θ_j are collected together into a *Fisher information matrix*. Start with the mixed partials of $\ell(\boldsymbol{\vartheta})$, $h_{jk} = \partial^2\ell(\boldsymbol{\vartheta})/\partial\theta_j\partial\theta_k$, and from these, build the *Hessian matrix* of second partial derivatives for $\ell(\boldsymbol{\vartheta})$,

$$\mathbf{H}_\ell = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1p} \\ h_{21} & h_{22} & \cdots & h_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ h_{p1} & h_{p2} & \cdots & h_{pp} \end{bmatrix}. \tag{5.4}$$

Here again, the Hessian elements are functions of the data, $X_1, \ldots, X_n$. Therefore, one can evaluate their negative expected values, say, $\mathcal{F}_{jk}(\boldsymbol{\vartheta}) = E[-h_{jk}] = E[-\partial^2\ell(\boldsymbol{\vartheta})/\partial\theta_j\partial\theta_k]$ and

array them together into the Fisher information matrix

$$\mathbf{F}(\boldsymbol{\vartheta}) = \begin{bmatrix} \mathcal{F}_{11}(\boldsymbol{\vartheta}) & \mathcal{F}_{12}(\boldsymbol{\vartheta}) & \dots & \mathcal{F}_{1p}(\boldsymbol{\vartheta}) \\ \mathcal{F}_{21}(\boldsymbol{\vartheta}) & \mathcal{F}_{22}(\boldsymbol{\vartheta}) & \dots & \mathcal{F}_{2p}(\boldsymbol{\vartheta}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{F}_{p1}(\boldsymbol{\vartheta}) & \mathcal{F}_{p2}(\boldsymbol{\vartheta}) & \dots & \mathcal{F}_{pp}(\boldsymbol{\vartheta}) \end{bmatrix}. \tag{5.5}$$

In effect, $\mathbf{F}(\boldsymbol{\vartheta})$ is the negative expected Hessian of the log-likelihood function. Note that, as is often the case, $\mathcal{F}_{jk}(\boldsymbol{\vartheta}) = \mathcal{F}_{kj}(\boldsymbol{\vartheta})$ so that the information matrix $\mathbf{F}(\boldsymbol{\vartheta})$ is symmetric. Since it is based on the expected values of log-likelihood derivatives, $\mathbf{F}(\boldsymbol{\vartheta})$ is also known as the *expected information matrix*.

5.2 Point estimation

To estimate an unknown parameter vector $\boldsymbol{\vartheta}$, we use quantitative information in the random sample, $\{X_1, X_2, \dots, X_n\}$ from $f_X(x|\boldsymbol{\vartheta})$. A variety of strategies can be applied to achieve this goal; what follows is a short review of established approaches. As mentioned in Section 3.3.1, the default notation for a point estimator of $\boldsymbol{\vartheta}$ places a circumflex accent ($\hat{}$) above the parameter; thus $\hat{\boldsymbol{\vartheta}}$ is a point estimator for $\boldsymbol{\vartheta}$ (unless a more-specific or traditional notation presents itself, such as $\overline{X}$ for a population mean μ).

It is important to note that an estimator, $\hat{\boldsymbol{\vartheta}}$, is based on data and, therefore, in the abstract is a function of the random variables $X_1, X_2, \dots, X_n$. Thus it is itself a random variable, with its own joint p.m.f. or p.d.f., say $f_{\hat{\boldsymbol{\vartheta}}}(\cdot)$. The distribution associated with any statistic from a random sample is called the *sampling distribution* of the statistic, and every individual point estimate will possess a consequent mean $E[\hat{\theta}_j]$ and variance $\mathrm{Var}[\hat{\theta}_j]$, along with between-estimate covariances $\mathrm{Cov}[\hat{\theta}_j, \hat{\theta}_k]$ $(j \neq k)$ in the multiparameter case. These are collected together into the estimator's covariance matrix

$$\mathrm{Var}[\hat{\boldsymbol{\vartheta}}] = \begin{bmatrix} \mathrm{Var}[\hat{\theta}_1] & \mathrm{Cov}[\hat{\theta}_1, \hat{\theta}_2] & \cdots & \mathrm{Cov}[\hat{\theta}_1, \hat{\theta}_p] \\ \mathrm{Cov}[\hat{\theta}_2, \hat{\theta}_1] & \mathrm{Var}[\hat{\theta}_2] & \cdots & \mathrm{Cov}[\hat{\theta}_2, \hat{\theta}_p] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[\hat{\theta}_p, \hat{\theta}_1] & \mathrm{Cov}[\hat{\theta}_p, \hat{\theta}_2] & \cdots & \mathrm{Var}[\hat{\theta}_p] \end{bmatrix}, \tag{5.6}$$

as in (2.12).

The standard deviation of a univariate point estimator is the square root of its variance, $\sqrt{\mathrm{Var}[\hat{\theta}_j]}$. This will often be a function of θ_j (and perhaps other elements of $\boldsymbol{\vartheta}$) and in practice must be estimated as well. To do so, replace any unknown quantities by their individual point estimators, that is, replace θ_j with $\hat{\theta}_j$ wherever it appears. The result is called the *standard error* of θ_j:

$$\mathrm{se}[\hat{\theta}_j] = \sqrt{\mathrm{Var}[\hat{\theta}_j]|_{\boldsymbol{\vartheta}=\hat{\boldsymbol{\vartheta}}}}, \tag{5.7}$$

where the vertical bar notation in the expression indicates evaluation at $\boldsymbol{\vartheta} = \hat{\boldsymbol{\vartheta}}$. The standard error can be interpreted as a measure of uncertainty associated with estimating the population parameter θ_j.

5.2.1 Bias

In practice, we would expect the theoretical mean of a point estimator to be (near) the value it is estimating, that is, $E[\hat{\theta}_j] = \theta_j$. When this occurs, $\hat{\theta}_j$ is called an *unbiased estimator* of θ_j. Conversely, if $E[\hat{\theta}_j] \neq \theta_j$, then $\hat{\theta}_j$ is a *biased estimator*. Unbiased estimators exhibit a kind of "scientific objectivity". This is considered an optimal feature if it is achieved.

An estimator's *bias* is the difference between its expected value and the target parameter: $\text{Bias}[\hat{\theta}_j] = E[\hat{\theta}_j] - \theta_j$. Obviously, an unbiased estimator has $\text{Bias}[\hat{\theta}_j] = 0$. To quantify variation about the target parameter, the *mean squared error* of an estimator is $\text{MSE}[\hat{\theta}_j] = E[(\hat{\theta}_j - \theta_j)^2]$, which can be shown to decompose into $\text{MSE}[\hat{\theta}_j] = \text{Var}[\hat{\theta}_j] + \text{Bias}^2[\hat{\theta}_j]$.

Example 5.2.1 **Sample mean from Normal population.** Take a random sample from a normal distribution: $X_i \sim \text{i.i.d. N}(\mu, \sigma^2)$, $i = 1, \ldots, n$. In Section 2.3.9, it was seen that the sample mean is then also normally distributed, as $\overline{X} \sim \text{N}\left(\mu, \sigma^2/n\right)$. Since $\overline{X}$ can be viewed as a point estimator of μ, this says that the sampling distribution of $\overline{X}$ is normal, with expected value equal to μ and with variance equal to σ^2/n. Further, we see $E[\overline{X}] = \mu$, so $\overline{X}$ is an unbiased estimator of μ.

To find the standard error under (5.7) of the point estimator $\overline{X}$, recognize that $\sqrt{\text{Var}[\overline{X}]} = \sqrt{\sigma^2/n}$. If σ^2 is unknown – as is almost always the case in practice – replace it with its point estimator: the sample variance from (3.5), $S^2 = \sum_{i=1}^{n} (X_i - \overline{X})^2/(n-1)$. The standard error of $\overline{X}$ is then $\text{se}[\overline{X}] = S/\sqrt{n}$. □

Since a point estimator is a function of the sample, $X_1, \ldots, X_n$, it also depends on the sample size n. (Indeed, some authors write $\hat{\boldsymbol{\vartheta}}_n$ to remind the reader of this.) As n grows large, it is natural to expect that the distributional properties of $\hat{\boldsymbol{\vartheta}}_n$ will be affected. It is also natural to ask whether or how the estimator performs as $n \to \infty$. The sampling distribution of $\hat{\boldsymbol{\vartheta}}_n$ as $n \to \infty$ is known as the *asymptotic distribution* of the estimator, and in many cases, it has a common or known form.

Example 5.2.2 **Asymptotics for the sample mean.** From the central limit theorem in Section 2.3.9, we saw that for a random sample from *any* p.m.f. or p.d.f. $f_X(x|\mu, \sigma^2)$ with finite mean $E[X_i] = \mu$ and finite variance $\text{Var}[X_i] = \sigma^2$, the sampling distribution of $\overline{X}$ approaches $\text{N}\left(\mu, \sigma^2/n\right)$ as $n \to \infty$. This says that the asymptotic distribution of $\overline{X}$ has a normal (Gaussian) form.

Further, because the expected value of this asymptotic distribution is μ, one can say that $\overline{X}$ is 'asymptotically unbiased' for μ. In very large samples, it is reasonable to expect that $\overline{X}$ will come close to μ and that it will do so in a manner approximating normal (Gaussian) variation. (If the original random sample possesses a parent normal distribution, this will be an *exact* relationship, as in Example 5.2.1.)

In practice, every case will differ: the central limit theorem can take effect fairly quickly for some nonnormal parent distributions, where approximate normality can be valid for n as small as 10 or 20. Other distributions, especially many discrete or highly asymmetric forms, can require n upwards of 100 or more before variation in $\overline{X}$ begins to appear normal. □

5.2.2 The method of moments

With only limited assumptions made on the distribution of the X_is, a simple, yet useful estimation approach can be derived. Known as the *method of moments* (MOM), the procedure

equates the first $p \geq 1$ population moments of X_i to their corresponding sample moments and then solves for the p unknown parameters.

Given $X_i \sim$ i.i.d. $f_X(x|\mathfrak{\vartheta})$, $i = 1, \ldots, n$, suppose that the (common) jth population moments are $E[X^j]$, $j = 1, \ldots, p$, and that each of these p moments depends on the elements of $\mathfrak{\vartheta}$. Let the corresponding sample moments be $M_j = \frac{1}{n}\sum_{i=1}^{n} X_i^j$. Then to find an MOM estimator for $\mathfrak{\vartheta}$, set each $E[X^j]$ equal to M_j and solve for the θ_js. This will be a p-dimensional system of equations defined by each individual *estimating equation* $E[X^j] = M_j$, $j = 1, \ldots, p$. Technically, all that is required to implement the MOM is knowledge of p distinct population moments for X. Full specification of $f_X(x|\theta)$ is not necessary.

Notice that the MOM construction for the estimating equations is not unique; one could instead set $E[X^{j+1}] = M_j$ for every j, or $E[X^{2j}] = M_j$, and so on. (However, one would need substantial motivation for doing so.)

Example 5.2.3 **Method of Moments for Bernoulli sample (Example 2.3.2, continued).** Suppose a random sample is taken of i.i.d. Bernoulli observations, each with constant, unknown probability π: $X_i \sim$ i.i.d. Bin$(1, \pi)$, $i = 1, \ldots, n$. To find an MOM estimator for π, recall that the first moment of a Bernoulli random variable is simply $E[X_i] = \pi$. Thus the MOM estimator is built from the single estimating equation that equates the first sample moment with π. But, the first sample moment is the sample mean, so this becomes $\frac{1}{n}\sum_{i=1}^{n} X_i = \pi$. The solution here is trivial: 'solve' for π and take as the MOM estimator

$$\hat{\pi}_{\text{MOM}} = \frac{\sum_{i=1}^{n} X_i}{n}.$$

Notice that $\hat{\pi}_{\text{MOM}}$ is the sample proportion, that is, the number of Bernoulli successes out of the total n subjects sampled. This is an intuitively natural estimator to use for this setting.

For instance, recall in Example 2.3.2 the purchasing study of a retail outlet's $n = 1024$ affinity customers who might make a purchase during a marketed sale. Suppose after the event concluded that $\sum_{i=1}^{n} X_i = 506$ of the customers actually made a sale purchase. The MOM estimator is then $\hat{\pi}_{\text{MOM}} = 506/1024$ or 49.41% of the outlet's affinity base. (These data will be studied further in the following examples.)

In passing, recall that the Bernoulli distribution is a special case of the binomial and that sums of i.i.d. binomials (and Bernoullis) are also binomial. Here, $Y = \sum_{i=1}^{n} X_i \sim$ Bin(n, π), and so one can also write the MOM estimator as $\hat{\pi}_{\text{MOM}} = Y/n$. □

5.2.3 Least squares/weighted least squares

Another traditional method for estimating an unknown parameter is known as the *method of least lquares* (LS). The method does not require specification of the likelihood and is thus quite general. Suppose generically that the observations each have expected value $E[X_i] = \theta$. Then, the LS method estimates θ by minimizing the objective quantity $\mathcal{D} = \sum_{i=1}^{n} (X_i - \theta)^2$, that is, the sum of squared deviations of each observation from θ.

Example 5.2.4 **Least squares estimator for a population mean.** Take a random sample $X_i \sim$ i.i.d. $f_X(x|\mu)$ $i = 1, \ldots, n$, where the unknown, finite population mean is $E[X_i] = \mu$ (and make no other assumptions on X_i). To estimate μ, the LS method minimizes $\mathcal{D} = \sum_{i=1}^{n} (X_i - \mu)^2$. Different strategies can be applied here to find the minimum.

For instance, expanding the square produces

$$\mathcal{D} = \sum_{i=1}^{n}(X_i^2 - 2\mu X_i + \mu^2) = \left(\sum_{i=1}^{n} X_i^2\right) - 2n\overline{X}\mu + n\mu^2 \,, \tag{5.8}$$

using the fact that $\sum_{i=1}^{n} X_i = n\overline{X}$.

Now, view $\mathcal{D}$ as a function of μ and in particular recognize that (5.8) describes a parabola: $\mathcal{D}(\mu) = a\mu^2 + b\mu + c$, with coefficients $a = n$, $b = -2n\overline{X}$, and $c = \sum_{i=1}^{n} X_i^2$. The vertex of the parabola occurs at $\mu = -b/(2a) = -(-2n\overline{X})/(2n) = \overline{X}$. Further, the parabola is convex, because its quadratic coefficient is strictly positive ($a = n > 0$). Thus the parabola in (5.8) attains a minimum at its vertex. But, this says that the LS objective function $\mathcal{D}(\mu)$ is minimized at the sample mean, so we take $\hat{\mu}_{\text{LS}} = \overline{X}$. The least squares estimator of a population mean is the sample mean. (Also see Exercise 5.3.) □

The LS method applies more generally when a function of $p > 1$ parameters, $g(\theta_1, \theta_2, \ldots, \theta_p)$, is chosen to model $E[X_i]$. Then, to estimate the unknown parameters, the LS approach minimizes the squared difference between the X_is and $g(\cdot)$:

$$\mathcal{D} = \sum_{i=1}^{n} \{X_i - g(\theta_1, \theta_2, \ldots, \theta_p)\}^2$$

This results in a p-dimensional system of equations which when solved produces a vector of estimators $\hat{\boldsymbol{\vartheta}} = [\hat{\theta}_1\, \hat{\theta}_2 \cdots \hat{\theta}_p]^{\mathrm{T}}$.

If it is felt that the observations contribute information about $\boldsymbol{\vartheta}$ in a heterogeneous manner, one can apply *weighted least squares* (WLS): minimize the weighted objective quantity

$$\mathcal{D}_{\mathbf{w}} = \sum_{i=1}^{n} w_i \{X_i - g(\theta_1, \theta_2, \ldots, \theta_p)\}^2 \,,$$

where the weights, $w_i \geq 0$, are chosen to account for the differential quality of each X_i. For example, suppose $\text{Var}[X_i]$ varies with i so that some observations are more variable and, therefore, less precise regarding the information they provide on $\boldsymbol{\vartheta}$. Then, we typically take the weights proportional to the reciprocals of these variances: $w_i \propto 1/\text{Var}[X_i]$. With this, a more variable (= less precisely measured) observation has lesser impact on $\hat{\boldsymbol{\vartheta}}$.

5.2.4 Maximum likelihood *

When the full parametric structure of a random sample can be specified via a likelihood function $L(\boldsymbol{\vartheta}; x_1, x_2, \ldots, x_n)$, a powerful estimation method may be brought to bear for estimating $\boldsymbol{\vartheta}$. Given the likelihood, ones finds the log-likelihood function $\ell(\boldsymbol{\vartheta})$ and then maximizes it with respect to the p unknown parameters. This is the *method of maximum likelihood* (ML). In most cases, determining the maximum likelihood estimators (MLEs) requires appeal to differential calculus; that is, set the first partial derivatives of $\ell(\boldsymbol{\vartheta})$ equal to zero and solve the resulting system of p estimating equations to find $\hat{\boldsymbol{\vartheta}}_{\text{ML}}$.

Example 5.2.5 **MLEs from normal population.** Take a random sample of normal (Gaussian) observations, $X_i \sim$ i.i.d. $\text{N}(\mu, \sigma^2)$, $i = 1, \ldots, n$. To find the MLE for $\boldsymbol{\vartheta} = [\mu, \sigma^2]^{\mathrm{T}}$, start

with the likelihood function. This is built from the normal p.d.f. in (2.34):

$$L(\mu, \sigma^2; x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x_i-\mu)^2}{2\sigma^2}\right\}$$
$$= \frac{1}{(2\pi)^{n/2}\sigma^n} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2\right\}.$$

It will be convenient to work with the reparameterization $\tau = \sigma^2 > 0$ in order to prevent confusion when taking derivatives. The log-likelihood is

$$\ell(\mu, \tau) = C + \log(\tau^{-n/2}) - \frac{1}{2\tau}\sum_{i=1}^{n}(x_i-\mu)^2 = C - \frac{n}{2}\log(\tau) - \frac{1}{2\tau}\sum_{i=1}^{n}(x_i-\mu)^2,$$

where C is a constant that does not affect the maximization. From this, the ML estimating equations are found from the partial derivatives of $\ell(\mu, \tau)$ with respect to each parameter. Set these equal to zero and then solve for μ and τ:

$$\frac{\partial \ell}{\partial \mu} = -\frac{2}{2\tau}\sum_{i=1}^{n}(x_i-\mu)\frac{\partial(-\mu)}{\partial\mu} = \frac{1}{\tau}\sum_{i=1}^{n}(x_i-\mu) = \frac{\sum_{i=1}^{n} x_i}{\tau} - \frac{n\mu}{\tau} = 0 \tag{5.9}$$

and

$$\frac{\partial \ell}{\partial \tau} = -\frac{n}{2\tau} + \frac{1}{2\tau^2}\sum_{i=1}^{n}(x_i-\mu)^2 = 0. \tag{5.10}$$

Solving (5.9) for μ (assuming $\tau \neq 0$) leads immediately to

$$\hat{\mu}_{\text{ML}} = \frac{\sum_{i=1}^{n} x_i}{n} = \bar{x}.$$

Here again, we find the estimate of the population mean is the sample mean.

Continuing on to the variance, because we know $\hat{\mu}_{\text{ML}} = \bar{x}$ in the simultaneous system of equations, we can substitute this for μ in (5.10) to find

$$\frac{\partial \ell}{\partial \tau} = -\frac{n}{2\tau} + \frac{1}{2\tau^2}\sum_{i=1}^{n}(x_i-\bar{x})^2 = 0,$$

that is, $n\tau = \sum_{i=1}^{n}(x_i-\bar{x})^2$. Solving for $\tau(=\sigma^2)$ gives

$$\hat{\sigma}^2_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2.$$

Curiously, this is *not* the sample variance, S^2, from (3.5): $\hat{\sigma}^2_{\text{ML}} = \frac{n-1}{n}S^2$. (See Example 5.2.6.) □

A technical aside: if applying calculus-based methods to find MLEs, one must verify that the resulting stationary points of $\ell(\boldsymbol{\vartheta})$ are maxima, that is, the proposed value for $\hat{\boldsymbol{\vartheta}}$ does indeed maximize $\ell(\boldsymbol{\vartheta})$. Generally, the solution is again to apply differential calculus. Construct the Hessian matrix $\mathbf{H}_\ell = \{h_{jk}\}$ of second partial derivatives, $h_{jk} = \partial^2\ell(\boldsymbol{\vartheta})/\partial\theta_j\partial\theta_k$, from (5.4), and

verify that $\mathbf{H}_\ell$ is negative definite. In the special case of $p = 2$, this simplifies to verifying (i) $h_{11} < 0$ and (ii) $h_{11}h_{22} - h_{12}^2 > 0$ (Khuri 2003, Section 7.7).

One must also verify that the log-likelihood does not reach a true maximum at the boundaries of the parameter space. This is usually a simple check: identify the (joint) values of $\boldsymbol{\vartheta}$ at its boundary and evaluate $\ell(\boldsymbol{\vartheta})$ at those points. Then, verify that the log-likelihood (or the full likelihood, if easier to evaluate) achieves a larger value at the proposed MLEs. (Most likelihood functions are stable enough that their stationary points are indeed global maxima, although one should always conduct the verification just in case.) Exercise 5.4 verifies that the MLEs $\hat{\mu}_{\text{ML}}$ and $\hat{\sigma}^2_{\text{ML}}$ for the normal case in Example 5.2.5 do indeed maximize the likelihood function.

Example 5.2.6 Variance estimation. Suppose a random sample produces observations $X_i \sim$ i.i.d. $f_X(x|\mu, \sigma^2)$ $i = 1, \ldots, n$. Assume the underlying p.m.f. or p.d.f. has a finite population mean $E[X_i] = \mu$ and a finite population variance σ^2. If the sample is obtained from a normal population, $\text{N}(\mu, \sigma^2)$, then as seen in Example 5.2.5, the MLE for μ is the sample mean $\overline{X}$. Further, from Example 5.2.4, it is also the LS estimator for μ, *and* from Example 5.2.1, it is an unbiased estimator for μ. (Hooray for the sample mean!)

Example 5.2.5 also showed that, however, the MLE for the variance σ^2 from a normal random sample is *not* the sample variance, S^2, from (3.5): $\hat{\sigma}^2_{\text{ML}} = \frac{n-1}{n}S^2$. To confuse the issue further, it can be shown (Exercise 5.6) that for any random sample where the population variance is finite, $E[S^2] = \sigma^2$. Thus S^2 is an unbiased estimator for σ^2, but $\hat{\sigma}^2_{\text{ML}}$ is not:

$$E\left[\hat{\sigma}^2_{\text{ML}}\right] = E\left[\frac{n-1}{n}S^2\right] = \frac{n-1}{n}E[S^2] = \frac{n-1}{n}\sigma^2 \neq \sigma^2.$$

The conundrum here is, which to choose for estimating σ^2? Standard practice generally favors the unbiased estimator S^2, and so the sample variance is employed in the various statistical operations in the following. Indeed, in **R**, application of the `var(x)` function to a sample of data in `x` will produce the unbiased estimator S^2. From this, the usual point estimator for σ is taken as the sample standard deviation, $S = \sqrt{S^2}$, available in **R** via the `sd()` function.

This has important implications. For example, the standard error of $\overline{X}$ was found in Example 5.2.1 as, technically, the square root of *an estimator of* σ^2, divided by $\sqrt{n}$. As established practice employs S^2 as the estimator of σ^2, this results in $\text{se}[\overline{X}] = S/\sqrt{n}$.

To illustrate, recall the data in Table 4.2 on ages at myocardial infarction ('heart attack') among $n = 126$ cardiac patients. In Example 4.1.7, a normal quantile plot suggested a reasonable fit of the normal model to these data. Thus to estimate the mean age of attack for this population of subjects, from Example 5.2.5, the MLE/unbiased estimator is simply the sample mean. In **R**, this is calculated via `mean(x)`, and for these data, this produces $\overline{X} = 62.8137$. The sample variance is then $S^2 = \frac{1}{125}\sum_{i=1}^{216}(X_i - 62.8137)^2$. Here, direct calculation is simplified by using the computing formulas from (3.6), although it is even faster to employ the computer: in **R**, `var(x)` gives $S^2 = 69.5908$, while `sd(x)` produces $S = 8.3421$. With these, the standard error of $\overline{X}$ can be found as $\text{se}[\overline{X}] = S/\sqrt{126} = 8.3421/11.2250 = 0.7432$.

It is worth acknowledging that with very large sample sizes the difference between the unbiased estimator S^2 and the MLE $\hat{\sigma}^2_{\text{ML}}$ will essentially vanish, because as $n \to \infty$, the two estimators converge to the same value.

If, further, the random sample is from a normal distribution, $X_i \sim$ i.i.d. $\text{N}(\mu, \sigma^2)$, $i = 1, \ldots, n$, the full sampling distribution of S^2 can be determined. Under normal sampling,

it can be shown (Casella and Berger 2002, Section 5.3) that

$$\frac{(n-1)S^2}{\sigma^2} \sim \chi^2(n-1).$$

This feature will prove useful in making inferences on both σ^2 and μ. □

For the sorts of complex, multiparameter models often employed in data analytics, construction and maximization of a multidimensional log-likelihood can require extensive calculation and is usually performed by computer. Nonetheless, ample benefits accrue from the effort: MLEs possess a number of desirable properties. For example, suppose interest focuses on some known function, $h(\boldsymbol{\vartheta})$, of the parameter vector $\boldsymbol{\vartheta}$. To find the MLE for $h(\boldsymbol{\vartheta})$, one can simply evaluate the function at the MLE for $\boldsymbol{\vartheta}$, that is, $\widehat{h(\boldsymbol{\vartheta})} = h(\hat{\boldsymbol{\vartheta}})$. This is known as the functional *invariance property* of MLEs (Casella and Berger 2002, Section 7.2).

Another important feature of MLEs is that the form of their asymptotic distribution is often known, allowing for operable large-sample approximations. Under certain regularity conditions (Lehmann and Casella 1998, Section 6.3), the ML vector $\hat{\boldsymbol{\vartheta}}$ will possess a p-variate normal distribution where, individually,

$$\hat{\theta}_j \dot{\sim} \mathrm{N}(\theta_j, \mathrm{Var}[\hat{\theta}_j]), \tag{5.11}$$

$j = 1, \ldots, p$. (Recall that the symbol $\dot{\sim}$ is read as 'is approximately distributed as.' The approximation improves as $n \to \infty$.) The covariance matrix of $\hat{\boldsymbol{\vartheta}}$ is found by inverting the Fisher information matrix, $\mathbf{F}(\boldsymbol{\vartheta})$, from (5.5). For shorthand notation, write $E[\hat{\boldsymbol{\vartheta}}] = \boldsymbol{\vartheta}$ and $\mathrm{Var}[\hat{\boldsymbol{\vartheta}}] = \mathbf{F}^{-1}(\boldsymbol{\vartheta})$. The large-sample variances are found as the diagonal elements of $\mathbf{F}^{-1}(\boldsymbol{\vartheta})$.

In the single-parameter case with $p = 1$, this is $\hat{\theta} \dot{\sim} \mathrm{N}(\theta, \mathrm{Var}[\hat{\theta}])$, where $\mathrm{Var}[\hat{\theta}]$ is simply the reciprocal of the Fisher information number

$$\mathrm{Var}[\hat{\theta}] \approx \frac{1}{\mathcal{F}(\theta)}. \tag{5.12}$$

5.3 Interval estimation

Parameter estimation is a descriptive step for using data to learn about the larger population. The next step, statistical inference, has two general forms: interval estimation and hypothesis testing. These are briefly reviewed in this section and the next section, respectively. In the former case, the concept is an extension of the point estimators introduced earlier, where an entire *interval* of plausible values is provided as an estimate for the unknown parameter. When constructed properly, an interval estimator informs the analyst on uncertainty in estimating θ: wider intervals suggest greater uncertainty (and less precision), narrower intervals suggest less uncertainty (more precision).

5.3.1 Confidence intervals

A *confidence interval* is a form of estimator for the unknown parameter θ that uses the data to construct an interval within which θ may lie. Formally, a confidence interval is a pair of values $L_\theta(X_1, \ldots, X_n)$ and $U_\theta(X_1, \ldots, X_n)$ that satisfy the probability statement

$$P[L_\theta(X_1, \ldots, X_n) < \theta < U_\theta(X_1, \ldots, X_n)] = 1 - \alpha, \tag{5.13}$$

where $1-\alpha$ is called the *confidence coefficient* or the *confidence level* of the interval. Typical values are 90%, 95%, or 99%, with 95% seen most often in practice. If both L_θ and U_θ are

finite, the confidence interval is *two sided*. If either limit is infinite, its finite counterpart is a *one-sided* confidence bound on θ.

It is important to recognize that confidence is not probability. The probability that a calculated interval actually contains the true value of θ is not $1-\alpha$. It is either 0 or 1. That is, suppose that we calculate a 95% interval for θ and find $23.08 < \theta < 56.26$. Clearly, the probability that θ is included in this interval is 0 (if it is not) or 1 (if it is). As no random variability is ascribed to θ, no other form of probability can be assigned to the statement $23.08 < \theta < 56.26$. Thus, instead of a probabilistic interpretation, we say that the interval 'covers' θ with confidence $1-\alpha$. This is a *frequentist interpretation* for coverage: if over repeated sampling one counts the number of times an interval covers the true θ, one finds that $100(1-\alpha)\%$ of the intervals cover correctly. Thus confidence is a measure of the interval estimator's frequency of correct coverage for the unknown parameter.

5.3.2 Single-sample intervals for normal (Gaussian) parameters

The theory and application of confidence intervals is well established for the normal (Gaussian) distribution setting. Take a random sample of normal observations, $X_i \sim$ i.i.d. $\mathrm{N}(\mu, \sigma^2)$, $i = 1, \ldots, n$. Assume both μ and σ^2 are unknown. For a confidence interval on μ, the goal is to produce a statement of the form in (5.13). To do so, recall that the ML (and LS) estimator for the population mean here is $\overline{X}$, with standard error $\mathrm{se}[\overline{X}] = S/\sqrt{n}$, and where S is the sample standard deviation. Statements about μ can be built from the distribution of $\overline{X}$, which from Section 2.3.9 is given by $\overline{X} \sim \mathrm{N}\left(\mu, \sigma^2/n\right)$. Standardizing to N(0,1) produces

$$Z = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} \sim \mathrm{N}(0, 1). \tag{5.14}$$

Now, we know that the standard normal upper-$\frac{\alpha}{2}$ critical point, $z_{\alpha/2}$, satisfies $P[Z > z_{\alpha/2}] = \alpha/2$; cf. Figure 2.5. Manipulating this 'tail area' relationship into an interval relationship leads to $P[-z_{\alpha/2} < Z < z_{\alpha/2}] = 1 - \alpha$, which for the $\overline{X}$ standardization in (5.14) is equivalent to

$$\begin{aligned} 1 - \alpha &= P\left[-z_{\alpha/2} < \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} < z_{\alpha/2}\right] = P\left[-z_{\alpha/2}\frac{\sigma}{\sqrt{n}} < \overline{X} - \mu < z_{\alpha/2}\frac{\sigma}{\sqrt{n}}\right] \\ &= P\left[\overline{X} - z_{\alpha/2}\frac{\sigma}{\sqrt{n}} < \mu < \overline{X} + z_{\alpha/2}\frac{\sigma}{\sqrt{n}}\right] \end{aligned} \tag{5.15}$$

(relying on the fact that $\sigma/\sqrt{n} > 0$). This satisfies (5.13) and, hence, is a valid $100(1-\alpha)\%$ confidence interval for μ, *if* σ *were known*. (Clearly, σ is a critical component of both the lower and upper endpoints.) In practice, however, σ is hardly ever known. Instead, as in Example 5.2.6, the sample standard deviation S is used to estimate it. In effect, the standardization in (5.14) becomes

$$T = \frac{\overline{X} - \mu}{S/\sqrt{n}}. \tag{5.16}$$

Notice that now, the centered difference $\overline{X} - \mu$ has been divided not by the standard deviation of $\overline{X}$, $\sqrt{\mathrm{Var}[\overline{X}]} = \sigma/\sqrt{n}$, but by the standard error $\mathrm{se}[\overline{X}] = S/\sqrt{n}$.

It can be shown (Exercise 5.7) that the quantity T in (5.16) satisfies the relationship in (2.41) that defines Student's t-distribution. Here $T \sim t(n-1)$. The degrees of freedom (d.f.), $\nu = n-1$, are derived from the χ^2 relationship for the variance estimator S^2 introduced in Example 5.2.6: for a normal random sample, $(n-1)S^2/\sigma^2 \sim \chi^2(n-1)$. In this case, we often say that $\overline{X}$ in (5.16) has been not 'standardized,' but 'Studentized' (Hartley 1938).

As a result, t-distribution critical points, $t_{\alpha/2}(n-1)$, can be substituted for the standard normal critical points in reconstructing (5.15). Start with $P[-t_{\alpha/2}(n-1) < T < t_{\alpha/2}(n-1)] = 1-\alpha$ as in Figure 5.1 and find

$$1-\alpha = P\left[-t_{\alpha/2}(n-1) < \frac{\overline{X}-\mu}{S/\sqrt{n}} < t_{\alpha/2}(n-1)\right]$$
$$= \cdots = P\left[\overline{X} - t_{\alpha/2}(n-1)\frac{S}{\sqrt{n}} < \mu < \overline{X} + t_{\alpha/2}(n-1)\frac{S}{\sqrt{n}}\right]. \qquad (5.17)$$

This then defines a t-distribution $100(1-\alpha)\%$ confidence interval for the mean from a normal random sample.

Notice the symmetric form of the interval in (5.17): a point estimator, $\overline{X}$, centered between the same separating quantity, $t_{\alpha/2}(n-1)S/\sqrt{n}$. Shorthand for this construction is

$$\overline{X} \pm t_{\alpha/2}(n-1)\frac{S}{\sqrt{n}}\,.$$

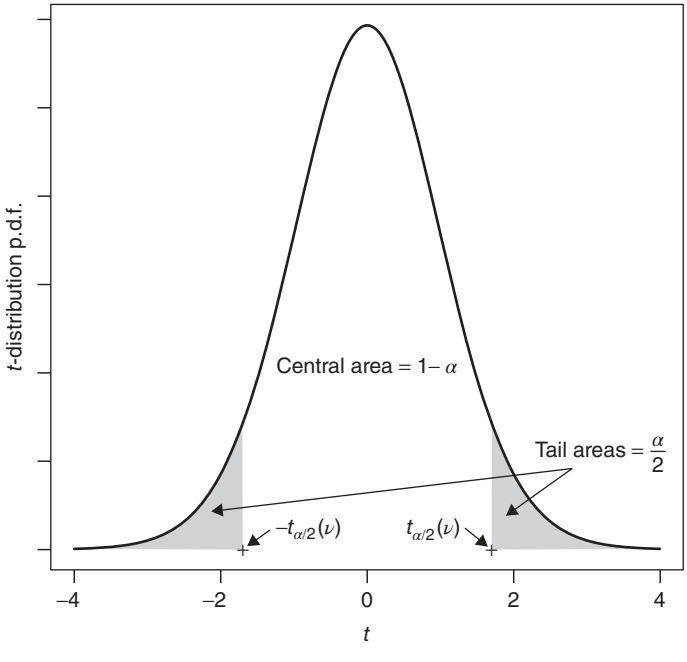

Figure 5.1 Bounding a t-distribution's p.d.f. area at critical points $\pm\, t_{\alpha/2}(\nu)$.

The separating quantity is called the *margin of error* (MOE) of the confidence interval. (More technically, the MOE is the half-width of a confidence interval for an unknown parameter.)

If only one-sided limits are desired on μ, the construction simplifies slightly. For an upper limit, start with the strict lower-tail relationship $P[-t_\alpha(n-1) < T] = 1-\alpha$ and manipulate it into

$$1-\alpha = P\left[-t_\alpha(n-1) < \frac{\overline{X}-\mu}{S/\sqrt{n}}\right] = P\left[-t_\alpha(n-1)\frac{S}{\sqrt{n}} < \overline{X}-\mu\right]$$
$$= P\left[\mu < \overline{X} + t_\alpha(n-1)\frac{S}{\sqrt{n}}\right]. \tag{5.18}$$

Thus the $100(1-\alpha)\%$ one-sided, upper, confidence limit for μ is $\overline{X} + t_\alpha(n-1)\frac{S}{\sqrt{n}}$. For the one-sided lower limit, reverse the construction to find

$$1-\alpha = P\left[\overline{X} - t_\alpha(n-1)\frac{S}{\sqrt{n}} < \mu\right]. \tag{5.19}$$

Example 5.3.1 Confidence interval on μ from a normal sample (Example 4.1.7, continued). Return to the myocardial infarction data in Table 4.2. In Example 4.1.7, a normal quantile plot suggested a possible fit of the normal model to these data. Thus to estimate the mean age of attack for this population the sample mean is $\overline{X} = 62.8137$, calculated in **R** via `mean(x)`. In Example 5.2.6, the standard error of $\overline{X}$ was found as $\text{se}[\overline{X}] = S/\sqrt{126} = 8.3421/11.2250 = 0.7432$, using `sd(x)` to find $S = 8.3421$.

These are precisely the components required to construct the t confidence interval for μ from (5.17). Set the confidence level to 95%. The necessary t critical point is then $t_{\alpha/2}(n-1) = t_{0.025}(125) = 1.9791$, found in **R** via `qt(0.025, 125, lower.tail=FALSE)`. This gives an MOE of

$$t_{0.025}(125)\frac{S}{\sqrt{n}} = (1.9791)(0.7432) = 1.4708,$$

with consequent 95% confidence interval for μ given by 62.8137 ± 1.4708, that is,

$$61.3429 < \mu < 64.2845.$$

With 95% confidence, we learn that the average age at which this population suffers its (first) myocardial infarction is between about 61.3 and 64.3 years. (As noted above, however, the value of $P[61.3429 < \mu < 64.2846]$ is either 0 or 1, and it is not 0.95.)

These various operations can be coded directly in **R** and assembled into a fast routine or function to find the confidence limits. One example is the standard **R** routine `t.test()`:

```
> t.test( age, conf.limit=0.95 )
```

with output (edited)

```
        One Sample t-test
  data:  age
  t = 84.5209, df = 125, p-value < 2.2e-16
  alternative hypothesis: true mean is not equal to 0
```

```
95 percent confidence interval:
 61.34289 64.28456
sample estimates:
mean of x
 62.81372
```

The confidence limits follow after '`95 percent confidence interval`' and they corroborate the direct calculations above. (Other components of this output are discussed in Example 5.4.1.) □

Although the normal population variance σ^2 is often viewed as a nuisance parameter, there are occasions when confidence limits may be desired for it as well. These rely directly on the χ^2 relationship for S^2 discussed in Example 5.2.6. Let $C^2 = (n-1)S^2/\sigma^2$ such that $C^2 \sim \chi^2(n-1)$. Then using χ^2 critical points (see Figure 2.6), the statement

$$1 - \alpha = P[\chi^2_{1-\alpha/2}(n-1) < C^2 < \chi^2_{\alpha/2}(n-1)] \tag{5.20}$$

can be inverted into

$$1 - \alpha = P\left[\frac{(n-1)S^2}{\chi^2_{\alpha/2}(n-1)} < \sigma^2 < \frac{(n-1)S^2}{\chi^2_{1-\alpha/2}(n-1)}\right]. \tag{5.21}$$

This defines a $100(1-\alpha)\%$ confidence interval for the variance from a normal random sample. (For a confidence interval on the standard deviation, σ, first apply the square root across all sides of the inequalities in (5.20).)

Notice that the interval for σ^2 in Equation (5.21) is not symmetric, although it was constructed from a symmetric assignment of tail areas. That is, equal $\alpha/2$ area was allocated to each tail in (5.20). While this is a natural choice, it is not unique: one could assign two times as much tail area to one side – or three times as much, or four times, and so on – and still produce a valid confidence statement similar to (5.21). In practice, the choice of tail area assignment is determined on a case-by-case basis, depending on which of the two limits requires more attention for the problem under study.

If no a priori information is available to guide the assignment, an equal-area split is simplest but not necessarily optimal. Theoretical constructions for building optimal confidence limits on a normal variance were given by Tate and Klett (1959). These require additional specifications to uniquely determine the optimal limits; see the article by Tate and Klett (1959), or Casella and Berger (2002, Exercise 9.52), for more details.

For the more-common calculation of confidence limits for μ as in (5.17), the issue of allocating tail areas is not a concern. The equal-tail area allocation used there is known to be optimal (Casella and Berger 2002, Section 9.3).

The operations used to construct the confidence intervals for both μ and σ^2 in (5.17) and (5.21), respectively, took advantage of a useful feature. The statistics used to form the intervals had familiar reference distributions. For μ, the T ratio in (5.16) was distributed as $t(n-1)$ while for σ^2, C^2 in (5.20) was distributed as $\chi^2(n-1)$. In both cases, the t and χ^2 reference distributions did not depend on any unknown parameters. Statistics that exhibit this useful feature are given a special name: they are called *pivotal quantities* (Casella and Berger 2002, Section 9.2.2). Although not always available, the 'pivoting' of such quantities is a popular strategy to produce confidence limits for many different distributions. See the examples given in Casella and Berger (2002, Section 9.2), Piegorsch and Bailer (1997, Section 2.6), and similar sources.

5.3.3 Two-sample intervals for normal (Gaussian) parameters

When the random sampling occurs across two independent samples, a confidence interval can be constructed to *compare* the means of each sample. In the normal case, suppose data from the first sample are $X_{1j} \sim$ i.i.d. $\mathrm{N}(\mu_1, \sigma_1^2)$, $j = 1, \ldots, n_1$, independent of data from the second sample $X_{2j} \sim$ i.i.d. $\mathrm{N}(\mu_2, \sigma_2^2)$, $j = 1, \ldots, n_2$. Assume both samples' sets of parameters μ_1, σ_1^2 and μ_2, σ_2^2 are unknown. Of interest is interval estimation of the difference in means $\Delta = \mu_1 - \mu_2$. An unbiased estimator for Δ is the difference in sample means, $\overline{X}_1 - \overline{X}_2$, with corresponding standard error

$$\mathrm{se}[\overline{X}_1 - \overline{X}_2] = \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}},$$

where the S_i^2s are the respective sample variances from each independent random sample ($i = 1, 2$). These various statistics are then employed in constructing the pivotal quantity

$$T = \frac{\overline{X}_1 - \overline{X}_2 - \Delta}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}\,. \tag{5.22}$$

The pivot T in (5.22) is approximately t-distributed: $T \dot{\sim} t(\nu_{\mathrm{WS}})$, with approximate d.f.

$$\nu_{\mathrm{WS}} = \left\lfloor \frac{\left(U_1 + U_2\right)^2}{\frac{1}{n_1 - 1}U_1^2 + \frac{1}{n_2 - 1}U_2^2} \right\rfloor, \tag{5.23}$$

and with $U_i = S_i^2/n_i$ ($i = 1, 2$). (The 'floor' notation in ν_{WS} reminds the analyst to round *down* to the nearest whole number.) This approximation for the d.f. was proposed by Smith (1936) and later by Welch (1938) and is a special case of a general method for moment-based d.f. estimation described by Satterthwaite (1946). It is known as the *Welch–Satterthwaite correction* for the d.f. of T.

Manipulating the pivot in (5.22) yields the approximate confidence limits (Exercise 5.15)

$$\overline{X}_1 - \overline{X}_2 \pm t_{\alpha/2}(\nu_{\mathrm{WS}})\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}\,. \tag{5.24}$$

The Welch–Satterthwaite approximation for the d.f. improves as $\min\{n_1, n_2\} \to \infty$. It is quite accurate, however, and the approximate confidence level for (5.24) will be very near $1-\alpha$ for sample sizes as low as $n_i = 10$.

Example 5.3.2 Two-sample confidence interval for Lung Function data. Taussig et al. (2003) discussed data on lung function response among 353 asthmatics living in a semi-arid environment, including possible differences between patients who smoked and those who did not. Recorded was the patient's percentage of expected forced expiratory volume in 1 s (known as 'FEV1') at the age of 26 years. Higher values indicate more robust lung function. A selection of the data appear in Table 5.1. (The full data set is available at http://www.wiley.com/go/piegorsch/data_analytics.)

Table 5.1 Percentage of expected forced expiratory volume in 1 s (FEV1) among asthma patients at the age of 26 years.

Smoking status	FEV1
No	90, 84, 103, 94, 88, 101, … , 78, 76, 81, 101, 93, 111
Yes	104, 82, 79, 93, 86, 69, … , 105, 99, 112, 135, 96, 85

Table 5.2 Summary statistics for FEV1 measurements from Table 5.1.

Smoking status	Sample size, n_i	Mean, $\overline{X}_i$	Variance, S_i^2
No ($i = 1$)	265	97.5962	121.9386
Yes ($i = 2$)	88	97.7500	150.7644

To compare the two groups, take X_1 as the FEV1 scores for nonsmokers and X_2 as those for smokers. Preliminary examination of the data (Exercise 5.13) indicates that a normal sampling assumption for both variables is reasonable, so assume $X_{1j} \sim$ i.i.d. $\mathrm{N}(\mu_1, \sigma_1^2)$, $j = 1, \ldots, n_1$, independent of $X_{2j} \sim$ i.i.d. $\mathrm{N}(\mu_2, \sigma_2^2)$, $j = 1, \ldots, n_2$.

To determine a plausible range of values for the mean difference $\Delta = \mu_1 - \mu_2$ under these assumptions, appeal to the confidence interval in (5.24). Summary statistics for the necessary calculations appear in Table 5.2. Appeal is made here to the Welch–Satterthwaite correction for the t reference distribution. From (5.23), this produces $\nu_{\mathrm{WS}} = \lfloor 136.76 \rfloor = 136$ d.f.

Set the confidence level to 90%. The corresponding critical point is $t_{0.10/2}(136) = 1.6561$. From (5.24), the limits are then produces the limits $-0.1538 \pm (1.6561)\sqrt{2.1734} = -0.1538 \pm 2.4415$. That is, with 90% confidence, we state that the difference in mean FEV1 scores rests in the interval $-2.5953 \leq \Delta \leq 2.2877$. As this interval contains $\Delta = 0$, it is plausible that there is no real difference between the two mean scores. (Also see Example 5.4.2.)

In **R**, this analysis can be achieved via the `t.test()` function (with output edited):

```
> t.test(X1, X2, conf.level=0.90, var.equal=FALSE)
        Welch Two Sample t-test
  data:  X1 and X2
  t = -0.1043, df = 136.758, p-value = 0.9171
  alternative hypothesis: true difference in means is not
                                                  equal to 0
  90 percent confidence interval:
   -2.595218  2.287670
  sample estimates:
  mean of x    mean of y
   97.59623     97.75000
```

The `var.equal=FALSE` option institutes the Welsh–Satterthwaite correction for the d.f., indicated by the '`Welch Two Sample t-test`' output header. The `conf.level=0.90` option calls for the 90% limits, which appear in the **R** output under '`90 percent confidence interval`.' The results corroborate the direct calculations given above. (Slight disparities occur due to rounding.) □

In the special case where $\sigma_1^2 = \sigma_2^2 = \sigma^2$ (say), the independent-samples confidence interval in (5.24) collapses to an exact construction. As the variance is equal in both groups, we *pool* the two sample variances into an (unbiased) estimator for σ^2,

$$S_p^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2} \,. \tag{5.25}$$

With this, a pivotal quantity is

$$T = \frac{\overline{X}_1 - \overline{X}_2 - \Delta}{S_p \sqrt{\dfrac{1}{n_1} + \dfrac{1}{n_2}}} \,. \tag{5.26}$$

Under the assumption of homogeneous variances for this independent/normal two-sample setting, the pivot in (5.26) is *exactly* t-distributed for any values of n_i, such that $T \sim t(n_1 + n_2 - 2)$. From this, straightforward manipulation of the pivot produces the exact $100(1-\alpha)\%$ confidence limits

$$\overline{X}_1 - \overline{X}_2 \pm t_{\alpha/2}(n_1 + n_2 - 2)\; S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}} \,. \tag{5.27}$$

In **R**, if `X1` and `X2` are vectors containing the data from each independent sample, then the command `t.test(X1, X2, conf.level=1-alpha, var.equal=TRUE)` can supply the confidence limits in (5.27), under "`95 percent confidence interval`."

In another special, two-sample case, the t-distribution again produces exact confidence limits. Suppose the observations are recorded in *matched pairs*, (X_{1j}, X_{2j}), with $X_{ij} \sim$ i.i.d. $\mathrm{N}(\mu_i, \sigma_i^2)$, $i = 1, 2$; $j = 1, \ldots, n$. To build a confidence interval for $\mu_D = \mu_1 - \mu_2$, we construct the difference variable $D_j = X_{1j} - X_{2j}$, where now $D_j \sim$ i.i.d. $\mathrm{N}(\mu_D, \sigma_D^2)$ and σ_D^2 is a function of the original variances and of $\mathrm{Cov}[X_{1j}, X_{2j}]$. (The exact expression is not required for the analysis – note that because the data appear in pairs, the assumption of independence between the two samples is untenable.) In effect, the differences now act as a single sample, allowing for appeal to arguments that produce single-sample limits as in (5.17). Find the mean difference $\overline{D}$ and the variance $S_D^2 = \sum_{j=1}^{n} (D_i - \overline{D})^2/(n-1)$, and construct a pivot similar to (5.16):

$$T_D = \frac{\overline{D} - \mu_D}{S_D/\sqrt{n}} \,.$$

Under the normal sampling assumptions, $T_D \sim t(n-1)$ (exactly). Thus a construction similar to (5.17) can be used to find $100(1-\alpha)\%$ limits on μ_D. These are simply

$$\overline{D} \pm t_{\alpha/2}(n-1)\frac{S_D}{\sqrt{n}} \,.$$

In **R**, if `X1` and `X2` are the vectors containing the matched pairs from each sample, then the command

```
> t.test( X1, X2, conf.level=1-alpha, paired=TRUE )
```

can supply these paired-t limits, under '`95 percent confidence interval`.'

In the general two-sample case, one can also construct confidence limits for comparing the two population variances σ_1^2 and σ_2^2. Suppose again that $X_{1j} \sim$ i.i.d. $\mathrm{N}(\mu_1, \sigma_1^2)$, $j = 1, \ldots, n_1$, independent of $X_{2j} \sim$ i.i.d. $\mathrm{N}(\mu_2, \sigma_2^2)$, $j = 1, \ldots, n_2$. Assume all the population parameters are unknown. It is possible to construct an (exact) confidence interval on the *ratio* of variances σ_1^2/σ_2^2, via appeal to the F-distribution from Section 2.3.10. Start with the independent sample variances S_1^2 and S_2^2, and recall that $C_i^2 = (n_i - 1)S_i^2/\sigma_i^2 \sim$ indep. $\chi^2(n_i - 1)$ for $i = 1, 2$. Since the F-distribution was defined as a scaled ratio of independent χ^2 variates, and because the S_i^2s and, hence, the C_i^2s are independent, let

$$F_{21} = \frac{C_2^2/(n_2 - 1)}{C_1^2/(n_1 - 1)} = \frac{S_2^2}{S_1^2} \times \frac{\sigma_1^2}{\sigma_2^2} \, .$$

Under the two-sample normal assumptions, $F_{21} \sim F(n_2 - 1, n_1 - 1)$. Then, in similar form to (5.21), the probability statement

$$1 - \alpha = P[F_{1-\alpha/2}(n_2 - 1, n_1 - 1) < F_{21} < F_{\alpha/2}(n_2 - 1, n_1 - 1)]$$

can be inverted into

$$1 - \alpha = P\left[F_{1-\alpha/2}(n_2 - 1, n_1 - 1)\frac{S_1^2}{S_2^2} < \frac{\sigma_1^2}{\sigma_2^2} < F_{\alpha/2}(n_2 - 1, n_1 - 1)\frac{S_1^2}{S_2^2}\right] . \tag{5.28}$$

Clearly then, (5.28) defines a $100(1-\alpha)\%$ confidence interval for σ_1^2/σ_2^2.

As with the χ^2-based interval for a single variance, the F-based interval for a variance ratio in (5.28) is not symmetric, although it is constructed from a symmetric assignment of tail areas. Thus it also suffers from the question of tail area assignment: while convenient, the equal-tail areas in (5.28) do not produce an optimal confidence interval. Theoretical constructions for building minimum-length confidence limits on a normal variance ratio were given by Levy and Narula (1974). These require additional specifications to uniquely determine the optimal limits; see the article by Levy and Narula (1974), or also Wilson and Tonascia (1971), for more details.

5.3.4 Wald intervals and likelihood intervals*

The 'pivotal' characteristic of the t-interval in (5.17) overlaps with an omnibus technique for constructing confidence limits, using MLEs from Section 5.2.4. The limits are known collectively as *Wald intervals*, based on Abraham Wald's (1943) pioneering work in this area. Their key feature relies on the fact that as the sample size grows large, the MLE will often be approximately normal. From this, a pivotal quantity can be constructed from which approximate confidence limits on the target parameter may be determined.

Take a random sample $X_i \sim$ i.i.d. $f_X(x|\theta)$, $i = 1, \ldots, n$. Under suitable regularity conditions, the large-sample distribution of the MLE $\hat{\theta}$ is given in (5.11) as $\hat{\theta} \sim \mathrm{N}(\theta, \mathrm{Var}[\hat{\theta}])$, where the variance $\mathrm{Var}[\hat{\theta}]$ is the reciprocal of the Fisher information number, as in (5.12). The corresponding standard error, $\mathrm{se}[\hat{\theta}]$, is the (estimated) square root of the variance in (5.7). With these, a pivotal quantity can be constructed similar to (5.16): standardize $\hat{\theta}$ by dividing $\mathrm{se}[\hat{\theta}]$ into the centered quantity $\hat{\theta} - \theta$. This produces the pivot

$$Z = \frac{\hat{\theta} - \theta}{\mathrm{se}[\hat{\theta}]} \, . \tag{5.29}$$

If $\hat{\theta} \dot{\sim} \mathrm{N}(\theta, \mathrm{Var}[\hat{\theta}])$, then the ratio in (5.29) will be approximately standard normal: $Z = (\hat{\theta} - \theta)/\mathrm{se}[\hat{\theta}] \dot{\sim} \mathrm{N}(0, 1)$. This large-sample reference distribution does not depend on θ, thus Z is an approximate pivotal quantity.

To assemble a confidence interval for θ, apply the pivot in an analogous manner to (5.15): start with $P[-z_{\alpha/2} < Z < z_{\alpha/2}] = 1 - \alpha$, which for the pivot in (5.29) gives the approximation

$$\begin{aligned} 1 - \alpha &\approx P\left[-z_{\alpha/2} < \frac{\hat{\theta} - \theta}{\mathrm{se}(\hat{\theta})} < z_{\alpha/2}\right] = P\left[-z_{\alpha/2}\mathrm{se}(\hat{\theta}) < \hat{\theta} - \theta < z_{\alpha/2}\mathrm{se}(\hat{\theta})\right] \\ &= \cdots = P\left[\hat{\theta} - z_{\alpha/2}\mathrm{se}(\hat{\theta}) < \theta < \hat{\theta} + z_{\alpha/2}\mathrm{se}(\hat{\theta})\right]. \end{aligned} \tag{5.30}$$

The Wald interval is the resulting, approximate, $100(1-\alpha)\%$ construct

$$\hat{\theta} \pm z_{\alpha/2}\mathrm{se}[\hat{\theta}]. \tag{5.31}$$

Similar to the application of the central limit theorem in Example 5.2.2, the Wald approximation in (5.30) improves as $n \to \infty$, and it applies for a broad range of parent distributions; however, its quality in practice will vary from case to case. The large-sample features of the MLE can take effect fairly quickly for some continuous parent distributions, where approximate normality can be valid for n as small as 10 or 20. Other distributions, especially many discrete forms, can require n upwards of 100 or more before variation in $\hat{\theta}$ begins to appear normal.

Example 5.3.3 MLE on π from a binomial sample. An important application of the Wald approach occurs when sampling from a binomial distribution, with some caveats. Suppose a random sample is taken with n Bernoulli trials, $X_i \sim$ i.i.d. $\mathrm{Bin}(1, \pi)$. To estimate π, consider the MLE. Recall that the sum, $Y = \sum_{i=1}^{n} X_i$, of these n binary observations is binomially distributed, $Y \sim \mathrm{Bin}(n, \pi)$. The associated likelihood function is simply the binomial p.m.f. from (2.17). This leads to a log-likelihood function of the form

$$\ell(\pi) = C + y\log(\pi) + (n - y)\log(1 - \pi)\,, \tag{5.32}$$

where C is a constant that does not affect the optimization.

Maximization of (5.32) proceeds by applying differential calculus. The log-likelihood derivative is

$$\ell'(\pi) = \frac{y}{\pi} - \frac{n - y}{1 - \pi}$$

which when set equal to zero produces the estimating equation $(1 - \pi)y - \pi(n - y) = 0$. Solving this for π yields $\hat{\pi}_{\mathrm{ML}} = Y/n$. This is the sample proportion, seen also with the MOM estimator in Example 5.2.3. To verify that this is a true MLE, find $\ell''(\pi) = -\pi^{-2}y - (1 - \pi)^{-2}(n - y)$. This is negative for any $y \in \{0, \ldots, n\}$ and any $\pi \in (0, 1)$; hence, the log-likelihood is strictly concave. Further, the log-likelihood is a minimum at the parameter boundaries: $\lim_{\pi \to 0}\ell(\pi) = \lim_{\pi \to 1}\ell(\pi) = -\infty$. Thus the stationary point at $\hat{\pi}_{\mathrm{ML}}$ must be a maximum. □

Example 5.3.4 Confidence intervals on π from a binomial sample (Example 5.3.3, continued). The binomial MLE $\hat{\pi}_{\mathrm{ML}} = Y/n$ possesses an approximate normal distribution for large n, with mean equal to π and with large-sample standard error built from the Fisher information number. The latter quantity here is $\mathcal{I}(\pi) = n/\{\pi(1 - \pi)\}$ (Exercise 5.1). Taking

the square root of the reciprocal Fisher information and using $\hat{\pi}_{\text{ML}}$ to estimate the unknown value of π produces the standard error

$$\text{se}[\hat{\pi}_{\text{ML}}] = \frac{1}{\sqrt{\mathcal{F}(\hat{\pi}_{\text{ML}})}} = \sqrt{\frac{\hat{\pi}_{\text{ML}}(1-\hat{\pi}_{\text{ML}})}{n}} \,. \tag{5.33}$$

With this, the Wald interval from (5.31) is

$$\hat{\pi}_{\text{ML}} \pm z_{\alpha/2}\sqrt{\frac{\hat{\pi}_{\text{ML}}(1-\hat{\pi}_{\text{ML}})}{n}} \,.$$

While simple and easy to apply, the actual coverage level for the basic Wald interval here is known to be highly erratic, even in large samples (Brown et al. 2001, 2002). (Typical recommendations for 'large sample' call for $n > 5/\min\{\pi, 1-\pi\}$, although this is not enough to overcome the erratic behavior of the interval.) Part of the problem arises from the use of a continuous (normal) distribution to approximate a discrete-valued statistic such as Y/n. A 'continuity correction' has been proposed to alleviate some of this concern – see the review in Blyth and Still (1983) – however, substantial irregularities still remain with use of this simple Wald interval for a binomial π.

Luckily, a wide variety of alternative intervals for π exist; see the presentation, and the accompanying discussion, in Brown et al. (2001). Here, two closed-form alternatives are presented for general use. The first is a very simple modification of the Wald interval given by Agresti and Coull (1998). It essentially replaces the ML point estimator by a slightly modified ratio,

$$\tilde{\pi} = \frac{Y + \frac{1}{2}z^2_{\alpha/2}}{n + z^2_{\alpha/2}},$$

and then operates under the same *estimator* $\pm$ *MOE* template. The MOE is again of the form $z_{\alpha/2}\text{se}[\tilde{\pi}]$, where

$$\text{se}[\tilde{\pi}] = \sqrt{\frac{\tilde{\pi}(1-\tilde{\pi})}{n + z^2_{\alpha/2}}} \,.$$

The final construction is

$$\tilde{\pi} \pm z_{\alpha/2}\sqrt{\frac{\tilde{\pi}(1-\tilde{\pi})}{n + z^2_{\alpha/2}}} \,. \tag{5.34}$$

This is known as an *Agresti–Coull* (*AC*) confidence interval for π.

Similar to the Wald interval, the AC interval in (5.34) is based on asymptotic arguments and thus is not valid with small samples. Brown et al. (2001) recommend that it be used only when $n > 40$.

Notice that for the popular confidence level of $1-\alpha = 0.95$, $z_{0.05/2} = z_{0.025} = 1.96 \approx 2$, from `qnorm(0.025, lower.tail=FALSE)`, and so a quick approximation for (5.34) employs $\tilde{\pi}' = (Y+2)/(n+4)$, with $\text{se}[\tilde{\pi}'] = \sqrt{\tilde{\pi}'(1-\tilde{\pi}')/(n+4)}$. This was, in fact, part of Agresti and Coull's original suggestion for an alternative interval estimator.

It is worth mentioning that the form of the AC point estimator $\tilde{\pi}$ (and also $\tilde{\pi}'$) is chosen purposefully: the addition of $\frac{1}{2}z^2_{\alpha/2}$ in the numerator and twice of it in the denominator pulls or *shrinks* the estimator away from the boundaries, $\pi = 0$ and $\pi = 1$, of the parameter space and

towards the central value of $\pi = \frac{1}{2}$. Near these boundaries, the Wald interval for π experiences much of its irregularity. Thus as a point estimator, $\tilde{\pi}$ is biased and, of course, deviates from the MLE and the MOM. If point estimation were the lone goal of the analysis, the MLE would be preferable. For interval estimation, however, the location of the interval's anchor is less crucial than the coverage quality of its eventual limits. As a result, the AC interval has gained substantial support in practical applications with sufficiently large samples.

Although not a likely occurrence in large-data analytics, the question does remain: what to do if $n \leq 40$? Again, a number of possibilities exist. One favored here is more complicated than (5.34), but still possesses a closed form. Owing to Wilson (1927), the interval is based on manipulating the log-likelihood derivative $\ell'(\pi)$ to produce valid confidence limits on π. Both Agresti and Coull (1998) and Brown et al. (2001) noted that the Wilson interval's coverage performance for bounding π exhibits noticeable stability. Including a continuity correction provides worthwhile improvement, although it also adds complexity to the final form:

$$\frac{\left(Y \pm \frac{1}{2}\right) + \frac{1}{2}z^2_{\alpha/2}}{n + z^2_{\alpha/2}} \pm z_{\alpha/2}\frac{\sqrt{\left(Y \pm \frac{1}{2}\right) - \frac{1}{n}\left(Y \pm \frac{1}{2}\right)^2 + \frac{1}{4}z^2_{\alpha/2}}}{n + z^2_{\alpha/2}},$$

where the repeated $\pm$ notation instructs the user to calculate the lower endpoint whenever a minus $(-)$ appears and to calculate the upper endpoint whenever a plus $(+)$ appears. (If $Y = 0$, set the lower endpoint to 0. If $Y = n$, set the upper endpoint to 1.) This is known as the *Wilson continuity-corrected* (WCC) interval for π.

In practice, the WCC interval exhibits stable small-sample coverage properties for sample sizes as low as $n \geq 5$. (With very small sample sizes, the coverage can be slightly conservative.) For any smaller sample size, the practical recommendation would be to recover more data before undertaking any statistical inferences on π.

Computation of these confidence limits is available in a variety of computer packages/platforms. In **R**, the AC interval and an *un*corrected Wilson interval are available via the `binom.confint()` function in the external *binom* package. The former is recommended for $n > 40$. (Although not recommended for use in any circumstance, the Wald interval is incorporated into the `prop.test()` function, similar to the `t.test()` function in Example 5.3.1.) For smaller samples sizes, direct coding is required for the WCC interval, although this would not be difficult for moderately experienced **R** users.

To illustrate, recall from Example 2.3.2 the study of a retail outlet's $n = 1024$ customers who might make a purchase during a sale. In Example 5.2.3, it was determined that $y = 506$ of the customers actually made a sale purchase, and so the MLE/MOM estimator is $\hat{\pi}_{\text{MOM}} = 506/1024$ or 49.41% of the customer base. For a 95% confidence interval, the sample size is large enough here to employ the AC form. In **R**, this gives

```
> binom.confint(x=506, n=1024, conf.level=0.95,
                          method='agresti-coull')
          method   x    n      mean     lower     upper
 1 agresti-coull 506 1024 0.4941406 0.4635975 0.5247276
```

The respective 95% AC limits appear under "`lower`" and "`upper`" ("`mean`" gives the MLE). They indicate that between 46.360% and 52.473% of the customer base was disposed to make a sale purchase. This information can be used by the outlet's marketing team when designing future sales events. (For more on this, see Example 5.4.4.) □

It is important to emphasize that instabilities with the Wald interval for single-sample binomial data arise primarily from complications when representing discrete binomial responses through a continuous normal distribution. In many other settings, the Wald approach operates quite well – at least with sufficiently large samples – and the binomial case is not representative of the method's larger applicability. The Wald methodology also extends easily to the multiparameter case, where it can be quite useful; see Equation (5.49).

An alternative strategy exists for using the likelihood function to build confidence intervals. This evaluates the likelihood at any potential value, θ, for the interval and compares it to the likelihood at the MLE $\hat{\theta}$. Specifically, start with the *likelihood ratio* (LR)

$$\Lambda(\theta) = \frac{L(\theta; X_1, \ldots, X_n)}{L(\hat{\theta}; X_1, \ldots, X_n)}. \tag{5.35}$$

As a function of θ, this ratio is always less than or equal to 1, since $\hat{\theta}$ maximizes L. Thus values of θ that drive $\Lambda(\theta)$ sufficiently close to 1 are a 'most likely' set of entries for the interval estimator. To quantify 'most likely,' note that (5.35) is a function of the data and, therefore, is itself a random quantity. In particular, in large samples, the transformation

$$G^2(\theta) = -2\log\{\Lambda(\theta)\}$$

satisfies $G^2(\theta) \dot{\sim} \chi^2(1)$, so that $P[G^2(\theta) \leq \chi^2_\alpha(1)] \approx 1 - \alpha$ (Wilks, 1938). Now, values of θ for which $G^2(\theta)$ is sufficiently small become plausible entries for a confidence interval, and thus this probability can be manipulated into a confidence statement that bounds θ. The result is called a $100(1-\alpha)\%$ *likelihood ratio (LR) interval* for θ.

Unfortunately, the LR interval can be difficult to implement, because it does not always produce closed-form expressions for the confidence limits. Indeed, in the general case, one cannot write the actual limits $L_\theta(X_1, \ldots, X_n) < \theta < U_\theta(X_1, \ldots, X_n)$ without more specific details about the nature of the likelihood. As a result, computer calculation is common. For example, a computationally intensive LR interval exists for the binomial parameter π in Example 5.3.4 and can be calculated in **R** via the `method='profile'` option in `binom.confint()`. See Exercise 5.19.

One can also construct confidence intervals on a case-by-case basis, sculpting each procedure to fit the features of the likelihood under study. For example, if $X \sim \text{Poisson}(\lambda)$, an equivalence relationship between Poisson and χ^2 cumulative distribution functions (c.d.f.s) can be used to build confidence limits for λ. Noted originally by Przyborowski and Wilenski (1935) and also by Garwood (1936), the relationship states that if $X \sim \text{Poisson}(\lambda)$, then $P[X > x] = P[V < 2\lambda]$, where $V \sim \chi^2(2x)$. From this, the following, simple confidence limits on λ can be derived:

$$\frac{1}{2}\chi^2_{1-(\alpha/2)}(2X) < \lambda < \frac{1}{2}\chi^2_{\alpha/2}(2X+2)\ . \tag{5.36}$$

If $X = 0$ in (5.36), set the lower endpoint to zero. Similar constructs exist for many other families of distributions; see the exposition in Piegorsch and Bailer (1997, Section 2.6) for more details.

5.3.5 Delta method intervals*

When domain-specific interest exists in some function, $h(\theta)$, of an unknown parameter, the ML invariance property ensures that $h(\hat{\theta})$ is the corresponding ML estimate. Building confidence

limits for $h(\theta)$ is somewhat more complicated, however. For instance, in the abstract, a $1-\alpha$ Wald interval for $h(\theta)$ is $h(\hat{\theta}) \pm z_{\alpha/2}\text{se}[h(\hat{\theta})]$. This will be a valid interval estimator if (i) $h(\hat{\theta})$ is (approximately) normal, so that the standard normal critical point $z_{\alpha/2}$ may be employed, and (ii) we can find the standard error of $h(\hat{\theta})$, at least to a good approximation.

More generally, one can appeal to the delta method from Section 2.3.9 for making inferences on any $h(\theta)$. In effect, when $\hat{\theta} \dot{\sim} \text{N}(\theta, \text{Var}[\hat{\theta}])$, the delta method gives $h(\hat{\theta}) \dot{\sim} \text{N}\left(h(\theta), \{h'(\theta)\}^2\text{Var}[\hat{\theta}]\right)$. The approximation improves as $n \to \infty$. From this, a delta method confidence interval is straightforward to derive: appeal to (5.7) and find

$$\text{se}[h(\hat{\theta})] = \sqrt{\text{Var}[h(\hat{\theta})]\Big|_{\theta=\hat{\theta}}} = \sqrt{\{h'(\hat{\theta})\}^2\text{Var}[\hat{\theta}]\Big|_{\theta=\hat{\theta}}} \approx |h'(\hat{\theta})|\ \text{se}[\hat{\theta}],$$

then construct the approximate Wald limits

$$h(\hat{\theta}) \pm z_{\alpha/2}|h'(\hat{\theta})|\text{se}[\hat{\theta}].$$

When there is more than one unknown parameter under study, a multivariate version of the delta method may be developed. The particulars extend beyond the scope here, but the general result may be stated: start with the vector of parameters $\boldsymbol{\vartheta} = [\theta_1 \cdots \theta_p]^{\text{T}}$, an unbiased estimator $\hat{\boldsymbol{\vartheta}} = [\hat{\theta}_1 \cdots \hat{\theta}_p]^{\text{T}}$, variances $\text{Var}[\hat{\theta}_j]$, and covariances $\text{Cov}[\hat{\theta}_j, \hat{\theta}_k]$ $(j \neq k)$. Assume interest focuses on some (univariate) function $h(\boldsymbol{\vartheta})$. Then under appropriate regularity conditions, the delta method gives, to first order, $E[h(\hat{\boldsymbol{\vartheta}})] \approx h(\boldsymbol{\vartheta})$ and

$$\text{Var}[h(\hat{\boldsymbol{\vartheta}})] \approx \sum_{j=1}^{p}\left(\frac{\partial h}{\partial \theta_j}\right)^2\text{Var}[\hat{\theta}_j] + 2\sum_{j=1}^{p-1}\sum_{k=j+1}^{p}\frac{\partial h}{\partial \theta_j}\frac{\partial h}{\partial \theta_k}\text{Cov}[\hat{\theta}_j, \hat{\theta}_k], \tag{5.37}$$

where $\partial h/\partial\theta_j$ is the partial derivative of $h(\boldsymbol{\vartheta})$ with respect to θ_j. For the simplest case where $p = 2$, (5.37) simplifies to

$$\text{Var}[h(\hat{\theta}_1, \hat{\theta}_2)] \approx \left(\frac{\partial h}{\partial \theta_1}\right)^2\text{Var}[\hat{\theta}_1] + 2\frac{\partial h}{\partial \theta_1}\frac{\partial h}{\partial \theta_2}\text{Cov}[\hat{\theta}_1, \hat{\theta}_2] + \left(\frac{\partial h}{\partial \theta_2}\right)^2\text{Var}[\hat{\theta}_2]. \tag{5.38}$$

If, as is common, any θ_j remains in the expression for the approximate variance, replace it with its estimator $\hat{\theta}_j$.

In large samples, approximate normality also holds: under suitable regularity conditions,

$$h(\boldsymbol{\vartheta}) \dot{\sim} \text{N}\left(h(\boldsymbol{\vartheta}), \text{Var}[h(\hat{\boldsymbol{\vartheta}})]\right),$$

where the variance is given by (5.37). This leads to approximate $1-\alpha$ confidence limits via the Wald construction

$$h(\hat{\boldsymbol{\vartheta}}) \pm z_{\alpha/2}\sqrt{\text{Var}[h(\hat{\boldsymbol{\vartheta}})]\Big|_{\boldsymbol{\vartheta}=\hat{\boldsymbol{\vartheta}}}}\ . \tag{5.39}$$

Example 5.3.5 Confidence interval for a ratio. Many operations in data analytics involve a ratio of parameters, $h(\theta_1, \theta_2) = \theta_1/\theta_2$. Ratios are notorious for their instability (especially if θ_2 is near zero), and any calculations using them must be performed with care. With this in mind, one can use the delta method to construct a confidence interval for θ_1/θ_2.

Begin with the MLE, $h(\hat{\theta}_1, \hat{\theta}_2) = \hat{\theta}_1/\hat{\theta}_2$. In large samples, this will be approximately normal, with mean θ_1/θ_2 and variance from (5.38):

$$\begin{aligned}\text{Var}\left[\hat{\theta}_1/\hat{\theta}_2\right] &\approx \left(\frac{\partial\{\theta_1/\theta_2\}}{\partial\theta_1}\right)^2 \text{Var}[\hat{\theta}_1] + 2\left(\frac{\partial\{\theta_1/\theta_2\}}{\partial\theta_1}\right)\left(\frac{\partial\{\theta_1/\theta_2\}}{\partial\theta_2}\right)\text{Cov}[\hat{\theta}_1, \hat{\theta}_2] \\ &\qquad + \left(\frac{\partial\{\theta_1/\theta_2\}}{\partial\theta_2}\right)^2 \text{Var}[\hat{\theta}_2] \qquad (5.40) \\ &= \frac{1}{\theta_2^2}\left\{\text{Var}[\hat{\theta}_1] - 2\frac{\theta_1}{\theta_2}\text{Cov}[\hat{\theta}_1, \hat{\theta}_2] + \frac{\theta_1^2}{\theta_2^2}\text{Var}[\hat{\theta}_2]\right\}.\end{aligned}$$

For use in practice, take the square root of (5.40) and replace the θ_js with their estimates to find $\text{se}[\hat{\theta}_1/\hat{\theta}_2]$. From this, a $1-\alpha$ Wald interval for the ratio is $\hat{\theta}_1/\hat{\theta}_2 \pm z_{\alpha/2}\text{se}[\hat{\theta}_1/\hat{\theta}_2]$.

One caveat: this Wald interval for θ_1/θ_2 can be unstable in small samples, depending on the values of θ_1 and θ_2 and on the underlying distribution of the data. In practice, an alternative confidence interval based on *Fieller's method* for a ratio (Fieller 1940) can operate with better stability and accuracy. The derivation is nontrivial, although manageable; see Buonaccorsi (2012). Fieller's theorem gives $1-\alpha$ limits for θ_1/θ_2 as

$$\begin{aligned}&\frac{\hat{\theta}_1}{\hat{\theta}_2} + \frac{\gamma}{1-\gamma}\left(\frac{\hat{\theta}_1}{\hat{\theta}_2} - \frac{\hat{\sigma}_{12}}{\hat{\sigma}_2^2}\right) \\ &\quad \pm \frac{z_{\alpha/2}}{(1-\gamma)|\hat{\theta}_2|}\sqrt{\hat{\sigma}_1^2 + \left(\frac{\hat{\theta}_1\hat{\sigma}_2^2}{\hat{\theta}_2} - 2\hat{\sigma}_{12}\right)\left(\frac{\hat{\theta}_1}{\hat{\theta}_2}\right) - \gamma\left(\hat{\sigma}_1^2 - \frac{\hat{\sigma}_{12}^2}{\hat{\sigma}_2^2}\right)},\end{aligned}$$

where $\hat{\sigma}_j^2 = \text{Var}[\hat{\theta}_j]$ $(j = 1, 2)$, $\hat{\sigma}_{12} = \text{Cov}[\hat{\theta}_1, \hat{\theta}_2]$, and $\gamma = z_{\alpha/2}^2\hat{\sigma}_2^2/\hat{\theta}_2^2$. □

5.3.6 Bootstrap intervals*

When the parent distribution of the data is unknown, likelihood-based methods for building confidence intervals are unavailable, because it is impossible to construct a complete likelihood for the unknown parameter. An alternative approach in this case approximates the unknown distribution of the data by simulating random outcomes a large number of times via computer. This is an application of the *Monte Carlo method*, a name coined by John von Neumann and Stanislaw Ulam while both were working at the Los Alamos National Laboratory in the 1940s (Anonymous 1949, p. 546). The term associates with the random outcomes in games of chance seen in Monte Carlo, the capital of Monaco and a well-known center for gambling. Other names for such random simulation of stochastic outcomes include *stochastic simulation*, *Monte Carlo simulation*, and *synthetic data generation*.

One specialized form of Monte Carlo simulation useful for building confidence intervals is the method of *bootstrap resampling* (Davison and Hinkley 1997). The bootstrap method is founded on an elegantly simple idea (Efron and Gong 1983): because the sampling distribution for a statistic is based on repeated samples with replacement – or 'resamples' – from the same population, one can use the computer to simulate repeated sampling, calculating the target statistic for each simulated sample. The resulting, simulated sampling distribution for the statistic is used to approximate its underlying, true sampling distribution. In the simplest case,

the empirical distribution of the data is used as the basis for the simulated resamples. These are drawn by computer from a theoretical distribution that matches the empirical distribution, and the statistic of interest is calculated for each simulated resample. The resampled values of the statistic provide an approximate distribution from which to construct confidence intervals.

To illustrate the approach, consider the following simple bootstrap confidence interval. Take a random sample, $\{X_1, \ldots, X_n\}$, from some population with unknown c.d.f. $F_X(x)$. The goal is to obtain an interval estimate of the unknown parameter θ based on an estimator $\hat{\theta}$ calculated from the data. Start with an estimate of $F_X(x)$ by counting how often any of the data values lie at or below a desired target argument x and dividing by n. This is called the *empirical c.d.f.*

$$\hat{F}_X(x) = \frac{\{\text{Number of } X_i \leq x\}}{n}.$$

If $\hat{F}_X(x)$ is a good estimate of $F_X(x)$, we can generate bootstrap samples as follows:

(1) Generate a bootstrap sample, say, $\{X_1^*, X_2^*, \ldots, X_n^*\}$ at random from the empirical c.d.f. $\hat{F}_X(x)$ using standard methods for simulating pseudorandom variates from c.d.f.s (Gentle 2003, Section 4.1).

(2) Calculate the statistic of interest from the bootstrap sample. Denote this as $\hat{\theta}^*$.

(3) Repeat steps (a) and (b) a large number, B, of times. Babu and Singh (1983) gave theoretical arguments for $B = n(\log n)^2$, although if $n < 60$, a common recommendation is to set at least $B = 2000$ for building confidence intervals.

(4) Assemble the target statistics from each of the bootstrap samples. Let $\hat{\theta}_1^*, \hat{\theta}_2^*, \ldots, \hat{\theta}_B^*$ be this collection of the resampled statistic.

This assemblage of bootstrapped statistics $\hat{\theta}_b^*$ $(b = 1, \ldots, B)$ represents an empirical estimate for the sampling distribution of $\hat{\theta}$. From this, we obtain a confidence interval on θ by selecting specified percentiles from the empirical distribution formed from the collection of $\hat{\theta}_b^*$s. This is known as the *percentile method* (DiCiccio and Romano 1988). For example, a 95% confidence interval on θ based on the percentile method takes the 2.5th percentile of the bootstrap distribution of the $\hat{\theta}_b^*$s for the lower bound and the 97.5th percentile for the upper bound.

In **R**, a variety of external packages can create bootstrap resamples. (One can also code the bootstrap directly, as desired.) For instance, the external *boot* package is recommended and ties directly to the textbook by Davison and Hinkley (1997).

5.4 Testing hypotheses

A second form of statistical inference useful in data analytics is based on the comparison of two competing hypotheses that describe the unknown parameter θ. This is known as *hypothesis testing*. Suppose a particular value of θ, say θ_o, defines a standard or objective condition for the underlying population. Then, the *null hypothesis* represents the condition that θ is θ_o, written as $H_o: \theta = \theta_o$. This is also called the 'no-effect hypothesis,' because θ_o generally indicates lack of some impact or effect such as no response to an external stimulus. The *alternative hypothesis* or *research hypothesis*, H_a, is a specification for θ that represents a plausible research alternative to H_o. (Some authors denote the alternative hypothesis as H_1.) The goal is determination of whether H_o or H_a represents a credible indication of the nature

of θ. To achieve this, the process is structured to arrive at a decision regarding H_o, that is, whether to reject it or accept it in light of the information about θ observed in the data.

Two fundamental probabilities lie at the core of testing these hypothesis statistically. The first is the probability of a false positive error: $\alpha = P[\text{reject } H_o | H_o \text{ true}]$. (Notice how the probability statement conditions on the unknown event that H_o is true.) This is the *Type I* or *false positive error rate*, and it is usually fixed in advance by the analyst. The second is the probability of a false negative error: $\beta = P[\text{accept } H_o | H_o \text{ false}]$. This is the *Type II* or *false negative error rate*. Associated with β is the *power* or *sensitivity* of the test: $P[\text{reject } H_o | H_o \text{ false}] = 1 - \beta$. For fixed α, the goal is to find a test of H_o (vs H_a) that minimizes β and thus maximizes power to the greatest extent possible.

Given a set of data, a test statistic that addresses the effect being studied in H_o is calculated. From this, a *rejection region* – an older term is *critical region* – is formed where, if the test statistic falls into this region, H_o is rejected in favor of H_a. (If not, we fail to reject H_o.) Equivalently, one can find the *P-value* of the test, defined as the probability under H_o of observing a test statistic as extreme as or more extreme than that actually observed. Note that 'more extreme' is defined in the context of H_a. For example, when testing $H_o: \theta = \theta_o$ versus a 'one-sided' alternative such as $H_a: \theta > \theta_o$, 'more extreme' corresponds to values of the test statistic supporting $\theta > \theta_o$. By contrast, for the 'two-sided' alternative $H_a: \theta \neq \theta_o$, 'more extreme' corresponds to values of the test statistic supporting either $\theta > \theta_o$ or $\theta < \theta_o$.

Small P-values indicate departure from H_o; thus for a fixed level of α, one rejects H_o when $P \leq \alpha$. Since the significance of departure from H_o is captured by whether or not α exceeds P, α is often called the *significance level* of the test. Indeed, the technical terminology has evolved to focus on the 'significance' of the test outcome; see Table 5.3.

Hypothesis tests may be derived under a number of different conditions. These correspond to analogous criteria for building confidence intervals. Indeed, the two concepts are tautologically related: suppose one is testing $H_o: \theta = \theta_o$ versus $H_a: \theta \neq \theta_o$. It is possible to build a $1 - \alpha$ confidence region for θ from the complement of the rejection region (often called the 'acceptance region'): if the confidence region fails to contain θ_o, one will reject H_o at the α level of significance and vice versa (Casella and Berger 2002, Section 9.2).

It is important to emphasize that the rationale for specifying a one-sided alternative must be determined *prior* to examining the data. The analyst cannot first study the observations, then use the information in, say, $\overline{X}$ to choose the direction for H_a in order to 'improve' the opportunity to reject H_o. One-sided hypotheses, tests, and confidence limits may be based only on a priori or existing domain-specific knowledge for the testing or estimation problem

Table 5.3 Terminology for hypothesis test outcomes.

H_o	H_a	Test outcome	Terminology
$\theta = \theta_o$	$\theta \neq \theta_o$	Reject H_o	θ significantly different from θ_o
		Fail to reject H_o	θ insignificantly (or not significantly) different from θ_o
$\theta = \theta_o$	$\theta > \theta_o$	Reject H_o	θ significantly greater than θ_o
		Fail to reject H_o	θ not significantly greater than θ_o
$\theta = \theta_o$	$\theta < \theta_o$	Reject H_o	θ significantly smaller than θ_o
		Fail to reject H_o	θ not significantly smaller than θ_o

at hand. (For example, a biomedical scientist may question whether proximity to a pollutant source leads to increased cancer in an exposed population. The underlying scientific rationale clearly calls for selection of a one-sided, increasing alternative hypothesis.) If no such prior subject-matter justification is available, however, the default selection is always a two-sided H_a. Any other use violates the basic probability statements from which the inferences were formed and will in fact lead to incorrect statistical inferences. While data can certainly be 'mined' in an exploratory manner to search for unrecognized features or trends, construction of formal statistical inferences cannot be based on 'snooping' through the concurrent observations.

5.4.1 Single-sample tests for normal (Gaussian) parameters

Perhaps the most well-known hypothesis test is the *t-test*, so-named because it is based on Student's *t*-distribution from Section 2.3.10. Many forms exist for the *t*-test; consider here the simple case of testing from a normal random sample. Let $X_i \sim$ i.i.d. $N(\mu, \sigma^2)$, $i = 1, \ldots, n$, for unknown μ and σ^2. Consider testing H_o: $\mu = \mu_o$ versus H_a: $\mu \neq \mu_o$ at significance level α.

For this *t*-test, estimate μ with its MLE from Example 5.2.5, $\overline{X}$, and use the unbiased sample variance, S^2, from (3.5) to estimate σ^2. The standard error of $\overline{X}$ is $\text{se}[\overline{X}] = S/\sqrt{n}$. From these, build the test statistic

$$T = \frac{\overline{X} - \mu_o}{S/\sqrt{n}}, \tag{5.41}$$

where $\overline{X} \sim N(\mu, \sigma^2/n)$. *If* H_o is true, the expected value of $\overline{X}$ is $\mu = \mu_o$ and then the statistic in (5.41) is a Studentized *t* random variable. Thus, conditioning on the event that H_o is true produces $T \sim t(n-1)$.

To build the rejection criterion, consider the region $|T| \geq t_{\alpha/2}(n-1)$. When H_o is true and $T \sim t(n-1)$, the corresponding false positive (Type I) error rate is

$$\begin{aligned} P[\text{reject } H_o | H_o \text{ true}] &= P[|T| \geq t_{\alpha/2}(n-1) | \mu = \mu_o] \\ &= P[|T| \geq t_{\alpha/2}(n-1) | T \sim t(n-1)], \end{aligned} \tag{5.42}$$

where the absolute value in the rejection criterion forces consideration of both the lower and upper tails of the $t(n-1)$ reference distribution. By the symmetry of the *t* p.d.f., however, the latter probability in (5.42) is just $2 \times P[T \geq t_{\alpha/2}(n-1) | T \sim t(n-1)]$, and from the definition of the *t* critical point, this becomes $(2)(\alpha/2) = \alpha$. So, (5.42) reduces to $P[\text{reject } H_o | H_o \text{ true}] = \alpha$, satisfying the false positive error requirement. This defines the *one-sample t-test* of H_o: $\mu = \mu_o$ versus H_a: $\mu \neq \mu_o$.

The corresponding *P*-value is the probability under H_o that the (null) reference distribution exceeds the observed value of the test statistic. Denote this latter quantity as t_{calc}. When H_o is true, the *T* statistic is referred to $t(n-1)$, so the *P*-value is

$$P = P[|t(n-1)| \geq |t_{\text{calc}}|] = 2P[t(n-1) \geq |t_{\text{calc}}|],$$

that is, appeal to the symmetry of the $t(n-1)$ p.d.f. and find twice its upper tail area past $|t_{\text{calc}}|$. In **R**, this can be calculated via the command

```
> 2*pt( abs(tcalc), df=n-1, lower.tail=FALSE )
```

The one-sample t-test satisfies the confidence interval/hypothesis test tautology mentioned earlier. When testing $H_o: \mu = \mu_o$ versus the two-sided alternative $H_a: \mu \neq \mu_o$ via a level-α t-test, the corresponding (two-sided) $1-\alpha$ confidence region for μ from (5.17) will fail to contain μ_o whenever the test rejects H_o and vice versa.

One-sided tests are similar. For instance, when testing $H_o: \mu = \mu_o$ against the one-sided alternative $H_a: \mu > \mu_o$, employ the rejection region $T \geq t_\alpha(n-1)$, with corresponding P-value $P = P[t(n-1) \geq t_{calc}]$. In **R**, find the P-value via `pt(tcalc, df=n-1, lower.tail=FALSE)`.

Example 5.4.1 ***t*-test for μ from a normal sample (Example 5.3.1, continued).** Consider again the myocardial infarction data in Table 4.2. The average age of attack for these data was seen to be $\overline{X} = 62.8137$, while the standard error of $\overline{X}$ was found as $\text{se}[\overline{X}] = S/\sqrt{126} = 8.3421/11.2250 = 0.7432$. In Example 4.1.7, a normal quantile plot suggested a tenable fit of the normal model to these data. Thus if interest existed in testing a particular value of μ_o for this population's mean age of attack, the t-test would be an option.

At the time of this writing, US adults are eligible for medical coverage through the Federal Medicare system after they turn age 65. Interest with this study population includes whether its true mean age-to-attack, μ, is earlier than the Medicare entry age. This translates to testing $H_o: \mu = 65$ versus the one-sided alternative $H_a: \mu < 65$. We reject H_o in favor of H_a via the t-test when $T \leq -t_\alpha(n-1)$. At $\alpha = 0.05$, the necessary critical point is $-t_{0.05}(125) = -1.6571$, found in **R** via `qt(0.05, 125, lower.tail=TRUE)`.

The pertinent test statistic here is

$$T = \frac{\overline{X} - \mu_o}{S/\sqrt{n}} = \frac{62.8137 - 65}{8.3421/\sqrt{126}},$$

producing $t_{calc} = -2.9418$. Since $t_{calc} = -2.9418 < -1.6571 = -t_{0.05}(125)$, we reject H_o for these data and conclude that the mean age of attack for this population is significantly lower than 65 years. Medicare planners can use this knowledge for planning medical coverage and care needs for this population of patients.

The P-value here is $P[t(n-1) \leq t_{calc}] = P[t(125) \leq -2.9418] = 0.0019$, found using the **R** command `pt(-2.9418, df=125, lower.tail=TRUE)`. Since $P = 0.0019 \leq 0.05 = \alpha$, the decision is again to reject H_o in favor of H_a.

The **R** routine `t.test()` provides all these calculations in one combined operation. Simply use

```
> t.test( X, alternative="less", mu=65, conf.level=0.95 )
```

with output (edited)

```
        One Sample t-test
data:  X
t = -2.9418, df = 125, p-value = 0.001945
alternative hypothesis: true mean is less than 65
95 percent confidence interval:
    -Inf  64.04526
sample estimates:
mean of x
 62.81372
```

In particular, the P-value of $P = 0.0019$ is given after '`p-value =`' (the test statistic is listed after '`t =`').

Note that with these same data, the `t.test()` output in Example 5.3.1 exhibited some marked differences. In that example, the **R** function was used only to find a 95% confidence interval for μ and not to construct a targeted hypothesis test. Thus no specific value was given for μ_o, and no `alternative=` option was supplied. As a result, the `t.test()` output in that example used the **R** defaults of $\mu_o = 0$ and a two-sided alternative (settings with little sensible interpretation with these data). Of course, the 95% confidence limits it provided were valid, and useful, because they gave a plausible range of values for the population's mean age of attack. The inference provided by the hypothesis test here is arguably more limited, if also more specific.

Speaking of confidence limits, notice in the `t.test()` output above that a set of confidence limits is provided after '`95 percent confidence interval`.' Because the `conf.level=0.95` option was entered, the confidence level was set to 95% while, because `alternative='less'` was specified, the limits are one sided. That is, they correspond to the one-sided upper bound from (5.18). This is indicated by the `-Inf` output for the lower confidence limit. As expected, the upper bound does not reach the null value of $\mu = 65$. □

If desired, it is also possible to fashion tests of the variance parameter in this single-sample, normal setting. Suppose interest targets a test of $H_o: \sigma^2 = \sigma_o^2$ versus $H_a: \sigma^2 \neq \sigma_o^2$. The χ^2 relationship employed to build the confidence limits on σ^2 in (5.21) also allows for construction of a test statistic. Under H_o, the statistic

$$C^2 = \frac{(n-1)S^2}{\sigma_o^2}$$

is distributed as $\chi^2(n-1)$. Thus the rejection region

$$C^2 \leq \chi^2_{1-\frac{\alpha}{2}}(n-1) \quad \text{or} \quad C^2 \geq \chi^2_{\alpha/2}(n-1)$$

can be shown to satisfy a level-α false positive error requirement. Similar to the caveat in Section 5.3.2, however, allocation here of equal $\alpha/2$ false positive error probability to both tails in the χ^2 reference distribution is done purely for convenience. It is generally suboptimal, and alternative tail area allocations should be considered on a case-by-case basis.

5.4.2 Two-sample tests for normal (Gaussian) parameters

Hypothesis tests can also be developed for testing across two independent samples. To establish the concepts, suppose data from the first sample are $X_{1j} \sim$ i.i.d. $N(\mu_1, \sigma_1^2)$, $j = 1, \ldots, n_1$, independent of data from the second sample $X_{2j} \sim$ i.i.d. $N(\mu_2, \sigma_2^2)$, $j = 1, \ldots, n_2$. Assume both samples' sets of parameters μ_1, σ_1^2 and μ_2, σ_2^2 are unknown. All of the tests discussed in this section for testing these various parameters will be constructed essentially by inverting their corresponding confidence intervals from Section 5.3.3.

Begin with tests for $H_o: \mu_1 = \mu_2$ versus $H_a: \mu_1 \neq \mu_2$, that is, testing whether the means of the two samples are equal. By focusing on the difference $\Delta = \mu_1 - \mu_2$, the hypotheses become $H_o: \Delta = 0$ versus $H_a: \Delta \neq 0$, for which the pivotal quantity from (5.22) leads directly to the

Studentized test statistic

$$T = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{1}{n_1}S_1^2 + \frac{1}{n_2}S_2^2}}\,. \tag{5.43}$$

If H_o is true, the T statistic in (5.43) has the approximate, null, reference distribution $T \dot{\sim} t(\nu_{WS})$, where the d.f. ν_{WS} are given by the Welch–Satterthwaite correction in (5.23). Reject H_o when $|T| \geq t_{\alpha/2}(\nu_{WS})$. The corresponding approximate P-value is

$$P \approx P[|t(\nu_{WS})| \geq |t_{calc}|] = 2P[t(\nu_{WS}) \geq |t_{calc}|],$$

where t_{calc} is the observed value of the test statistic in (5.43). In **R**, find the P-value via `2*pt(abs(tcalc),df=nuWS,lower.tail=FALSE)`, where `tcalc` is the calculated test statistic and `nuWS` are the Welch–Satterthwaite d.f. from (5.23).

As above, the Welch–Satterthwaite approximation for the d.f. improves as $\min\{n_1, n_2\} \to \infty$. It is quite accurate, however, and the approximate false positive error rate will be very near α for sample sizes as low as $n_i = 10$.

An aside: readers may wonder what happened to 'Δ_o' in the Studentized test statistic (5.43). That is, if mimicking the Studentizing operation in the analogous statistic from (5.41), one would expect the numerator for T in (5.43) to appear as $\overline{X}_1 - \overline{X}_2 - \Delta_o$. In fact, it does: the null value Δ_o is simply 0 here, leading to a numerator of the form $\overline{X}_1 - \overline{X}_2 - 0 = \overline{X}_1 - \overline{X}_2$. Indeed, to test for any other value of Δ_o in H_o: $\mu_1 - \mu_2 = \Delta_o$, one simply subtracts the specified value of Δ_o in the numerator of (5.43).

One-sided tests are similar. For instance, when testing H_o: $\mu_1 = \mu_2$ against the alternative H_a: $\mu_1 > \mu_2$, reconstruct the hypotheses as H_o: $\Delta = 0$ versus H_a: $\Delta > 0$ and use the rejection region $T \geq t_\alpha(\nu_{WS})$, with corresponding P-value $P = P[t(\nu_{WS}) \geq t_{calc}]$. In **R**, find the P-value via `pt(tcalc, df=nuWS, lower.tail=FALSE)`.

Example 5.4.2 **Two-sample *t*-test for Lung Function data (Example 5.3.2, continued).** Return to the lung function data from Table 5.1, measuring 'FEV1' in 26-year-old asthmatics.

In Example 5.3.2, a 90% confidence interval for the difference in mean FEV1 scores was seen to contain the value $\mu_1 - \mu_2 = 0$, indicating a lack of any difference in mean FEV1 between smokers and nonsmokers. To formalize this, test the hypothesis H_o: $\mu_1 = \mu_2$ against H_a: $\mu_1 \neq \mu_2$ at the 10% significance level.

The necessary statistics were given in Table 5.2, from which the test statistic in (5.43) evaluates to

$$\frac{97.5962 - 97.7500}{\sqrt{\frac{121.9386}{265} + \frac{150.7644}{88}}} = \frac{-0.1538}{\sqrt{2.1734}}$$

or simply $t_{calc} = -0.1043$. The corresponding Welch–Satterthwaite correction for the d.f. gave $\nu_{WS} = 136$, so the P-value is $P \approx 2P[t(136) \geq |-0.1043|] = 0.9171$. As this is greater than $\alpha = 0.10$, we fail to reject H_o and conclude that no significant difference exists between the mean FEV1 scores. Notice that this is consistent with the inferences reached using the confidence interval in Example 5.3.2, illustrating the tautology between confidence intervals and hypothesis tests.

To perform these calculations directly in **R**, use

```
> t.test( X1, X2, conf.level=0.90, var.equal=FALSE )
```

The output is identical to that seen in Example 5.3.2, where the output statistics corroborate those calculated earlier. While decreases in FEV1 are often observed among older, lifelong smokers, the inferences here suggest that such effects are not as evident among individuals in their 20s, at least for this population of asthmatics. □

If one assumes that the two population variances are equal, $\sigma_1^2 = \sigma_2^2 = \sigma^2$ (say), the independent-samples t-test in (5.43) collapses to an exact construction. The numerator of the test statistic remains the same, and the denominator now employs the pooled variance estimator, S_p^2, from (5.25). The result is an appropriate modification of the pivotal quantity in (5.26) for testing $\mathrm{H_o}$: $\Delta = 0$:

$$T = \frac{\overline{X}_1 - \overline{X}_2}{S_p\sqrt{\dfrac{1}{n_1} + \dfrac{1}{n_2}}}. \tag{5.44}$$

Under a homogeneous-variance assumption, the null reference distribution for T in (5.44) is *exactly* $T \sim t(n_1 + n_2 - 2)$. Reject $\mathrm{H_o}$ against $\mathrm{H_a}$: $\Delta \neq 0$ when $|T| \geq t_{\alpha/2}(n_1 + n_2 - 2)$. In **R**, if `X1` and `X2` are vectors containing the data from each independent sample, then the command `t.test(X1, X2, var.equal=TRUE)` can conduct the test and supply pertinent statistics.

The corresponding exact P-value is

$$P = P[|t(n_1 + n_2 - 2)| \geq |t_{\mathrm{calc}}|] = 2P[t(n_1 + n_2 - 2) \geq |t_{\mathrm{calc}}|],$$

where t_{calc} is the observed value of the test statistic in (5.44). In **R**, use

```
> 2 * pt( abs(tcalc), df=n1+n2-2, lower.tail=FALSE )
```

One-sided tests are similar. For instance, when testing $\mathrm{H_o}$: $\mu_1 = \mu_2$ against the one-sided alternative $\mathrm{H_a}$: $\mu_1 > \mu_2$, reconstruct the hypotheses as $\mathrm{H_o}$: $\Delta = 0$ versus $\mathrm{H_a}$: $\Delta > 0$ and employ the rejection region $T \geq t_\alpha(n_1 + n_2 - 2)$. The corresponding P-value is $P = P[t(n_1 + n_2 - 2) \geq t_{\mathrm{calc}}]$. For the P-value in **R**, use `pt(tcalc, df=n1+n2-2, lower.tail=FALSE)` or simply call

```
> t.test( X1, X2, var.equal=TRUE, alt='greater' )
```

to conduct the test and acquire the pertinent statistics.

A specialized t-test may also be developed for the particular case where observations are recorded in *matched pairs*, (X_{1j}, X_{2j}), with $X_{ij} \sim$ i.i.d. $\mathrm{N}(\mu_i, \sigma_i^2)$, $i = 1, 2$; $j = 1, \ldots, n$.

As in Section 5.3.3, we translate the paired data to differences $D_j = X_{1j} - X_{2j}$, where $D_j \sim$ i.i.d. $\mathrm{N}(\mu_D, \sigma_D^2)$. Here again, the differences act as a single sample, allowing for construction of a paired t statistic essentially similar to the single-sample Studentized statistic in (5.41).

For this paired-samples t-test, the two-sided hypotheses are $\mathrm{H_o}$: $\mu_D = 0$ versus $\mathrm{H_a}$: $\mu_D \neq 0$, where $\mu_D = \mu_1 - \mu_2$. The test statistic is then

$$T_D = \frac{\overline{D}}{S_D/\sqrt{n}}. \tag{5.45}$$

Reject $\mathrm{H_o}$ when $|T_D| \geq t_{\alpha/2}(n-1)$. The corresponding exact P-value is

$$P = P[|t(n-1)| \geq |t_{\mathrm{calc}}|] = 2P[t(n-1) \geq |t_{\mathrm{calc}}|],$$

where t_{calc} is the observed value of the test statistic in (5.45). In **R**, use

```
> 2 * pt( abs(tcalc), df=n-1, lower.tail=FALSE )
```

One-sided tests are similar. For instance, when testing H_o: $\mu_1 = \mu_2$ against the one-sided alternative H_a: $\mu_1 > \mu_2$, reconstruct the hypotheses as H_o: $\mu_D = 0$ versus H_a: $\mu_D > 0$ and employ the rejection region $T_D \geq t_\alpha(n-1)$. The corresponding P-value is $P = P[t(n-1) \geq t_{\text{calc}}]$. For the P-value in **R**, use `pt(tcalc, df=n-1, lower.tail=FALSE)` or simply call

```
> t.test( X1, X2, paired=TRUE, alt='greater' )
```

to conduct the test and acquire the pertinent statistics.

In the general two-sample normal setting, one can also test hypotheses for the two population variances σ_1^2 and σ_2^2. Suppose again that $X_{1j} \sim$ i.i.d. $\text{N}(\mu_1, \sigma_1^2)$, $j = 1, \ldots, n_1$, independent of $X_{2j} \sim$ i.i.d. $\text{N}(\mu_2, \sigma_2^2)$, $j = 1, \ldots, n_2$. Assume that all the population parameters are unknown and focus interest on testing H_o: $\sigma_1^2 = \sigma_2^2$ versus H_a: $\sigma_1^2 \neq \sigma_2^2$. Notice that this is equivalent to testing the variance ratio: H_o: $\sigma_1^2/\sigma_2^2 = 1$ versus H_a: $\sigma_1^2/\sigma_2^2 \neq 1$.

Similar to the (exact) confidence interval on σ_1^2/σ_2^2 in (5.28), one builds the test statistic from an F ratio

$$F_{12} = \frac{S_1^2}{S_2^2},$$

where S_1^2 and S_2^2 are the independent sample variances. Under H_o, $F_{12} \sim F(n_1 - 1, n_2 - 1)$; therefore, for a level-α test, reject H_o in favor of H_a when

$$F_{12} \leq F_{1-\frac{\alpha}{2}}(n_1 - 1, n_2 - 1) \text{ or } F_{12} \geq F_{\alpha/2}(n_1 - 1, n_2 - 1). \tag{5.46}$$

Using a clever manipulation of the relationship between F_{12} and $1/F_{12}$ – see Section 2.3.10 – one can simplify the rejection region here. Let $S^2_{(2)} = \max\{S_1^2, S_2^2\}$, $S^2_{(1)} = \min\{S_1^2, S_2^2\}$, and denote $n^*_{(j)}$ as the sample size associated with each $S^2_{(j)}$ $(j = 1, 2)$. Then, the rejection region in (5.46) is equivalent to

$$\frac{S^2_{(2)}}{S^2_{(1)}} \geq F_{\alpha/2}(n^*_{(2)} - 1, n^*_{(1)} - 1).$$

In either case, the corresponding P-value is $P = 2\ P[F(n^*_{(2)}, n^*_{(1)}) \geq S^2_{(2)}/S^2_{(1)}]$.

In **R**, the `var.test()` function can perform these operations with independent, two-sample, normal data.

5.4.3 Walds tests, likelihood ratio tests, and 'exact' tests*

The tautologous relationship between confidence intervals and hypothesis tests also applies to the general Wald statistic in Section 5.3.4, and Wald intervals are easily inverted to produce hypothesis tests. From a random sample $X_i \sim$ i.i.d. $f_X(x|\theta)$, $i = 1, \ldots, n$, consider tests on the generic parameter θ via H_o: $\theta = \theta_\text{o}$ versus H_a: $\theta \neq \theta_\text{o}$. To do so, find the MLE $\hat{\theta}$ and its standard error $\text{se}[\hat{\theta}]$. In large samples, $\hat{\theta} \dot{\sim} \text{N}(\theta, \text{se}^2[\hat{\theta}])$, so consider the standardized test statistic

$$Z = \frac{\hat{\theta} - \theta_\text{o}}{\text{se}[\hat{\theta}]}\ . \tag{5.47}$$

If H_o is true, the (approximate) expected value of $\hat{\theta}$ is $\theta = \theta_o$, and then this Z statistic is (approximately) standard normal. That is, conditioning on the event that H_o is true produces $Z \dot{\sim} N(0, 1)$.

Similar to the t-test construction in (5.42), the consequent rejection region is $|Z| \geq z_{\alpha/2}$ and the approximate false positive (Type I) error rate is

$$P[\text{reject } H_o | H_o \text{ true}] = P[|Z| \geq z_{\alpha/2} | \theta = \theta_o] \approx 2\{1 - \Phi(z_{\alpha/2})\}, \tag{5.48}$$

where $\Phi(z)$ is the standard normal c.d.f. from (2.35). Using the definition of $z_{\alpha/2}$, (5.48) reduces to

$$P[\text{reject } H_o | H_o \text{ true}] \approx 2\left\{1 - \left(1 - \frac{\alpha}{2}\right)\right\} = \alpha,$$

satisfying the false positive error requirement, at least approximately. (The approximation improves as $n \to \infty$.) This defines the *Wald test* of $H_o: \theta = \theta_o$ versus $H_a: \theta \neq \theta_o$, via the rejection region $|Z| \geq z_{\alpha/2}$.

The corresponding Wald test P-value is the probability under H_o that the (null) reference distribution exceeds the observed value of the test statistic. Denote this latter quantity as z_{calc}. The P-value is then

$$P = P[|Z| \geq |z_{calc}|] \approx 2\{1 - \Phi(|z_{calc}|)\},$$

that is, twice the upper tail area past $|z_{calc}|$ from a standard normal reference distribution. In **R**, this can be calculated via `2*pnorm(abs(zcalc), lower.tail=FALSE)`.

It is interesting to note that an equivalent rejection region for the two-sided Wald test can be constructed from a χ^2 critical point. Recall from Section 2.3.10 that if $Z \sim N(0, 1)$, then $Z^2 \sim \chi^2(1)$. Thus the two-sided Wald rejection region can also be written as $Z^2 \geq \chi^2_\alpha(1)$. The corresponding approximate P-value is $P[\chi^2(1) \geq z^2_{calc}]$.

The tautology between confidence intervals and hypothesis tests remains in effect here. When testing $H_o: \theta = \theta_o$ versus the two-sided alternative $H_a: \theta \neq \theta_o$ via a level-α Wald test, the corresponding two-sided $1-\alpha$ confidence region for θ will fail to contain θ_o whenever the test rejects H_o and vice versa.

For testing $H_o: \theta = \theta_o$ against the one-sided alternative $H_a: \theta > \theta_o$, employ the rejection region $Z \geq z_\alpha$, with corresponding approximate P-value $P \approx 1 - \Phi(z_{calc})$. In **R**, find the P-value via `pnorm(zcalc, lower.tail=FALSE)`.

In similar fashion, one can invert LR intervals into hypothesis tests of $H_o: \theta = \theta_o$ versus $H_a: \theta \neq \theta_o$. Start with the test statistic

$$G^2 = -2 \log \left\{ \frac{L(\theta_o; X_1, \ldots, X_n)}{L(\hat{\theta}; X_1, \ldots, X_n)} \right\}.$$

Under H_o, $G^2 \dot{\sim} \chi^2(1)$, so reject H_o in favor of H_a when $G^2 \geq \chi^2_\alpha(1)$. The corresponding, approximate P-value is $P \approx P[\chi^2(1) \geq g^2_{calc}]$, where g^2_{calc} is the observed value of the LR statistic.

Alert readers will notice that both Z^2 and G^2 appeal in large samples to the same reference distribution, $\chi^2(1)$. This is no coincidence: in most settings, the limiting value and large-sample reference distribution for both these likelihood-based methods will be the same. We call this a form of *asymptotic equivalence* among the test statistics. As $n \to \infty$, the methods provide essentially the same inference on θ, although, depending on the specific likelihood under study, they can vary substantially in small samples.

Extensions to tests for the multiparameter case are also possible with these likelihood-based approaches; see, for example, Cox (1988).

Example 5.4.3 Wald test and LR test for μ from a normal sample. With a single sample from a normal distribution, the Wald test collapses to the t-test from Example 5.4.1. To see this, start with a normal random sample $X_i \sim$ i.i.d. N(μ, σ^2), $i = 1, \ldots, n$, for unknown μ and σ^2. Set the hypotheses as H_o: $\mu = \mu_o$ versus H_a: $\mu \neq \mu_o$ and operate at significance level α. The Wald statistic from (5.47) takes the MLE of μ, subtracts the null value μ_o, and divides this difference by the standard error of the MLE. In this normal case, however, the MLE of μ is $\overline{X}$, with standard error se$[\overline{X}] = S/\sqrt{n}$. The Wald statistic is, therefore,

$$\frac{\overline{X} - \mu_o}{S/\sqrt{n}}.$$

Clearly, this is the T statistic in (5.41), so that the two test statistics are one and the same.

The Wald procedure approximates the large-sample, null, reference distribution for T as N(0,1), producing a large-sample rejection region of the form $|T| \geq z_{\alpha/2}$. Since we know from Section 5.4.1 that the exact reference distribution for T is $t(n-1)$, however, the exact rejection region is $|T| \geq t_{\alpha/2}(n-1)$. Indeed, as $n \to \infty$, $t_{\alpha/2}(n-1) \to z_{\alpha/2}$, and thus the two rejection regions will in fact converge to one another.

It is interesting to note that, here, the LR test for H_o: $\mu = \mu_o$ produces a rejection region of the form

$$\frac{n(\overline{X} - \mu_o)^2}{S^2} \geq F_\alpha(1, n-1)$$

(Casella and Berger 2002, Section 8.2), where $F_\alpha(1, n-1)$ is the upper-α critical point from an F-distribution with 1 and $n-1$ d.f. Observe that this rejection region is equivalent to $T^2 \geq t_\alpha^2(n-1)$, because a squared $t(n-1)$ variate is distributed as $F(1, n-1)$. Thus in this simple normal setting, the t-, Wald, and LR tests for μ all coincide. □

Example 5.4.4 Tests for π from a binomial sample. When sampling from a binomial distribution, interest often exists in testing whether the probability parameter π equals a particular π_o. Suppose a random sample is taken of n Bernoulli trials, $X_i \sim$ i.i.d. Bin$(1, \pi)$, such that the sum, $Y = \sum_{i=1}^n X_i$, is $Y \sim$ Bin(n, π). The basic Wald test of H_o: $\pi = \pi_o$ versus H_a: $\pi \neq \pi_o$ can be constructed from the MLE $\hat{\pi}_{ML} = Y/n$, using the standard error in (5.33). The test statistic is

$$W = \frac{\hat{\pi}_{ML} - \pi_o}{\text{se}[\hat{\pi}_{ML}]},$$

referenced in large samples to $W \dot{\sim}$ N(0, 1). This can be conducted in **R** via the `prop.test()` function.

As seen with the binomial confidence intervals in Example 5.3.4, however, the Wald approach can exhibit instabilities with a single binomial sample, and alternative methods are indicated. One could construct an LR test for H_o here (Exercise 5.30), and for sufficiently large n, this exhibits reasonable properties.

Another option valid for any sample size is a computer-intensive strategy known as an 'exact test' of H_o. As the binomial likelihood is defined over a discrete, finite, sample space, it is feasible in theory to enumerate all possible configurations for Y under H_o and compare these

to the outcome actually observed. The corresponding P-value is the probability of recovering a configuration as extreme as or more extreme (with respect to H_a) than the observed outcome.

In **R**, the binomial exact test is available via the `binom.test()` function. To illustrate, recall from Example 2.3.2 the study of a retail outlet's $n = 1024$ affinity customers who might make a purchase during a sale. In Example 5.2.3, it was determined that $Y = 506$ of the customers made a purchase, so the MLE was $\hat{\pi}_{ML} = 506/1024$ or 49.41% of the affinity base. The original question raised by the outlet concerned calculations with $\pi = 50\%$ (Example 2.3.2), so consider testing $H_o: \pi = \frac{1}{2}$ versus $H_a: \pi \neq \frac{1}{2}$ via the binomial exact test. Set $\alpha = 0.05$. The **R** command

```
> binom.test( x=506, n=1024, p=0.50, conf.level=0.95,
                                    alternative='two.sided' )
```

produces

```
        Exact binomial test
data:  506 and 1024
number of successes = 506, number of trials = 1024,
                                              p-value = 0.7311
alternative hypothesis: true probability of success is not
                                                   equal to 0.5
95 percent confidence interval:
 0.4630844 0.5252306
sample estimates:
probability of success
             0.4941406
```

The exact test provides a P-value of $P = 0.7311$ (following "`p-value =`") for testing H_o against H_a. As this is clearly greater than $\alpha = 0.05$, we fail to reject H_o with these data and conclude that the probability of a customer participating in the sale does not appear to differ significantly from 50%. One might call this a "50–50 chance," indicating no swing in the customers' purchasing probability away from 50% in response to the sale. (The outlet's marketing department could use this information when planning future sales events such as this.) □

Notice in the previous example that the `binom.test()` output also provided a "`95 percent confidence interval`" for the unknown value of π. This corresponds to an interval given by Clopper and Pearson (1934). It literally inverts the binomial exact test to produce the confidence limits, taking advantage of the test-interval tautology discussed earlier. The method is *very* conservative – Brown et al. (2001) called it "wastefully" so – and is not generally recommended for modern, practical use. Other possibilities, mentioned in Example 5.3.4, would be preferred.

Exact tests can also be developed for hypotheses that involve two independent binomial samples. Discussion of these methods is, however, deferred to the introduction of contingency tables in Section 8.3.3.

5.5 Multiple inferences*

An important consideration in data analytics is that of *multiple inferences* or *multiple comparisons*, that is, when two or more inferences are performed on a single set of data. The multiplicity problem is common when there are many population parameters under study and

confidence intervals or hypothesis tests are desired on each. For instance, suppose a hypothesis test is conducted on each of p parameters from the same set of data. Without correction for the multiplicity of tests being undertaken, there will be p opportunities to make a false positive error. Thus the *experimentwise* or *familywise false positive error rate* (FWER) will be much larger than the *pointwise* significance level, α. (Analogous concerns regarding the confidence coefficient, $1-\alpha$, arise when constructing multiple pointwise confidence intervals.) To account for this error inflation, some adjustment is required.

The simplest way to adjust for multiple inferences is to build the multiplicity into the estimation or testing scenario; for example, construct joint hypothesis tests or joint confidence regions that simultaneously contain all p parameters of interest. All of the inferential methods presented in the preceding sections can be extended in this manner. For instance, a simultaneous, p-dimensional, $1-\alpha$ Wald confidence region for a vector of unknown parameters $\boldsymbol{\vartheta}$ is the ellipsoid defined by the matrix inequality

$$(\hat{\boldsymbol{\vartheta}} - \boldsymbol{\vartheta})^{\mathrm{T}} \mathbf{F}(\hat{\boldsymbol{\vartheta}})(\hat{\boldsymbol{\vartheta}} - \boldsymbol{\vartheta}) \leq \chi^2_\alpha(p), \tag{5.49}$$

where $\mathbf{F}(\hat{\boldsymbol{\vartheta}})$ is the Fisher information matrix evaluated at the MLE $\hat{\boldsymbol{\vartheta}}$.

Joint calculations are not always possible, however, or they may be less useful in practice than the simpler statements a pointwise construction can provide. For instance, a set of p intervals on the individual parameters, θ_j ($j = 1, \ldots, p$), of $\boldsymbol{\vartheta}$ is often easier to interpret than a p-dimensional confidence ellipsoid. In such cases, it is still possible to adjust for the multiplicity, using one of a variety of probability inequalities (Galambos and Simonelli 1996). The next section describes perhaps the most popular of these, the *Bonferroni inequality*.

5.5.1 Bonferroni multiplicity adjustment

Bonferroni's inequality (Bonferroni 1936) is in essence a statement relating the joint probability of a set of multiple events to the individual event probabilities. Let $P[\mathcal{E}_1\mathcal{E}_2 \cdots \mathcal{E}_p]$ denote the probability that the p events $\mathcal{E}_j$ ($j = 1, \ldots, p$) occur simultaneously. Then, if $\mathcal{E}_j^c$ denotes the complementary event that $\mathcal{E}_j$ did *not* occur, Bonferroni's inequality may be written as

$$P[\mathcal{E}_1\mathcal{E}_2 \cdots \mathcal{E}_p] \geq 1 - \sum_{j=1}^{p} P[\mathcal{E}_j^c]. \tag{5.50}$$

Translated into a multiple inference statement, Bonferroni's inequality says the familywise probability that any set of events occurs can be bounded below by 1 minus the sum of the probabilities that each individual event did not occur. Clearly, in the special case where each $P[\mathcal{E}_j^c]$ equals the same constant, say, $P[\mathcal{E}_j^c] = \gamma$ for all $j = 1, \ldots, p$, (5.50) simplifies to

$$P[\mathcal{E}_1\mathcal{E}_2 \cdots \mathcal{E}_p] \geq 1 - p\gamma.$$

In the context of a set of p confidence intervals, $\mathcal{E}_j$ corresponds to the event that θ_j is covered correctly by the jth interval ($j = 1, \ldots, p$). So, if the goal is to set the simultaneous confidence level among p individual confidence intervals no smaller than $1-\alpha$, using $\gamma = \alpha/p$ produces the desired result. That is, if each complementary event – here, failure to cover θ_j – occurs with probability α/p, the simultaneous collection of coverage events occurs with probability at least $1-\alpha$, from (5.50). This is called *minimal simultaneous coverage*.

Example 5.5.1 Simultaneous confidence interval on μ and σ^2 (Example 5.3.1, continued). Suppose a random sample is taken from a normal distribution with unknown mean μ and unknown variance σ^2: $X_i \sim$ i.i.d. N(μ, σ^2), $i = 1, \ldots, n$. Equations (5.17) and (5.21), respectively, define pointwise $1 - \alpha$ confidence limits for each parameter. To construct a *joint* confidence region for both μ and σ^2 with minimal, simultaneous $1 - \alpha$ confidence, one can appeal to the Bonferroni inequality.

To do so, recognize that there are $p = 2$ parameters here for which a simultaneous inference is desired. Thus, if we set the confidence level for the μ interval from (5.17) to $1 - \frac{\alpha}{2}$ and also set the confidence level for the σ^2 interval from (5.21) to $1 - \frac{\alpha}{2}$, the respective probabilities of noncoverage, that is, the $\mathcal{E}_j^c$ events in (5.50) – are each $\gamma = \alpha/2$. Thus from Bonferroni's inequality, the joint coverage probability will meet or exceed $1 - (2)(\frac{\alpha}{2}) = 1 - \alpha$. This translates to the joint, minimal $1-\alpha$ statement

$$P\left[\overline{X} - t_{\alpha/4}(n-1)\frac{S}{\sqrt{n}} < \mu < \overline{X} + t_{\alpha/4}(n-1)\frac{S}{\sqrt{n}}\right.$$
$$\left. \text{and } \frac{(n-1)S^2}{\chi^2_{\alpha/4}(n-1)} < \sigma^2 < \frac{(n-1)S^2}{\chi^2_{1-\frac{\alpha}{4}}(n-1)}\right] \geq 1 - \alpha. \tag{5.51}$$

To illustrate, recall the myocardial infarction data in Table 4.2. There, the sample mean was $\overline{X} = 62.8137$ (years) and the sample variance was $\sigma^2 = 69.5908$. A minimal 95% set of joint confidence limits for both μ and σ^2 here requires the critical points $t_{0.05/4}(125) = t_{0.0125}(125) = 2.2687$, $\chi^2_{0.05/4}(125) = \chi^2_{0.0125}(125) = 163.0876$, and $\chi^2_{1-\frac{0.05}{4}}(125) = \chi^2_{0.9875}(125) = 92.27$. Applied in (5.51), these lead to the joint limits

$$62.8137 - (2.2687)\frac{8.3421}{\sqrt{126}} < \mu < 62.8137 + (2.2687)\frac{8.3421}{\sqrt{126}}$$

and

$$\frac{(125)(69.5908)}{163.0876} < \sigma^2 < \frac{(125)(69.5908)}{92.2700}$$

or simply $61.1277 < \mu < 64.4997$ (years) and $53.3385 < \sigma^2 < 94.2760$. Notice that the interval for μ has widened versus that in Example 5.3.1: by extending joint coverage to both μ and σ^2, the Bonferroni correction here has made each individual inference somewhat less-precise. This is a common trade-off with multiplicity adjustments; a corollary of what Clarke et al. (2009, Exercise 1.2) call the 'no-free-lunch theorem.' (There is even a website: http://www.no-free-lunch.org/ !)

The following are some caveats:

(a) The solution in (5.51) is not unique, in that allocation of noncoverage can be modified to suit the user's needs. If, for example, greater interest existed in covering μ than σ^2, the noncoverage allocation could be varied so that $\alpha/3$ was allocated to the μ interval – that is, $\alpha/6$ in each tail – and $2\alpha/3$ was allocated to the σ^2 interval. The Bonferroni lower bound is then (still) $1 - \left(\frac{2\alpha}{3} + \frac{\alpha}{3}\right) = 1 - \alpha$.

(b) As noted in Section 5.3.2, the equal-tail interval (5.21) for σ^2 is known to be suboptimal. For greater precision in practice, an interval such as that by Tate and Klett (1959) could be applied instead within the Bonferroni adjustment.

(c) Many options are available to the analyst when manipulating pointwise intervals for μ and σ^2 into a joint confidence region, and the construction here is only a first step. See, for example, Casella and Berger (2002, Exercise 9.14) for some other alternatives. □

Similar considerations apply in the *multiple testing* scenario. Suppose $m > 1$ null hypotheses H_{oj} ($j = 1, \ldots, m$) are under study. Define each $\mathcal{E}_j$ as the event that H_{oj} is (correctly) not rejected when it is true. Then $\mathcal{E}_j^c$ represents the event that H_{oj} is rejected when it is true, a false positive (Type I) error on the jth test. $P[\mathcal{E}_j^c]$ becomes the associated false positive error rate. If the goal is to set the simultaneous FWER among the m individual tests no larger than α, using the corrected pointwise rate $\gamma = \alpha/m$ in Bonferroni's inequality produces the desired result. In effect, one replaces α with α/m in the jth critical point used to test H_{oj}. (Or, one multiplies each pointwise P-value by m. A useful **R** function in this regard is `p.adjust()`.)

In general, for both confidence intervals and hypothesis tests, the multiple events $\mathcal{E}_j$ will be correlated, because they are derived from the same set of data. This does not affect implementation of the Bonferroni inequality, endowing it an omnibus quality. Coupled with its simplicity, this has earned the correction wide popularity for controlling multiplicity with simultaneous confidence levels or FWERs.

The 'no-free-lunch theorem' still applies, however. Convergence to the Bonferroni lower bound in (5.50) can be poor, depending on the underlying correlation structure among the $\mathcal{E}_j$s, and thus the actual simultaneous confidence level or FWER can become very conservative (i.e., much larger than $1-\alpha$ or much smaller than α, respectively). One cannot determine the stringency of the resulting inferences without studying the actual probability structure of the $\mathcal{E}_j$s, which must be done on a case-by-case basis. Further, even if the bound is fairly tight, as the number of comparisons grows the eventual inferences become more draconian: confidence limits widen and critical points grow large compared to the single-comparison, pointwise case (see Exercise 5.35). The price of controlling for multiplicity is generally lower precision or lower power in the eventual intervals or tests, respectively. With small numbers of comparisons, this trade-off is considered worthwhile. In many large-scale knowledge-discovery investigations, however, hundreds or even thousands of hypothesis tests are performed on a single set of data. Pushing m this high exceeds the practicality and the design of the Bonferroni adjustment, requiring a different metric for multiplicity correction. One possibility is discussed in the next section.

Beyond Bonferroni's inequality, there is a much larger theory and practice of multiple comparisons and other simultaneous inferences. A full description exceeds the scope of this section; interested readers should refer to dedicated texts on the topic such as Liu (2010), Hsu (1996), or Hochberg and Tamhane (1987). In particular, for executing multiple comparisons in **R**, see Bretz et al. (2011), including their description of the external *multcomp* package, or Dudoit and van der Laan (2008) and their discussion of the external *multtest* package.

5.5.2 False discovery rate

As mentioned previously, modern knowledge discovery often takes an exploratory data analysis (EDA) approach where formal, confirmatory inferences on population parameters

become a lesser goal, replaced by issues of exploratory *feature assessment* (Hastie et al. 2009, Section 18.7). In effect, the hypothesis testing paradigm is coopted into a broad search-and-discover process, prompting the 'discovery' terminology popular in this setting. For example, in genome-wide association studies (GWAS) or some large-scale astronomy experiments, many different hypotheses may be tested in order to identify veiled threads or patterns within a large data set. Here, 'many' is not just 10 or even 100; it can be in the many thousands. This sort of large-scale/high-dimensional multiple testing requires a different multiplicity adjustment criterion than the FWER in Section 5.5.1.

Towards this end, Benjamini and Hochberg (1995) modified the traditional false positive FWER approach and asked, can one reduce the conservatism of Bonferroni-type corrections in order to highlight new 'discoveries' (rejected, false, null hypotheses), if one is willing to allow for more false discoveries (rejected, true, null hypotheses)? Their answer became known as the problem of controlling the *false discovery rate* (FDR). Core constituents of this approach share components with those in formal hypothesis testing, but they are deployed in a different manner.

To establish the notation: suppose a large number, m, of hypotheses $\mathrm{H}_{\mathrm{o}j}$ ($j = 1, \ldots, m$) are under study. Of these, let m_{o} (an unknown quantity) be true, representing no new discoveries. Given a decision rule for rejecting each $\mathrm{H}_{\mathrm{o}j}$, calculate the associated P-value, P_j. Let R_m be the number of null hypotheses that are actually rejected. Of these, let V_m be the number of rejected null hypotheses for which $\mathrm{H}_{\mathrm{o}j}$ was true, that is, the number of false discoveries. (As they depend on the data, both R_m and V_m are random variables. Obviously, however, V_m is not observed in practice.)

The ratio V_m/R_m is the false discovery proportion, described by Seeger (1968) following on work by Eklund and Seeger (1965). Benjamini and Hochberg (1995) took this notion and defined their false discovery rate based on the expected proportion

$$\mathrm{FDR} = E\left[\frac{V_m}{R_m}\middle| R_m > 0\right] P[R_m > 0],$$

where the expectation is taken with respect to the joint distribution of the data. (Some authors write FDR in the simpler form $E[V_m/R_m]$, although they must then explicitly define $V_m/R_m = 0$ when $R_m = 0$.) By contrast, the FWER is simply $P[V_m \geq 1]$, and it can be shown that FDR $\leq$ FWER (Clarke et al. 2009, Section 11.4.1).

The Benjamini and Hochberg (henceforth, BH) strategy replaces the more-stringent FWER with the FDR. The analyst begins by specifying a maximum FDR, denoted by convention as α. (The notation can be confusing: α here is *not* the overall false positive rate; rather, it is the maximum proportion of false discoveries one is on average willing to accept. In practice, values for it can reach up or past 15%, possibly higher, depending on the goals of the exploratory study.) Next, order the m P-values into $P_{(1)} \leq P_{(1)} \leq \cdots \leq P_{(m)}$ and find

$$K = \min\left\{ j \in \{1, \ldots, m\} \middle| P_{(j)} \leq \frac{j\alpha}{m} \right\}. \tag{5.52}$$

Assuming the index K exists for a given data set, reject (or 'discover') any hypothesis whose $P_{(j)} \leq P_{(K)}$. Benjamini and Hochberg showed that if the P_js are independent, the P-value threshold based on (5.52) controls the FDR in that

$$\mathrm{FDR} \leq \frac{m_{\mathrm{o}}}{m}\alpha \leq \alpha.$$

Because m_o is unknown, so is this upper bound; however, methods exist to estimate m_o and/or the realized FDR; see, for example, Hastie et al. (2009, Alg. 18.3) or Storey (2011).

It is worth noting that the concept of ordering P-values to control multiple error rates is a time-tested maneuver in simultaneous inference. For example, an approach similar to (5.52) was also studied by Seeger (1968) and Simes (1986) to improve the Bonferroni inequality when testing the global null hypothesis that *all* the H_{oj}s hold. Indeed, when $m_o = m$, the method coincides with (5.52).

Example 5.5.2 FDR analysis with gene expression data. In the analysis of genetic microarray data, a set of $m > 1$ genes is often studied to determine if there is some change in the genes between two distinguishing conditions, for example, two different strains of a microorganism, or cancer patients versus normal controls, and so on (Nguyen et al. 2002). A two-sample t-test (or other appropriate statistical test) is conducted to compare the jth gene's expression between the two conditions and the corresponding P-value, P_j, is recorded. The number of genes under study often is very large, and the question of identifying differential expressions among them is usually more exploratory than confirmatory. As a result, control of the FDR is popular here.

For example, Table 5.4 presents a selection of $m = 1000$ ordered $P_{(j)}$-values from a two-group gene comparison described in Broberg (2003, Table 2), where roughly 10% of the genes exhibited differential expression. (The full set of P-values is available at http://www.wiley.com/go/piegorsch/data_analytics.) The table also lists the corresponding BH thresholds from (5.52) at $\alpha = 0.15$, and a marker indicating which of the ordered P-values fall below the threshold.

In the table, the genes with the 12 smallest P-values lie below the BH threshold. These would be considered indicators of a potentially 'significant' difference between the two conditions, and worth further, targeted study. □

A variety of alternative error rates have evolved from (or in some cases, preceded) the FDR, each allowing for different levels of control to suit different application needs; see,

Table 5.4 Selection of ordered P-values, $P_{(j)}$, from gene expression analysis of $m = 1000$ genes in Example 5.5.2, with BH thresholds from (5.52).

Ordered $P_{(j)}$-value	Threshold at $\alpha = 0.15$	Above/below threshold
0.00004	0.00015	✓
0.00010	0.00030	✓
0.00012	0.00045	✓
⋮	⋮	⋮
0.00113	0.00150	✓
0.00135	0.00165	✓
0.00171	0.00180	✓
0.00221	0.00195	✗
0.00258	0.00210	✗
⋮	⋮	⋮
0.99874	0.14985	✗
0.99948	0.15000	✗

Markers indicate genes whose $P_{(j)}$-values fall below (✓) or above (✗) the threshold.

for example, Clarke et al. (2009, Section 11.4.1). Among these, a closely related rate is the *positive false discovery rate* or pFDR (Storey 2002, 2003):

$$\text{pFDR} = E\left[\frac{V_m}{R_m}\middle| R_m > 0\right].$$

The pFDR measures the proportion of false discoveries that occur on average, presuming that rejections ('discoveries') do in fact occur. It can be shown that FDR $\leq$ pFDR $\leq$ FWER, and, indeed, as $m \to \infty$, $E[\text{FDR}] \approx E[\text{pFDR}] \approx E[V_m]/E[R_m]$. (The latter ratio is known as the *marginal false discovery rate* or mFDR.) Thus for testing problems when the number of hypotheses is extremely large, these discovery rates will be similar.

Since its introduction, numerous extensions of the basic BH algorithm have appeared. Typically, these either modify the threshold below which each $P_{(j)}$ marks a discovery, or first estimate m_0 and then adaptively modify the selection rule using this estimator. Hwang et al. (2011) reviewed a number of these and gave guidance on their operating characteristics in different settings. In fact, applications of the false discovery rate and its sequelae have become widespread in large-scale data analytics. Instructive reviews are available in Farcomeni (2008), Goeman and Solari (2011), and Storey (2011).

Exercises

5.1 Suppose a single observation is taken from a binomial distribution: $Y \sim \text{Bin}(n, \pi)$. Find the Fisher information number $\mathcal{I}(\pi)$ from (5.3) for this single observation.

5.2 Take a random sample of normal observations, $X_i \sim$ i.i.d. $\text{N}(\mu, \sigma^2)$, $i = 1, \ldots, n$. Find expressions for the MOM estimators of μ and σ^2. How do these compare to the MLEs from Example 5.2.5?

5.3 In Example 5.2.4, use differential calculus to show that $\overline{X}$ is the LS estimator for μ. Be sure to verify that the estimator does indeed minimize the LS objective function.

5.4 Verify that the MLEs for the unknown normal mean and variance in Example 5.2.5 do in fact maximize the likelihood (or the log-likelihood) function.

5.5 Take a random sample of Poisson observations, $X_i \sim$ i.i.d. $\text{Poisson}(\lambda)$, $i = 1, \ldots, n$. Find the MLE, $\hat{\lambda}$, for λ. Be sure to verify that it is a true maximum. Given $\hat{\lambda}$, what is the MLE of $\psi = \log(\lambda)$? Why?

5.6 Let $X_i \sim$ i.i.d. $f_X(x|\mu, \sigma^2)$ $i = 1, \ldots, n$. Assume the underlying p.m.f. or p.d.f. has a finite population mean $E[X_i] = \mu$ and a finite population variance σ^2. Show that the expected value of the sample variance S^2 from (3.5) equals σ^2, so that S^2 is an unbiased estimator of σ^2 (Casella and Berger 2002, Section 5.2). (*Hint*: Use the computing form for S^2 from (3.6), and recall that for any random variable W with a finite mean and a finite variance, $E(W^2) = \text{Var}(W) + E^2(W)$.)

5.7 Show that the Studentized ratio in (5.16) satisfies the relationship in (2.41) that defines a t-distributed random variable. (*Hint*: Recall the χ^2 relationship for S^2 introduced in Example 5.2.6.)

5.8 Consider again the circulatory-disease mortality data from Table 3.4. (The complete data are available at http://www.wiley.com/go/piegorsch/data_analytics.) In

Example 4.1.3, it was established that the standardized mortality rates (SMRs) exhibited a clear right skew, for which a (natural) logarithmic transform can bring the variation closer to normal, as in Exercise 4.7b. Apply the log transform to the SMRs and calculate a 90% confidence interval on the mean for these transformed data using (5.17). How would you translate these confidence limits back into the original SMR scale? Is there any (theoretical) aspect of this reverse transform that has pragmatic value?

5.9 Return to the husbands' heights data in Exercise 3.2.

(a) Plot a histogram (with an overlaid kernel density estimate and accompanying rug plot) and a normal quantile plot. Do the data appear symmetric and normal?

(b) If the data appear roughly normal, calculate a 99% confidence interval for the mean height in this population using (5.17).

5.10 Verify that by beginning with (5.20), the confidence statement for σ^2 in (5.21) results. How would you modify this to produce a $100(1-\alpha)\%$ one-sided, upper, confidence limit for σ^2?

5.11 Return to the myocardial infarction data in Table 4.2, as seen in Example 5.3.1. Suppose interest also exists in calculating a 95% confidence interval on the variance σ^2.

(a) Calculate an equal-tail area interval as per (5.21).

(b) Assume that there is twice as much interest in the lower limit as there is in the upper limit, so that tail areas should be allocated asymmetrically. Make this assignment, determine the form of the resulting confidence interval, and calculate it for these data.

(c) Refer to Tate and Klett (1959) and explore how to calculate their 'minimum length' confidence interval for these data.

5.12 Repeat the calculations in Exercise 5.11 for the following data sets. Set your confidence level to $1-\alpha = 0.90$.

(a) The hazard vulnerability data from Exercise 3.3.

(b) The log-transformed standardized mortality rates from Exercise 5.8.

5.13 Return to the lung function data in Example 5.3.2. Validate the normal sampling assumptions made on the data as follows.

(a) Plot histograms for both samples. (Be sure to indicate the binning algorithm you employ.) Overlay kernel density estimates and rug plots. Do the histograms/density estimates appear symmetric and bell shaped?

(b) Graph side-by-side boxplots. Do the boxes corroborate indications from the histograms?

(c) Graph normal quantile plots for both samples. Do the plots appear linear?

(d) If you are familiar with the Shapiro–Wilk test (Royston 1982; Shapiro and Wilk 1965) for normality, conduct the test separately for both samples at $\alpha = 0.05$. (In **R**, use the `shapiro.test()` function.)

5.14 Toraason et al. (2006) reported data on DNA damage in workers' peripheral blood lymphocytes after exposure to the solvent 1-bromopropane ($CH_3CH_2CH_2Br$) at two independent workplace facilities, labeled as 'A' and 'B.' The damage was measured by a laboratory assay, where higher values indicate greater damage to the DNA. The data among all workers reporting assay results are as follows:

Facility	Assay scores
A	1739, 2408, 2009, 4023, 2760, 4068, 2218, 2603, 3773, 2598, 2911, 3162, 2570, 3794, 2758, 3348, 2129, 3453, 2821, 2536, 3281, 2766, 2619, 2835, 2913, 4892, 3156, 2675, 3802, 3928, 2408, 3190, 3341, 2956, 3557, 5007, 3693, 3051, 3093, 3615
B	2367, 3113, 2500, 3085, 2557, 2182, 2389, 2671, 2816, 3054, 3486, 4175, 2987, 2909, 2909, 2631, 3031, 2825, 2562, 2374

Assay data such as these often skew to the right, for which a (natural) logarithmic transform brings variation closer to normal. Thus let $X_1 = \log\{\text{Facility A scores}\}$, independent of $X_2 = \log\{\text{Facility B scores}\}$.

(a) Plot histograms for both samples of log-transformed data. (Be sure to indicate the binning algorithm you employ.) Overlay a kernel density estimate and include a rug plot. Do the histograms/density estimates appear symmetric and bell shaped?

(b) Graph side-by-side boxplots. Do the boxes corroborate indications from the histograms?

(c) Graph normal quantile plots for both samples. Do the plots appear linear?

(d) If you are familiar with the Shapiro–Wilk test (Royston 1982; Shapiro and Wilk 1965) for normality, conduct the test separately for both samples at $\alpha = 0.05$. (In **R**, use the `shapiro.test()` function.)

(e) Assume that the normal sampling assumption for both variables is tenable and find a 90% confidence interval for the difference in mean log-DNA damage scores. Make no assumptions about the variances σ_1^2 and σ_2^2.

5.15 Manipulate the pivot in (5.22) to produce an approximate probability statement that contains the t-based confidence limits for $\Delta = \mu_1 - \mu_2$ in (5.24).

5.16 The Australian postal service (Australia Post) performs screening inspections to identify potentially contaminated items and quarantine them. In a study of parcels entering the system over a 12-month period $n = 2\,862\,399$ parcels were inspected. Of these, $Y = 7919$ were intercepted as having high biosecurity risk (Decrouez and Robinson 2012). Assume $Y \sim \text{Bin}(n, \pi)$. Calculate the following confidence intervals for the binomial parameter π. Set your confidence level to $1-\alpha = 0.99$.

(a) The Agresti–Coull confidence interval.

(b) The Wilson continuity-corrected confidence interval.

(c) The LR confidence interval.

5.17 In an ecotoxicological experiment with the potential carcinogen Aflatoxicol, rainbow trout (*Oncorhynchus mykiss*) embryos were exposed to 0.250 ppm of the compound and later examined for development of liver tumors. Out of $n = 338$ exposed trout, $Y = 286$ exhibited the tumor (Roy and Kaiser 2013). Assume $Y \sim \text{Bin}(n, \pi)$, and calculate a 95% Agresti–Coull confidence interval for π.

5.18 *Altmetric data* are used in scientific publishing to determine patterns of scholarly and social interest in published articles. The altmetric scores – say, number of Twitter tweets about an article after its appearance – are seen as potential impact metrics. For instance, Thelwall et al. (2013) reported data on how often a published article receives a higher altmetric score than adjacent articles appearing immediately before and immediately after, using a database of published articles in the biomedical sciences. Those that did were viewed as 'successes' in attracting greater altmetric attention than colocated articles in the same journal. The result is a proportion: $Y = \{\text{number of successes}\}$ over $n = \{\text{number of tested articles}\}$. For the following altmetric scores from this database, assume $Y \sim \text{Bin}(n, \pi)$ and find a 90% Agresti–Coull confidence interval for π.

(a) Using Twitter tweets as the altmetric score, $Y = 24\,315$ and $n = 42\,891$.

(b) Using Facebook posts as the altmetric score, $Y = 3229$ and $n = 5612$.

(c) Using Google+ posts as the altmetric score, $Y = 426$ and $n = 804$.

5.19 Return to the customer purchasing data in Example 5.3.4, where $Y = 506$ customers out of $n = 1024$ made a purchase during a sale. Calculate the following confidence intervals for the binomial parameter π and compare them to the Agresti–Coull interval presented in the example. As there, set your confidence level to $1-\alpha = 0.95$.

(a) The LR confidence interval.

(b) The Wilson continuity-corrected confidence interval.

5.20 From a study of respiratory afflictions similar to that in Example 5.3.2, Taussig et al. (2003) described data on the numbers of lower respiratory tract illnesses observed in the first 3 years of life among children with asthma. The counts, presented as a frequency table, are

Number of illnesses:	0	1	2	3	4	5	6	7	≥8
Frequency:	267	175	79	47	17	13	2	1	0

Thus of the $n = 601$ children studied, 267 reported $Y_i = 0$ illnesses, 175 reported $Y_i = 1$ illness, 79 reported $Y_i = 2$ illnesses, and so on.

(a) Construct a bar chart of the observed frequencies. What pattern do you see?

(b) Assume $Y_i \sim$ i.i.d. Poisson(λ), $i = 1, \ldots, 601$, and calculate a 95% confidence interval for the mean number of illnesses, λ, in this population of children, via (5.36).

5.21 Continuing with the ratio of parameters, θ_1/θ_2, from Example 5.3.5, notice that $\theta_1/\theta_2 = \exp\{\log(\theta_1/\theta_2)\}$. To derive an alternative confidence interval on the ratio, consider the function $h(\theta_1, \theta_2) = \log(\theta_1/\theta_2) = \log(\theta_1) - \log(\theta_2)$ and use the delta

method in Section 5.3.5 to approximate $\text{Var}[h(\theta_1, \theta_2)] = \text{Var}[\log(\theta_1/\theta_2)]$. With this, develop a $100(1-\alpha)\%$ Wald confidence interval for $\log(\theta_1/\theta_2)$. Use the result to construct another form of $100(1-\alpha)\%$ Wald confidence interval for θ_1/θ_2.

5.22 If, for a given set of data, you reject a null hypothesis at the $\alpha = 0.05$ significance level, would you also reject at the $\alpha = 0.10$ significance level (with the same set of data)? Why or why not?

5.23 Return to the hazard vulnerability data from Exercise 3.3 and let μ be the population mean of the listed index values. Assume $X_i \sim \text{i.i.d. } \text{N}(\mu, \sigma^2)$, $i = 1, \ldots, n$. Test H_o: $\mu = 1$ versus H_a: $\mu \neq 1$ at $\alpha = 0.05$.

5.24 Continuing with the log-transformed circulatory-disease mortality data in Exercise 5.8, assume the transformed data represent a random sample from $\text{N}(\mu, \sigma^2)$. Test H_o: $\mu = 2.3$ versus H_a: $\mu \neq 2.3$ at $\alpha = 0.01$.

5.25 Return to the two-sample data from Exercise 5.14 on DNA damage in workers' peripheral blood lymphocytes. Conduct a test for whether any difference exists in mean (log) scores between the two facilities. Operate at the 10% significance level. Comment on how your result compares to the inferences from that earlier exercise.

5.26 Recall the husbands' and wives' heights data from Exercise 3.12.

(a) As with the husbands' heights in Exercise 5.9, plot a histogram (overlay a kernel density estimate) and normal quantile plot for the wives' heights. Do the data appear roughly symmetric and normal?

(b) Assess whether heights of married couples are equal in this population. (The data are paired, so conduct a paired t-test.) The null hypothesis is obviously H_o: $\mu_D = 0$, where $\mu_D = \mu_{\text{male}} - \mu_{\text{female}}$. What choice would you make for the alternative hypothesis H_a? (Why?) Employ it here. Operate at $\alpha = 0.01$.

(c) Take a close look at the wives' heights. Do you see anything odd with the data?

5.27 Return to the Lung Function data in Examples 5.3.2 and 5.4.2. Recall that no assumptions were made there concerning equality of the variances.

(a) Conduct a test for homogeneity of variances on the data: that is, test H_o: $\sigma_1^2 = \sigma_2^2$ versus H_a: $\sigma_1^2 \neq \sigma_2^2$ at $\alpha = 0.01$.

(b) If you conclude that the variances are not significantly different, return to Example 5.3.2 and calculate a 90% confidence interval for the difference in means, $\mu_1 - \mu_2$, using (5.27). Compare the result to that seen in the exercise.

(c) If you conclude that the variances are not significantly different, return to Example 5.4.2 and perform a test of equality between the means using (5.44). As there, operate at $\alpha = 0.10$. Compare the result to that seen in the example.

5.28 Return to the two-sample DNA damage data from Exercises 5.14 and 5.25. Recall that no assumptions were made there concerning equality of the variances.

(a) Conduct a test for homogeneity of variances on the log-transformed data: that is, test H_o: $\sigma_1^2 = \sigma_2^2$ versus H_a: $\sigma_1^2 \neq \sigma_2^2$ at $\alpha = 0.01$.

(b) If you conclude that the variances are not significantly different, return to Exercise 5.14 and calculate a 90% confidence interval for the mean (log) difference, Δ, using (5.27). Compare the result to that seen in the example.

(c) If you conclude that the variances are not significantly different, return to Exercise 5.25 and perform a test of equality between the means using (5.44). As there, operate at a 10% significance level. Compare the result to that seen in the exercise.

5.29 Two-sample hypothesis tests are often employed in online marketing, via what is referred to as *A–B testing*. The approach can be applied in a number of ways. Suppose that a web site designer tests whether a new site layout (variant 'B') attracts more customers, compared to the current site (the 'control' variant 'A'). A volunteer is randomly shown either variant A or variant B and the amount of time spent (in seconds) viewing the site is recorded. The same volunteer is then shown the other variant and the amount of time spent on that site is recorded. This is repeated for $n = 500$ volunteers. The data are matched pairs (X_{Aj}, X_{Bj}), $j = 1, \ldots, 500$, producing differences $D_j = X_{Aj} - X_{Bj}$ for a paired t-test, as in (5.45). Suppose this experiment produced a mean difference of $\overline{D} = 12.6306 - 17.8446 = -5.2140$ s, with sample standard deviation $S_D = 4.2749$ s. Test whether the new (variant B) web site held volunteers' attentions significantly longer than the control (variant A). Operate at $\alpha = 0.01$.

5.30 Let $Y \sim \text{Bin}(n, \pi)$. Construct an LR test for $\text{H}_\text{o}: \pi = \pi_\text{o}$ versus $\text{H}_\text{a}: \pi \neq \pi_\text{o}$. Use this to test $\pi = \frac{1}{2}$ in Example 5.4.4 and compare the result to the conclusions drawn there.

5.31 Let $Y \sim \text{Bin}(n, \pi)$. Suppose $n > 40$. How would you modify the recommended $1-\alpha$ Agresti–Coull confidence interval for π into a level-α test of $\text{H}_\text{o}: \pi = \pi_\text{o}$ versus $\text{H}_\text{a}: \pi \neq \pi_\text{o}$?

5.32 Recall the postal quarantine study described in Exercise 5.16, where $n = 2\,862\,399$ parcels were inspected in a 12-month period, of which $Y = 7919$ had high biosecurity risk. Conduct a binomial exact test to determine if the true probability of intercepting a high-risk parcel is below 1%. Operate at $\alpha = 0.05$. What do you determine?

5.33 Follow on the suggestion in Example 5.5.1 about unequal Bonferroni allocations and modify the adjustment with the myocardial infarction data from Table 4.2. Allocate $\alpha/3$ noncoverage probability to the μ interval and $2\alpha/3$ noncoverage probability to the σ^2 interval. For simplicity, continue to use the equal-tail interval from (5.21) for σ^2. How do the results compare with those from the example?

5.34 The ecotoxicological data in Exercise 5.17 were part of a larger experiment on rainbow trout carcinogenesis. The full study employed five different exposures (in ppm) of aflatoxicol. At each exposure, x_j, the number of trout with the tumor, Y_j, was recorded, out of n_j fish exposed ($j = 1, \ldots, 5$). From Roy and Kaiser (2013), the data are

Dose, x_j	0.010	0.025	0.050	0.100	0.250
Tumor-bearing trout, Y_j	25	132	226	281	286
Number exposed, n_j	347	346	353	355	338

Assume $Y_j \sim$ indep. Bin(n, π_j), $j = 1, \ldots, 5$, and calculate an Agresti–Coull confidence interval for each π_j. As the data come from the same experiment, adjust each pointwise confidence interval for multiplicity via a Bonferroni correction so that the overall simultaneous confidence level is no smaller than $1-\alpha = 0.95$. What patterns emerge?

5.35 Consider testing a multiple series of hypotheses H_{0j} versus H_{aj}, $j = 1, \ldots, m$. Assume that for a single comparison, rejection occurs when a test statistic T_j exceeds the pointwise critical point $t_{\alpha/2}(\nu)$ from a $t(\nu)$ reference distribution. Apply a Bonferroni correction to the critical point and examine its progression as m changes for the following cases. Set $\alpha = 0.05$. (Notice that, as described here, rejection becomes more demanding as the corrected critical point grows.)

(a) Set $\nu = 4$ and progress $m = 1, 5, 10, 50, 100, 500$.

(b) Set $\nu = 12$ and progress $m = 1, 5, 10, 50, 100, 500$.

(c) Set $\nu = 25$ and progress $m = 1, 5, 10, 50, 100, 500$.

5.36 For the gene-expression microarray study in Example 5.5.2, data were also generated from $m = 1000$ genes where roughly 5% of the genes exhibited differential expression. The corresponding P-values are available online at http://www.wiley.com/go/piegorsch/data_analytics; a sample follows:

$$\text{Ordered } P_{(j)}\text{:} \quad 0.00007 \quad 0.00009 \quad 0.00039 \quad \cdots \quad 0.99807 \quad 0.99834$$

Perform a multiple testing examination of these P-values by applying the Benjamini and Hochberg (1995) FDR correction, at $\alpha = 0.15$. Which genes' P-values indicate a potential discovery?

Part II

STATISTICAL LEARNING AND DATA ANALYTICS

6

Techniques for supervised learning: simple linear regression

The specialized statistical learning techniques required for large-scale data analytics separate into one of two basic genera: supervised or unsupervised. Supervised learning techniques will be described in this and the following three chapters; the current chapter focuses on one of the simplest forms of supervised learning, linear regression analysis. Unsupervised methods are described in Chapters 10 and 11.

6.1 What is "supervised learning?"

The term "supervised learning" is rooted in statistical learning/machine learning parlance, where it describes the analysis of data via a focused structure – sometimes called "learning with a teacher" (Kantardzic 2003, Section 4.3). The observations may belong to a set of training data, used to identify a model that relates the inputs, usually a vector or matrix of predictor/feature variables $\mathbf{X}$, to an output response variable Y. The "teacher" is an optimality or fitness criterion – such as least squares minimization or likelihood maximization – that guides selection of the final model. Learning occurs when the input predictors are processed through the model to discover unknown dependencies between $\mathbf{X}$ and Y.

Two basic species of supervised statistical learning are *regression analysis*, described in this and the next two chapters, and *classification analysis*, described in Chapter 9. Both are predictive in nature: they combine input variables with the eventual model to forecast or classify future realizations of the outcome variable. The next section begins with the simplest form of predictive model, *simple linear regression* (SLR), where a single quantitative variable, x, is used to describe a single outcome variable, Y.

Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery,
First Edition. Walter W. Piegorsch.

www.wiley.com/go/piegorsch/data_analytics

6.2 Simple linear regression

6.2.1 The simple linear model

The SLR paradigm involves a single predictor variable, x, also called an *input variable* or *feature variable*, used to describe an output or response variable, Y, via a simple linear model. Formally, assume data are collected in matched pairs $(x_i, Y_i), i = 1, \ldots, n$, where the response variable is modeled as

$$Y_i \sim \text{indep. } \mathrm{N}\big(\mu(x_i), \sigma^2\big) \ , \tag{6.1}$$

and where the predictor x_i is fixed and nonstochastic. The simple linear model specifies a linear relationship between $E[Y_i] = \mu(x_i)$ and x_i:

$$\mu(x_i) = \beta_0 + \beta_1 x_i \ , \tag{6.2}$$

where β_0 and β_1 are *regression coefficients* representing the Y-intercept and slope, respectively, of the mean response $\mu(x_i)$.

The regression coefficients are assumed unknown and must be estimated from the data. Most analysts employ the method of least squares (LS; see Section 5.2.3): construct the objective quantity $\mathcal{D} = \sum_{i=1}^{n} \{Y_i - (\beta_0 + \beta_1 x_i)\}^2$ and minimize $\mathcal{D}$ with respect to both β_0 and β_1. This produces a system of two equations with two unknowns:

$$\begin{aligned} \sum_{i=1}^{n} Y_i &= n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i Y_i &= \beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2 \ . \end{aligned} \tag{6.3}$$

The equations in (6.3) are known as the *normal equations* for the SLR model. The solution to this system of equations produces the unique LS estimators (Exercise 6.1)

$$\begin{aligned} \hat{\beta}_1 &= \frac{\sum_{i=1}^{n}(x_i - \overline{x})Y_i}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \text{ and} \\ \hat{\beta}_0 &= \overline{Y} - \hat{\beta}_1 \overline{x} \ . \end{aligned} \tag{6.4}$$

It can be shown that the maximum likelihood estimators (MLEs; see Section 5.2.4) for β_0 and β_1 are identical to these LS estimators. See Exercise 6.1.

To estimate the mean response in (6.2), simply apply the LS estimators for the regression coefficients to the model equation. This gives

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_0 x_i \ , \tag{6.5}$$

which are called the *fitted values* from the SLR. More generally, the estimated mean response at any predictor level x is $\hat{\mu}(x) = \hat{\beta}_0 + \hat{\beta}_0 x$.

Example 6.2.1 UK cancer mortality and local employment. To study national indicators of public health, fiscal, and societal trends, the United Kingdom collects data on mortality and other associated socioeconomic factors. For example, in a study similar to that in Example 3.5.1, cancer mortality rates among persons younger than 75 years of

Table 6.1 Selected data pairs (x_i, Y_i) with $x = \{\text{Employment rate}\}$ (as average quarterly percentage of adults over 16 gainfully employed) and $Y = \{\text{Cancer log-mortality rates per 100 000 population}\}$ (3-year running average), from a larger set of $n = 342$ paired observations recorded throughout the United Kingdom in 2008.

(90.3, 3.85)	(87.7, 4.45)	(87.6, 4.68)	⋯	(61.3, 5.05)	(60.7, 4.94)	(58.6, 4.72)

Source: http://data.gov.uk/dataset/ni_151_-_employment_rate.

age (per 100 000 population; averaged over the previous 3 years) were determined for $n = 342$ locations – specifically, principal local and regional authorities known as "local councils" – throughout the United Kingdom in 2008. Also recorded was each locality's employment rate (as average quarterly % of persons over 16 years of age in gainful employment). The employment rate is used here as a surrogate measure for the health of the local economy.

Study of a possible connection between cancer mortality and a locality's economic health is, perhaps, provocative: does higher employment bring lower cancer mortality, on average, to a community? (And if so, what is the latent connection? Are the citizens more active at work, better able to access quality health care, and/or more proactive about disease screenings? If a significant effect were identified, the potential for further knowledge discovery is intriguing.) To study this possibility, consider an SLR between the two variables via the model in (6.1). View $x = \{\text{Employment rate}\}$ as the predictor and mortality as the target response. As seen, for example, in Exercise 3.17, mortality rates often exhibit a right skew, so operate here with a logarithmic transform of the raw rates: $Y = \log\{\text{Mortality}\}$.

The data values, as (x_i, Y_i) pairs, appear in Table 6.1. (As above, only a selection of the data is given in the table. The complete set is available at http://www.wiley.com/go/piegorsch/data_analytics.)

A first step for any regression analysis is to *plot the data*. Figure 6.1 presents a scatterplot of the (x_i, Y_i) pairs, with the estimated LS line, $\hat{\mu}(x) = \hat{\beta}_0 + \hat{\beta}_1 x$ (derived later), overlaid. The plot shows a generally broad scatter; however, as employment increases, the speculated decrease in log-mortality appears clear. (A number of other interesting features are evident in the scatterplot; these are explored throughout the remainder of this and the next chapter.)

To quantify the potential effects seen in Figure 6.1, we can calculate LS estimators of the intercept and slope under the SLR model. For the slope, the data give $\sum_{i=1}^{n}(x_i - \overline{x})Y_i = -153.2633$ and $\sum_{i=1}^{n}(x_i - \overline{x})^2 = 10\,143.4665$. Thus

$$\hat{\beta}_1 = \frac{-153.2633}{10\,143.4665} = -0.0151.$$

As anticipated, the estimated slope is negative: an increase in employment of 1% relates on average to a change in log-mortality of −0.0151 units. This translates to $e^{-0.0151} = 0.9850$ or about 1.5% fewer cancer deaths.

We also find $\overline{Y} = 4.6884$ and $\overline{x} = 76.3968$. Thus the estimated intercept is $\hat{\beta}_0 = 4.6884 - (-0.0151)(76.3968) = 5.8428$.

Statistical software packages readily calculate these quantities. For instance, in **R**, the `lm()` function is used to perform linear regression. The syntax takes the response variable, say, `Y`, and regresses against the predictor variable, `x`, via the call `lm(Y~x)`. For `Y` = log{Mortality} and `x` = {Employment rate}, this produces the simple output

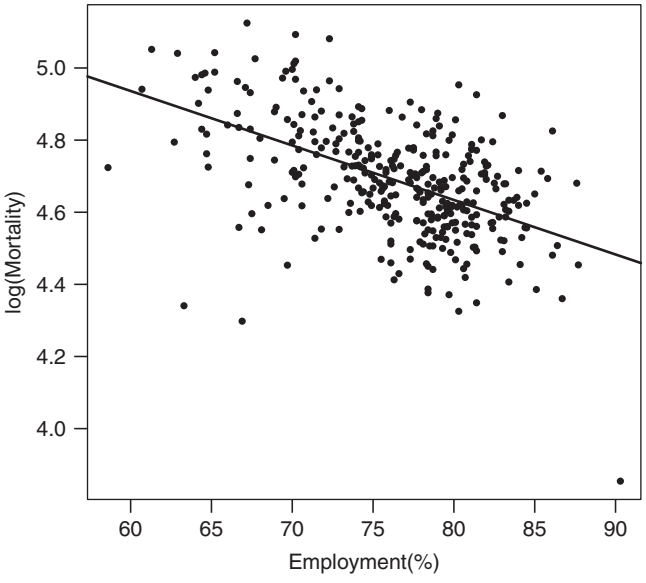

Figure 6.1 Scatterplot for UK cancer mortality data in Example 6.2.1. Least squares regression line $\hat{\mu}(x)$ is overlaid. Source: Data from http://data.gov.uk/dataset/ni_151_-_employment_rate.

```
Call:
    lm( formula = Y ~ x )

Coefficients:
  (Intercept)             x
    5.8427513     -0.0151096
```

The LS estimates are given in the final row: $\hat{\beta}_0$ under `(Intercept)` and $\hat{\beta}_1$ under `x` (for the predictor variable named in the call to `lm()`). □

The LS estimators in (6.4) possess a number of important qualities. One can show that $E[\hat{\beta}_j] = \beta_j$ for $j = 0, 1$; hence, the estimators are unbiased. Further, their sampling variances are

$$\begin{aligned} \mathrm{Var}[\hat{\beta}_0] &= \sigma^2 \left\{ \frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right\}, \text{ and} \\ \mathrm{Var}[\hat{\beta}_1] &= \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \end{aligned} \tag{6.6}$$

(see Exercise 6.2).

A famous result in linear model theory known as the *Gauss–Markov theorem* (Christensen 2011, Section 2.3) relates that the variances in (6.6) are the minimum possible among all unbiased estimators for β_0 and β_1. Thus the $\hat{\beta}_j$s are known as "best linear unbiased estimators" (BLUEs).

The complete sampling distribution for each $\hat{\beta}_j$ also follows from the SLR model in (6.1):

$$\hat{\beta}_j \sim \mathrm{N}\big(\beta_j, \mathrm{Var}[\hat{\beta}_j]\big)\,, \; j = 0, 1, \tag{6.7}$$

where $\text{Var}[\hat{\beta}_j]$ is given in (6.6). The covariance between the two estimators is $\text{Cov}[\hat{\beta}_0, \hat{\beta}_1] = -\sigma^2\bar{x}/\sum_{i=1}^n (x_i - \bar{x})^2$.

Using (6.7), one finds $\hat{\mu}(x) \sim \text{N}\big(\beta_0 + \beta_1 x, \text{Var}[\hat{\mu}(x)]\big)$ for any x, where

$$\text{Var}[\hat{\mu}(x)] = \sigma^2 \left\{ \frac{1}{n} + \frac{(x - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right\} \tag{6.8}$$

(again, see Exercise 6.2).

The variance parameter, σ^2, in (6.1) is usually unknown. It can be estimated by imitating the strategy employed with the simple sample variance S^2 in Section 3.3.2: sum the squared deviations between the observed values, Y_i, and the estimated response under the posited model. Then, divide by the number of degrees of freedom (d.f.) in the resulting sum of squares. Under the SLR model, the estimated responses are the fitted values $\hat{Y}_i$, thus the pertinent deviations are

$$e_i = Y_i - \hat{Y}_i \tag{6.9}$$

($i = 1, \ldots, n$), better known as the *residuals* from the model fit. Summing the squared residuals produces the aptly named *residual sum of squares*, also called the *error sum of squares* (SSE)

$$\text{SSE} = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2. \tag{6.10}$$

The SSE here has $n-2$ d.f. (called *error degrees of freedom*). In effect, this is the number of observations minus the number of parameters estimated from $\mu(x)$. Dividing the SSE by its d.f. produces a *mean squared error* (MSE):

$$\text{MSE} = \frac{\sum_{i=1}^n e_i^2}{n-2} = \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n-2}\,.$$

Just as the similarly constructed sample variance S^2 estimates the population variance in a single random sample, the MSE here estimates the population variance for the SLR model. Importantly, the MSE is an unbiased estimator of σ^2; that is, $E[\text{MSE}] = \sigma^2$. This follows from the larger fact that the MSE has its own sampling distribution:

$$\frac{(n-2)\text{MSE}}{\sigma^2} \sim \chi^2(n-2), \tag{6.11}$$

where under (6.1) both $\hat{\beta}_0$ and $\hat{\beta}_1$ are independent of this MSE-based χ^2 variable (Casella and Berger 2002, Section 11.3.4).

The residuals, e_i, in (6.9) are useful for more than just constructing the SSE. In particular, they help in assessing quality of the model fit and in other regression diagnostics. This is discussed further in Section 6.3.

For quantifying the uncertainty associated with estimation of these various regression quantities, and for conducting inferences on them, we calculate the estimators' standard errors. Recall from (5.7) that the standard error of a point estimator is simply the square root of its estimated sampling variance. For example, $\text{se}[\hat{\beta}_1]$ is the square root of $\text{Var}[\hat{\beta}_1]$ in (6.6), after inserting an appropriate estimate for the unknown variance parameter σ^2. In the latter instance,

we use the unbiased estimator MSE, producing

$$\text{se}[\hat{\beta}_1] = \sqrt{\frac{\text{MSE}}{\sum_{i=1}^{n}(x_i - \overline{x})^2}}. \tag{6.12}$$

Now, from (6.7), $\hat{\beta}_1$ is normally distributed so that $Z = (\hat{\beta}_1 - \beta_1)/\sqrt{\text{Var}[\hat{\beta}_1]} \sim \text{N}(0,1)$. Further, this Z is independent of the scaled MSE in (6.11). Incorporating the standard error in (6.12), one can build from these a t-distributed ratio in a similar manner to (5.16):

$$\frac{\hat{\beta}_1 - \beta_1}{se[\hat{\beta}_1]} \sim t(n-2) \tag{6.13}$$

(see Exercise 6.5). Using (6.13), confidence regions and hypothesis tests for β_1 are readily constructed. For instance, similar to the single-sample interval in (5.17), a $1-\alpha$ confidence interval for β_1 is based on

$$P\left\{\hat{\beta}_1 - t_{\alpha/2}(n-2)\ \text{se}[\hat{\beta}_1] < \beta_1 < \hat{\beta}_1 + t_{\alpha/2}(n-2)\ \text{se}[\hat{\beta}_1]\right\} = 1-\alpha\ , \tag{6.14}$$

that is, $\hat{\beta}_1 \pm t_{\alpha/2}(n-2)\ \text{se}[\hat{\beta}_1]$, which mimics the popular "Wald" form from (5.31). (Again, see Exercise 6.5.)

Hypothesis tests on β_1 are similarly developed. Notice in particular that when $\beta_1 = 0$, the mean response in (6.2) simplifies to just $\mu(x) = \beta_0$, that is, a constant that *does not depend upon* x. Thus, many analysts focus attention on testing whether or not $\beta_1 = 0$, that is, H_o: $\beta_1 = 0$ versus H_a: $\beta_1 \neq 0$. Again using the $t(n-2)$ relationship in (6.13), a corresponding t-statistic can be constructed by setting β_1 to its null value of 0:

$$T_1 = \frac{\hat{\beta}_1}{\text{se}[\hat{\beta}_1]}. \tag{6.15}$$

When H_o: $\beta_1 = 0$ is true, the null reference distribution for (6.15) is $T_1 \sim t(n-2)$. Reject the null hypothesis in favor of the two-sided alternative in H_a when $|T_1| \geq t_{\alpha/2}(n-2)$. The corresponding, two-sided P-value is

$$P = 2P[t(n-2) \geq |t_{1\text{calc}}|],$$

where $t_{1\text{calc}}$ is the observed value of the test statistic in (6.15). For one-sided tests of H_o: $\beta_1 = 0$ against, say, H_a: $\beta_1 > 0$, reject H_o in favor of H_a when $T_1 \geq t_\alpha(n-2)$. The corresponding, one-sided P-value is then $P = P[t(n-2) \geq t_{1\text{calc}}]$. One-sided tests against H_a: $\beta_1 < 0$ are similar. (Recall that the rationale for performing a one-sided test must be made *prior* to examining the data.)

In **R**, these various quantities are built into the `lm` object from the call to `lm(Y~x)`. For instance, `coef(lm(Y ~ x))` lists the LS estimates, `confint(lm(Y ~ x))` gives $1-\alpha$ confidence intervals, `fitted(lm(Y ~ x))` calculates fitted values, while `summary(lm(Y ~ x))` provides a detailed output with t-statistics, standard errors, and P-values. Of course, users may also calculate any of these quantities directly in **R** by appeal to the program's base arithmetic functions.

Example 6.2.2 UK cancer mortality and local employment (Example 6.2.1, continued). Return to the UK cancer mortality data in Table 6.1, where $Y = \log\{\text{Mortality}\}$ due to cancer was regressed on $x = \{\text{Employment rate}\}$, via the SLR model in (6.1) and (6.2). The LS point

estimates were found to be $\hat{\beta}_0 = 5.8428$ and $\hat{\beta}_1 = -0.0151$. These are used to calculate the fitted values $\hat{Y}_i = 5.8428 - 0.0151 x_i$, leading to an MSE of

$$\text{MSE} = \frac{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2}{n-2} = \frac{6.1428}{340} = 0.0181.$$

Recall also that $\sum_{i=1}^{n} (x_i - \bar{x})^2 = 10\,143.4665$.

The parameter of primary interest here is the slope, β_1. To construct its 95% confidence interval, $\hat{\beta}_1 \pm t_{\alpha/2}(n-2)\,\text{se}[\hat{\beta}_1]$, condition the analysis on the observed pattern of employment rates and start by calculating the standard error $\text{se}[\hat{\beta}_1]$. From (6.12), this is $\sqrt{0.0181/10\,143.4665} = 1.336 \times 10^{-3}$. With $t_{0.025}(340) = 1.9670$, the 95% interval calculates to

$$-0.0151 \pm (1.9670)(1.336 \times 10^{-3}) = -0.0151 \pm 0.0026$$

or $-0.0177 < \beta_1 < -0.0125$. In **R**, find this as the lower row in the output from the `confint()` function:

```
> confint( lm(Y ~ x) )
                  2.5 %      97.5 %
 (Intercept)   5.6416923   6.0438103
  x           -0.0177347  -0.0124844
```

One might alternatively perform an hypothesis test here, based on the original speculation that increasing employment may lead to decreases in cancer mortality. As phrased, this is a valid a priori rationale for testing H_o: $\beta_1 = 0$ versus the one-sided alternative H_a: $\beta_1 < 0$.

Set the significance level to $\alpha = 0.05$. The test statistic from (6.15) is

$$t_{1\text{calc}} = \frac{\hat{\beta}_1}{\text{se}[\hat{\beta}_1]} = \frac{-0.0151}{1.336 \times 10^{-3}} = -11.3024.$$

(Using the higher precision available in, e.g., **R**, a more-accurate value is $t_{1\text{calc}} = -11.3214$; see the following output.) The consequent, one-sided P-value is $P[t(340) < -11.3214]$ which is available in **R** via `pt(-11.3214,df=340)`. This gives $P \approx 10^{-25}$, which is well below α. We therefore reject H_o and conclude, perhaps intriguingly, that log-mortality decreases significantly as weekly earnings increase for these UK communities. Further study to understand what might underly this feature may be warranted.

These various statistics appear throughout the **R** output (edited) from a call to `summary(lm(Y ~ x))`:

```
 Call:
      lm(formula = Y ~ x)
 Coefficients:
                Estimate  Std. Error   t value    Pr(>|t|)
 (Intercept)   5.8427513   0.1022178   57.1598  < 2.22e-16
  x           -0.0151096   0.0013346  -11.3214  < 2.22e-16

 Residual standard error: 0.134414 on 340 degrees of freedom
 Multiple R-squared:  0.27378
```

The P-values are given in the final column of the central output, under the header `Pr(>|t|)`. Notice that the P-value for testing H_o: $\beta_1 = 0$ is two sided, as indicated by the absolute-value

signs in the header. Dividing by 2 here would recover the one-sided P-value, although, in either case, the value is so significant that the result is best reported as simply $P < 0.0001$. □

Any of these various operations may also be applied for building inferences on β_0 or on $\mu(x)$. For instance, from (6.6), the standard error of $\hat{\beta}_0$ is

$$\text{se}[\hat{\beta}_0] = \sqrt{\text{MSE}\left\{\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}\right\}}. \tag{6.16}$$

Use this in the $1 - \alpha$ confidence interval $\hat{\beta}_0 \pm t_{\alpha/2}(n-2)\ \text{se}[\hat{\beta}_0]$. (Hypothesis tests for β_0 are similarly developed.) Or, from (6.8), the standard error of $\hat{\mu}(x) = \hat{\beta}_0 + \hat{\beta}_1 x$ is

$$\text{se}[\hat{\mu}(x)] = \sqrt{\text{MSE}\left\{\frac{1}{n} + \frac{(x - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}\right\}}, \tag{6.17}$$

with corresponding $1 - \alpha$ confidence interval at any (pointwise) value of x given by

$$\hat{\mu}(x) \pm t_{\alpha/2}(n-2)\ \text{se}[\hat{\mu}(x)]. \tag{6.18}$$

Prediction of a future observation, say, $\hat{Y}(x)$ at a given x is also possible; the point estimate is again $\hat{Y}(x) = \hat{\beta}_0 + \hat{\beta}_1 x$, but with a correspondingly more complex standard error $\text{se}[\hat{Y}(x)]$ that takes into account the prediction feature. See Kutner et al. (2005, Section 2.5) for full details.

A useful summary statistic in regression analysis compares the residual variation with the total variation in the sample. Recognize that if no regression model were being fit to the Y_is, the sum of squares used to measure variability would be the numerator of the sample variance: $\sum_{i=1}^n (Y_i - \overline{Y})^2$. Denote this as the *total sum of squares*, or SSTo. The analogous quantity under the SLR model is the SSE from (6.10). Their difference, SSTo − SSE, represents variation in Y_i that is accounted for by the regression; thus the ratio

$$R^2 = \frac{\text{SSTo} - \text{SSE}}{\text{SSTo}} = 1 - \frac{\text{SSE}}{\text{SSTo}}. \tag{6.19}$$

can be viewed as the percentage variation in Y_i accounted for by variation in the x_is. (Notice that $0 \le R^2 \le 1$.) This quantity is known as the *coefficient of determination*. Its intuitive interpretation makes it an often-used (and sometimes, over-used) summary for the value of the SLR. Numerically, R^2 can be shown to equal the squared sample correlation between x_i and Y_i from (3.9), motivating use of the R symbol.

It is worth noting that the interpretive quality of the R^2 statistic varies from setting to setting, and will be domain dependent. A linear regression producing R^2 near 80% might indicate a quality fit for an ecological data set; by contrast, the same $R^2 = 0.80$ could border on substandard for a chemometric regression analyses. "How high is high?" for R^2 depends on the nature of error variation seen in each subject-matter application.

Example 6.2.3 Cancer mortality and local employment (Example 6.2.1, continued). Return to the UK cancer mortality data in Table 6.1, where $Y = \log\{\text{Mortality}\}$ due to cancer was regressed on $x = \{\text{Employment rate}\}$, via the SLR model in (6.1) and (6.2). Recall that the error sum of squares for this model was SSE = 6.1428. As $\overline{Y}$ was seen to be 4.6884, the

total sum of squares can be calculated as $\text{SSTo} = \sum_{i=1}^{342} (Y_i - 4.6884)^2 = 8.4586$. This gives

$$R^2 = 1 - \frac{6.1428}{8.4586} = 0.2738$$

or 27.38%. This is a low value, suggesting that variability in the data remains substantial even after incorporating the effects of x_i. (Small R^2 values are not altogether unusual with some biomedical or, for that matter, socioeconomic studies in humans. Simply put, people tend to be highly variable in their social, economic, and biological responses.) Recall, however, that the slope coefficient was seen in Example 6.2.2 to be highly significant: decreasing employment rates appear to significantly affect cancer (log-)mortality in these communities, although the large amount of unexplained variation suggests that additional factors may also be at play. (See Example 7.1.1.)

R^2 is routinely displayed by most regression software programs. For example, **R** provides it near the bottom of the display from the `summary()` command (cf. the output in Example 6.2.2). □

6.2.2 Multiple inferences and simultaneous confidence bands

Recall from Section 5.5 that adjustments for multiplicity are prescribed whenever more than one inference is performed on a single set of data. This issue is a pertinent one in regression analysis, because there are often many different parameters or model features under study. For instance, analysts may wish to perform *joint* inferences on both of the unknown regression parameters, β_0 and β_1. The various tests and confidence intervals presented in the previous section were all constructed to be *pointwise* in nature, however. If multiple inferences are conducted on the same data, some correction for multiplicity is in order.

Perhaps the simplest way to form multiplicity-adjusted, joint confidence intervals for both β_0 and β_1 is to apply the Bonferroni correction from Section 5.5.1. The approach mimics that in Example 5.5.1, where joint intervals were constructed for the two unknown parameters, μ and σ^2, from an independent, identically distributed (i.i.d.) normal sample. With the SLR model in (6.1), we again have two unknown parameters, β_0 and β_1, so we apply the Bonferroni correction by simply replacing α with $\alpha/2$ wherever it appears in the two sets of confidence limits. This produces the joint (minimal) $1 - \alpha$ confidence rectangle

$$\begin{aligned} &\hat{\beta}_0 \pm t_{\alpha/4}(n-2)\ \text{se}[\hat{\beta}_0], \text{ and} \\ &\hat{\beta}_1 \pm t_{\alpha/4}(n-2)\ \text{se}[\hat{\beta}_1]. \end{aligned} \tag{6.20}$$

It is possible to develop a less-conservative, two-dimensional, joint confidence region for β_0 and β_1 by extending the Wald region in (5.49). The resulting $1 - \alpha$ region takes the form of an ellipse, defined by the inequality

$$\frac{\text{se}^2[\hat{\beta}_1](\beta_0 - \hat{\beta}_0)^2 - 2C_{01}(\beta_0 - \hat{\beta}_0)(\beta_1 - \hat{\beta}_1) + \text{se}^2[\hat{\beta}_0](\beta_1 - \hat{\beta}_1)^2}{\text{se}^2[\hat{\beta}_0]\text{se}^2[\hat{\beta}_1] - C_{01}^2} \leq 2F_\alpha(2, n-2), \tag{6.21}$$

where the standard errors $\text{se}[\hat{\beta}_0]$ and $\text{se}[\hat{\beta}_1]$ are given in (6.16) and (6.12), respectively, and C_{01} is the estimated covariance between $\hat{\beta}_0$ and $\hat{\beta}_1$:

$$C_{01} = \widehat{\text{Cov}}[\hat{\beta}_0, \hat{\beta}_1] = \frac{-\bar{x}\ \text{MSE}}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

(cf. Exericse 6.2). The estimated variance–covariance terms may be accessed in **R** by applying the `vcov()` function to the `lm` object from the model fit. This produces the full 2×2 estimated covariance matrix for $\hat{\beta}_0$ and $\hat{\beta}_1$; cf. (5.6).

Although more complex in both its construction and its implementation, the joint confidence ellipse in (6.21) often provides far better precision for joint inferences on the regression coefficients than the conservative Bonferroni rectangle.

Example 6.2.4 Cancer mortality and local employment (Example 6.2.1, continued). Return to the UK cancer mortality data in Table 6.1, where $Y = \log\{\text{Mortality}\}$ due to cancer was regressed on $x = \{\text{Employment rate}\}$, via the SLR model in (6.1) and (6.2). As β_0 represents the log-mortality when a community's employment rate is zero – an arguably undesirable level – inferences on the intercept may be of limited socioeconomic interest here. Nonetheless, construction of a joint confidence region for both β_0 and β_1 is instructive.

Begin by setting the confidence level to $1 - \alpha = 0.95$. The various quantities required for the calculation of both the Bonferroni rectangle in (6.20) and the confidence ellipse in (6.21) were previously calculated or displayed in Examples 6.2.1 and 6.2.2, save for the estimated covariance C_{01}. Direct calculation gives

$$C_{01} = \frac{-\bar{x}\,\text{MSE}}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = -\frac{(76.3968)(0.0181)}{10\,143.4665} = -1.361 \times 10^{-4}.$$

Alternatively, one can apply **R**'s `vcov()` function:

```
> vcov( lm(Y ~ x) )
              (Intercept)             x
(Intercept)   0.010448485  -1.36075e-04
x            -0.000136075   1.78116e-06
```

Notice that this gives a correlation between $\hat{\beta}_0$ and $\hat{\beta}_1$ close to its lower limit of -1:

$$\text{Corr}[\hat{\beta}_0, \hat{\beta}_1] = \frac{C_{01}}{\text{se}[\hat{\beta}_0]\text{se}[\hat{\beta}_1]} = \frac{-1.361 \times 10^{-4}}{\sqrt{1.045 \times 10^{-2}}\sqrt{1.781 \times 10^{-6}}} = -0.9975.$$

For a twofold multiplicity adjustment, the (minimal) 95% Bonferroni limits require $t_{0.05/4}(340) = 2.2514$, leading to the joint confidence rectangle

$$\begin{aligned} 5.8428 - (2.2514)(0.1022) = 5.6127 &< \beta_0 \\ &< 6.0729 = 5.8428 + (2.2514)(0.1022), \\ -0.0151 - (2.2514)(0.0013) = -0.0181 &< \beta_1 \\ &< -0.0121 = -0.0151 + (2.2514)(0.0013). \end{aligned}$$

The alternative, joint 95% confidence ellipse requires $F_{0.05}(2, 340) = 3.0223$. The ellipse bounds a region containing all values of β_0 and β_1 that satisfy

$$\begin{aligned} &\frac{(1.78 \times 10^{-6})(\beta_0 - 5.8428)^2}{(0.0104)(1.78 \times 10^{-6}) - (-0.000136)^2} \\ &- \frac{(2)(-0.000136)(\beta_0 - 5.8428)(\beta_1 + 0.0151)}{(0.0104)(1.78 \times 10^{-6}) - (-0.000136)^2} \\ &+ \frac{(0.0104)(\beta_1 + 0.0151)^2}{(0.0104)(1.78 \times 10^{-6}) - (-0.000136)^2} \leq (2)(3.0223). \end{aligned}$$

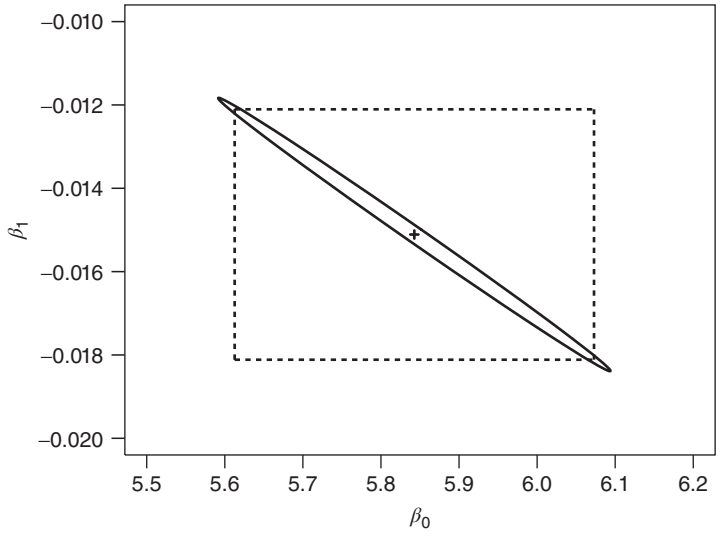

Figure 6.2 95% joint confidence regions for (β_0, β_1) with UK cancer mortality data in Example 6.2.4: joint confidence ellipse from (6.21) (solid curve) and joint Bonferroni rectangle from (6.20) (dashed lines). Least squares point estimate marked by a cross (+). Source: Data from http://data.gov.uk/dataset/ni_151_-_employment_rate.

Figure 6.2 overlays the two regions in (β_0, β_1) space. The especially strong correlation $\text{Corr}[\hat{\beta}_0, \hat{\beta}_1]$ approaching -1 induces heavy eccentricity (narrow elongation) in the confidence ellipse. By contrast, the Bonferroni rectangle remains relatively wide, illustrating its extreme conservatism with these data.

Construction of the confidence ellipse in **R** is facilitated by the use of the `ellipse()` function in the external *ellipse* package. □

Issues of multiplicity adjustment become even more complex when considering inferences on the mean response in (6.2). While estimation of $\mu(x)$ via $\hat{\mu}(x)$ at any x is straightforward, construction of, say, confidence limits on $\mu(x)$ requires more care. The pointwise limits in (6.18) are valid for any value of x, but for *only a single such value*. If inferences are desired at more than one x, correction for multiplicity is required.

The simplest approach for making such corrections is to again apply a Bonferroni correction from Section 5.5.1. Suppose interest centers on joint $1 - \alpha$ confidence bounds for $\mu(x)$ at an a priori set of $H > 1$ predictor values x_h, $h = 1, \ldots, H$. (These can include any of the original predictors, x_i, or an entirely new set of values, or any combination thereof.) Then, the Bonferroni adjustment simply modifies the pointwise limits in (6.18) by dividing H into α at every x_h:

$$\hat{\mu}(x_h) \pm t_{\alpha/(2H)}(n-2)\ \text{se}[\hat{\mu}(x_h)] \tag{6.22}$$

$(h = 1, \ldots, H)$, where $\text{se}[\hat{\mu}(x_h)]$ is given by (6.17). It is straightforward to verify that for fixed α, $t_{\alpha/(2H)}(n-2)$ grows as H increases, forcing the joint intervals to widen. This again reinforces the potential conservatism of the Bonferroni bounds.

A useful alternative to the Bonferroni adjustment for multiple inferences on $\mu(x)$ applies a method for constructing confidence bounds at *every* value of x on the real line. Owing to Working and Hotelling (1929) and Scheffé (1953), the approach creates a $1 - \alpha$ *simultaneous confidence band* around the line represented by $\mu(x)$. For purposes of calculation, the

Working–Hotelling–Scheffé (WHS) bands are almost identical to the Bonferroni confidence bounds; the only change from (6.22) is to modify the critical point:

$$\hat{\mu}(x) \pm \sqrt{2F_{\alpha}(2, n-2)}\ \mathrm{se}[\hat{\mu}(x)]. \tag{6.23}$$

The properties of the confidence band in (6.23) are worth reemphasizing: the band provides simultaneous $1-\alpha$ coverage on $\mu(x)$ for *all* values of x. This includes any predictor values that were not originally considered by the analyst. Thus, while the Bonferroni bounds in (6.22) require specification of the H target x_hs prior to viewing the data, the WHS bands can be applied in either an a priori or an a posteriori manner to any values of x.

Of course, the "no-free-lunch theorem" (Section 5.5.1) still applies: the width of the WHS bands in (6.23) is usually larger than that of the Bonferroni bounds in (6.22). That is,

$$t_{\alpha/(2H)}(n-2) < \sqrt{2F_{\alpha}(2, n-2)}$$

for most values of H. This need not always be the case, however (Schwager 1984). For any known set of H target predictors x_h, analysts should always check that the Bonferroni bounds are indeed tighter than the WHS band for their intended application. Of course, when interest centers on the entire mean response line, or on any *post hoc* confidence statements, the WHS band is the proper inference vehicle.

Example 6.2.5 Cancer mortality and local employment (Example 6.2.1, continued). Return to the UK cancer mortality data in Table 6.1, where $Y = \log\{\text{Mortality}\}$ due to cancer was regressed on $x = \{\text{Employment rate}\}$, via the SLR model in (6.1) and (6.2). Construction of a WHS simultaneous band for $\mu(x)$ from (6.23) is relatively straightforward, since essentially all its components have been calculated in the previous examples. The LS estimator for the mean response is $\hat{\mu}(x) = 5.8428 - 0.0151x$, with

$$\mathrm{se}[\hat{\mu}(x)] = \sqrt{0.0181\left\{\frac{1}{342} + \frac{(x-76.3968)^2}{10\,143.4665}\right\}}$$

from (6.17). For a 95% band, the pertinent WHS critical point is $\sqrt{2F_{0.05}(2, 340)} = \sqrt{(2)(3.0223)} = 2.4586$. The resulting simultaneous bands are represented by

$$5.8428 - 0.0151x \pm 2.4586\sqrt{0.0181\left\{\frac{1}{340} + \frac{(x-76.3968)^2}{10\,143.4665}\right\}}$$

for all x. Figure 6.3 plots the bands, along with the centering LS regression line, overlayed on the original scatterplot. Notice that the bands are hyperbolic in shape, with minimum width at $x = \bar{x} = 76.3968$. (This is also recognizable from their mathematical expression, above.) They also fail to admit a horizontal line between their upper and lower bounds, consistent with the earlier inference from Example 6.2.2 that β_1 is negatively valued. □

The observation in Example 6.2.5 that the WHS confidence bands for $\mu(x)$ achieve minimum width at $x = \bar{x}$ also extends to the Bonferroni bounds in (6.22). Both have widths proportional to $\mathrm{se}[\hat{\mu}(x)]$, and as can be seen in (6.17), this standard error (i) is minimized at $x = \bar{x}$, and (ii) expands hyperbolically and symmetrically as x diverges from $\bar{x}$. This has the effect of widening either confidence statement for choices of x farther away from the center of

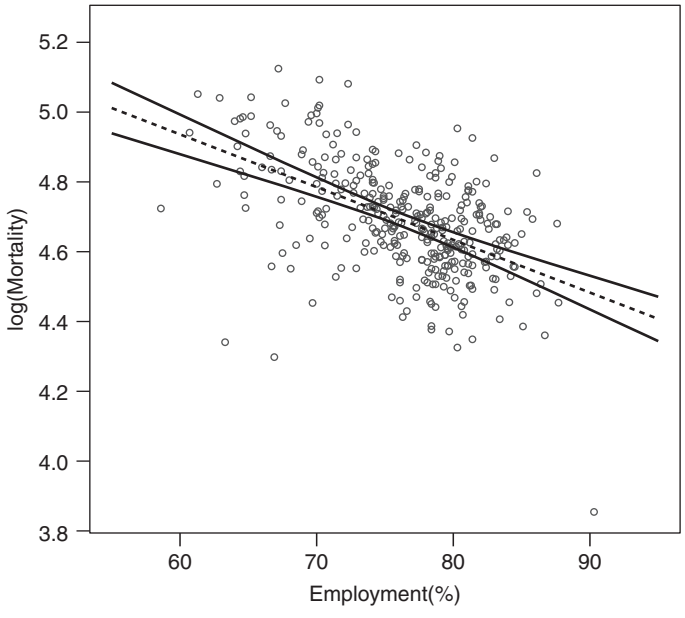

Figure 6.3 Ninety-five percent WHS simultaneous confidence bands (solid curves) for $\mu(x)$ with UK cancer mortality data in Example 6.2.5. Least squares estimator $\hat{\mu}(x)$ given by dashed line. Original scatterplot from Figure 6.1 superimposed. Source: Data from http://data.gov.uk/dataset/ni_151_-_employment_rate.

the x-range. In a certain sense, this is a desirable quality: as x draws away from $\bar{x}$, the strength of the regression relationship imparted by the larger data cloud weakens. The resulting standard errors represent this by widening the confidence intervals/bands.

Indeed, values of x outside the range of the observed predictor values represent true *extrapolations* and are subject to question. It is typically unclear whether any given model will apply outside the range of the observations, because by definition, no data are available to certify the model's use. Unless the analyst is assured that the model is valid at points away from the data cloud, any such extrapolated inferences – either simultaneous or pointwise – may have questionable validity.

6.3 Regression diagnostics

The quality of an SLR fit can vary, and it is good analytic practice to examine the fit for potential violations of the model assumption, outlying observations, or other unusual features. A variety of diagnostic tools are available for this task. One of the most basic, and also most useful, is analysis of the residuals, e_i, from (6.9). In effect, the residuals estimate variation in the Y_is that remains – hence their name – after the SLR effect has been accounted for via the model fit.

A simple tool to study residual variation graphs e_i against the fitted values $\hat{Y}_i$. (For the SLR model, one could equivalently plot e_i against the predictor variable x_i.) Known as a *residual*

plot, this graphic very effectively visualizes the quality of an SLR fit. When the model is correct, $E[e_i] = 0$ (Exercise 6.3d). Thus we expect the residual plot to display essentially random scatter about $e = 0$. In fact, under (6.1) the residuals should imitate random normal (Gaussian) noise. Figure 6.4 gives a prototypical example.

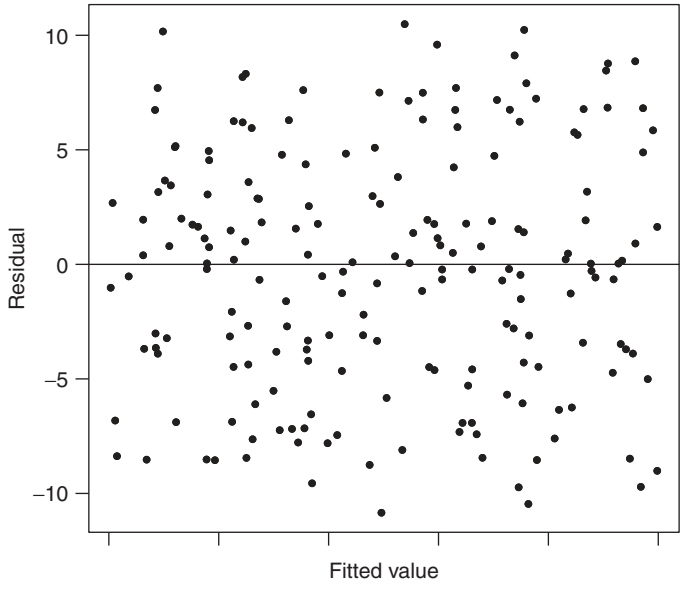

Figure 6.4 Prototypical residual plot with desired random scatter about $e = 0$. Horizontal line at $e = 0$ included for reference.

By contrast, if a residual plot displays a clear pattern – of which there are many different types – then some model violation or discrepancy may be indicated. A common violation is heterogeneous variance, i.e. a departure from the assumption that $\text{Var}[Y_i] = \sigma^2$ is constant. This can be quickly recognized in a residual plot by variation(s) in the width of the residual pattern. For instance, if $\text{Var}[Y_i]$ increases with increasing x_i, a plot of e_i vs. x_i will show a broadening of the residuals around $e = 0$ as x_i increases. If the variation drops with increasing x_i, then the pattern will be reversed. Plots against the fitted values, $\hat{Y}_i$, will show similar effects; Figure 6.5 illustrates the phenomenon. In this case, some remedial action to account for the heterogeneous variability would be required. (Possible remedies are discussed in Section 6.4.)

Another form of model violation observable in a residual plot is departure from linearity, where the mean response takes some form of nonlinear function. For instance, if $E[Y]$ is quadratic in x but an SLR model is fit, the fitted values will under- and overestimate the true response in a recognizable, alternating pattern; Figure 6.6 gives a typical example. Here again, some remedial action would be required, such as expanding the SLR model to include a quadratic term in x; see Section 7.2.

The residuals are also valuable in assessing the quality of the normality assumption in (6.1). If normality is valid, the raw residuals should display normal variation, at least to a good approximation. Histograms, density estimates, or stemplots of the e_is may help in visualizing the effect. Or, the normal quantile plot discussed in Section 4.1.5 could be applied: under

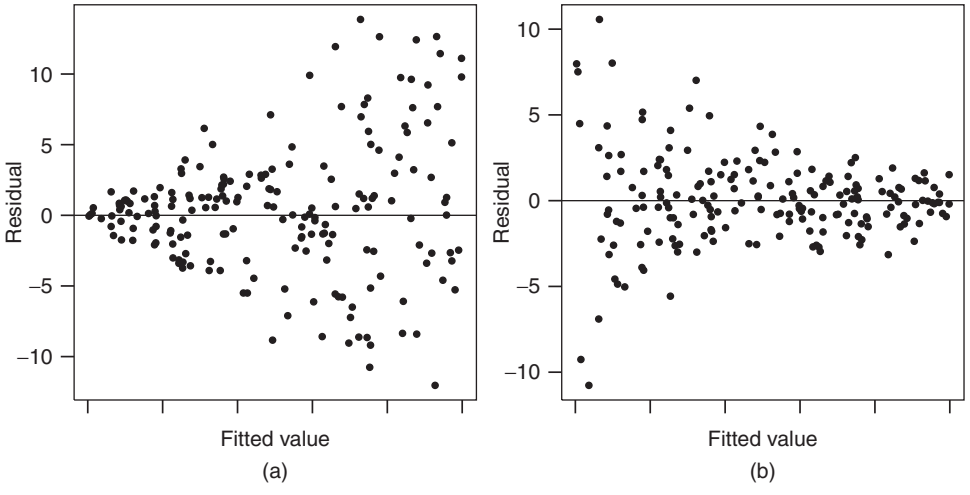

Figure 6.5 Residual plots indicating heterogeneous variation: (a) increasing variation or (b) decreasing variation as fitted values increase. Horizontal lines at $e = 0$ included for reference.

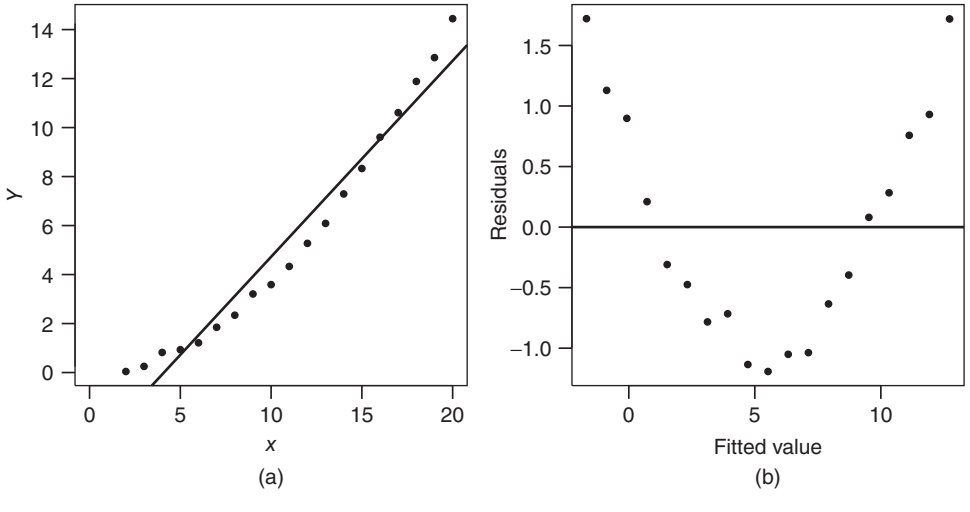

Figure 6.6 (a) Scatterplot and (b) corresponding residual plot indicating curvilinear mean response. LS line for SLR model overlaid on the scatterplot, and horizontal $e = 0$ line overlaid on the residual plot, for reference.

normality, a normal quantile plot of the residuals will appear roughly linear. If the plot deviates strongly from normal variation, however, a transformation of the original Y_is such as the power transform in (3.13) (or, if the data justify it, consideration of a generalized linear model as in Chapter 8) may be warranted.

Residual plots can also aid in the identification of potential *outliers*. Extending the concept of a univariate outlier from Section 3.4.1, a residual that is far away from $e = 0$ in either a

positive or negative direction, and also lies far afield from the bulk of the residual cloud, may indicate an outlying data pair with respect to the SLR model.

Raw residuals are scale/measurement dependent, however. In one data set, an absolute residual of $|e_i| = 8.2$ may be less egregious than another data set's residual of $|e_i| = 0.7$. To adjust for this, we stabilize the raw residuals to a commensurate scale. The operation essentially "Studentizes" the residual, much like that seen with the sample mean in (5.16): subtract the true mean and divide by the standard error. The former is trivial: $E[e_i] = 0$. To find the latter, start with the variance. Under the SLR model,

$$\text{Var}[e_i] = (1 - h_{ii})\sigma^2$$

(Kutner et al. 2005, Section 6.4), where h_{ii} is taken from a special matrix known as a *hat matrix*. (The hat matrix is described in more detail in Section 7.1.1.) For use with the SLR model here, one can show (Exercise 7.8)

$$h_{ii} = \frac{nx_i^2 - 2x_i \sum_{\ell=1}^n x_\ell + \sum_{\ell=1}^n x_\ell^2}{n \sum_{\ell=1}^n x_\ell^2 - \left(\sum_{\ell=1}^n x_\ell\right)^2} \tag{6.24}$$

($i = 1, \ldots, n$). As σ^2 is unknown in $\text{Var}[e_i]$, we estimate it with the MSE. Taking square roots then produces the standard error $\text{se}[e_i] = \sqrt{(\text{MSE})(1 - h_{ii})}$.

With this, the Studentized residual becomes

$$\frac{e_i}{\sqrt{(\text{MSE})(1 - h_{ii})}} .$$

A further refinement for stabilizing the residuals involves examination of how *case deletion* of individual observations affects the model fit. Clearly, one would expect the SLR fit to change more drastically when extreme, outlying observations are deleted, than when observations more in line with the rest of the data are removed. Formally then, denote the predicted value under the SLR model when the ith observation is removed from the data as $\hat{Y}_{i[-i]}$. If this value deviates particularly far from Y_i, the original observation could be a potential outlier. To quantify this, the *(raw) deleted residual* is $Y_i - \hat{Y}_{i[-i]}$ and its Studentized version is $t_i = (Y_i - \hat{Y}_{i[-i]})/\text{se}[Y_i - \hat{Y}_{i[-i]}]$. Extremely large values of t_i identify potential outlying response values.

At first glance, it appears that calculation of the t_is requires recalculation of the SLR fit for every deleted observation. A series of important computing formulae avoid this drawback, however. First, the deleted residual $Y_i - \hat{Y}_{i[-i]}$ can be shown to equal $e_i/(1 - h_{ii})$, where h_{ii} is given in (6.24). From this, the standard error is simpler to derive. In the end, the complete *Studentized deleted residual* becomes

$$t_i = e_i \sqrt{\frac{n - p - 2}{(\text{SSE})(1 - h_{ii}) - e_i^2}} \tag{6.25}$$

(Kutner et al. 2005, Section 10.2), where p is the number of predictor/feature variables fit in the model. (This general notation for p will be useful in future chapters. For the SLR setting, obviously, $p = 1$.)

Studentized deleted residuals follow t-distributions, with $t_i \sim t(n - p - 2)$. Thus if an individual observation's Studentized deleted residual exceeds, say, the upper-$\frac{\alpha}{2}$ critical point from

this reference distribution, it can be viewed as a possible outlier. Applying a Bonferroni correction (Section 5.5.1) when testing across all n observations leads to the following outlier criterion for an SLR model fit: consider the ith observed response to be a potential outlier if

$$|t_i| \geq t_{\alpha/(2n)}(n - p - 2), \tag{6.26}$$

where the Studentized deleted residual t_i is given in (6.25).

Example 6.3.1 UK cancer mortality and employment (Example 6.2.1, continued). Return to the UK cancer mortality data in Table 6.1, where $Y = \log\{\text{Mortality}\}$ due to cancer was regressed on $x = \{\text{Employment rate}\}$, via the SLR model in (6.1) and (6.2). The residuals e_i from the SLR fit can be computed in **R** by applying the `resid()` function to the `lm` object, for example, `resid(lm(Y ~ x))`. As an initial diagnostic, Figure 6.7 displays a normal quantile plot (Section 4.1.5) for the raw residuals. This is achieved in **R** via the sample commands

```
> qqnorm( resid(lm(Y ~ x)) )
> qqline( resid(lm(Y ~ x)) )
```

The `qqline()` command supplies a reference line passing through the first and third quartiles. Visually, departures from this line seen over the bulk of the data range help to indicate a lack of normality in the residuals.

The quantile pattern appears generally consistent with a normal distribution: for the most part, the quantiles line up and coincide with the overlaid normal reference line. Three unusual

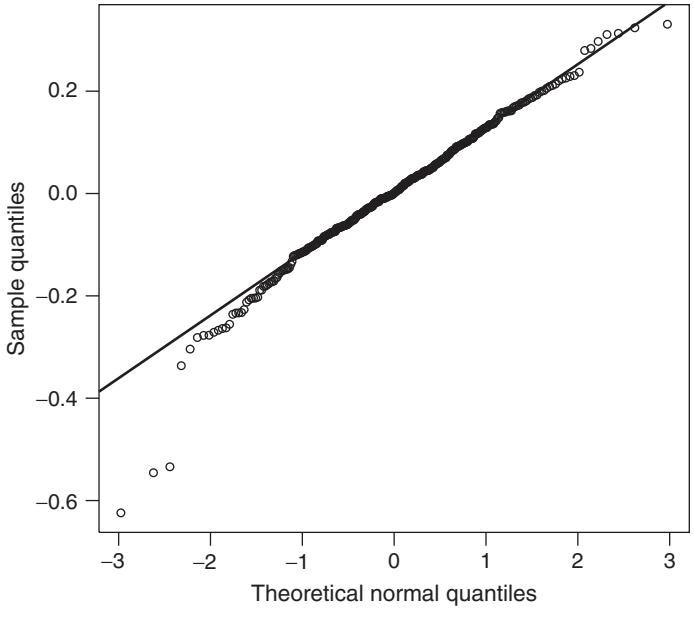

Figure 6.7 Normal quantile plot for raw residuals, e_i, from SLR fit with UK cancer mortality data in Example 6.3.1. Solid line is normal reference line. Source: Data from http://data.gov.uk/dataset/ni_151_-_employment_rate.

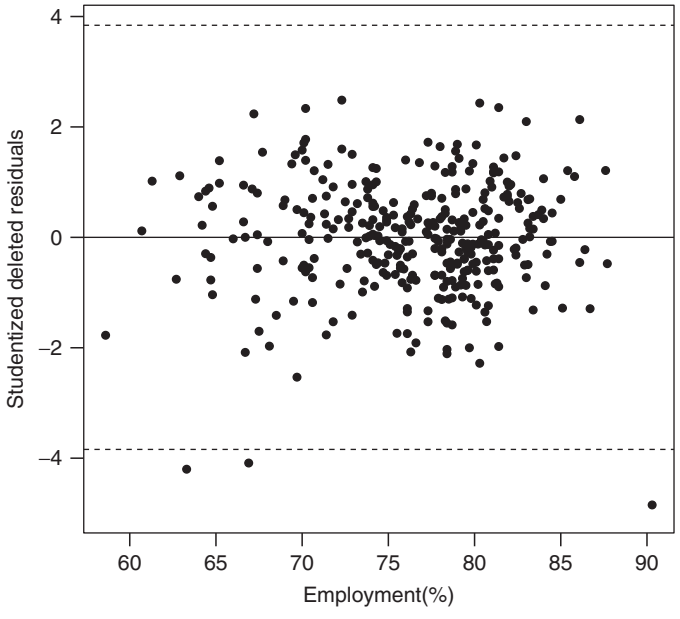

Figure 6.8 Studentized deleted residual plot from SLR fit with cancer mortality data in Example 6.3.1. Dashed lines are 5% exceedance levels, $\pm t_{0.05/684}(339) = 3.8411$. Solid horizontal line at $t = 0$ included for reference. Source: Data from http://data.gov.uk/dataset/ni_151_-_employment_rate.

points appear in the lower tail of the distribution, however. These help to prompt the next step in the diagnostic process, outlier detection.

As a diagnostic tool, residual visualization with these data provides an opportunity for some intriguing knowledge discovery. The raw residual plot and the Studentized deleted residual plot present roughly the same pattern (Exercise 6.8), although the latter is preferred because it adjusts for possible scale issues. The Studentized deleted residuals, t_i, may be accessed in **R** via `rstudent(lm(Y ~ x))`. These are plotted in Figure 6.8 against employment rate x_i, where the scatter around $t = 0$ appears generally random. No troublesome patterns are evidenced in the general scatter.

The plot also displays exceedance limits from (6.26) at $\alpha = 0.05$ for identifying potential outliers: with $n = 342$ and $p = 1$, $\pm t_{0.05/684}(339) = \pm 3.8411$. Three communities drop below the lower limit in the plot (none go above the upper limit), suggesting potential outlying status.

The fact that the three potential outliers in Figure 6.8 all drop *below* the exceedance limit is intriguing: these three communities all exhibit lower-than-expected log-mortality. They correspond to the three lowest-lying points in Figure 6.1 and are the three unusual, lower-tail points marked by the normal quantile plot in Figure 6.7. Closer inspection of the data shows that the three localities are (starting with the lowest residual): the City of London, the Royal Borough of Kensington & Chelsea (RBKC), and the City of Westminster. All are London Boroughs (LBs: administrative districts that make up the larger London metropolitan area). In fact, the three adjoin at the core of central London. The City of London is the smallest, barely over a square mile in area and with only about 7000 residents. It is, however, the business and commercial heart of London and a leading center of global finance. (This may explain its elevated

employment rate of 90.3%, the largest in the UK sample.) The other two localities are larger and have much higher population densities. The three councils also report some of the highest median earnings in London and across the United Kingdom: for 2008, the City of London was first in median earnings, Westminster was fifth, and the RBKC was forty-seventh. (Example 7.1.1 explores some of these factors in more detail.) Given these three communities' similar geographic and socioeconomic features, their potential outlying status may be motivation for further study by British sociologists, economists, and medical researchers. □

An observation need not be an "outlier" to affect the fit of a regression model. For instance, if a single x_i rests far from the bulk of the other predictor values, it can literally lever the regression fit. We say that an x_i's *leverage* is its ability to strongly influence the fit of the regression. (This is usually seen as a detriment.) Some instructive online applications that illustrate the leverage effect are available at

http://www.amstat.org/publications/jse/v6n3/applets/regression.html

and

http://www.stat.tamu.edu/jhardin/applets/signed/Outreg.html.

Readers are encouraged to experiment with these and see how a single x_i can manipulate an SLR fit.

The Studentized deleted residual in (6.25) gives guidance on ways to quantify leverage, via its component "hat" value h_{ii} from (6.24). These hat values can be shown to lie between 0 and 1. Recognize then that as $h_{ii} \to 1$, $|t_i|$ will grow larger, while the reverse occurs as $h_{ii} \to 0$. Other factors can affect the value of $|t_i|$, of course, but this nonetheless indicates the impact h_{ii} has on an observation's status: small values of h_{ii} will associate with observations closer in some manner to the core of the data, while larger values may act otherwise.

A practical rule-of-thumb for gauging the leverage of a predictor value is to flag any x_i whose hat value exceeds twice the average of all the h_{ii}s. In fact, the sum $\sum_{i=1}^{n} h_{ii}$ always equals 1 plus the number of predictor/feature variables fit in the model (Kutner et al. 2005, Section 10.3), or simply $p + 1$ using the notation introduced in (6.25). Thus, the leverage rule-of-thumb simplifies to

$$h_{ii} > \frac{2(p+1)}{n} ,$$

which reduces to $h_{ii} > 4/n$ for the $p = 1$ SLR setting.

Notice that the hat values from (6.24) do not depend on the Y_is – that is, they are functions only of the predictor values – and thus this leverage criterion can be examined for any x_i prior to observing the response values. This can be a useful exercise if the design of the predictor spacings is determined before collecting the data.

Example 6.3.2 UK cancer mortality and employment (Example 6.2.1, continued). Return to the UK cancer mortality data in Table 6.1, where $Y = \log\{\text{Mortality}\}$ due to cancer was regressed on $x = \{\text{Employment rate}\}$, via the SLR model in (6.1) and (6.2). To study the potential leverage of the employment rate values, we calculate the h_{ii}s in (6.24). In **R**, employ the `hatvalues()` function. Combined with **R**'s logical indexing capability – i.e., using square brackets to identify selected elements of an array or object – we can isolate those observations with high-leverage predictors. For these data, $4/n = 4/342 = 0.0117$, and we find 30 predictors in the sample that exhibit high leverage (output edited):

```
> hii <- hatvalues( lm(Y ~ x) )
> p <- length( coef(lm(Y ~ x)) ) - 1
> n <- length(x)
```

```
> cbind( x[ hii > 2*(p+1)/n ], hii[ hii > 2*(p=1)/n ] )
     [,1]       [,2]              [,1]       [,2]
  1  90.3 0.02198052          61  64.2 0.01758973
 12  64.4 0.01711270          64  64.7 0.01641194
 14  64.8 0.01618230          71  64.8 0.01618230
 20  66.9 0.01181531          79  64.4 0.01711270
 25  58.6 0.03414856          85  64.7 0.01641194
 26  66.7 0.01219375          87  64.0 0.01807464
 28  66.6 0.01238593         123  86.4 0.01278888
 30  60.7 0.02721439         184  86.1 0.01220605
 33  63.3 0.01983395         195  86.7 0.01338946
 36  61.3 0.02539291         222  66.6 0.01238593
 44  65.2 0.01528346         225  64.6 0.01664356
 45  62.9 0.02088265         253  66.0 0.01358040
 46  66.7 0.01219375         286  86.1 0.01220605
 54  65.2 0.01528346         307  87.7 0.01551954
 58  62.7 0.02141883         318  87.6 0.01529766
```

The initial (unlabeled column) indices in the above output are the values of i that satisfy the selection criterion $h_{ii} > 2(p+1)/n = 4/n$. The second column under `[,1]` gives the selected x_is, and the third column under `[,2]` lists the corresponding hat values h_{ii}.

Figure 6.9 reproduces the original scatterplot and LS line, now with the high-leverage points highlighted. As expected, high-leverage points lie at the extremes of the predictor range. Notice, however, that many of the point fall generally in line with the overall trend in the data; indeed, the only points that stand out blatantly are the same three LB localities identified by the outlier analysis in Example 6.3.1. (How coincidental this is would require deeper study into the nature of those localities' individual socioeconomic and health factors. Outliers need not be high-leverage points and vice versa.) □

Beyond simple outlier status of Y_i or the leverage potential of x_i, data pairs in an SLR may also be studied for their specific influence on the model fit. Such "influential observations" combine features such as high leverage, separation from the general trend, or other compelling features that may be worthy of identification. A useful comprehensive measure that quantifies the influence of the ith observation on the collection of $\hat{Y}_i$s is known as *Cook's distance* (Cook 1977). The statistic compares every fitted value $\hat{Y}_m$ ($m = 1, \ldots, n$) with its prediction $\hat{Y}_{m[-i]}$ when the ith observation is deleted. Squaring and summing their differences gives a case-deletion measure of discrepancy at each i:

$$\sum_{m=1}^{n} (\hat{Y}_m - \hat{Y}_{m[-i]})^2.$$

Scaling by the product $(p+1)$MSE then produces Cook's distance measure for the ith observation:

$$D_i = \frac{\sum_{m=1}^{n} (\hat{Y}_m - \hat{Y}_{m[-i]})^2}{(p+1)\text{MSE}}.$$

For comparative purposes, D_i is often referred to an F random variable. (Quality of the F-approximation can fluctuate but it suffices for use as a diagnostic tool.) An observation is viewed as "influential" if the probability $P[F(p+1, n-p-1) \leq D_i]$ exceeds a defined cutoff. Recommendations for this threshold vary, but a common theme is that the probability

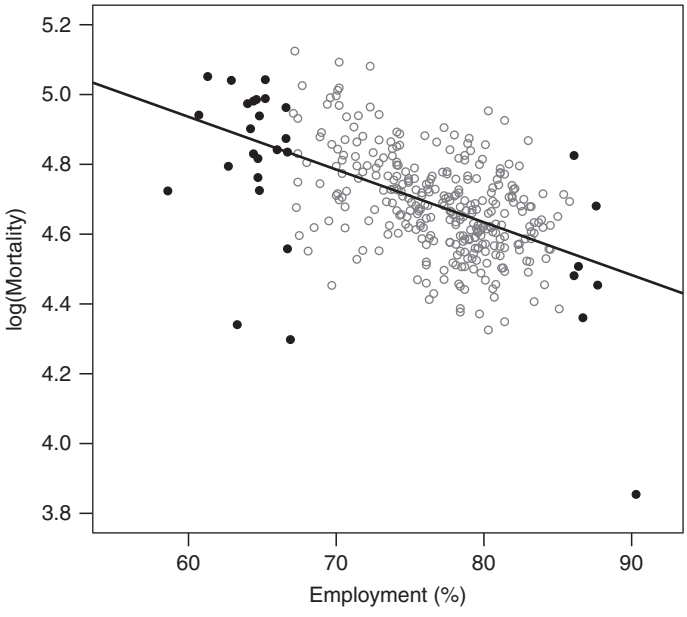

Figure 6.9 Scatterplot for UK cancer mortality data in Example 6.3.2, with high-leverage points identified by solid circles. (Gray circles are points below the leverage threshold.) Least squares regression line $\hat{\mu}(x)$ is overlaid. Source: Data from http://data.gov.uk/dataset/ni_151_-_employment_rate.

should exceed at least $1/2$ for an observation to be influential. For small n, the threshold is sometimes pushed down as low as $1/5$ or even $1/10$. (Recall for the SLR case that $p = 1$.)

Although the D_i measure is based on case deletion, it does not require repeated recalculation of the SLR fit. As with the Studentized deleted residuals, a convenient computing formula is available:

$$D_i = \left(\frac{e_i^2}{[p+1]\text{MSE}} \right) \left(\frac{h_{ii}}{[1-h_{ii}]^2} \right). \tag{6.27}$$

The formula illustrates the distance's features: both e_i and h_{ii} enter into its computation and as either (or both) depart from zero, D_i grows larger. Thus departure from the central trend and/or high leverage will produce a high influence value.

Example 6.3.3 UK cancer mortality and employment (Example 6.2.1, continued). Return to the UK cancer mortality data in Table 6.1, where $Y = \log\{\text{Mortality}\}$ due to cancer was regressed on $x = \{\text{Employment rate}\}$, via the SLR model in (6.1) and (6.2). To study the influence of the various data points using Cook's distance in (6.27), one can compute D_i for each data pair and compare it to an $F(p+1, n-p-1) = F(2, 340)$ distribution. In **R**, this employs the eponymous `cooks.distance()` function.

As with the leverage measure in Example 6.3.2, calculation of every D_i can grow tedious for large data sets, so it is more valuable to identify only those points with high influence. Here, use as the criterion $P[F(2, 340) > D_i] > \frac{1}{2}$. Sample **R** code is

```
> Di = cooks.distance( lm(Y ~ x) )
> which( pf(Di, p+1, n-p-1, lower=T) > .5)
```

(The `which()` function identifies which indices satisfy a logical query. The `pf()` function calculates F probabilities.) Interestingly, this results in *no* observations meeting the high-influence criterion. The largest value of D_i in the data set is $D_1 = 0.2476$, corresponding to the most extreme outlier in Example 6.3.1 (the City of London LB), and also the point farthest along and lowest in Figure 6.1. These extreme-appearing features are not sufficient, however, to flag it – or any other data point – as highly influential under Cook's distance: $P[F(2, 340) \leq 0.2476]$ is only 0.22 and not near the threshold of $\frac{1}{2}$. □

In practice, when influential observations are flagged by any of these regression diagnostics, it is common to repeat the analysis without the questionable data points. Then, examine whether and how the LS estimates and other analytic outcomes change as a result. (Exercise 6.8 applies this strategy to the UK mortality versus employment data in Examples 6.3.1–6.3.3.) Very large differences suggest that the outliers, influential points, and so on may require special attention, or at least a reexamination of how they were sampled, to verify that their inclusion in the analysis is warranted. In the process, further study of these unusual observations may provide for new knowledge discovery.

A variety of other diagnostic tools are available for assessing influence in regression analysis, and the presentation here is intended only as a brief introduction. For more on regression diagnostics, see Kutner et al. (2005, Section 10.4) or the classic texts by Belsley et al. (1980) and Cook and Weisberg (1982).

6.4 Weighted least squares (WLS) regression

An important, and not uncommon, violation of the basic SLR assumptions is heterogeneous variation in the Y_is, that is, departure from the assumption that $\text{Var}[Y_i] = \sigma^2$ is constant. This was illustrated conceptually by the residual diagnostic plots in Figure 6.5. If left unaccounted, differential variation in $\text{Var}[Y_i]$ can detrimentally affect the inferences from an SLR fit.

When heterogeneous variation is present, a common remedial tactic is to transform the original responses. As noted in Section 3.4.3, moving from Y_i to $g(Y_i)$ will change the distributional properties of the observations, a possible consequence of which can be reversion to homogeneous variation (at least to a good approximation). Of course, this may also change the underlying parent distribution of the Y_is: data that began as normal or approximately normal may be driven away from this status, creating another model violation. (In large samples, normality may still be valid to a good approximation, as per the delta method theorem in Section 2.3.9.) On the other hand, if the variance heterogeneity exists concomitantly *with* nonnormal variation, then the transformation may act to address both problems. The natural logarithm is a popular choice, as seen in Example 6.2.1. The various alternatives in Section 3.4.3 can provide similar remediation along these lines.

When a transformation to homogeneous variation cannot be found, or when the desired variance-stabilizing transformation disrupts normality in the data, the standard remedy is to apply weighted least squares (WLS) from Section 5.2.3. That is, include heterogeneous weights, w_i, in the LS normal equations (6.3) to account for the differential variation. The *weighted normal equations* in the SLR setting are

$$\begin{aligned} \sum_{i=1}^{n} w_i Y_i &= \beta_0 \sum_{i=1}^{n} w_i + \beta_1 \sum_{i=1}^{n} w_i x_i \\ \sum_{i=1}^{n} w_i x_i Y_i &= \beta_0 \sum_{i=1}^{n} w_i x_i + \beta_1 \sum_{i=1}^{n} w_i x_i^2 \,. \end{aligned} \tag{6.28}$$

For the remainder of this section, assume $\text{Var}[Y_i] = \sigma_i^2$ across all $i = 1, \ldots, n$. A propitious choice for w_i in (6.28) weights each observation inversely proportional to its variation, so take $w_i \propto 1/\sigma_i^2$. Thus the more precision – that is, lower variance – an observation exhibits, the higher its weight.

Solving the weighted normal equations for the unknown regression parameters produces WLS estimators

$$\tilde{\beta}_1 = \frac{\sum_{i=1}^n w_i x_i Y_i - w_+^{-1} \sum_{i=1}^n w_i x_i \sum_{j=1}^n w_j Y_j}{\sum_{i=1}^n w_i x_i^2 - w_+^{-1} \left(\sum_{i=1}^n w_i x_i\right)^2} \text{ and}$$

$$\tilde{\beta}_0 = \frac{\sum_{i=1}^n w_i Y_i - \tilde{\beta}_1 \sum_{i=1}^n w_i x_i}{w_+},$$

where $w_+ = \sum_{i=1}^n w_i$.

The standard error of $\tilde{\beta}_1$ is

$$\text{se}[\tilde{\beta}_1] = \frac{\sqrt{\widetilde{\text{MSE}}}}{\sqrt{\sum_{i=1}^n w_i x_i^2 - w_+^{-1} \left(\sum_{i=1}^n w_i x_i\right)^2}},$$

where $\widetilde{\text{MSE}}$ is the weighted mean square $\sum_{i=1}^n w_i (Y_i - \tilde{Y}_i)^2/(n-2)$ and $\tilde{Y}_i = \tilde{\beta}_0 + \tilde{\beta}_1 x_i$ are the WLS fitted values ($i = 1, \ldots, n$). An approximate $1 - \alpha$ confidence interval for β_1 is then

$$\tilde{\beta}_1 \pm t_{\alpha/2}(n-2)\text{se}[\tilde{\beta}_1].$$

When the σ_i^2 values are unknown, the weights must be estimated or calculated under some sensible supposition. For instance, the variance may increase proportionally with some positive function $h(x_i)$. Simple examples include $h(x_i) = x_i^2$ or $h(x_i) = x_i$ for $x_i > 0$. If so, set the weights inversely proportional to variance: $w_i = 1/h(x_i)$ for all i. More complex strategies are also available; see, for example, Kutner et al. (2005, Section 11.1).

Example 6.4.1 Baseball batting averages. Statistical analytics increasingly support strategic planning with professional sports teams. Popular examples such as the motion picture *Moneyball* (http://www.sonypictures.com/movies/moneyball/) and the associated book (Lewis 2003) advertise and encourage this phenomenon. Indeed, North American Major League Baseball is a heavy user of "sabermetrics" – the use of statistical methods to analyze and predict baseball player and team performance (http://sabr.org/sabermetrics) – as *Moneyball* helped to illustrate.

For example, Friendly (1991, Section A.2) discusses data on performance of $n = 322$ Major League Baseball players, not including pitchers, after the 1986 season. Included among the data are the players' career batting averages, along with the number of years they had played in the major leagues. One might question whether increased tenure in the major leagues leads to a higher batting average. To study this, let $Y = \{\text{Career batting average} \times 1000\}$ and $x = \{\text{Years played}\}$. Table 6.2 presents a selection of the original data. (The complete set is available at http://www.wiley.com/go/piegorsch/data_analytics. The larger database is also available as part of the external *vcd* package in **R**.)

Conditioning the analysis on the observed pattern of years played, a scatterplot of the data (Figure 6.10a) indicates a clear upward trend. The raw residuals, $Y_i - \hat{Y}_i$, from the unweighted SLR fit suggest a pattern of decreasing variance with increasing response, however (Figure 6.10b). To adjust for this, consider a WLS–SLR fit of these data.

Table 6.2 Selected data from a larger set of $n = 322$ observations on Major League Baseball players' career batting performance through 1986 season (see text for details).

Player code	Years played	Hits	At bats	Batting average
AN	2	42	214	0.196
AT1	10	1300	4631	0.281
⋮	⋮	⋮	⋮	⋮
WW	11	1457	4908	0.297

Source: Friendly (1991).

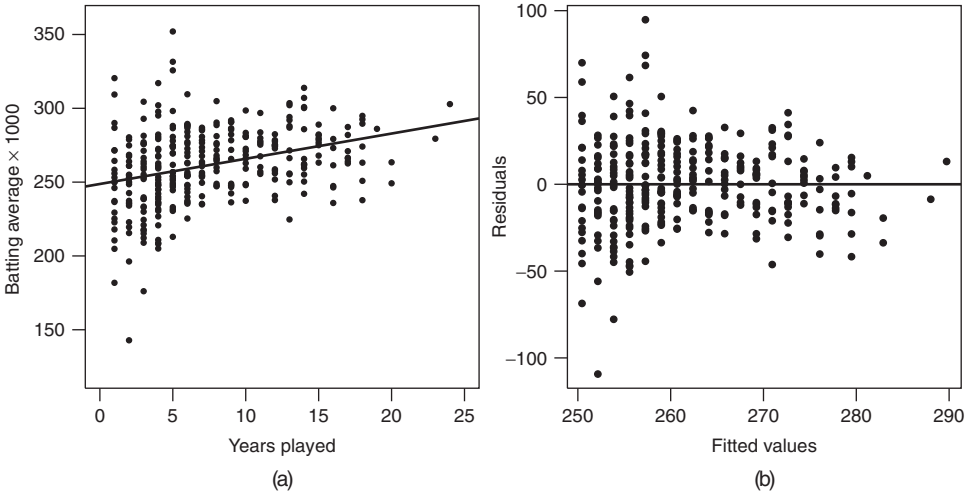

Figure 6.10 (a) Scatterplot for baseball batting average data in Example 6.4.1 with unweighted LS regression line overlaid and (b) raw residual plot from unweighted LS regression. Source: Data from http://vincentarelbundock.github.io/Rdatasets/doc/vcd/Baseball.html. Graphic adapted from Friendly (1991).

A plausible assumption to make on the heterogeneous variation here is that the variance is inversely proportional to the (positive) number of years played, that is, $\text{Var}[Y_i] \propto 1/x_i$. This leads to weights of the form $w_i \propto x_i$. The following are the sample **R** code; note the use of the `weights=` option in the `lm` command:

```
> Y <- 1000*hits/atbat; x <- years
> baseballW.lm <- lm( Y ~ x, weights = x )
> summary( baseballW.lm )
> confint( baseballW.lm )
```

This produces (output edited)

```
Call:
     lm(formula = Y ~ x, weights = x)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  253.7371     2.7194  93.308  < 2e-16
 x             1.2405     0.2295   5.405 1.27e-07
```

and

```
                  2.5 %      97.5 %
(Intercept)  248.3870012  259.087166
  x            0.7889613    1.692028
```

where, in particular, an approximate (pointwise) 95% confidence interval on the slope parameter is given by $0.7890 \leq \beta_1 \leq 1.6920$. Besides the implication that a significant relationship appears between x and Y, the interval indicates that every additional year of experience yields on an average an increase of between 0.8 and 1.7 points in batting average for these professional baseball players.

In passing, it is worth speculating on possible factors that underlie this analysis. A natural conclusion is that a player's skill in batting improves as he become more mature, and the data seem to support this. Other factors may also be at work, however. For instance, is there a selective effect as careers develop? Figure 6.10a shows fewer observations for larger values of x, particularly past $x \geq 19$ years or so. Players whose batting skills do not advance sufficiently as they age – and, for example, as they negotiate for greater salary and benefits – may face release from team owners who insist on increased batting performance. Indeed, the figure also shows that few players exhibit batting averages below 0.200 – a minimum level to perform at the major-league level – and of these, all are young players with no more than $x = 3$ years of major-league experience. Failure to improve low batting averages may contribute to limited tenure in baseball's major leagues.

Similar statistics are now actively collected and explored by baseball sabermetricians, leading to further knowledge discoveries regarding players' performance. Examples include, but are not limited to, Kvam (2011) and Piette and Jensen (2012).

Exercises 6.10 and 7.16a explore further aspects of the WLS fit with these data. □

6.5 Correlation analysis

The SLR model is best suited for assessing the (linear) aspects of the mean response, $E[Y]$, and predicting its value(s) from the explanatory x-variable. When the analytic focus is not one of predictor-versus-response, however, a different model may be appropriate. For instance, if interest exists in simply evaluating the *association* between two paired random variables X and Y, data analysts often study the (linear) *correlation* between the two variables. The analysis is more in keeping with the approach in Section 3.3.3 than with the regression focus of this chapter; however, the similarities between two model formulations make it useful to present further details on correlation analysis here.

6.5.1 The correlation coefficient

It is important to emphasize that the underlying model for bivariate correlation analysis differs technically from the SLR structure in (6.1). Formally, the paired observations (X_i, Y_i), $i = 1, \ldots, n$, are viewed as realizations of a bivariate random vector with joint probability

density function $f_{X,Y}(x, y)$. Interest centers on estimation and inferences for the correlation between X and Y,

$$\rho = \frac{\text{Cov}[X, Y]}{\sigma_X \sigma_Y} ,$$

where σ_X and σ_Y are the population standard deviations of X and Y, respectively. The specific form taken for $f_{X,Y}$ is usually the bivariate normal from (2.39). Then, the MLE of ρ is precisely the product-moment correlation coefficient from Section 3.3.3:

$$r_{XY} = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2 \sum_{i=1}^{n}(Y_i - \overline{Y})^2}} . \tag{6.29}$$

As discussed in Chapter 3, $-1 \leq r_{XY} \leq 1$ and the statistic measures *linear* association between the paired variables. When $r_{XY} = 0$, the data suggest no apparent relationship between X_i and Y_i, while as $|r_{XY}| \to 1$, a linear relationship between them is indicated. Directionality (positive or negative association) is taken from the sign of r_{XY}.

A natural hypothesis to test under this bivariate normal model is whether H_o: $\rho = 0$ versus either two-sided (H_a: $\rho \neq 0$) or one-sided departures (e.g., H_a: $\rho > 0$ for positive association), depending on any a priori subject-matter input. To do so, construct the t-statistic

$$T = \frac{r_{XY}\sqrt{n-2}}{\sqrt{1 - r_{XY}^2}} . \tag{6.30}$$

Under H_o: $\rho = 0$, $T \sim t(n-2)$. Reject H_o in favor of H_a: $\rho \neq 0$ when $|T| \geq t_{\alpha/2}(n-2)$. The corresponding P-value is $2P[t(n-2) \geq |t_{\text{calc}}|]$, where t_{calc} is the observed value of the statistic in (6.30). One-sided tests are similar.

Example 6.5.1 **College admissions (Example 3.3.6, continued).** Recall the paired college admissions data in Example 3.3.6 relating high school class rank (%) to reported ACT score from a set of $n = 705$ admissions files at a large US public university.

After transformation of the class ranks via a logit function, the correlation between $X = \text{logit}\{\text{Class rank}\}$ and $Y = \{\text{ACT score}\}$ was found in Example 3.4.4 as $r_{XY} = 0.4597$. For example, use `cor(log(rank/(100-rank),ACT)` in **R**. It was implied that an observed correlation near 0.45 represented a moderate level of positive association. To formalize this, assume the bivariate normal model holds for the (X, Y) pairs, and consider testing H_o: $\rho = 0$. For the alternative, one assumes these two admissions measures would correlate positively, if at all, so set H_a: $\rho > 0$. Let $\alpha = 0.01$.

Given $r_{XY} = 0.4597$, the test statistic in (6.30) calculates to

$$t_{\text{calc}} = \frac{0.4597\sqrt{703}}{\sqrt{1 - 0.4597^2}} = 13.7247.$$

Using the greater accuracy available in, for example, **R**, a more-precise value is $t_{\text{calc}} = 13.7240$; see the following output. The corresponding one-sided P-value is $P = P[t(703) > 13.7240]$, which is vanishingly small. (In **R**, `pt(13.7240,df=703, lower=F)` gives 1.89×10^{-38}.) Thus P is far less than α and we reject H_o; conclude that a significantly positive correlation is indeed evidenced between these two admissions measures (after suitable transformation of the ranks).

The complete analysis is available in **R** via the `cor.test()` function (output edited):

```
> X = log( rank/(100-rank) )
> cor.test( x=X, y=ACT, alternative ='greater' )
          Pearson's product-moment correlation
  data:  X and ACT
  t = 13.7240, df = 703, p-value < 2.2e-16
  alternative hypothesis: true correlation is greater than 0
  sample estimates:
        cor
  0.4596824
```

□

Confidence intervals for ρ require additional effort, as the unknown parameter does not appear explicitly in the t-statistic from (6.30). The established approach, due to Fisher (1921), applies a transformation to the observed correlation:

$$Z = \frac{1}{2}\log\left(\frac{1+r_{XY}}{1-r_{XY}}\right), \tag{6.31}$$

which is an alternative expression for the inverse hyperbolic tangent of r_{XY}: $Z = \tanh^{-1}(r_{XY})$. Fisher used this to find

$$Z \dot{\sim} \mathrm{N}\left(\zeta, \frac{1}{n-3}\right),$$

where $\zeta = \tanh^{-1}(\rho)$. (Fisher determined that the approximation is very accurate, even for sample sizes as small as $n = 8$.) From this, an approximate $1-\alpha$ confidence interval on ζ is simply

$$Z_L = Z - \frac{z_{\alpha/2}}{\sqrt{n-3}} < \zeta < Z + \frac{z_{\alpha/2}}{\sqrt{n-3}} = Z_U,$$

where $z_{\alpha/2}$ is the upper-$\frac{\alpha}{2}$ critical point from the standard normal distribution. Applying the inverse transformation

$$\rho = \tanh(Z) = \frac{\mathrm{e}^{2Z}-1}{\mathrm{e}^{2Z}+1}$$

to the limits Z_L and Z_U produces an approximate $1-\alpha$ confidence interval on ρ:

$$\frac{\mathrm{e}^{2Z_L}-1}{\mathrm{e}^{2Z_L}+1} < \rho < \frac{\mathrm{e}^{2Z_U}-1}{\mathrm{e}^{2Z_U}+1}.$$

Example 6.5.2 College admissions (Example 6.5.1, continued). Continuing with the paired college admissions data, recall that the observed correlation between $X = \mathrm{logit}\{\text{Class rank}\}$ and $Y = \{\text{ACT score}\}$ was found to be $r_{XY} = 0.4597$. For a 95% confidence interval on the true correlation ρ under the bivariate normal model, apply Fisher's Z-transformation from (6.31):

$$Z = \frac{1}{2}\log\left(\frac{1+0.4597}{1-0.4597}\right) = \frac{1}{2}\log(2.7015) = 0.4969.$$

With this, the upper and lower Z-transformed 95% limits are

$$Z_L = 0.4969 - \frac{z_{0.025}}{\sqrt{705-3}} = 0.4229$$

and

$$Z_U = 0.4969 + \frac{z_{0.025}}{\sqrt{705-3}} = 0.5709.$$

Next, apply the inverse Z-transform (the hyperbolic tangent) to each limit to achieve the final 95% limits on ρ:

$$\frac{e^{(2)(0.4229)}-1}{e^{(2)(0.4229)}+1} = 0.3994 < \rho < 0.5160 = \frac{e^{(2)(0.5709)}-1}{e^{(2)(0.5709)}+1}.$$

We again see that a clear positive correlation exists between the two admission measures.

These confidence limits are available in **R** as part of the `cor.test()` output (previously suppressed), under "`95 percent confidence interval`:"

```
> X = log( rank/(100-rank) )
> cor.test( x=X, y=ACT, alternative='two-sided' )
          Pearson's product-moment correlation
  data:  X and ACT
  t = 13.724, df = 703, p-value < 2.2e-16
  95 percent confidence interval:
   0.3993997 0.5160072
  sample estimates:
        cor
  0.4596824
```

A variety of external **R** packages also perform Fisher's Z-transformation and the associated correlation analysis, including the `CIr()` function in the *psychometric* package and the `fisherz()` suite in the *psych* package. □

6.5.2 Rank correlation

When the assumption of bivariate normality for the random pair (X, Y) is questionable or untenable, a *rank-based* alternative exists for testing H_o: No association between the two variables. Known as *Spearman's rank correlation* (Spearman 1904b), the method essentially replaces the observations with their ranks (largest to smallest, within each variable) and then computes the product moment correlation (6.29) on these ranks. The resulting rank correlation, r_S, can then be assessed against a reference distribution based on how the ranks should permute if no correlation is present, that is, under H_o (Kvam and Vidakovic 2007, Section 7.3). The calculations are straightforward, if tedious, and most analysts employ the computer for estimation and testing. For example, in **R**, simply apply the `cor()` function with the `method="spearman"` option.

The Spearman rank test can suffer if many ties exist among the observations, which is not uncommon with the large data sets seen in modern data analytics. To compensate, a t-approximation is available if the sample size exceeds about 10: simply calculate the rank correlation r_S and apply it in the t-statistic from (6.30):

$$T_S = \frac{r_S\sqrt{n-2}}{\sqrt{1-r_S^2}}\,. \tag{6.32}$$

Under H_o, $T_S \dot{\sim} t(n-2)$. Reject H_o in favor of some association when $|T_S| \geq t_{\alpha/2}(n-2)$. The corresponding P-value is $2P[t(n-2) \geq |t_{S\text{calc}}|]$, where $t_{S\text{calc}}$ is the observed value of the statistic in (6.32). One-sided tests are similar.

Example 6.5.3 College admissions (Example 6.5.1, continued). Continuing with the paired college admissions data from Table 3.2, recall that concern over a large skew led to consideration of a logit transformation of the $n = 705$ original class ranks, because the skew would affect the quality of the bivariate normal-based analysis. Moving to Spearman's rank correlation can assuage this concern, however. So, consider estimating the rank correlation between the original variables $X = \{\text{Class rank}\}$ and $Y = \{\text{ACT score}\}$. This is quickly accomplished in **R**:

```
> cor( x=class.rank, y=ACT, method='spearman' )
 [1] 0.4392875
```

We find $r_S = 0.4393$, which is, again, a moderate level of positive association.

To test for significant, one-sided, positive association, set $\alpha = 0.01$. The Spearman rank test is available as a part of the `cor.test()` output after including the `method="spearman"` option (and, because this large data set contains many tied ranks, the `exact=FALSE` option to institute the large-sample t-approximation):

```
> cor.test( x=class.rank, y=ACT, method='spearman',
                            alternative ='greater', exact=FALSE )
          Spearman's rank correlation rho
  data:  class.rank and ACT
  S = 32745789, p-value < 2.2e-16
  alternative hypothesis: true rho is greater than 0
  sample estimates:
        rho
  0.4392875
```

Notice that the test statistic is given here as `S = 32745789`, which is the (very large!) Spearman rank-based test statistic (the sum of the squared differences in the ranks). As the `exact=FALSE` option was included, however, the P-value is instead based on the t-approximation. Moving to the `exact=TRUE` option would produce the same test statistic but would incur the **R** warning "`Cannot compute exact p-values with ties.`"

From the **R** output, the one-sided P-value is again vanishingly small: $P < 2.2 \times 10^{-16}$, which best reported as simply $P < 0.0001$. A strongly significant, positive correlation is evidenced between the two (original) variables. □

Exercises

6.1 Return to the SLR model from (6.1) and (6.2).

(a) Verify the forms of the normal equations in (6.3) by differentiating the LS objective quantity $\mathcal{D} = \sum_{i=1}^{n} \{Y_i - (\beta_0 + \beta_1 x_i)\}^2$ with respect to both β_0 and β_1.

(b) Show that the solution of the LS normal equations in (6.3) produces the point estimators $\hat{\beta}_0$ and $\hat{\beta}_1$ in (6.4). (*Hint*: Show that $\sum_{i=1}^{n}(x_i - \bar{x})^2 = \sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2$.)

(c) Construct the log-likelihood function (see Section 5.2.4) for β_0 and β_1 under the SLR model.

(d) Show that by jointly maximizing the log-likelihood, the MLEs for β_0 and β_1 are identical to the LS estimators derived in Exercise 6.1.

6.2 Show that for the SLR model in (6.1) and (6.2), the LS estimators from (6.4) possess the following sampling qualities.

(a) Both estimators are unbiased, that is, $E[\hat{\beta}_j] = \beta_j$ for $j = 0, 1$. (*Hint*: Show that $\hat{\beta}_0$ and $\hat{\beta}_1$ can each be written in the form $\sum_{i=1}^{n} \kappa_i Y_i$ for some constants κ_i, and then take advantage of features for sums of normally distributed random variables discussed in Section 2.3.9. Use this result as necessary throughout the exercise.)

(b) Verify the forms of the sampling variances in (6.6).

(c) $\text{Cov}[\hat{\beta}_0, \hat{\beta}_1] = -\sigma^2 \overline{x} / \sum_{i=1}^{n} (x_i - \overline{x})^2$.

(d) Both estimators are normally distributed: $\hat{\beta}_j \sim \text{N}\left(\beta_j, \text{Var}[\hat{\beta}_j]\right), j = 0, 1$, where the $\text{Var}[\hat{\beta}_j]$ quantities are given by (6.6).

(e) $\hat{\mu}(x) \sim \text{N}\left(\beta_0 + \beta_1 x, \text{Var}[\hat{\mu}(x)]\right)$ for any x, where $\text{Var}[\hat{\mu}(x)]$ is given by (6.8). (*Hint*: As above, show that $\hat{\mu}(x)$ can be written in the form $\sum_{i=1}^{n} \kappa_i Y_i$ for some constants κ_i, and then take advantage of features for sums of normally distributed random variables discussed in Section 2.3.9.)

6.3 Show that for the SLR model in (6.1) and (6.2), the fitted values $\hat{Y}_i$ from (6.5) and the residuals e_i from (6.9) satisfy the following relationships. (*Hint*: Where necessary, appeal to the relationships defined by the normal equations from (6.3).)

(a) $\sum_{i=1}^{n} e_i = 0$ and from this, $\sum_{i=1}^{n} Y_i = \sum_{i=1}^{n} \hat{Y}_i$.

(b) $\sum_{i=1}^{n} x_i e_i = 0$.

(c) $\sum_{i=1}^{n} \hat{Y}_i e_i = 0$.

(d) $E[e_i] = 0$.

6.4 For the SLR model in (6.1) and (6.2), are the LS estimators from (6.4) necessarily independent (in the statistical sense, as per Section 2.1.2)? Why or why not? If not, is there a feature or condition that ensures independence?

6.5 Use the normality of $\hat{\beta}_1$ from (6.7), the χ^2 feature of the scaled MSE from (6.11), and the independence between these two random variables to establish that $T = (\hat{\beta}_1 - \beta_1) / se[\hat{\beta}_1] \sim t(n-2)$. With this, manipulate the relationship $P[-t_{\alpha/2}(n-2) < T < t_{\alpha/2}(n-2)] = 1 - \alpha$ into the two-sided confidence interval in (6.14). (*Hint*: Imitate the operations in (5.17).) Can you achieve similar results for $\hat{\beta}_0$?

6.6 Vise (2012) reported data on business characteristics of $n = 250$ US trucking and delivery companies for calendar year 2011. Among the characteristics studied were the number of drivers and the number of power units each company employed. The data are available online at http://www.wiley.com/go/piegorsch/data_analytics; a sample follows:

Drivers:	102 000	125 497	24 335	···	392	386
Power units:	98 908	52 287	15 602	···	386	390

Examine these data as follows:

(a) Consider the application of the SLR model in (6.1) and (6.2). As the variables are counts, and because they range over many degrees of magnitude, it is natural to employ some form of transformation. Here, apply a logarithmic transform to both variables and plot $Y = \log\{\text{Units}\}$ versus $x = \log\{\text{Drivers}\}$. Does the resulting relationship appear linear? What is the coefficient of determination (R^2) with the transformed data?

(b) Regress $Y = \log\{\text{Units}\}$ on $x = \log\{\text{Drivers}\}$ via the SLR model. Use the results to test whether an increase in (log-)number of drivers increases the mean (log-)number of power units. Operate at $\alpha = 0.01$.

(c) Under the SLR model, construct a 90% confidence region for β_1.

(d) Under the SLR model, construct a joint 90% confidence region for β_0 and β_1. How does your inference on β_1 compare with the pointwise inference in Exercise 6.6c?

(e) Under the SLR model, construct and plot a 90% simultaneous WHS confidence band on the mean response for all x.

(f) Find the raw residuals e_i from the SLR fit in Exercise 6.6b, and plot them against x_i. Also construct a normal quantile plot of the e_is. Do any untoward patterns emerge?

(g) Calculate the Studentized deleted residuals t_i from the SLR fit in Exercise 6.6b and plot these against x_i. Do any points appear to be potential outliers at $\alpha = 0.10$?

(h) Find the diagonal elements h_{ii} ($i = 1, \ldots, n$) from the hat matrix under the SLR fit in Exercise 6.6b. Use these to identify if any of the x_is appear to be high leverage points.

(i) Find Cook's distance D_i ($i = 1, \ldots, n$) using (6.27), and identify if any data pairs appear influential, using the criterion $P[F(p+1, n-p-1) \leq D_i] > \frac{1}{2}$ with $p = 1$.

(j) If a new company with 530 drivers were to be formed, how many power units would you expect the company would require, based on the SLR fit in Exercise 6.6? Also give a 99% confidence interval for this value.

6.7 Return to the UK cancer mortality data in Table 6.1, and construct a set of multiplicity-adjusted confidence bounds on the mean log(Mortality) response at the following $H = 5$ values of $x = \{\text{Employment rate}\}$: $x_h = 50, 60, 70, 80, 90\%$. Operate at a familywise confidence level of 90%.

6.8 Return to the UK cancer mortality data in Table 6.1, and perform the following diagnostic operations.

(a) Calculate the raw residuals e_i and the Studentized deleted residuals t_i from the SLR fit. Plot both against (i) the fitted values $\hat{Y}_i$ and (ii) the predictor values x_i.

How do the patterns differ among these various plots? For the raw residual plots in particular, would the outlying features discussed in Example 6.3.1 be as evident?

(b) Supplement the normal quantile plot in Figure 6.7 by constructing a boxplot and a histogram of the raw residuals (inserting rug plots at bottom). What patterns emerge? Do these corroborate the indications in Example 6.3.1?

(c) Remove the three potential outliers identified in Example 6.3.1, and fit a new SLR model to the remaining 339 paired observations. Calculate the new LS estimates for β_0 and β_1. Plot the remaining points and overlay the new LS line. How do the results differ from those seen using the full 342-point data set? Repeat the effort by removing only the most extreme potential outlier (the City of London LB). Do your calculations bring into question any of the three points?

(d) Remove the 30 leverage points identified in Example 6.3.2, and fit a new SLR model to the remaining 312 paired observations. Calculate the new LS estimates for β_0 and β_1. Plot the remaining points and overlay the new LS line. How do the results differ from those seen using the full 342-point data set?

6.9 A famous exercise in data mining involves airline on-time performance for US commercial flights (see http://stat-computing.org/dataexpo/2009/). The data set is massive, with almost 120 million records; for illustration purposes, consider here only data on flights reporting actual arrival delays (positive times, in minutes) from calendar year 2008, and only over distances less than 3000 miles. These arrival delay times exhibit a right skew, so take the response variable as $Y = \log\{\text{Arrival delay}\}$ and regress this against $x = \{\text{Flight distance}\}$. The data are available online at http://www.wiley.com/go/piegorsch/data_analytics; a sample is as follows:

Distance (miles):	810	515	515	$\cdots$	215	533	533
Arrival delay (min):	2	14	34	$\cdots$	2	14	9

Examine these data as follows:

(a) Even when reduced as described above, the sample size here is almost 3×10^6. To simplify the exercise, select from the data set a random sample of $n = 2000$ data pairs. Calculate $Y = \log\{\text{Arrival delay}\}$ and plot this against $x = \{\text{Flight distance}\}$. What pattern emerges? (In **R**, use the `sample()` function with the `replace=F` option. If you cannot download or access the original file, download a sample file with 2000 randomly selected data pairs at http://www.wiley.com/go/piegorsch/data_analytics. Note that the sample file has already applied the logarithmic transform to the delay times.)

(b) Regress Y on x via the SLR model in (6.1) and (6.2). Test whether increasing flight distance increases the mean (log-)arrival delay. Operate at $\alpha = 0.05$.

(c) Under the SLR model, construct a 90% confidence region for β_1.

(d) Under the SLR model, construct and plot a 90% simultaneous WHS confidence band on the mean response for all x.

(e) Find the raw residuals e_i from the SLR fit in Exercise 6.9b and plot them against the fitted values $\hat{Y}_i$. What patterns emerge?

(f) Calculate the Studentized deleted residuals t_i from the SLR fit in Exercise 6.9b and plot these against $\hat{Y}_i$. Determine if any points are potential outliers by assessing the Studentized deleted residuals against the exceedance limits $\pm t_{\alpha/(2n)}(n - p - 2)$, where $p = 1$. (Set $\alpha = 0.10$. Can you see any possible problems with these Bonferroni-adjusted limits here?)

(g) Find Cook's distance D_i $(i = 1, \ldots, n)$ using (6.27) and identify if any data pairs appear influential, using the criterion $P[F(p + 1, n - p - 1) \leq D_i] > \frac{1}{2}$ at $p = 1$.

(h) Suppose you were meeting a flight from Miami, FL, to Seattle, WA (distance = 2734 miles), and you were informed the flight's arrival was delayed. How long a delay would you expect to encounter based on the analysis of your sample in Exercise 6.9b? Also give a 99% confidence interval for this value.

(i) Repeat your analysis in Exercise 6.9b with a new random sample of $n = 2000$ data pairs. (If computer resources permit, do so a number of times.) Do you recover the same results, at least qualitatively?

(j) What factors might contribute to model-assumption violations with these data?

6.10 Return to the baseball batting average data from Table 6.2 and consider the following, additional aspects of the WLS analysis for these data.

(a) What is the interpretation of the estimated intercept term, $\hat{\beta}_0$? From the Example, find a 95% confidence interval for the intercept and interpret the results. (Should any adjustment be made to these quantities?)

(b) Recover the residuals, $\tilde{e}_i = Y_i - \tilde{Y}_i$, from the WLS–SLR fit and construct (i) a histogram of the $\tilde{e}_i$s (include a rug plot at bottom) and (ii) a normal quantile plot of the $\tilde{e}_i$s. Do these suggest strong departures from the normal parent assumptions?

(c) Plot the residuals, $\tilde{e}_i$, against x_i. Notice that the pattern of variance heterogeneity has not changed much, which is not unreasonable: the WLS solution is not designed to remove the heterogeneity, merely to adjust the estimators for it.

(d) Find the Studentized deleted residuals from the WLS–SLR fit (in **R**, use `rstudent()`). Plot these against x_i. Determine if any points are potential outliers by assessing the Studentized deleted residuals against the approximate exceedance limits $\pm t_{\alpha/(2n)}(n - p - 2)$, where $p = 1$. Set $\alpha = 0.05$. Do any points appear to be potential outliers?

6.11 In 2013, the online site payscale.com released data on median, annual, full-time earnings (in dollars, to the nearest hundred) among $n = 606$ private colleges and universities in the United States; see http://www.payscale.com/college-salary-report-2013. For each college, reported were $x = \{\text{Starting salaries}\}$ and $Y = \{\text{Mid-career salaries}\}$ of their alumni. ("Starting" employees were defined as having 5 years or less of experience in their field, and "Mid-career" employees were defined as having 10 years or more of experience.) The data values are available online at

http://www.wiley.com/go/piegorsch/data_analytics; a sample is as follows:

x = Starting salary ($):	34 200	30 200	$\cdots$	49 700	58 300
Y = Mid-career salary ($):	42 300	43 400	$\cdots$	111 000	137 000

Analyze these data as follows:

(a) Plot Y against x. Does the relationship appear linear? Do any features of the plot stand out?

(b) Conditioning the analysis on the observed pattern of starting salaries, regress Y on x via the SLR model in (6.1) and (6.2). Find the raw residuals $e_i = Y_i - \hat{Y}_i$ from the SLR fit and plot them against x_i. What pattern emerges?

(c) Apply a WLS SLR fit to these data. For your weights, set $w_i = 1/x_i$. Use the results to test if increasing one's starting salary also increases the associated mid-career salary, on average. Operate at $\alpha = 0.05$.

(d) Estimate the mean increase in annual mid-career salary associated with a $1000 rise in annual starting salary for these private-college graduates, based on your WLS–SLR fit in Exercise 6.11c. Also give an approximate, 95% confidence interval for this value.

6.12 In a review of annual operating characteristics of US natural gas utilities for calendar year 2011, data on $n = 245$ natural gas suppliers were collected (Anonymous 2012). Among the features recorded were $Y = \{\text{Miles of gas mains}\}$ and $x = \{\text{Numbers of customers}\}$ for each company. The data are available online at http://www.wiley.com/go/piegorsch/data_analytics; the following is a sample:

x = Customers:	5 801 000	4 461 363	4 295 741	$\cdots$	2183	1077
Y = Mains (miles):	497 738	80 159	6567	$\cdots$	100	48

Examine these data as follows:

(a) Plot Y against x. Does the relationship appear linear? Do any features of the plot stand out?

(b) Regress Y on x via the SLR model in (6.1) and (6.2). Find the raw residuals $e_i = Y_i - \hat{Y}_i$ from the SLR fit and plot them against x_i. What pattern emerges?

(c) Apply a WLS–SLR fit to these data. For your weights, set $w_i = 1/x_i$. Use the results to test if increasing the number of customers increases mean miles of gas mains. Operate at $\alpha = 0.01$.

(d) If two other companies, the first with $x = 20\,000$ customers and the second with $x = 100\,000$ customers, were to be studied, how many miles of gas mains would you expect each company would require, based on your WLS–SLR fit in

Exercise 6.12c? Also give multiplicity-adjusted, approximate, 99% confidence intervals for these values.

6.13 Under the bivariate normal model in Section 6.5, find the conditional p.d.f. of $Y \mid X = x$ (see Section 2.1.2), and show that this is $Y \mid X = x \sim \mathrm{N}(\omega + \psi x, \tau^2)$, with $\omega = \mu_Y - \mu_X\psi$, $\psi = \rho\sigma_Y/\sigma_X$, $\tau^2 = \sigma_Y^2(1-\rho^2)$, and where μ_X, μ_Y and σ_X, σ_Y are the population means and standard deviations, respectively, and ρ is the correlation of X and Y.

6.14 As part of an environmental management effort, the UK records and archives data on residual waste collected per household for $n = 383$ nationwide communities (http://data.gov.uk/dataset/ni_151_-_employment_rate). Reported are the paired variables $X = \{\text{Collected waste for calendar year 2008}\}$ and $Y = \{\text{Collected waste for calendar year 2009}\}$ (both in kg/household) for each community. The data are available online at http://www.wiley.com/go/piegorsch/data_analytics; the following is a sample:

2008 waste:	1006	995	992	$\cdots$	386	361	345
2009 waste:	991	835	857	$\cdots$	346	387	340

To assess potential association between the patterns of generated waste between the years 2008 and 2009, examine these data as follows:

(a) Plot the paired data. Is any pattern apparent?

(b) Calculate the correlation coefficient (6.29) between the 2008 and 2009 data. What does it suggest? Is this surprising?

(c) Conduct a test of whether or not the corresponding population correlation coefficient under the bivariate normal model is zero. Operate at $\alpha = 0.01$.

(d) Calculate a 99% confidence interval for the population correlation coefficient under the bivariate normal model. What does the interval indicate?

6.15 From an ecological study of longleaf pine (*Pinus palustris*) characteristics, Chen et al. (2004, Section 8.6) discussed data from a sample of $n = 396$ trees in a North American forest. Recoded were the paired variables $X = \{\text{Diameter at breast height (in cm)}\}$ and $Y = \{\text{Height (in ft)}\}$ for each tree. The data are available online at http://www.wiley.com/go/piegorsch/data_analytics; The following is a sample:

Diameters (cm):	15.9	22.0	56.9	$\cdots$	6.0	3.8	8.0
Heights (ft):	28.0	26.0	119.0	$\cdots$	12.0	3.5	9.0

(a) Examine the original variables X and Y to determine if they appear to exhibit normal-distribution features. Plot histograms (include rug plots) and construct normal quantile plots for each variable. Do either or both appear normal?

(b) Calculate the Spearman rank correlation, r_S, for these data. One might expect that diameter and height of the tree are positively associated, so use r_S to test this proposal with these data. Operate at $\alpha = 0.01$.

7

Techniques for supervised learning: multiple linear regression

7.1 Multiple linear regression

The simple linear regression (SLR) model in (6.1) and (6.2) can be extended to accommodate multiple predictor variables, creating a *multiple linear regression* (MLR) model. The response variable Y_i remains normally distributed, with

$$Y_i \sim \text{indep. } \mathrm{N}(\mu_i, \sigma^2), \tag{7.1}$$

but now the mean response $E[Y_i] = \mu_i$ is expanded into a function of p fixed predictor variables x_{ij} $(j = 1, \ldots, p)$:

$$\mu_i = \mu(x_{i1}, \ldots, x_{ip}) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}. \tag{7.2}$$

Here, μ_i is modeled as a *linear predictor*, that is, a linear combination of the p predictor variables along with a constant 'intercept' term β_0. For the latter, we often view its corresponding 'predictor' as the constant regressor $x_{i0} = 1$.

The β_js now represent $p + 1$ unknown regression coefficients. For $j \geq 1$, each β_j is interpreted as the change in $E[Y_i]$ for a unit increase in the corresponding x_{ij} – the 'slope' of the jth predictor – assuming all the other x-variables are held fixed. (If it is not possible to vary one predictor while holding all others constant, this interpretation may not make sense. An example occurs when the x_{ij}s are taken as polynomial terms: $x_{ij} = x_i^j$. See Section 7.2.) As previously, $\sigma^2 = \mathrm{Var}[Y_i]$ is the unknown, constant variance term.

Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery,
First Edition. Walter W. Piegorsch.

www.wiley.com/go/piegorsch/data_analytics

7.1.1 Matrix formulation

Some simplification in the description of the MLR model is available by moving to vector and matrix notation, similar to that presented in Section 2.2. (See Appendix A for a review of vector and matrix terminology.) The Y_is are collected into the $n \times 1$ *response vector* $\mathbf{Y} = [Y_1 \cdots Y_n]^{\mathrm{T}}$, while the $p + 1$ predictor values for each ith observation are placed into individual $1 \times (p + 1)$ *row* vectors $\mathbf{x}_i = [1 \;\; x_{i1} \;\; x_{i2} \;\; \cdots \;\; x_{ip}]$ $(i = 1, \ldots, n)$. (Notice inclusion of a leading constant, $x_{i0} = 1$, to represent the intercept term.) These are collected into the *design matrix*

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix}. \tag{7.3}$$

The use of matrix notation also simplifies the larger model formulation. Define the expectation operator so that it applies to a vector componentwise, that is, $E[\mathbf{Y}] = [E(Y_1) \cdots E(Y_n)]^{\mathrm{T}}$. Similarly, define the covariance matrix of $\mathbf{Y}$ as $\mathrm{Var}[\mathbf{Y}]$ from (2.12). Note that this provides $E[\mathbf{AY}] = \mathbf{A}E[\mathbf{Y}]$ and $\mathrm{Var}[\mathbf{AY}] = \mathbf{A}\mathrm{Var}[\mathbf{Y}]\mathbf{A}^{\mathrm{T}}$ for any conformable, nonstochastic matrix $\mathbf{A}$.

Using this matrix notation, $E[\mathbf{Y}]$ is simply $[\mu_1 \cdots \mu_n]^{\mathrm{T}}$. Under (7.2), this translates to the matrix expression

$$E[\mathbf{Y}] = \mathbf{X}\boldsymbol{\beta}, \tag{7.4}$$

where $\boldsymbol{\beta} = [\beta_0 \;\; \beta_1 \cdots \;\; \beta_p]^{\mathrm{T}}$ is the $(p + 1)$-vector of unknown regression coefficients. Each element of the expectation vector in (7.4) is simply the mean response (7.2) at the ith observation. In vector notation, these can be expressed as the linear combination

$$\mu(\mathbf{x}_i) = \mathbf{x}_i\boldsymbol{\beta}.$$

Also, for the MLR model in (7.1), $\mathrm{Var}[\mathbf{Y}]$ is simply $\sigma^2\mathbf{I}$. (Recall that $\mathbf{I} = \mathrm{diag}\{1, 1, \ldots, 1\}$ is the identity matrix from Section A.1.)

Point estimation again proceeds via least squares (LS). The LS estimator $\hat{\boldsymbol{\beta}}$ satisfies the MLR normal equations, which are $(p + 1)$-variate extensions of (6.3):

$$\mathbf{X}^{\mathrm{T}}\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^{\mathrm{T}}\mathbf{Y} \tag{7.5}$$

(Kutner et al. 2005, Section 6.3). When the rank of the $n \times (p + 1)$ design matrix $\mathbf{X}$ is equal to $p + 1$, the $(p + 1) \times (p + 1)$ inverse matrix $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$ is well defined. This is used in the LS solution to (7.5):

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}. \tag{7.6}$$

As in the SLR case, under (7.1), this LS estimator corresponds to the maximum likelihood estimate for $\boldsymbol{\beta}$ and is again 'best linear unbiased' via an extension of the Gauss–Markov theorem from Section 6.2.1. Also, its sampling distribution under the normal parent model is again normal: $\hat{\boldsymbol{\beta}} \sim \mathrm{N}_p(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1})$.

The fitted values $\hat{Y}_i$ are collected into the $n \times 1$ column vector $\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$. Notice, however, that from (7.6), this is

$$\hat{\mathbf{Y}} = \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y},$$

that is, another linear combination of the Y_is (seen also with the SLR case). Write this as $\hat{\mathbf{Y}} = \mathbf{H}\mathbf{Y}$, where the $n \times n$ matrix

$$\mathbf{H} = \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}} \tag{7.7}$$

is called the *hat matrix* of the LS fit. Its diagonal elements are denoted by h_{ii}, explaining the notation in (6.24). The hat matrix possesses many interesting qualities; see Kutner et al. (2005, Section 5.11). For instance, its trace equals the number of β-parameters: $\mathrm{tr}(\mathbf{H}) = p + 1$.

To estimate σ^2, mimic the approach taken in the SLR case and start with the residuals $e_i = Y_i - \hat{Y}_i$. In vector notation, this is $\mathbf{e} = \mathbf{Y} - \hat{\mathbf{Y}}$ for $\mathbf{e} = [e_1 \cdots e_n]^{\mathrm{T}}$. Notice that the residuals may also be written as

$$\mathbf{e} = (\mathbf{I} - \mathbf{H})\mathbf{Y},$$

that is, another linear combination of the Y_is. Next, write the residual (error) sum of squares as $\mathrm{SSE} = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 = \mathbf{e}^{\mathrm{T}}\mathbf{e} = \mathbf{Y}^{\mathrm{T}}(\mathbf{I} - \mathbf{H})\mathbf{Y}$ and divide by the corresponding error degrees of freedom (d.f.), say, df_E, to find the mean squared error (MSE):

$$\mathrm{MSE} = \frac{\mathrm{SSE}}{\mathrm{df_E}} = \frac{\mathbf{e}^{\mathrm{T}}\mathbf{e}}{n - p - 1}$$

(see Exercise 7.7). Notice the change in $\mathrm{df_E}$ from the SLR model: in effect, estimation of each new β_j results in a loss of one additional d.f. for error, so we have gone from $\mathrm{df}_E = n - 2$ to $\mathrm{df}_E = n - p - 1$.

The MSE remains unbiased for estimating the variance parameter: $E[\mathrm{MSE}] = \sigma^2$. Also, it remains statistically independent of $\hat{\boldsymbol{\beta}}$ and χ^2-distributed (after appropriate scaling):

$$\frac{(n - p - 1)\mathrm{MSE}}{\sigma^2} \sim \chi^2(n - p - 1). \tag{7.8}$$

As the covariance matrix of the LS estimator for $\boldsymbol{\beta}$ has the particularly simple form $\mathrm{Var}[\hat{\boldsymbol{\beta}}] = \sigma^2(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$, substituting the MSE for σ^2 produces the estimated covariance matrix

$$\widehat{\mathrm{Var}}[\hat{\boldsymbol{\beta}}] = \mathrm{MSE}\ (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}. \tag{7.9}$$

The jth diagonal element of $\widehat{\mathrm{Var}}[\hat{\boldsymbol{\beta}}]$ is the square of the individual standard error $\mathrm{se}[\hat{\beta}_j]$, while the (j, k)th off-diagonal element is the estimated covariance $\widehat{\mathrm{Cov}}[\hat{\beta}_j, \hat{\beta}_k]$ $(j \neq k)$.

An MLR extension also can be constructed for the coefficient of determination from (6.19). Called the *coefficient of multiple determination*, it has the same structure and form:

$$R^2 = 1 - \frac{\mathrm{SSE}}{\mathrm{SSTo}}\ , \tag{7.10}$$

where $\mathrm{SSTo} = \sum_{i=1}^{n}(Y_i - \overline{Y})^2$. Similar to its SLR counterpart, R^2 here represents the percentage variation in Y_i that is accounted for by variation in the aggregate collection of p predictor variables.

7.1.2 Weighted least squares for the MLR model

If each observation Y_i in the MLR model can be assigned a heterogeneous weight $w_i > 0$, as in Section 6.4, we apply weighted least squares (WLS). In matrix notation, the weights are collected into a diagonal *weight matrix*: $\mathbf{W} = \mathrm{diag}\{w_1, w_2, \ldots, w_n\}$. For known $\mathbf{W}$, the weighted normal equations – generalizing (6.28) – are

$$(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X})\tilde{\boldsymbol{\beta}} = \mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{Y}.$$

The solution (when the inverse exists) is clearly

$$\tilde{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{Y}, \tag{7.11}$$

extending the LS result in (7.6). The estimated covariance matrix is

$$\widehat{\mathrm{Var}}[\tilde{\boldsymbol{\beta}}] = \mathrm{MSE}\ (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X})^{-1}.$$

As in the unweighted case, the WLS estimator $\tilde{\boldsymbol{\beta}}$ corresponds to the maximum likelihood estimator for $\boldsymbol{\beta}$ under a normal error assumption and is again 'best linear unbiased.' Kutner et al. (2005, Section 11.1) give further details on WLS estimation for the MLR model.

7.1.3 Inferences under the MLR model

Similar to the SLR setting, inferences on any individual β_j may be constructed to assess whether the corresponding predictor variable x_j is important in modeling $E[Y_i]$. The statistical features of the LS estimators induce marginal t-distributions for the $\hat{\beta}_j$s:

$$\frac{\hat{\beta}_j - \beta_j}{\mathrm{se}[\hat{\beta}_j]} \sim t(n-p-1), \tag{7.12}$$

where $\mathrm{se}[\hat{\beta}_j]$ is the square root of the jth diagonal element from the estimated covariance matrix in (7.9). From this, a pointwise $1-\alpha$ confidence interval on β_j has the recognizable 'Wald' form

$$\hat{\beta}_j \pm t_{\alpha/2}(n-p-1)\ \mathrm{se}[\hat{\beta}_j], \tag{7.13}$$

for any single $j = 0, \ldots, p$.

For simultaneous confidence statements on a prescribed subset of the β_js, extensions of the Bonferroni approach from Section 6.2.2 may be applied. Suppose the indices comprising the desired subset of β_js are $j_1, j_2, \ldots, j_q$ ($q \le p+1$). To apply a q-fold Bonferroni correction to the pointwise limits in (7.13), simply construct the q joint, minimal, $1-\alpha$ confidence statements

$$\hat{\beta}_j \pm t_{\alpha/(2q)}(n-p-1)\ \mathrm{se}[\hat{\beta}_j], \quad \text{for all } j = j_1, j_2, \ldots, j_q. \tag{7.14}$$

(In effect, this produces a q-dimensional, hyperrectangular, confidence region.)

If the target collection of the β_js cannot be prespecified, so that *post hoc* inference are likely, construct the Bonferroni bounds with $p+1$ replacing q in (7.14) and then select any subset of $q \le p+1$ confidence limits. Alternatively, a joint $1-\alpha$ confidence ellipsoid can be built for the entire $(p+1)$-vector of regression parameters, extending the two-dimensional ellipse in (6.21). This is defined by a matrix inequality similar to (5.49)

$$(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^{\mathrm{T}}\widehat{\mathrm{Var}}[\hat{\boldsymbol{\beta}}]^{-1}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \le (p+1)F_{\alpha}(p+1, n-p-1).$$

Both approaches are conservative and will produce minimal $1-\alpha$ joint confidence bounds on the *post hoc* subset. (The $1-\alpha$ confidence ellipsoid will be exact when $q = p+1$, however.)

For a confidence interval on the mean response in (7.2) at any given *row* vector $\mathbf{x} = [1 \;\; x_1 \;\; x_2 \;\; \cdots \;\; x_p]$, begin with the LS point estimate

$$\hat{\mu}(\mathbf{x}) = \mathbf{x}\hat{\boldsymbol{\beta}} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p.$$

Then, mimic the construction in (6.18) and construct the pointwise $100(1-\alpha)\%$ t-interval

$$\hat{\mu}(\mathbf{x}) \pm t_{\alpha/2}(n-p-1)\ \mathrm{se}[\hat{\mu}(\mathbf{x})], \tag{7.15}$$

where $\mathrm{se}[\hat{\mu}(\mathbf{x})] = \sqrt{(\mathrm{MSE})\mathbf{x}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{x}^{\mathrm{T}}}$. For multiple intervals at a prescribed collection of $H \geq 1$ vectors $\mathbf{x}_h$ $(h = 1, \ldots, H)$, apply a Bonferroni adjustment to (7.15). This gives the joint, minimal, $100(1-\alpha)\%$ hyperrectangle

$$\hat{\mu}(\mathbf{x}_h) \ \pm t_{\alpha/(2Hq)}(n-p-1)\ \mathrm{se}[\hat{\mu}(\mathbf{x}_h)],$$

for all $h = 1, \ldots, H$. Extending this to *all possible* vectors $\mathbf{x}$, the WHS method from Section 6.2.2 gives a simultaneous $1-\alpha$ confidence surface as

$$\hat{\mu}(\mathbf{x}) \pm \sqrt{(p+1)F_{\alpha}(p+1, n-p-1)}\ \mathrm{se}[\hat{\mu}(\mathbf{x})].$$

This WHS construction may also be applied for *post hoc* $1-\alpha$ confidence statements on any subcollection of $\mu(\mathbf{x})$s.

Hypothesis tests on the β_j parameters may be similarly extended from the SLR setting. For testing $\mathrm{H_o}$:$\beta_j = 0$ against $\mathrm{H_a}$: $\beta_j \neq 0$, appeal to the t-distribution relationship in (7.12) and reject $\mathrm{H_o}$ when the test statistic

$$T_j = \frac{\hat{\beta}_j}{\mathrm{se}[\hat{\beta}_j]} \tag{7.16}$$

exceeds the two-sided critical point $t_{\alpha/2}(n-p-1)$ in absolute value. The corresponding P-value is

$$P = 2P[t(n-p-1) \geq |t_{j\mathrm{calc}}|],$$

where $t_{j\mathrm{calc}}$ is the observed value of T_j in (7.16). (One-sided tests are similar.) Note that this tests the significance of the jth predictor variable given that all the other predictor variables are present in the model. This is termed a *partial test* of $\mathrm{H_o}$.

The partial t-test from (7.16) is a pointwise test, in that it does not adjust for any other inferences being performed on the same set of data. For joint, coordinated testing on whole subsets or subcollections of the β-parameters, a general strategy is available. Suppose a null hypothesis fixes a subset of $q \leq p+1$ β_js equal to zero. For convenience, arrange the predictors so that this is the final group of q regression coefficients, that is,

$$\mathrm{H_o}\colon\ \beta_{p-q+1} = \beta_{p-q+2} = \cdots = \beta_p = 0. \tag{7.17}$$

The alternative hypothesis here is that at least one of these β_js is nonzero.

To test $\mathrm{H_o}$ in (7.17), we construct a *discrepancy measure* that quantifies the fit of the *full model* (FM) with all $p+1$ β-parameters, versus the fit of the *reduced model* (RM) under

H_o with only $p + 1 - q$ (nonzero) β-parameters. A useful discrepancy measure for the MLR model involves the sum of squared errors (SSE) under the FM. Denote this as

$$\text{SSE(FM)} = \sum_{i=1}^{n} \{Y_i - \hat{Y}_i(\text{FM})\}^2.$$

Next, compare SSE(FM) to the SSE calculated when the RM under (7.17) holds:

$$\text{SSE(RM)} = \sum_{i=1}^{n} \{Y_i - \hat{Y}_i(\text{RM})\}^2.$$

The two SSEs quantify the relative quality of each model's fit to the data: if H_o is false, we expect SSE(RM) to greatly exceed SSE(FM), because the model under which it is fit fails to include important predictor variables. If H_o is true, we expect SSE(RM) to be roughly equal to SSE(FM). Notice that SSE(RM) will always exceed SSE(FM), because adding one (or more) predictor variables to the model always drops the SSE. The issue here is whether the drop from RM to FM is so large as to call H_o in question.

Corresponding to these terms, also write the error d.f. as $\text{df}_E(\text{FM})$ and $\text{df}_E(\text{RM})$, respectively. The difference between the two is $\Delta_E = \text{df}_E(\text{RM}) - \text{df}_E(\text{FM})$. Here, this is $\Delta_E = (n + q - p - 1) - (n - p - 1) = q$, that is, the number of parameters constrained by H_o.

The test statistic built from this discrepancy measure is

$$F = \frac{\{\text{SSE(RM)} - \text{SSE(FM)}\}/\Delta_E}{\text{SSE(FM)}/\text{df}_E(\text{FM})}. \tag{7.18}$$

Under H_o in (7.17), $F \sim F(\Delta_E, \text{df}_E[\text{FM}])$. Reject H_o in favor of any departure when $F_{\text{calc}} \geq F_\alpha(\Delta_E, \text{df}_E[\text{FM}])$. The corresponding P-value is $P\{F(\Delta_E, \text{df}_E[\text{FM}]) \geq F_{\text{calc}}\}$.

If the null hypothesis is H_o: $\beta_1 = \beta_2 = \cdots = \beta_p = 0$, the RM is simply $E[Y_i] = \beta_0$, that is, no effect due to any regressor variables. This is known as the 'full' F-test for the MLR. **R** list the full F-test statistic and its associated P-value at the bottom of the `summary()` output when applied to an `lm` object (previously suppressed in output above).

For the discrepancy approach based on (7.18) to be valid, the parameters represented under RM must be a true subset of those under FM. We say then that the models are 'nested.' If the relationship between RM and FM does not satisfy a nested hierarchy, (7.18) under H_o may not follow (or even approximate) an F-distribution. The family of models is then said to be 'separate' (Cox 1961; 1962); inferences for testing separate families is still a developing area of data analytic research (Cox 2013).

Notice that when testing H_o in (7.17), one could alternatively apply a simple Bonferroni correction to the collection of q individual t-statistics based on (7.16), with $j = p - q + 1, p - q + 2, \ldots, p$. This has the advantage of identifying *which* of the specific β_js are significant (if any), something the F-statistic in (7.18) cannot supply. Of course, the 'no-free-lunch theorem' (Section 5.5.1) still applies: the q Bonferroni-based inferences will be conservative, with familywise false positive error (FWE) exceeding α (and, greatly exceeding it for large q). If appropriate, consideration could instead be applied to controlling the false discovery rate (FDR; see Section 5.5.2). Analysts must decide which of these approaches will be most suitable on a case-by-case basis in consultation with the domain-specific expert(s).

Example 7.1.1 Cancer mortality multiple regression. Consider again the UK cancer mortality data from Example 6.2.1. Recall that the SLR fit on $Y = \log\{\text{Mortality}\}$ with the single

predictor variable $x = \{\text{Employment rate}\}$ produced a highly significant regression coefficient and also a relatively low value of R^2. This suggested that additional factors might be important in modeling $E[Y]$. Towards that end, a series of further locality-specific predictor variables were also recorded as part of the UK government's data collecting effort. Including the original employment predictor, these were

x_1 = Employment rate (average quarterly percentage of persons over 16 gainfully employed)
x_2 = Median weekly earnings (£)
x_3 = Unpaid volunteering (percentage of citizenry participating once or more per month)
x_4 = Civic activity (percentage of citizenry participating in local decision-making)
x_5 = Smoking cessation (persons per 100 000 over 16 who stopped smoking in past 4 weeks)
x_6 = Emergency bed-days (number of beds × days in use at hospital emergency rooms)

Predictors x_1 and x_2 represent economic factors, predictors x_3 and x_3 represent lifestyle factors, and predictors x_5 and x_6 represent health factors, all associated with socioeconomic activity in each locality. Table 7.1 presents the data. (As above, only a selection of the data is given in the table. The complete set is available at http://www.wiley.com/go/piegorsch/data_analytics.)

Table 7.1 Selected data from a larger set of $n = 342$ observations recorded in localities throughout the United Kingdom in 2008.

Y_i	x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	x_{i6}
47.20	843.8	25.7	90.3	524	23.6	2 419
152.27	535.9	13.4	67.7	1277	16.0	102 700
⋮	⋮	⋮	⋮	⋮	⋮	⋮
92.75	375.8	16.2	78.9	684	30.6	51 450
106.72	418.5	12.3	72.6	684	23.6	51 603

See Example 7.1.1 for Y- and x- variable descriptions.

An initial step in the analysis of these data is visual inspection: Figure 7.1 displays a scatterplot matrix with all the variables. A variety of patterns are evidenced, some of which suggest strong potential for the additional variables to impact the mean response.

To study this further, apply the MLR model in (7.1) to Y_i with the $p = 6$ predictor variables in Table 7.1. In **R**, the command for an MLR fit is again `lm()`, with syntax extended after the tilde (`~`) to include the additive variables. For example,

```
> mlrFM.lm <- lm( Y ~ employmt + earnings + volunteer
                              + civic + stopsmk + emergbeds )
> summary( mlrFM.lm )
```

The call to `summary()` produces the following **R** output (edited):

```
Call:
 lm(formula = Y ~ employmt + earnings + volunteer + civic
                                      + stopsmk + emergbeds)
```

```
Coefficients:
              Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)  5.714e+00   1.411e-01   40.489   < 2e-16
 employmt   -7.500e-03   1.689e-03   -4.439  1.23e-05
 earnings   -4.369e-04   1.063e-04   -4.108  5.01e-05
 volunteer  -1.142e-02   2.415e-03   -4.730  3.32e-06
 civic      -5.110e-03   3.095e-03   -1.651  0.099656
 stopsmk     1.015e-04   2.956e-05    3.433  0.000671
 emergbeds   6.320e-08   5.844e-08    1.082  0.280229
```

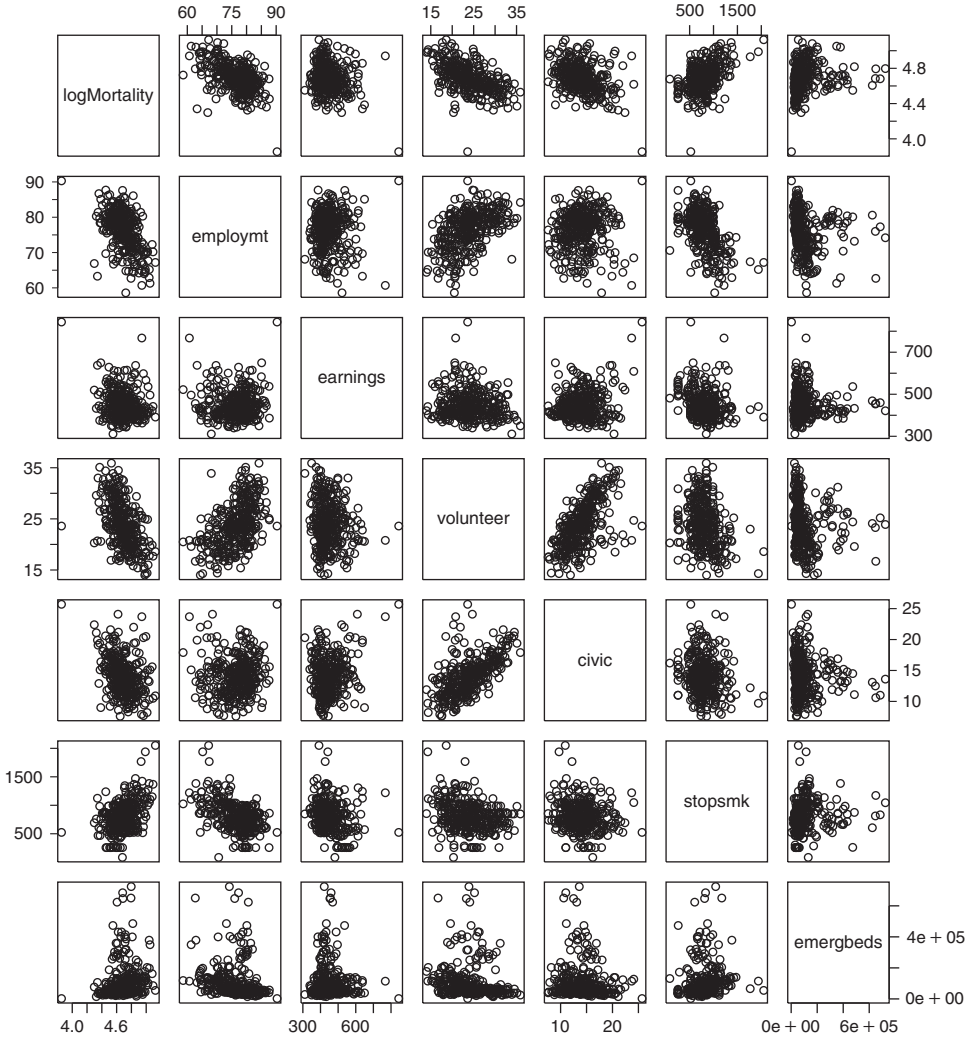

Figure 7.1 Scatterplot matrix for data in Example 7.1.1. Variables are $Y = \log\{\text{Mortality}\}$, $x_1 = \{\text{Employment rate}\}$, $x_2 = \{\text{Weekly earnings}\}$, $x_3 = \{\text{Volunteering \%}\}$, $x_4 = \{\text{Civic participation \%}\}$, $x_5 = \{\text{Smoking cessation rate}\}$, and $x_6 = \{\text{Emergency bed-days}\}$. Source: Data from http://data.gov.uk/dataset/ni_122_-_mortality_from_all_cancers_at_ages_under_75.

```
Residual standard error: 0.1163 on 335 degrees of freedom
Multiple R-squared: 0.4639,     Adjusted R-squared: 0.4543
F-statistic: 48.31 on 6 and 335 DF,  p-value: < 2.2e-16
```

The analysis identifies a significant regression effect: the full F-test of

$$\mathrm{H_o}\colon\ \beta_1 = \beta_2 = \cdots = \beta_6 = 0 \tag{7.19}$$

shows a highly significant $P < 2.2 \times 10^{-16}$ (see the bottom of the output). Indeed, most of the predictor variables appear to significantly impact log-mortality: two-sided P-values from the (pointwise) partial t-tests on each variable are given in the final column of the `Coefficients` output, under `Pr(>|t|)`.

To formalize this indication, consider testing (7.19) via partial t-tests on each $\mathrm{H_o}$:$\beta_j = 0$, $j = 1, \ldots, 6$ (i.e., ignoring the intercept), including a Bonferroni correction for the sixfold multiplicity. To do so, simply multiply the pointwise partial P-values by the number of comparisons, here $q = 6$. (If the multiplication drives an adjusted P-value past 1.0, report it as $P = 1.0$.) The consequent, Bonferroni-adjusted P-values appear in Table 7.2, calculated using the **R** code

```
> p1 <- length( coef(mlrFM.lm) )
> p.adjust( summary(mlrFM.lm)$coefficients[,4][2:p1],method='bonf' )
```

Table 7.2 Bonferroni-adjusted P-values for testing $\mathrm{H_o}$:$\beta_j = 0$ ($j = 1, \ldots, 6$) with cancer mortality data from Example 7.1.1.

Predictor variable	Adjusted P-value
x_1 = Employment rate	$P_1 = 7.36 \times 10^{-5}$
x_2 = Weekly earnings	$P_2 = 0.0003$
x_3 = Volunteering %	$P_3 = 1.99 \times 10^{-5}$
x_4 = Civic activity %	$P_4 = 0.5979$
x_5 = Smoking cessation rate	$P_5 = 0.0040$
x_6 = Emergency bed-days	$P_6 = 1.0$

At a 5% false positive rate, only the adjusted P-values for $x_4 = \{\text{Civic activity}\}$ and $x_6 = \{\text{Emergency bed-days}\}$ in Table 7.2 fail to show significant differences from $\mathrm{H_o}$:$\beta_j = 0$. Thus we conclude that $x_2 = \{\text{Weekly earnings}\}$, $x_3 = \{\text{Volunteering \%}\}$, and $x_5 = \{\text{Smoking cessation}\}$ show significant effects, along with $x_1 = \{\text{Employment rate}\}$.

For finer resolution, we refit the MLR model with just the $p = 4$ significant variables. In order to keep the subscripts consistent, redesignate x_5 as $x_4' = \{\text{Smoking cessation}\}$.

In **R**, the MLR fit is achieved via

```
> mlrRM.lm=lm( Y ~ employmt + earnings + volunteer + stopsmk )
```

or, apply the `update()` function. The consequent output (edited) from `summary()` is

```
Call:
  lm(formula = Y ~ employmt + earnings + volunteer + stopsmk)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.672e+00  1.324e-01  42.834  < 2e-16
employmt    -6.544e-03  1.517e-03  -4.315 2.10e-05
earnings    -5.041e-04  9.939e-05  -5.072 6.53e-07
volunteer   -1.439e-02  1.673e-03  -8.601 3.02e-16
stopsmk      1.082e-04  2.940e-05   3.679 0.000272

Residual standard error: 0.1167 on 337 degrees of freedom
Multiple R-squared: 0.4573,     Adjusted R-squared: 0.4508
F-statistic: 70.99 on 4 and 337 DF,  p-value: < 2.2e-16
```

Notice that the R^2 value for the MLR fit has risen considerably from the earlier SLR fit in Example 6.2.3: 45.73% versus 27.38%, respectively.

From the signs of the estimated regression coefficients, the economic variables, Employment rate and Weekly earnings, and the remaining lifestyle variable, Volunteering %, decrease log-mortality as each rises (with all others held fixed), as might be expected. The remaining health variable, Smoking cessation, increases log-mortality as it rises, however. This seems counterintuitive: as more individuals stop smoking, one might expect the log-mortality to drop. One possible speculation here is that a high smoking cessation rate may indicate a community with a larger inceptive population of smokers, and with more smokers, there are more cancer deaths. Also, the crude rate does not take account for recidivism among individuals who quit but then return to smoking, which could eventually increase observed cancer mortality. The perplexing result from this data-mining exercise is cause for further investigation (and possible knowledge discovery) into how various factors affect cancer mortality for these British communities.

The next example continues study of these data with analysis of the residuals from the four-predictor MLR fit. □

Example 7.1.2 Cancer mortality multiple regression (Example 7.1.1, continued). Continue with the MLR analysis of the UK cancer mortality data from Example 7.1.1. No analysis is complete without a check of the model fit via study of the residuals. Figure 7.2 displays a normal quantile plot of the raw residuals, along with a plot of Studentized deleted residuals, from the final four-variable fit in Example 7.1.1. Overlayed in the Studentized residual plot are exceedance bounds that mark the t-critical points $\pm t_{\alpha/(2n)}(n-p-2)$ from (6.26) to identify unusual observations. At $\alpha = 0.05$, the pertinent critical points are $\pm t_{0.05/684}(336) = 3.8415$.

The normal quantile plot in Figure 7.2 mimics that seen with its $p = 1$ SLR cousin in Figure 6.7. The pattern appears generally consistent with a normal distribution, but three unusual points appear in the lower tail of the distribution. As might be expected, these are the same three potential outliers identified in Example 6.3.1: the City of London, the RBKC, and the City of Westminster. The corresponding Studentized deleted residual plot gives similar corroboration, although now only two points drop below the lower 5% exceedance level: the City of London and the RBKC. The City of Westminster's Studentized deleted residual now lies just above the lower exceedance line. Nonetheless, the indications from Example 6.3.1 for these three communities remain valid: they appear to exhibit different, and potentially intriguing, features regarding log-cancer mortality. □

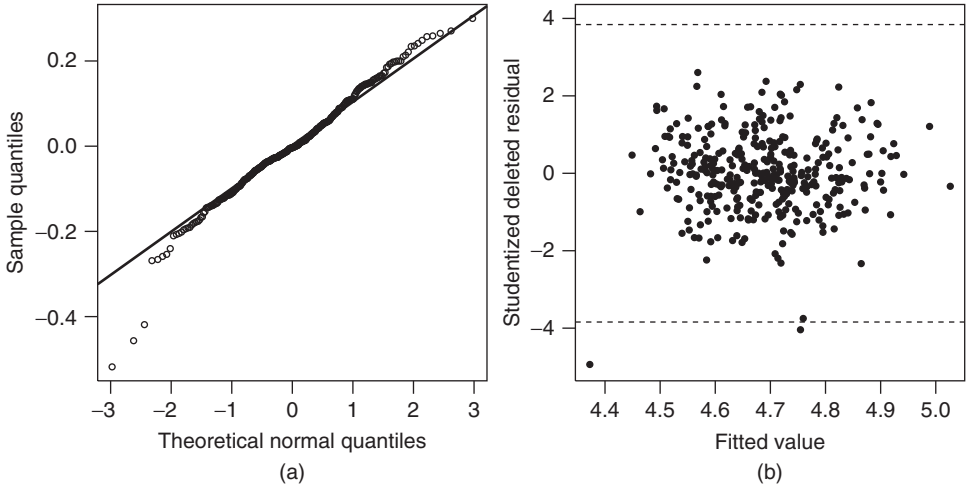

Figure 7.2 (a) Normal quantile plot for raw residuals with overlaid normal reference line and (b) Studentized deleted residual plot from MLR fit in Example 7.1.2. Variables are $Y = \log\{\text{Mortality}\}$, $x_1 = \{\text{Employment rate}\}$, $x_2 = \{\text{Weekly earnings}\}$, $x_3 = \{\text{Volunteering \%}\}$, and $x_4' = \{\text{Smoking cessation rate}\}$. Dashed lines in (b) are 5% exceedance levels for Studentized deleted residuals, $\pm t_{0.05/684}(336) = 3.8415$. Source: Data from http://data.gov.uk/dataset/ni_122_-_mortality_from_all_cancers_at_ages_under_75.

7.1.4 Multicollinearity

An important consideration when employing MLR models is the differing contributions of the various x-variables. It can sometimes be the case that information in one of the predictors closely duplicates or overlaps with information in another. The effect is known as *multicollinearity*. In the extreme, one predictor may be a strict linear function of another, say, $x_{im} = \gamma_0 + \gamma_1 x_{ij}$ at all i for some $j \neq m$. Then in the MLR model equation (7.2),

$$\begin{aligned}\mu_i &= \beta_0 + \beta_1 x_{i1} + \cdots + \beta_j x_{ij} + \cdots + \beta_m x_{im} + \cdots + \beta_p x_{ip} \\ &= \beta_0 + \beta_1 x_{i1} + \cdots + \beta_j x_{ij} + \cdots + \beta_m(\gamma_0 + \gamma_1 x_{ij}) + \cdots + \beta_p x_{ip}.\end{aligned}$$

But then, the mean response becomes

$$\mu_i = (\beta_0 + \beta_m\gamma_0) + \beta_1 x_{i1} + \cdots + (\beta_j + \beta_m\gamma_1)x_{ij} + \cdots + \beta_p x_{ip},$$

that is, x_{im} *drops* from the model, changing the intercept to $\beta_0 + \beta_m\gamma_0$ and the regression coefficient for x_{ij} to $\beta_j + \beta_m\gamma_1$. Thus there is no need to fit x_{im}. Its contribution is already represented in μ_i. (The same effect occurs if x_{im} is a linear combination of two or more other predictors already in the model. In matrix terms, this sort of perfect collinearity drops the column rank of $\mathbf{X}$ below $p + 1$, making $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ singular and impossible to invert.) When this occurs, we say x_{im} is *aliased* with the other predictor(s), because it replicates existing information already contained in μ_i. Most modern linear regression programs will identify an aliased predictor and remove it before fitting the model.

Even when no aliasing is present, one (or more) of the predictors may closely approximate information in the others. When this occurs, the strong multicollinearity leads to a variety of undesirable instabilities with the LS estimators for $\hat{\boldsymbol{\beta}}$ (Kutner et al. 2005, Section 7.6).

A number of strategies exist to identify multicollinearity in a set of posited predictor variables. A convenient way to visualize overlaps between the predictor variables is to build a scatterplot matrix among all the predictors. Clear redundancies will appear as strong patterns in the individual pairwise plots. Correspondingly, calculation of the pairwise correlations between the predictor variables – via, for example, the `cor()` function in **R** – will give a similar numerical indication, at least for any linear associations.

A more-formal numerical diagnostic is known as the variance inflation factor (VIF). When heavy multicollinearity exists among a set of predictor variables, the variances (and their square roots, the standard errors) of the LS estimators inflate. This increases the width of any confidence regions on $\boldsymbol{\beta}$ and $\mu(\mathbf{x})$ and decreases the power of any associated hypothesis tests. The VIFs are used to measure how much each predictor contributes to variance inflation (Kutner et al. 2005, Section 10.5). A VIF is calculated for each jth predictor as

$$\mathrm{VIF}_j = \frac{1}{1 - R_j^2} \ , \tag{7.20}$$

where R_j^2 is the coefficient of multiple determination found from regressing the jth predictor, x_{ij}, on the other $p - 1$ predictor variables ($j = 1, \ldots, p$). In **R**, the `vif()` function from the external *car* package computes VIFs.

In practice, a set of predictors whose maximum VIF exceeds 10 is felt to be highly multicollinear. Also check the mean, $\overline{\mathrm{VIF}} = \sum_{j=1}^{p} \mathrm{VIF}_j / p$, which should not grow far above 1. (In the latter case, recommendations vary, although a $\overline{\mathrm{VIF}}$ greater than about 6 or 7 is usually cause for concern.)

Example 7.1.3 Cancer mortality multiple regression (Example 7.1.1, continued). Consider again the UK cancer mortality data from Example 7.1.1. The MLR fit on $Y = \log\{\text{Mortality}\}$ employed the $p = 4$ predictor variables $x_1 = \{\text{Employment rate}\}$, $x_2 = \{\text{Weekly earnings}\}$, $x_3 = \{\text{Volunteering \%}\}$, and $x_4' = \{\text{Smoking cessation rate}\}$. Focusing on pertinent components of the scatterplot matrix in Figure 7.1 shows that these four predictors display a variety of pairwise patterns, with mostly diffuse pairwise spreads. The pair x_1 and x_4' exhibit perhaps the most substantial linear relationship in the scatterplot – suggesting the largest potential overlap – but even here the dispersion is fairly large.

Turning to the VIFs, the following sample **R** code and output give the individual values and their average

```
> library( car )
> vif( mlrRM.lm )
 employmt  earnings volunteer   stopsmk
 1.712690  1.086523  1.372481  1.438172

> mean( vif(mlrRM.lm) )
 [1] 1.402467
```

None of the VIFs exceed 1.8, so their maximum is well below the practical action limit of 10. Their mean is also well below 6, with $\overline{\mathrm{VIF}} = 1.4$. In general, this analysis indicates no serious concerns regarding multicollinearity with this suite of predictor variables. □

Remedies for multicollinearity are primarily related to predictor variable selection and design (and to a certain extent, simple common sense):

- To the best extent possible, identify predictors that provide separate domain-specific information (this may seem obvious, but is nonetheless worth stating).
- Avoid inclusion of predictors that reproduce very similar information (e.g., hospital admissions and hospital bed-days in a health care utilization study).
- Always try to limit the number, p, of predictor variables to a manageable total. As more variables are added to an MLR model, opportunities for multicollinearity will naturally increase.

One can also try transforming highly collinear predictor variables (cf. Section 3.4.3) to decrease their VIFs. The logarithm is fairly popular in this regard, although its success rate can vary. Alternatively, applying methods of multivariate data reduction to the set of predictors may make them more amenable for regression modeling. A popular data reduction approach is principal component analysis (PCA), discussed in Section 10.2. This leads to a methodology known as principal component regression (Hastie et al. 2009, Section 3.5.1). Or, if the focus is on prediction of μ_i, a form of penalized optimization known as *ridge regression* can be applied to address effects of multicollinearity; this is discussed in Section 7.4.2.

Note that the VIFs are determined computationally from only the x_{ij}s, so they can be computed before acquisition of the responses Y_i. Thus, where possible, it is good practice to check the VIFs for multicollinearity before final data acquisition. Predictors exhibiting high multicollinearity can be reassessed as to their anticipated value to the analysis and removed or transformed if felt to be marginal or nonessential.

7.2 Polynomial regression

A special case of the MLR mean response in (7.2) that employs only a single predictor, x, is the polynomial regression model

$$\mu_i = \mu(x_i) = \beta_0 + \beta_1(x_i - \overline{x}) + \beta_2(x_i - \overline{x})^2 + \cdots + \beta_p(x_i - \overline{x})^p. \tag{7.21}$$

The polynomial form is useful when the mean response is affected by x in a manner more complex than described by simple linear relationships. Centering the predictor about its mean $\overline{x}$ helps to avoid potentially high multicollinearity in the multiple regression (Bradley and Srivastava 1979) and is commonly instituted. If the values of x_i can be chosen in advance, special forms called *orthogonal polynomials* exist, which can eliminate multicollinearity among the polynomial regressors. See, for example, the review by Narula (1979).

Continue to assume $Y_i \sim$ indep. $\mathrm{N}(\mu[x_i], \sigma^2)$, $i = 1, \ldots, n$. Also assume that $p + 1 < n$ and that at least $p + 1$ distinguishable values exist among the x_is. (These latter assumptions are not unreasonable, because values of p greater than 3 are often difficult to motivate from subject-matter arguments.) In practice, it usually does not make sense to include a higher-order term without also including all the lower orders. Thus whenever a certain order of polynomial, p, is specified in a regression model, all the lower order terms at $p - 1, p - 2, \ldots, 2$, and 1 are generally also included.

Under these assumptions, the statistical machinery employed for fitting and analyzing an MLR becomes available for fitting and analyzing polynomial regression models, with

$x_{ij} = (x_i - \bar{x})^j$. LS estimators, standard errors, tests and confidence intervals, regression diagnostics, and so on all follow using the general MLR equations described in the previous section. This underlies much of the appeal for the use of polynomial models: although curvilinear regression relationships are often more complex than can be described by a simple polynomial function, (7.21) may provide an approximation to the true nonlinear relationship over the range of x under study (Piegorsch and Bailer, 2005, Section 1.5). This applies even for the SLR case of $p = 1$. Accepting the value of the approximation, the various methods of MLR analysis may then be applied for studying the curvilinear response. For instance, to test if there is any effect on $E[Y]$ due to the x-variable, assess $\mathrm{H_o}$: $\beta_1 = \beta_2 = \cdots = \beta_p = 0$ via an F-statistic as in (7.18). Or, partial tests on the individual regression coefficients follow from standard t-statistics. For instance, if $p = 2$, a test of $\mathrm{H_o}$:$\beta_2 = 0$ using $T_2 = |\hat{\beta}_2|/\mathrm{se}[\hat{\beta}_2]$ assesses whether quadratic curvature is required in a model that already has a linear term present. As mentioned at the beginning of Section 7.1, however, in the polynomial context, β_j loses its interpretation as change in response with the jth predictor when holding the other predictors fixed. Clearly, one cannot vary $(x_i - \bar{x})^j$ while holding $(x_i - \bar{x})$, … , $(x_i - \bar{x})^{j-1}$, $(x_i - \bar{x})^{j+1}$, … , $(x_i - \bar{x})^p$ fixed.

Exercises 7.10–7.12 explore features of polynomial regression in more detail.

7.3 Feature selection

In many informatic studies with multiple predictor variables, the number of potential x-variables can become quite large. It is natural in such studies to search for possible predictors that represent pertinent features associated with the supervised learning effort and include these in the multiple regression equation. Also called *attribute section* or *variable selection*, the identification of important predictors is a common component of the exploratory model-building exercise.

Combinations of the predictors can also be assessed, including higher-order polynomial terms such as x_j^2 and cross-product terms such as $x_j x_k$. (The cross-products are often called 'interactions.') Note that, as in Section 7.2, it does not usually make sense to include a higher-order term in a model without also including all its lower-order siblings. Thus, for example, given predictors x_1, x_2, and x_3, if next including $x_4 = x_2 x_3$, one should typically retain x_2 and x_3. Similarly, if including $x_5 = (x_3 - \bar{x}_3)^2$, one should retain x_3, and so on. Many of these may turn out to be uninteresting, however, in that they contribute insignificantly to modeling the mean response. Substantial multicollinearity may also exist among large collections of x-variables, as they can often overlap in the predictive information they provide. Thus, the selection effort must not be conducted in a cavalier manner.

Clearly, when prior, domain-specific knowledge indicates that certain predictor variables are expected to impact the mean response, they should be included and assessed in the MLR model fit. When reason for doubt exists about some of the potential predictors, however, and/or when the analysis has more of an exploratory and less of a confirmatory character, specification and selection of predictor variables can itself become part of the exploratory effort. Of course, determination of the final set of regressors should be made via careful and purposeful analysis of the data; selecting predictor variables for further study is only a first step in constructing the MLR model. (Any consequent inferences must also be viewed as conditional on the selection effort. If the selection process involves some form of formal inferences, these can disrupt the final, unconditional (familywise) error rates or confidence levels. Berk et al. (2013) discussed some of the theoretical complexities associated with making inferences after variable selection; also see Potscher (1991).)

7.3.1 R^2_p plots

Perhaps the simplest strategy for regression variable selection is to quantify the value of each potential variable configuration and then choose that configuration that optimizes some fixed score or metric, say, the coefficient of multiple determination from the MLR fit, R^2, in (7.10). Suppose there are $Q > 1$ distinct variable configurations under consideration, each with a different combination of one or more potential predictors. Assume an intercept term, β_0, is always included in the model, so that the total number of regression coefficients being estimated is $p + 1 \geq 2$. (The requirement to always include β_0 is made for convenience and if relaxed, does not substantively affect the methods described later.) For example, at $p = 1$, the models all contain an intercept and a single predictor x_j; at $p = 2$, they all contain an intercept and either two predictors x_j and x_k $(j \neq k)$ or if desired, a predictor x_j and its mean-adjusted square $(x_j - \bar{x}_j)^2$; at $p = 3$, they contain three predictors, x_j, x_k, andx_ℓ, or two predictors and one of their squares, or two predictors and their cross-product, and so on.

Let $R^2_{[p]}$ denote the *optimum* score for R^2 among all variable configurations under consideration with exactly p terms based on the predictor variables and any functions of them, along with an intercept β_0, in the model for μ_i. (For R^2, the optimum is always the largest R^2 at that p, but other measures may require the smallest value; see the following text.) By adding more terms and increasing p, we expect $R^2_{[p]}$ also to rise because as noted in Section 7.1.3, the corresponding $\text{SSE}_{[p]}$ in (7.10) will always drop. For large-enough p, however, the drop in SSE and consequent rise in R^2 will likely become slight, even infinitesimal. This will appear on a plot of $R^2_{[p]}$ versus p as a leveling of the $R^2_{[p]}$ curve with increasing p. If the point of this 'diminishing return' is visually obvious, select the variable configuration corresponding to the maximum R^2 at that p.

Example 7.3.1 **Cancer mortality multiple regression (Example 7.1.1, continued).** Return to the UK cancer mortality data in Example 7.1.1. The MLR fit on $Y = \log\{\text{Mortality}\}$ previously employed the $p = 4$ predictor variables $x_1 = \{\text{Employment rate}\}$, $x_2 = \{\text{Weekly earnings}\}$, $x_3 = \{\text{Volunteering \%}\}$, and $x'_4 = \{\text{Smoking cessation rate}\}$. Now consider as additional predictors the associated quadratic terms $(x_j - \bar{x}_j)^2$ $(j = 1, \ldots, 4)$, along with all six second-order cross-products (interactions) $x_j x_k$ $(j \neq k)$. This produces a suite of 14 possible predictor variables and, including the intercept, up to $p = 15$ regression parameters to be estimated from the data.

For selecting among the consequent variable configurations, apply the R^2 measure as described earlier and construct an $R^2_{[p]}$ plot to inspect the pattern as p increases. To facilitate the computations, **R** provides the `leaps()` function in the external *leaps* package. This finds all possible R^2_p values from input of the observation vector and the design matrix of all constructed predictor variables. Here, given the additional, second-order predictors `x1sq=(x1-mean(x1))^2`, …, `x12=x1*x2`, …, and so on, sample **R** code is

```
> require( leaps )
> Xmtx <- cbind( x1,x2,x3,x4,x1sq,x2sq,x3sq,x4sq,
                           x12,x13,x14,x23,x24,x34 )
> mlrBaseFM.r2 <- leaps( x=Xmtx, y=Y, method='r2' )
> plot( mlrBaseFM.r2$r2 ~ I(mlrBaseFM.r2$size - 1) )
```

which yields the R^2_p plot. (The `size` attribute in a `leaps` object is the total number of regression parameters, $p + 1$, so the code in the `plot` command above subtracts 1 from it.

Also included is the 'Identity' function `I()` to protect the direct subtraction operation.) The consequent maximum values $R^2_{[p]}$ can be isolated via, for example,

```
by( data=mlrBaseFM.r2$r2,
              INDICES=factor(mlrBaseFM.r2$size - 1), FUN=max )
```

Connecting these with line segments produces Figure 7.3. As expected, one sees a sharp rise when moving from $p = 1$ to multiple predictors and then a flattening as p increases further. By about $p = 5$, $R^2_{[p]}$ has clearly leveled out. This suggests that an MLR model with five predictors (and an intercept, so `size` $= p + 1 = 6$) produces about as much practical improvement in R^2 as can be extracted from these data.

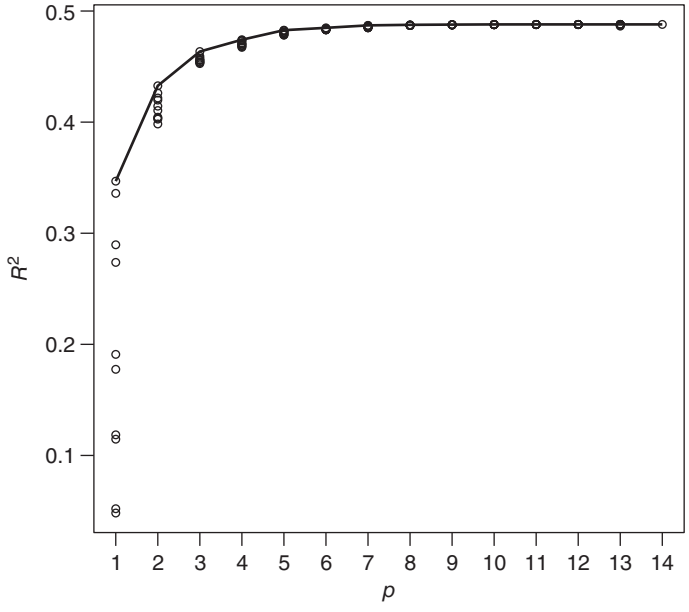

Figure 7.3 Variable selection plot of R^2_p versus p for MLR data in Example 7.3.1. Variables are $Y = \log\{\text{Mortality}\}$, $x_1 = \{\text{Employment rate}\}$, $x_2 = \{\text{Weekly earnings}\}$, $x_3 = \{\text{Volunteering \%}\}$, and $x'_4 = \{\text{Smoking cessation rate}\}$, plus all quadratic terms and second-order cross-products. Solid line connects values of maximum values $R^2_{[p]}$. Source: Data from http://data.gov.uk/dataset/ni_122_-_mortality_from_all_cancers_at_ages_under_75.

Closer inspection of the data set identifies $R^2_{[5]} = 0.4827$, with corresponding predictors $x_3, x'_4, (x_2 - \bar{x}_2)^2, x_1x_2$, and $x_1x'_4$. (Find this selection using

```
> maxR2 <- max( mlrBaseFM.r2$r2[mlrBaseFM.r2$size==6])
> index <- which( mlrBaseFM.r2$r2 == maxR2 )
> mlrBaseFM.r2$which[index,]
```

from the `leaps` object.) As x_2^2 and x_1x_2 are components of the selected configuration, one should override removal of the original x_1 and x_2 variables and also include these in any

further MLR fit. In fact, from the `mlrBaseFM.r2` object, we find that this configuration has the highest R^2 at $p = 7$, with $R^2_{[7]} = 0.4871$.

Exercise 7.13 explores this MLR analysis further, now with the seven predictor variables $x_1, x_2, x_3, x_4', (x_2 - \bar{x}_2)^2, x_1x_2$, and x_1x_4'. □

The R^2 measure's predisposition to increase as variables are added to the model is often viewed as a drawback, because it allows a naïve analyst to increase the measure simply by adding irrelevant or arbitrary variables. To correct for this, we can adjust R^2 to use mean squares instead of sums of squares. Known colloquially as the *adjusted* R^2, the corrected measure divides each sum of squares in (7.10) by its corresponding d.f.:

$$R^2_A = 1 - \frac{\text{SSE}/(n-p-1)}{\text{SSTo}/(n-1)}. \tag{7.22}$$

With this, let $R^2_{A[p]}$ be the largest value of R^2_A among all variable configurations under consideration containing p regression parameters (and an intercept). The adjustment in (7.22) no longer increases strictly with p, necessarily, although it often follows a roughly increasing pattern when plotted against p. As with the $R^2_{[p]}$ plot, visual inspection of $R^2_{A[p]}$ versus p can identify a point of 'diminishing return.' The corresponding collection of regressor variables may then be selected for further study. (See Exercise 7.13.) In **R**, use the `leaps()` function with the `method='adjr2'` option.

A variety of other, more-sophisticated diagnostic scores have been constructed to measure the quality of an MLR fit to which a graphical inspection strategy may be applied. For example, suppose the collection of potential predictor variables contains $P > 1$ candidates, including polynomial powers and cross-product interactions. Mallows (1973) suggested a measure using the MSE from a baseline, 'full' model with all the P predictors (and the intercept), MSE_P. He compared MSE_P with the SSE of an aspirant RM (which may or may not contain an intercept) with $p \leq P + 1$ terms, SSE_p, via what we now call *Mallows' C_p statistic*

$$C_p = \frac{\text{SSE}_p}{\text{MSE}_P} - (n - 2p).$$

Predictor configurations that describe the mean response well correspond to $E[C_p] \approx p$, while poor-fitting configurations produce large values well above p; the measure decreases towards p as the explanatory quality improves. Thus a value of C_p close to but not greatly exceeding p identifies a collection of model terms for further study. Plotting C_p versus p again gives a visual diagnostic for such selection: choose predictor configuration(s) at the smallest p such that $C_p \approx p$, without greatly exceeding it. To find C_p in **R**, use the `leaps()` function with the `method='Cp'` option.

Or, focusing on prediction of a future Y_i via the case-deletion strategy mentioned in Section 6.3, Allen (1974) constructed a *prediction sum of squares*,

$$\text{PRESS}_p = \sum_{i=1}^{n} \left(Y_i - \hat{Y}_{i[-i]}\right)^2,$$

where $\hat{Y}_{i[-i]}$ is the ith predicted value under the aspirant configuration of p terms when the ith observation is deleted from the data. If prediction is an important goal of the MLR analysis, variable configurations with a low PRESS_p become candidates for more-focused analysis. As above, plotting PRESS_p against p provides a facilitative visual device. For a further discussion on this and related aspects of MLR variable selection, see Kutner et al. (2005, Section 9.3).

7.3.2 Information criteria: AIC and BIC

A popular criterion in contemporary data analytics for variable and model selection focuses on maximizing statistical information in the fitted model. Here 'information' is used in a formal sense and is different from (but related to) the sample entropy in Section 3.4.2 and the Fisher information in Section 5.1. For selecting models with p predictor variables, define the *Akaike information criterion* (AIC; Akaike 1973) as

$$\text{AIC}_p = -2\hat{\ell}_p + 2\nu_p, \tag{7.23}$$

where $\hat{\ell}_p = \ell(\hat{\boldsymbol{\beta}}_p)$ is the value of the maximized log-likelihood function (Section 5.1) under the model with $(p+1) \times 1$ parameter vector $\boldsymbol{\beta}_p$, and $\nu_p \geq 1$ is the number of unknown parameters being estimated when fitting that model. For example, if the model contains p predictors along with an intercept and an unknown variance term, $\nu_p = p + 2$. Then, let $\text{AIC}_{[p]}$ denote the optimum value of AIC_p (here, the smallest) among all considered variable configurations containing p regressors.

In passing, note that (7.23) is the 'lower-is-better' form of the AIC. Some authors multiply (7.23) by -1 or $-\frac{1}{2}$ to achieve a 'higher-is-better' form. Analysts must be careful to expressly identify which form of AIC under which they operate.

In effect, the AIC begins by assessing the maximized log-likelihood for each competitor model (a natural criterion for defining the quality of a parametric model). Then, it multiplies by -2 to view the process as a minimization and finally adds a positive 'penalty' term that accounts for the number of parameters used to achieve the model fit. In most cases – not limited to the MLR setting – adding variables to a model improves the fit and increases the maximized likelihood, but sometimes only slightly. The AIC penalizes those models that achieve high likelihoods simply by including a 'kitchen sink' full of inconsequential parameters.

Given a set of models representing different variable configurations, an AIC-based model-selection strategy computes the optimal (smallest) AIC, $\text{AIC}_{[p]}$, from (7.23) at each p and then selects that model for which it is lowest across all p. For the MLR setting, AIC_p will order potential variable configurations identically to Mallows' C_p. No form of sampling distribution is involved, so there are no formal hypotheses to test or P-values to calculate.

A wide variety of other measures exist that compete with the AIC, most of which modify the penalty term in (7.23) for different levels of selectivity. A simple adjustment for small-samples known as the *corrected AIC*, or AICc, takes

$$\text{AICc}_p = \text{AIC}_p + \frac{2\nu_p(\nu_p + 1)}{n - \nu_p - 1} \tag{7.24}$$

(Hurvich and Tsai 1989). For the large data sets seen in modern data analytics, n will often exceed ν_p by many orders of magnitude, so that the AIC_p term in (7.24) will dominate. If so, AIC and AICc produce essentially the same results with very large n.

Another popular competitor is the *Bayesian information criterion* (BIC) due to Schwarz (1978):

$$\text{BIC}_p = -2\hat{\ell}_p + \nu_p \log(n).$$

(Some authors use the alternative moniker *Schwarz's Bayesian Criterion*, or SBC. Notice that $\text{BIC}_p = \text{AIC}_p + \nu_p(\log\{n\} - 2)$.) Since its penalty term weights ν_p more heavily than the AIC for most values of n, the BIC tends to favor models with smaller numbers of parameters. This is seen by many to be a favorable property.

Many other 'information criterion (IC)' variants exist, with sometimes-different and sometimes-overlapping features for assessing the quality of a model. These include the Takeuchi information criterion (TIC; Takeuchi 1976), the Kullback information criterion (KIC; Cavanaugh 1999), the deviance information criterion (DIC; Spiegelhalter et al. 2002), the focused information criterion (FIC; Claeskens and Hjort 2003), and many more. (One might say that an entire 'alphabet soup' of ICs has been, and continues to be, developed.) Final determination of which IC to employ is in the hands of the data analyst and should be made with the eventual, specific target inference(s) in mind. Either of the originals, AIC or BIC, can serve as useful defaults when no domain-specific motivation guides this choice.

In **R**, the `AIC()` function computes AIC for a variety of model classes and **R** objects. A similar `BIC()` function exists for finding the BIC. Objects may even be nested in the command to compare their AICs. As `help(AIC)` warns, however,

```
The log-likelihood and hence the AIC/BIC is only defined up to an
additive constant. Different constants have conventionally be used
for different purposes...
```

Thus, analysts should study the function structure and output carefully to ensure that the desired quantities are being produced. The next section discusses ways to approach this via automated selection.

7.3.3 Automated variable selection

As the number of possible regressors, p, increases, the number of possible variable configurations, 2^p, grows with it exponentially. Analysts run the risk of *overfitting* the data: a model with too many predictor variables will suitably accommodate many of the particular features in a given set of training data, but usually at the cost of poor predictive ability with future data sets.

To assuage these concerns, we can turn to automated or semiautomated variable selection procedures. These systematically study all possible models – or some prespecified subgroup, such as all possible second-order models – available from a given set of original predictor variables. Then, an operating subset of the larger set of variables is chosen that optimizes some preselected score. The result is one (or more) putative models that the analyst can evaluate further in detail.

One such strategy aims to find a few (usually one or two) of the best-performing predictor configurations at each p, so that the analyst can focus attention on these various subsets. In this sense, the process is known as *(best) subset selection*. 'Best' here is usually defined as minimization of the SSE for a fixed p; the largest desired value of p and the maximum number of output configurations per p are specified in advance.

In a sense, the R_p^2 plot and **R**'s `leaps()` function from Section 7.3.1 imitate a subset selection strategy, although, as presented above, only one configuration is chosen per p (corresponding to the largest R^2 at each p) and the analyst participates more directly in selection via the visualized plot.

To limit the (likely) large number of calculations required in automated subset selection, clever computational algorithms have evolved to flag the best subsets more efficiently. One of the best known is *leaps and bounds* (Furnival 1971; Furnival and Wilson 1974), a form of branch-and-bound algorithm (see Gatu and Kontoghiorghes 2006). Computationally, it can be quite efficient for values of p up to about 35 or 40.

Subset selection is available in **R** via a number of external packages. For instance, the *leaps* package offers its `regsubsets()` function, which provides graphical outputs as a pathway for visualizing large configurations of potentially best models.

Example 7.3.2 Cancer mortality multiple regression (Example 7.3.1, continued). Return to the UK cancer mortality data and the variable selection operations in Example 7.3.1. Recall that for the MLR fit on $Y = \log\{\text{Mortality}\}$, the single predictor variables $x_1 = \{\text{Employment rate}\}$, $x_2 = \{\text{Weekly earnings}\}$, $x_3 = \{\text{Volunteering \%}\}$, and $x_4' = \{\text{Smoking cessation rate}\}$ were augmented with the additional, second-order predictors $(x_j - \bar{x}_j)^2$ $(j = 1, \ldots, 4)$, and $x_j x_k$ $(j \neq k)$. Suppose we wish to conduct subset selection on this collection of 14 predictor variables, where the subsets contain no more than $p = 7$ predictors (for comparison with the results in Example 7.3.1). Also, we ask the computer to provide the *two* best subsets at each p. (As above, we always include the intercept term.) Sample **R** code using the `regsubsets()` function from the external *leaps* package is

```
> require( leaps )
> mlrBaseFM.rss <- regsubsets( x=Xmtx, y=Y, nbest=2, nvmax=7 )
> summary( mlrBaseFM.rss )
```

(where the corresponding **R** components from Example 7.3.1 remain in place). The various options to the function specify the design matrix **X** with the 14 predictors (`x=`), the response variable (`y=`), the number of 'best' subsets per p (`nbest=`), and the maximum number of variables per configuration (`nvmax=`). This produces the output (edited) in Figure 7.4.

In the `regsubsets` output, the first column indicates the number of predictors, p, and the second column indicates the best `(1)` and the next-best `(2)` variable configuration at that value of p. The asterisks (`*`) mark which of the predictors are selected for that configuration. For example, the best seven-variable configuration contains the predictors $x_1, x_2, x_3, x_4', (x_2 - \bar{x}_2)^2, x_1x_2$, and x_1x_4', which corresponds to results found using the R_p^2 plot in Example 7.3.1. For a given p, the analyst can decide which of these various configurations deserve deeper study in the MLR model. Additional visualization is available via the `plot()` command in *leaps*, which orders variable subsets by a selection criterion such as BIC or R^2. (Try it: `plot(mlrBaseFM.rss,scale='BIC')`.)

Of additional interest in the `regsubsets` output above is the patterns that appear among the various best subsets. Notice that the $(x_1 - \bar{x}_1)^2, (x_3 - \bar{x}_3)^2, (x_4' - \bar{x}_4')^2$, and x_3x_4' variables are *never* selected, suggesting that they would have limited value if employed in the MLR model. Further, x_2 is only selected once p has risen to 7, although its corresponding quadratic term $(x_2 - \bar{x}_2)^2$ appears in configurations with p as low as 3. As we usually require lower-order terms to accompany their higher-order companions when the latter are entered into an MLR model, this interesting anteposition could call for deeper analysis. Indeed, a variety of lesser patterns also surface in the output; their further study could make for intriguing knowledge discovery into the connections between these variables and log(Mortality) in these British communities. □

Another automated strategy approaches the feature selection process in a 'stepwise' manner. Rather than branch through the space of potential variable configurations, stepwise strategies take a single path based on a set of step-up and step-down criteria. Although less efficient than subset section, they can be useful in practice when p is very large and the subset approach becomes computationally infeasible.

```
Subset selection object
14 Variables  (and intercept)
2 subsets of each size up to [p =] 7
Selection Algorithm: exhaustive
         x1  x2  x3  x4 x1sq x2sq x3sq x4sq x12 x13 x14 x23 x24 x34
1 ( 1 )                                          *
1 ( 2 )                                                  *
2 ( 1 )                                      *   *
2 ( 2 ) *                                                *
3 ( 1 )              *       *                   *
3 ( 2 )          *                               *                 *
4 ( 1 )          *   *       *                           *
4 ( 2 )                      *                   *   *             *
5 ( 1 )          *   *       *                   *       *
5 ( 2 )          *   *       *                           *   *
6 ( 1 )          *   *       *                   *   *   *
6 ( 2 ) *        *   *       *                   *       *
7 ( 1 ) *   *    *   *       *                   *       *
7 ( 2 )     *    *   *       *                   *   *   *
```

Figure 7.4 **R** output from best subset selection analysis for UK cancer mortality data in Example 7.3.2. Source: Data from http://data.gov.uk/dataset/ni_122_-_mortality_from_all_cancers_at_ages_under_75.

A popular stepwise variant is *forward stepwise regression*, which applies both step-up and step-down criteria to achieve the recommended predictor list for further examination. As in Section 7.3.1, suppose there are a total of P potential variables under study. From a simple progenitor model – usually just the lone intercept – the forward stepwise algorithm proceeds as follows (Kutner et al., 2005, Section 9.4):

FS.0 Fix an 'entry' significance level, α_{en}, for admission of aspirant variables, and an 'exit' significance level, α_{ex}, for deletion of unnecessary ones. To avoid cycling, $\alpha_{\text{en}} < \alpha_{\text{ex}}$ is required.

FS.1 Fit P SLR models, one for each jth predictor variable ($j = 1, \ldots, P$). From each fit, compute the t-statistics for testing that the individual jth predictor is significant, via (7.16):

$$\frac{T_j - \hat{\beta}_j}{\text{se}[\hat{\beta}_j]}.$$

Find the largest $|T_j|$ and calculate its corresponding two-sided P-value P_j. Admit the associated x_j if $P_j \leq \alpha_{\text{en}}$. (Notice that no adjustments are made here for multiplicity.) If no $|T_j|$ can satisfy $P_j \leq \alpha_{\text{en}}$, terminate the stepwise procedure.

FS.2 Denote the selected x-variable by x_{j^*}. Fit all the $P - 1$ possible MLR models at $p = 2$, using the two predictors x_{j^*} and x_j ($j \neq j^*$). Compute the partial t-statistics T_j for testing that the newly added jth predictor ($j \neq j^*$) in each MLR is significant via (7.16). Find the largest $|T_j|$ and calculate its corresponding two-sided P-value P_j. Admit the associated x_j if $P_j \leq \alpha_{\text{en}}$. If no $|T_j|$ can satisfy $P_j \leq \alpha_{\text{en}}$, terminate the stepwise procedure.

FS.3 Now check for step-down variable exiting: in the selected two-variable MLR model, compute the partial t-statistic T_{j^*} via (7.16) for testing that the earlier j^*th predictor remains significant. Find its two-sided P-value P_{j^*} and *remove* x_{j^*} if $P_{j^*} > \alpha_{\text{ex}}$. Otherwise, continue with step-up variable admission.

FS.4 In the general case, suppose $J < P$ variables currently reside in the selected configuration, indexed by $j^* = 1, \ldots, J$. Fit all possible MLR models at $p = J + 1$, using the J selected predictors x_{j^*} and a newly added x_j $(j \neq j^*)$. Compute the partial t-statistics T_j for testing that the newly added jth predictor $(j \neq j^*)$ in each MLR is significant via (7.16). Find the largest $|T_j|$ and calculate its corresponding two-sided P-value P_j. Admit the associated x_j if $P_j \leq \alpha_{\text{en}}$. If no $|T_j|$ can satisfy $P_j \leq \alpha_{\text{en}}$, terminate the stepwise procedure.

FS.5 Check for step-down variable exiting. Denote the newly added predictor in the current $(J + 1)$-variable MLR model as $x_{j'}$. Compute the J partial t-statistics T_{j^*} via (7.16) for $j^* \neq j'$. Find the *smallest* $|T_{j^*}|$ and calculate its corresponding two-sided P-value P_{j^*}. Remove x_{j^*} if $P_{j^*} > \alpha_{\text{ex}}$. Then, continue with variable addition in Step FS.4.

In practice, choices for the entry and exit levels can vary greatly. Values in the 5–15% range are not uncommon, although increasing into the 20–35% range may be reasonable if the goal is to produce a large list of putative predictors for deeper examination.

A number of other strategies exist for automated variable section. A simple variant of forward stepwise regression is *forward selection*, where no step-down exiting is performed (i.e., Steps FS.3 and FS.5 are omitted from the above FS algorithm). Or, one can start with *all* P predictor variables in the MLR model and perform *backward elimination*: find the largest two-sided partial P-value among all the P predictors, and if that P-value exceeds a predetermined α_{ex}, remove the corresponding x-variable from the model. Then, repeat the operation on the remaining $P - 1$ predictors. Continue with step-down elimination until all remaining predictors fail the exit test.

Backward elimination is favored by some analysts because it tends to retain more of the pertinent predictors. (In early stages of forward stepwise algorithms, many of the important predictors have yet to enter into the model. This inflates the SSE and MSE, which in turn drives the entry t-statistics closer to zero. Consequently, step-up admission becomes more difficult and some important predictors may get lost along the way. Such is the lot when performing complex feature selection.) See, for example, Kutner et al. (2005, Section 9.4) or Hastie et al. (2009, Section 3.3) for further commentary.

Example 7.3.3 Cancer mortality multiple regression (Example 7.3.1,continued). Return to the UK cancer mortality data and the variable selection operations in Examples 7.3.1 and 7.3.2. Recall that for the MLR fit on $Y = \log\{\text{Mortality}\}$, the single predictor variables $x_1 = \{\text{Employment rate}\}$, $x_2 = \{\text{Weekly earnings}\}$, $x_3 = \{\text{Volunteering \%}\}$, and $x_4' = \{\text{Smoking cessation rate}\}$ were augmented with the additional, second-order predictors $(x_j - \bar{x}_j)^2$ $(j = 1, \ldots, 4)$ and $x_j x_k$ $(j \neq k)$.

To illustrate feature selection, consider simple backward elimination on the full set of 14 potential predictor variables. A convenient way to conduct the calculations is via **R**'s `drop1()` function. This fits all the individual predictors in a supplied MLR model and computes a table of the changes in fit when any term is singly dropped from the model. A `test='F'` option applies a partial F-test; with one numerator d.f., this is equivalent to the desired partial t-test

and gives identical P-values. Repeated application of `drop1()` to the serially reducing models mimics the backward elimination strategy.

For example, start with the full, 14-predictor model via the sample code

```
> mlrBaseFM.lm <- lm( Y ~ x1 + x2 + x3 + x4 + x1sq + x2sq +
                          x3sq + x4sq + x12 + x13 + x14 +
                          x23 + x24 + x34 )
> mlr14.drop <- drop1( mlrBaseFM.lm, test='F' )
> Pvals14 <- mlr14.drop[,6]
> row.names(mlr14.drop)[which(Pvals14==max(Pvals14,na.rm=T))]
```

The `mlr14.drop` object contains the table of pointwise partial P-values for testing each individual variable's impact on the model. The remaining commands simply output the name of the predictor associated with the largest P-value (output suppressed). If its corresponding P-value is greater than α_{ex}, remove that predictor from the model. (If not, stop and retain the entire set of 14 predictors.) Here, we operate with the elimination level set to $\alpha_{ex} = 0.15$.

Next, eliminate the single predictor recommended for removal by taking advantage of **R**'s `update()` function, and 'back' down to the next set of 13 P-values. Sample code with these data is

```
> mlr13.drop <- drop1( update(mlrBaseFM.lm, .~.-x34), test='F' )
> Pvals13 <- mlr13.drop[,6]
> row.names(mlr13.drop)[which(Pvals13==max(Pvals13,na.rm=T))]
```

Continue until no further predictor variables are flagged for elimination.

Table 7.3 summarizes the results of the backward elimination steps for these data. As seen therein, selected second-order cross-products are the first to be eliminated, followed by many of the quadratic terms. By the time $p = 7$ variables remain, the first-order term $x_2 = \{\text{Weekly earnings}\}$ is marked for elimination.

At this point, with $\alpha_{ex} = 0.15$, backward elimination produces a quandary: variable x_2 presents the highest remaining P-value at $P_2 = 0.163$. As this is (just) greater than α_{ex}, the algorithm calls for elimination of x_2. It still retains, however, the predictors $(x_2 - \bar{x}_2)^2$ and x_1x_2. Recall that standard practice requires retention of any lower-order terms that support their corresponding higher-order terms when left in a model. Thus x_2 should be retained, even though backward elimination now flags it for removal.

Consequently, the analyst has a number of options:

1. stop here and study the seven-parameter configuration x_1, x_2, x_3, x_4', $(x_2 - \bar{x}_2)^2$, x_1x_2, and x_1x_4' in greater detail. (A familiar collection for this set of features/regressors!);

2. ignore the call to eliminate x_2 until such time – if ever – that both $(x_2 - \bar{x}_2)^2$ and x_1x_2 are removed, and continue backward elimination by removing the highest P-values not associated with the single x_2 variable (there is no `keep=` option in `drop1()`, which would simplify such a strategy); or

3. blindly remove x_2 and continue with the elimination; should either or both second-order terms with x_2 be retained when elimination concludes; attempt to interpret the model without the supporting x_2 lower-order term.

Option (i) is perhaps most reasonable, because the larger goal of this analysis is to identify a feature configuration worthy of continued study, and this subset seems to be popular.

Table 7.3 Summary of backward elimination steps for variable selection with cancer mortality MLR analysis in Example 7.3.3.

No. variables in model	Model components	Maximum P-value	Targeted x-variable	Action
14	(Full model)	$P_{14} = 0.998$	x_3x_4'	Eliminate x_3x_4'
13	$x_1, \ldots, x_4'$, all squares, $x_1x_2, \ldots, x_2x_4'$	$P_{13} = 0.984$	x_2x_4'	Eliminate x_2x_4'
12	$x_1, \ldots, x_4'$, all squares, $x_1x_2, \ldots, x_2x_3$	$P_{12} = 0.967$	x_2x_3	Eliminate x_2x_3
11	$x_1, \ldots, x_4'$, all squares, $x_1x_2, \ldots, x_1x_4'$	$P_8 = 0.927$	$(x_4' - \overline{x}_4')^2$	Eliminate $(x_4' - \overline{x}_4')^2$
10	$x_1, \ldots, x_4'$, $(x_1 - \overline{x}_1)^2$, $(x_2 - \overline{x}_2)^2$, $(x_3 - \overline{x}_3)^2$, $x_1x_2, \ldots, x_1x_4'$	$P_5 = 0.751$	$(x_1 - \overline{x}_1)^2$	Eliminate $(x_1 - \overline{x}_1)^2$
9	$x_1, \ldots, x_4'$, $(x_2 - \overline{x}_2)^2$, $(x_3 - \overline{x}_3)^2$, $x_1x_2, \ldots, x_1x_4'$	$P_6 = 0.715$	$(x_3 - \overline{x}_3)^2$	Eliminate $(x_3 - \overline{x}_3)^2$
8	$x_1, \ldots, x_4'$, $(x_2 - \overline{x}_2)^2$, $x_1x_2, \ldots, x_1x_4'$	$P_7 = 0.571$	x_1x_3	Eliminate x_1x_3
7	$x_1, \ldots, x_4'$, $(x_2 - \overline{x}_2)^2$, x_1x_2, x_1x_4'	$P_2 = 0.163$	x_2	(see text)

Note: Elimination level set to $\alpha_{ex} = 0.15$.

(Indeed, a sensible omnibus strategy would apply a variety of different selection algorithms to the collection of potential feature variables and examine how/where they agree or disagree in their recommendations.) Option (ii) could also be considered if one were willing to make the required adjustment to the backward elimination strategy. Option (iii) is not recommended. □

As mentioned in the forward stepwise regression algorithm, stepwise methods that rely on P-values or test statistics typically do not adjust for multiplicity throughout their many (!) decision points/hypotheses regarding variable retention or deletion. As a result, little, if any, inferential value exists in the resulting quantities: the P-values are essentially being used as

objective scores for making the variable selection decisions. A reasonable alternative measure from which to select features that has gained some favor replaces the P-values (or partial t-tests or other traditional devices) with information criteria such as the AIC or BIC. One still applies forward selection, backward elimination, or a combination of the two, but now variables enter or exit based on how small the values of AIC or BIC can become.

In **R**, for example, the `step()` function can conduct forward selection (via its `direction='forward'` option), backward elimination (`direction='backward'`), or both (`direction='both'`) with the goal of minimizing the AIC in the final reported model. (AIC corresponds to the function's `k=2` option. To minimize BIC, input instead `k=log(n)`.) As with any feature-selection output, the final, reported variable configuration is then used as a springboard for greater examination of how the variables affect the observed response.

Example 7.3.4 Cancer mortality multiple regression (Example 7.3.1, continued). Return to the UK cancer mortality data and the variable selection operations in Examples 7.3.1–7.3.3. Recall that for the MLR fit on $Y = \log\{\text{Mortality}\}$, the single predictor variables $x_1 = \{\text{Employment rate}\}$, $x_2 = \{\text{Weekly earnings}\}$, $x_3 = \{\text{Volunteering \%}\}$, and $x_4' = \{\text{Smoking cessation rate}\}$ were augmented with the additional, second-order predictors $(x_j - \bar{x}_j)^2$ $(j = 1, \ldots, 4)$ and $x_j x_k$ $(j \neq k)$.

To illustrate feature selection via IC-based stepwise methods and to compare with the results in Example 7.3.3, consider again simple backward elimination on the full set of 14 potential predictor variables. Now, however, employ minimum-AIC as the selection criterion using `step()` in **R**. Given an `lm` object, say, `mlr14.lm` containing the full set of 14 predictors, the command is simply

```
> step( mlr14.lm, direction='backward', k=2 )
```

The consequent output is a thread of model fits that reports the values of AIC as terms are dropped singly from the full base model in `mlr14.lm`. The complete output is condensed and edited here for space considerations: the initial, base fit is reported as

```
Start:  AIC=-1464.19
Y ~ x1 + x2 + x3 + x4 + x1sq + x2sq + x3sq + x4sq + x12 + x13
                                          + x14 + x23 + x24 + x34
       Df   Sum of Sq       RSS        AIC
- x34   1 0.000000055 4.3312160 -1466.1851
- x24   1 0.000004861 4.3312208 -1466.1848
- x23   1 0.000017250 4.3312332 -1466.1838
- x4sq  1 0.000064731 4.3312807 -1466.1800
- x1    1 0.001205777 4.3324217 -1466.0900
- x1sq  1 0.001234972 4.3324509 -1466.0876
- x3sq  1 0.002005920 4.3332218 -1466.0268
- x13   1 0.005742240 4.3369582 -1465.7320
- x2    1 0.006244338 4.3374603 -1465.6924
- x3    1 0.012891450 4.3441074 -1465.1687
- x12   1 0.015313809 4.3465297 -1464.9781
<none>                4.3312159 -1464.1852
- x14   1 0.026781625 4.3579976 -1464.0769
- x4    1 0.026818793 4.3580347 -1464.0740
- x2sq  1 0.097974879 4.4291908 -1458.5351
```

The output indicates that greatest reduction in AIC occurs with elimination of $x_3 x_4'$.

The program next automatically eliminates x_3x_4' and steps down to assess whether further single-variable elimination can decrease the AIC. After nine step-down eliminations (output suppressed), we achieve

```
Step:  AIC=-1478.73
Y ~ x3 + x4 + x2sq + x12 + x14
       Df  Sum of Sq       RSS        AIC
<none>                4.3752500 -1478.7257
- x12   1 0.07259252 4.4478425 -1475.0979
- x2sq  1 0.15528962 4.5305396 -1468.7976
- x14   1 0.16254693 4.5377969 -1468.2502
- x4    1 0.27616605 4.6514161 -1459.7925
- x3    1 0.92637882 5.3016288 -1415.0445
```

at which point the operation terminates. **R** finds that no additional single-variable deletions can further decrease the AIC.

The reported minimum-AIC configuration from `step()` lists the five predictors x_3, x_4', $(x_2 - \bar{x}_2)^2$, x_1x_2, and x_1x_4'. Here again, however, the inclusion of selected second-order terms requires us to also include their supporting first-order terms. Thus the selected model presents $x_1, x_2, x_3, x_4', (x_2 - \bar{x}_2)^2, x_1x_2$, and x_1x_4' for further examination (see Exercise 7.13). We once again recover the same seven-predictor model seen in previous variable selection calculations with these data. □

The philosophy of applying an automated search method for variable selection in an MLR analysis is not without controversy. Certainly, the automated approach allows for faster and more-efficient study of a large collection of potential predictor variables. A primary argument against it, however, is its commissioning of the decision-making process to the computer – which operates without any domain-specific experience – and away from a data analyst who brings pertinent subject-matter knowledge to the selection process.

Indeed, automated selection necessarily induces selection bias in the eventual model chosen for the MLR (Freedman 1983), often substantially so. *P*-values for the final regression coefficients may be overstated, and associated predictor variables can appear statistically significant when they do not in fact contribute meaningfully to the regression. These concerns are difficult to dismiss. Clearly, blind or inattentive assignment to the computer for final decision(s) on which MLR predictors to employ is foolhardy; no automated variable selection procedure should ever replace informed scientific judgment. In the end, the analyst must ensure that use of computer-driven, automated, feature selection is intended only as a tool and not as the guiding force in the regression modeling effort.

7.4 Alternative regression methods*

When the basic assumptions of the MLR model in (7.1) – statistical independence of the data, normal parent distributions, linear relationships, additive effects, and so on – are in doubt, the quality of the resulting estimators and/or inferences can suffer. A number of alternative fitting strategies can be applied in such cases to provide remedial or robust estimates of the various regression features. A selected assortment is presented in this section.

7.4.1 Loess

If the actual shape/linearity of the mean response $\mu(\mathbf{x})$ is in doubt, a form of nonparametric regression can be applied that draws a smooth curve or surface through the data. Information on $\mu(\mathbf{x})$ is taken from those Y_is at $\mathbf{x}_i$s located near to $\mathbf{x}$, producing a 'locally weighted smoother' for $\mu(\mathbf{x})$. For the single-predictor ($p = 1$) setting, Cleveland (1979) introduced *locally weighted scatterplot smoothing* or 'lowess.' This later evolved for the case of $p \geq 2$ predictor variables into a form of *locally weighted polynomial regression* (Cleveland and Devlin 1988) to estimate the mean response surface. The modern term for the method is *loess*, after the slit-like deposits formed along river banks that resemble 'surfaces' of sorts (Cleveland et al. 1992).

To introduce the concepts, begin with the $p = 1$ case, where Y_i is regressed on a single predictor variable x_i, $i = 1, \ldots, n$. Take $\mu(x) = E[Y]$ and assume $\mu(x)$ is a smooth-but-unknown function of x. Similar to the moving average smoother from Section 3.5.2, loess constructs a local window or neighborhood around each x and fits a straight line to the data pairs (x_i, Y_i) within that window via weighed least squares (WLS, from Section 7.1.2). The weights $w_i = w_i(x)$ vary locally for any given x, and x_is close to x in the window are given heavier weight. Each local window is defined as a fixed proportion $q \in (0, 1)$ of those original x_is closest to x. The local linear fits are then connected across a fine grid of x-values. This produces a smoothed predictor, $\tilde{\mu}(x)$, for the mean response at any value of x in the range of the original data.

In effect, q serves as a smoothing parameter: small values of q draw the local windows tighter and produce a 'bumpy' loess fit; larger values smooth out the fit over the broader range of the data. In practice, q is usually taken between $q = 0.2$ and $q = 0.8$, depending on the particular visualization needs of the analysis. If desired, the first-order linear fit can be replaced with a higher-order polynomial in each local window; local quadratic smoothing is common.

In essence, loess assumes that the mean response $\mu(x)$ can be approximated accurately in some small neighborhood of x via a linear or quadratic (or other polynomial) function. As it is relatively simple to fit low-order polynomials via the MLR approach, the loess algorithm's increased complexity is less onerous than it may at first appear.

Formally, let $d_q[x]$ be the distance from x to the farthest predictor – the $\lfloor qn \rfloor$th 'nearest neighbor' – in its local window. Extending the triangular kernel smoother seen in Section 4.1.4, define the *tricube kernel* as

$$K_{\text{Cu}}(t) = (1 - |t|^3)^3 \; I_{(-1,1)}(t). \tag{7.25}$$

From this, the local weights at the given x are taken as

$$w_i(x) = \begin{cases} K_{\text{Cu}}\left(|x - x_i|/d_q[x]\right) & \text{if } |x - x_i| \leq d_q[x] \\ 0 & \text{otherwise} \end{cases}. \tag{7.26}$$

The loess predicted value $\tilde{\mu}(x)$ is then found by calculating local linear WLS estimates $\tilde{\beta}_{0x}$ and $\tilde{\beta}_{1x}$ using the weights in (7.26) and setting $\tilde{\mu}(x) = \tilde{\beta}_{0x} + \tilde{\beta}_{1x}x$. Repeating this procedure over a dense collection of x values produces smoothed loess predicted values, $\tilde{\mu}(x)$. Plotting these against x helps to visualize the shape, perturbations, and overall nature of the mean response, without call to any specific parametric assumptions on $\mu(x)$.

The loess algorithm extends to local quadratic smoothing by simply applying WLS with a parabolic response $\beta_0 + \beta_1 x + \beta_2 x^2$ at each x. The local WLS estimates become $\tilde{\mu}(x) = \tilde{\beta}_{0x} + \tilde{\beta}_{1x} x + \tilde{\beta}_{2x} x^2$. Local quadratic smoothing can add flexibility to the loess predicted values, with only a slight increase in computational burden.

If the distribution of Y_i can be assumed normal (Gaussian) with homogeneous variation, the loess fitting procedure stops after one WLS pass. If, however, the variation is thought to exhibit heavier-than-normal tail behavior (while remaining symmetric) or if greater robustness is desired in the predicted values, the fit is iteratively updated: from the WLS-loess fitted values $\tilde{Y}_i = \tilde{\mu}(x_i)$, define the initial loess residuals as $\tilde{e}_i = Y_i - \tilde{Y}_i$. The goal is to update the original weights $w_i(x)$ at each x to new values, $\tilde{w}_i(x)$, that diminish the effects of outlying or extreme observations.

First, calculate the absolute residuals $|\tilde{e}_i|$ and find their median, denoted by $\tilde{Q}_{|2|}$. Then, for the *bisquare kernel*

$$K_{\text{Sq}}(t) = (1 - t^2)^2 I_{(-1,1)}(t), \tag{7.27}$$

define the updated weights at each x as

$$\tilde{w}_i(x) = \begin{cases} K_{\text{Sq}}\left(\tilde{e}_i / 6\tilde{Q}_{|2|}\right) & \text{if } \tilde{e}_i \le 6\tilde{Q}_{|2|} \\ 0 & \text{otherwise} \end{cases}. \tag{7.28}$$

With the new weights in (7.28), perform an additional WLS fit to find updated, robust estimates $\breve{\beta}_{0x}$ and $\breve{\beta}_{1x}$ for the SLR parameters in the local window. From these, calculate the updated predicted value as $\breve{\mu}(x) = \breve{\beta}_{0x} + \breve{\beta}_{1x} x$. (If a quadratic polynomial was employed in the initial WLS fit, continue to use it for the regression model in each local window.) To ensure that this robust extension properly downweights any extreme observations, repeat the reweighting a second time to produce the final, robust predicted values $\breve{\mu}(x)$. Plot these against x to visualize the pattern of mean response in the data.

Loess is most appropriate when the observations are dense and numerous, so that the local smoother can adequately estimate the mean response over the range of the data. It may not perform as well if the sample size is small or if the Y_is are sparsely arranged. (This can be especially problematic when $p > 1$; see the following text.)

Example 7.4.1 Type Ia supernovae cosmology. In astronomy, exploding stars known as Type Ia supernovae (SNe Ia) possess very stable luminosity patterns. This allows astronomers to use the objects as a sort of 'standard candle' to calibrate and measure astronomical distances.

As part of a larger study into characteristics of $n = 580$ Type Ia Supernovae, Suzuki et al. (2012) gave data on the relationship between an SNe Ia's redshift and its distance modulus. A stellar object's redshift is the lengthening of its emitted radiation due to differential motion in space: redder if receding, bluer if approaching. In effect, this allows its use as a surrogate measure for the object's relative velocity. By contrast, an object's distance modulus is the difference between its apparent and absolute magnitudes, which produces a measure of 'distance.' Scatterplots of the distance modulus against the redshift are known as 'Hubble diagrams,' from which predictions on an object's distance can be made.

The data appear in Table 7.4. (As above, only a selection is given in the table. The complete set is available at http://www.wiley.com/go/piegorsch/data_analytics.) Figure 7.5 displays the corresponding Hubble diagram. The scatterplot indicates a concave-increasing trend in the

distance modulus. To model curvilinearity in the plot, one could attempt a quadratic polynomial fit as per Section 7.2. For these data, however, the quadratic's strict form of curvature cannot adequately accommodate the concave response in Figure 7.5 (see Exercise 7.17). As an alternative, consider here construction of a loess fit to provide a smooth prediction curve.

Table 7.4 Selected data from a larger set of $n = 580$ paired observations on Type Ia supernovae characteristics.

Supernova code	1993ah	1993ag	1993o	$\cdots$	2002kd	2002ki
x = Redshift	0.02849	0.05004	0.05293	$\cdots$	0.73500	1.14000
Y = Distance modulus	35.34658	36.68237	36.81769	$\cdots$	43.09184	44.19695

Source: Suzuki et al. (2012).

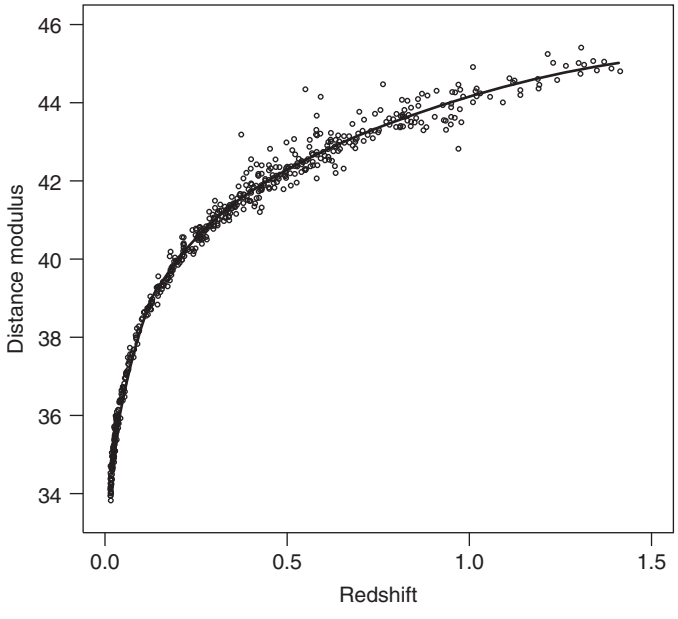

Figure 7.5 Scatterplot of $Y = \{\text{Distance modulus}\}$ versus $x = \{\text{Redshift}\}$ for $n = 580$ Type Ia supernovae from Example 7.4.1. Solid curve overlays robust quadratic loess prediction for $\mu(x)$ with smoothing parameter set to $q = 0.5$. Source: Data from http://supernova.lbl.gov/Union/. Graphic adapted from Suzuki et al. (2012).

For the loess predicted values, work with a second-order smoother, that is, impose no shape assumptions on the mean response $E[Y] = \mu(x)$ but build the loess smooth based on local quadratic regression within each window. In **R**, the `loess()` function can perform second-order loess regression. Sample **R** code is given as follows:

```
> Ytilde.loess <- loess( Y ~ x, span=.5, degree=2,
                                  family='symmetric' )
> Ysmooth <- predict( Ytilde.loess, data.frame(x=seq(0,1.5,.01)) )
```

In the call to `loess()`, the `span=.5` option sets the smoothing parameter to $q = 0.5$ (the default is $q = 0.75$), the `degree=2` option employs a quadratic fit within each window and the `family='symmetric'` option calls for multiple iterations with updated weights as in (7.28) (the default is a single iteration, via `family='gaussian'`). The `family='symmetric'` option was employed here to diminish the effects of possible outlying observations in the data.

The smoothed predicted values $\breve{\mu}(x)$ reside in `Ysmooth`. Plotted against x, this gives a smooth prediction curve and also helps to visualize the trend in the data without forcing any parametric assumptions on $\mu(x)$. This is displayed as an overlay (solid curve) in Figure 7.5.

Exercise 7.17 further explores the loess fit with these data. □

It is fairly straightforward to extend the loess approach to more than one predictor variable: essentially, all the components described above for the single-predictor case apply with multiple predictors. The only substantial change is a need to choose a priori the formal metric, $d[\mathbf{x}_i, \mathbf{x}_h]$, that defines 'distance' between two p-dimensional points $\mathbf{x}_i$ and $\mathbf{x}_h$ in each local window. (Many distance metrics exist. Table 9.8 lists some of the options.) Loess operations usually default to Euclidean, 'as-the-crow-flies' distance

$$d[\mathbf{x}_i, \mathbf{x}_h] = \sqrt{\sum_{j=1}^{p} (x_{ij} - x_{hj})^2}.$$

If the predictor variables differ substantially in their scales or units, it is common to divide them first by a scaling quantity such as their individual standard deviations.

Given a specification for the distance metric, $d[\mathbf{x}_i, \mathbf{x}_h]$, multivariable loess smoothing continues to apply a local WLS–MLR fit to the data in each $(p-1)$-dimensional window or span around the target predictor vector $\mathbf{x}$. The local window is again defined as the fixed proportion q of original $\mathbf{x}_i$s closest to $\mathbf{x}$, where q is typically taken in the range $0.2 \le q \le 0.8$.

Let $d_q[\mathbf{x}]$ now be the distance, defined by the choice for $d[\cdot, \cdot]$, from $\mathbf{x}$ to the farthest predictor vector in its local window. The weights for the WLS–MLR fit in the window around $\mathbf{x}$ are then taken as

$$w_i(x) = \begin{cases} K_{\text{Cu}}\left(d[\mathbf{x}, \mathbf{x}_i]/d_q[\mathbf{x}]\right) & \text{if } d[\mathbf{x}, \mathbf{x}_i] \le d_q[\mathbf{x}] \\ 0 & \text{otherwise,} \end{cases}$$

where $K_{\text{Cu}}(\cdot)$ is the tricube kernel from (7.25). The loess predicted value $\tilde{\mu}(\mathbf{x})$ is then found by calculating the (first-order) WLS–MLR parameter estimates $\tilde{\boldsymbol{\beta}}_{\mathbf{x}}$ from (7.11) in the local window about $\mathbf{x}$ and setting $\tilde{\mu}(\mathbf{x}) = \mathbf{x}\tilde{\boldsymbol{\beta}}_{\mathbf{x}}$. If desired, one can move to second-order MLR models by including quadratic terms and linear cross-products in the regression equation. For example, with two predictor variables x_1 and x_2, the full second-order response surface

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 \tag{7.29}$$

can be fit within each local window. Graphing $\tilde{\mu}(\mathbf{x})$ against $\mathbf{x}$ using surface plots, contour plots, or other dimension-reducing displays can help to visualize the smoothed surface.

If concern exists over potential outliers or heavy-tailed (symmetric) variation, apply the robust extension to the loess fit. As above, find the WLS-loess fitted values $\tilde{Y}_i = \tilde{\mu}(\mathbf{x}_i)$ and calculate the absolute initial loess residuals $|\tilde{e}_i| = |Y_i - \tilde{Y}_i|$. Find the corresponding median absolute residual $\tilde{Q}_{|2|}$ and set the updated weights at each $\mathbf{x}$ to

$$\tilde{w}_i(\mathbf{x}) = \begin{cases} K_{\text{Sq}}\left(\tilde{e}_i/6\tilde{Q}_{|2|}\right) & \text{if } \tilde{e}_i \le 6\tilde{Q}_{|2|} \\ 0 & \text{otherwise,} \end{cases} \tag{7.30}$$

where $K_{Sq}(t)$ is the bisquare kernel in (7.27). With the new weights in (7.30), recover the updated predicted value $\check{\mu}(\mathbf{x}) = \mathbf{x}\check{\boldsymbol{\beta}}_{\mathbf{x}}$. (If a second-order response surface was employed in the initial WLS fit, continue to use it for the MLR model in each local window.) To ensure that this robust extension properly downweights any extreme observations, repeat the reweighting for second time to produce the final, robust predicted values $\check{\mu}(\mathbf{x})$.

A warning: when $p > 1$, loess smoothing can suffer from a 'curse of dimensionality' (Clarke et al. 2009, Section 1.0.1). For instance, moving from one x-variable to two only doubles the number of predictors, but it in effect *squares* the space within which each predictor vector $\mathbf{x}$ lies. A dense collection of n points in one-dimensional space may appear decidedly more sparse in two (or more) dimensions, so a higher-dimensional space must be filled with many more points if the analytic goal includes estimation/visualization of the mean response that progresses through it. The 'curse' only exacerbates as p grows: unless n is *very* large and the observations densely fill the prediction space, it is usually advisable to apply loess smoothing to data sets with just a limited number of predictor variables.

Example 7.4.2 Automobile fuel economy (Example 4.2.6, continued). To illustrate application of loess smoothing with $p = 2$ predictor variables, consider again the automobile fuel economy data in Table 4.6. Recall that these data gave automobile miles driven per gallon of gasoline (MPG) performance for $n = 652$ automatic-transmission vehicles from the 2011 model year.

In Example 4.2.6, focus was on visualizing the data; here, consider extending this to how $Y = \text{MPG}$ relates to number of cylinders and engine displacement (L) when the latter are now viewed as two predictor variables. Rather than assuming any specific form for the mean response $\mu(x_1, x_2)$, however, employ a multidimensional loess smooth. The two scales of measurement for the predictors are somewhat different, so normalize the variables first by their standard deviations:

$$x_1 = \frac{\text{Number of cylinders}}{1.7604} \quad \text{and} \quad x_2 = \frac{\text{Engine displacement}}{1.3268}.$$

One typically starts with a simple scatterplot matrix of the data; however, the plots with $Y = \text{MPG}$ essentially replicate information already seen in the bubble plots from Figure 4.17. Thus a simple x_1, x_2 scatterplot will suffice (see Figure 7.6). The scatterplot shows that the two scaled predictors vary over a band or swath of values from lower left to upper right. (The correlation between the two predictors is 0.908. This suggests possible multicollinearity, but in fact the interrelationship between x_1 and x_2 here is not deleterious; see Exercise 7.5.)

For the loess fit, the potential complexity in the relationships among all the variables argues for use of robust, second-order smoothing. Set the smoothing parameter to $q = 0.7$. Sample **R** code is

```
> MPG.loess <- loess( Y ~ x1 + x2 + x1:x2, span=0.7,
                                  degree=2, family='symmetric' )

> x1grid <- seq( 0.75, 9, length=100 )
> x2grid <- seq( 0.75, 6, length=100 )
> Ysmooth <- matrix(0, nrow=100, ncol=100)
> for(i in 1:100) {
                    for(j in 1:100) {
          Ysmooth[i,j] <- predict( MPG.loess,
              newdata=data.frame(x1=x1grid[i], x2=x2grid[j] )  )
```

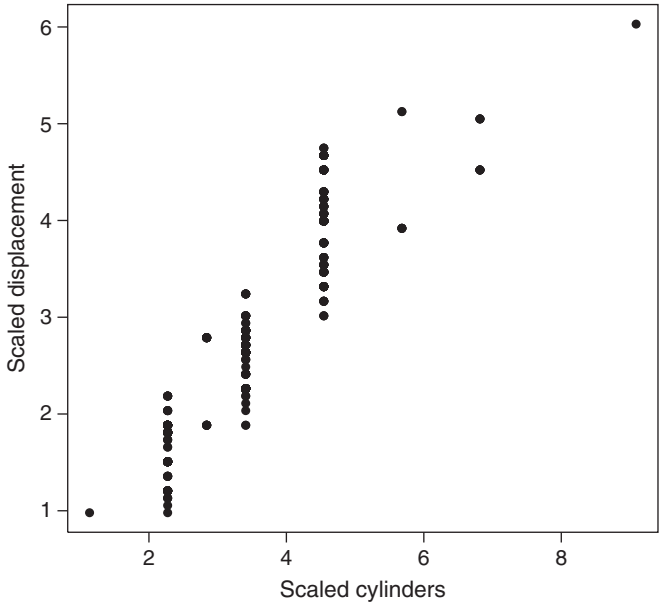

Figure 7.6 Scatterplot of $x_2 = \{\text{scaled Engine Displacement}\}$ against $x_1 = \{\text{scaled Number of Cylinders}\}$, from Example 7.4.2. Scaling is by individual standard deviations. Source: Data from http://www.fueleconomy.gov/feg/download.shtml.

```
        if ( x2grid[j] < max(0.8, -.75+0.7*x1grid[i]) ||
                    x2grid[j] > min(6.1, 1.1*x1grid[i]) )
                                Ysmooth[i,j] <- NA
                            } # end for j loop
                    }  # end for i loop
> contour( x=x1grid, y=x2grid, z=Ysmooth, levels=seq(12,44) )
```

In the originating call to `loess()`, the model `formula` now includes the two predictor variables and, for completeness, an interaction term `x1:x2` (a shorter, more efficient syntax in **R** is `Y ~ x1*x2`). The code lays out a polygonal grid of (x_1, x_2) points, over which the $\mu(x_1, x_2)$ surface is to be estimated. The complicated `for` statement restricts the construction to that swath of points. This corresponds roughly to the two-dimensional range of the predictor values. (A simpler rectangular grid would produce some extreme extrapolations: no automobiles are produced with, say, 16 cylinders but only 2 L of engine displacement.)

Figure 7.7 presents the subsequent plot from `contour()`. An overall decreasing pattern in estimated MPG is seen from bottom to top and, to a lesser extent, from right to left in the plot. This is consistent with our general expectations; however, the contours also appear to show a ridge-like structure arching down the surface as displacement increases. The relationship between mean MPG and the two predictors exhibits some intriguing complexities that may require additional study. (Exercise 7.20 explores the loess fit with these data in further detail.)

Readers are encouraged to experiment with **R**'s other plotting routines to better visualize the predicted contours/surface. For example,

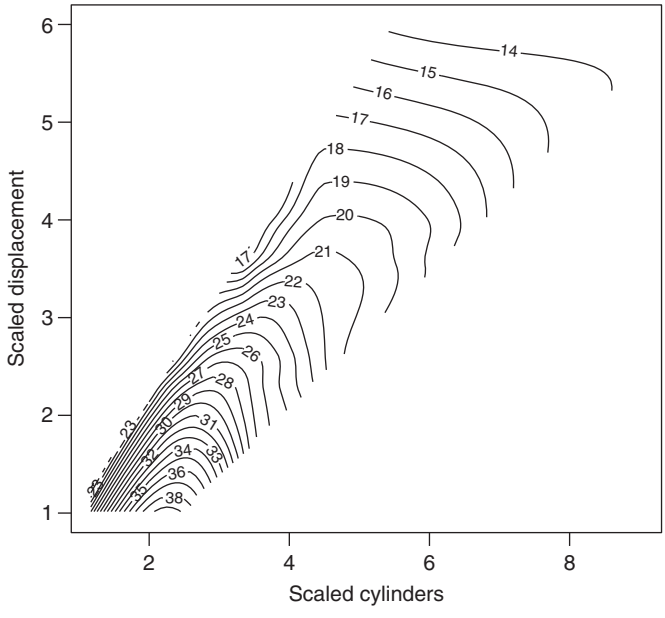

Figure 7.7 Contours of estimated mean MPG using robust, second-order, loess prediction in Example 7.4.2. Smoothing parameter is set to $q = 0.7$. Source: Data from http://www.fueleconomy.gov/feg/download.shtml.

```
> filled.contour( x=x1grid, y=x2grid, z=Ysmooth,
                        color.palette=terrain.colors )
```

gives a colored plot (not shown) that may improve the view of the changing surface features. □

A substantial literature exists on loess and its use in data analytics; useful sources include Gijbels and Prosdocimi (2010), Givens and Hoeting (2013, Section 11.4.1), and the early introduction by Cleveland et al. (1992).

7.4.2 Regularization: ridge regression

In some cases, the basic regression model may exhibit ill-posed or otherwise unstable features, such as high multicollinearity/correlations among the predictor variables, or very large numbers of predictors with p close to or even exceeding n. In situations such as these, the $\mathbf{X}^T\mathbf{X}$ matrix at the core of the ordinary LS estimator in (7.6) can be driven to a state of *ill condition* (Gentle 2007, Section 9.4.1). The pejorative term here is meaningful. If $\mathbf{X}^T\mathbf{X}$ is ill-conditioned, it can appear almost singular. The resulting inverse matrix is numerically unstable and detrimentally affects the LS estimator in (7.6): small changes in the data can lead to wild swings in the estimated β-parameters or in the predicted values.

An approach for reducing the effects of ill-conditioning is known under the larger moniker of *regularization* (Lukas 2012). The goal is to stabilize the regression parameter estimates by penalizing those that grow too large or unwieldy. This essentially drives the point estimates

closer to each other and eventually toward zero. Since the estimates literally shrink to the origin, the process with MLR data is often called *shrinkage regression* (Sundberg 2012).

Formally, to regularize a multicollinear or otherwise ill-conditioned set of MLR predictors, we apply a form of penalized least squares: add a penalization term to the usual error sum of squares (SSE) and minimize the resulting objective quantity

$$\mathcal{D}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left\{ Y_i - \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) \right\}^2 + \kappa \sum_{j=1}^{p} \mathcal{Q}(\beta_j). \tag{7.31}$$

Here $\mathcal{Q}(\beta_j)$ is a positive penalty function and $\kappa \geq 0$ is an associated tuning or regularization parameter. The added penalty term is sometimes referred to as a Tikhonov factor, and the general approach as *Tikhonov regularization*, after Tikhonov (1963). Properly constructed, the added factor acts to penalize large (absolute) values of the regression parameters and shrink the resulting estimates toward zero. A common choice employs the quadratic penalty function

$$\mathcal{Q}(\beta_j) = \beta_j^2.$$

Readers familiar with constrained minimization will recognize the expression in (7.31) as a form of objective quantity for an optimization method known as *Lagrange multipliers* (Hughes-Hallett et al. 2013, Section 15.3), where κ is the 'multiplier.' An equivalent formulation calls for minimization of the SSE subject to an upper-bound constraint involving $\boldsymbol{\beta}$. That is, minimize $\sum_{i=1}^{n} \{Y_i - (\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij})\}^2$ subject to $\sum_{j=1}^{p} \mathcal{Q}(\beta_j) \leq t_\kappa$. (The notation on the bound t_κ indicates that it and the tuning parameter/multiplier κ will be interrelated. The technical details exceed the scope here; interested readers may refer, for example, to Clarke et al. (2009, Section 10.3.1).)

In practice, it is common to standardize the predictors so that any differences in scale do not interfere with the regularization. Thus we work with the z-scores $z_{ij} = (x_{ij} - \overline{x}_j)/s_j$, where $\overline{x}_j$ is the arithmetic mean of the jth predictor and s_j is the corresponding standard deviation. (Some programs alternatively employ $z_{ij}\sqrt{n/(n-1)}$ or $z_{ij}/\sqrt{n-1}$; the latter gives predictors whose squares sum to 1.)

When the predictors are centered, the LS estimator for the intercept term β_0 is simply $\overline{Y}$. Indeed, because relocating the response variable by adding an arbitrary constant can affect the results, it is also common to center the Y_is: say, $U_i = Y_i - \overline{Y}$. This adds further numerical simplification, since the intercept term no longer enters into the regularization adjustment. (Alternatively, one can simply choose not to include β_0 in the regularization penalty term.)

The subsequent equations simplify somewhat after applying matrix notation, as in Section 7.1.1. Denote the corresponding design matrix (now *without* a leading column of ones for the intercept) as

$$\mathbf{Z} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1p} \\ z_{21} & z_{22} & \cdots & z_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{np} \end{bmatrix}$$

and the corresponding, centered, response vector as $\mathbf{U} = [U_1 \;\; U_2 \cdots U_n]^{\mathrm{T}}$. The revised MLR model takes the form $E[\mathbf{U}] = \mathbf{Z}\boldsymbol{\beta}$ where the $p \times 1$ vector of regression parameters becomes $\boldsymbol{\beta} = [\beta_1 \;\; \beta_2 \;\; \cdots \;\; \beta_p]^{\mathrm{T}}$. With these, the objective function with a quadratic penalty is simply

$$\mathcal{D}(\boldsymbol{\beta}) = (\mathbf{U} - \mathbf{Z}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{U} - \mathbf{Z}\boldsymbol{\beta}) + \kappa \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta}.$$

The resulting normal equations reduce to

$$(\mathbf{Z}^{\mathrm{T}}\mathbf{Z} + \kappa\mathbf{I})\hat{\boldsymbol{\beta}} = \mathbf{Z}^{\mathrm{T}}\mathbf{U},$$

where $\mathbf{I}$ is a $p \times p$ identity matrix. Notice that these have essentially the same form as the original MLR normal equations in (7.5), except that the (constant) tuning parameter κ has been added to the diagonal of the (potentially unstable) $\mathbf{Z}^{\mathrm{T}}\mathbf{Z}$ matrix. This diagonal 'ridge' gives the method its name in statistical parlance: *ridge regression*.

No matter what the condition of $\mathbf{Z}^{\mathrm{T}}\mathbf{Z}$, for any $\kappa > 0$, the diagonal increment in $\mathbf{Z}^{\mathrm{T}}\mathbf{Z} + \kappa\mathbf{I}$ stabilizes the matrix so effectively that it always possesses an inverse. Thus the solution to the ridge regression normal equations is always defined:

$$\hat{\boldsymbol{\beta}}_{\kappa} = (\mathbf{Z}^{\mathrm{T}}\mathbf{Z} + \kappa\mathbf{I})^{-1}\mathbf{Z}^{\mathrm{T}}\mathbf{U}. \tag{7.32}$$

The universal existence of the ridge regression estimator and its consequent ability to address issues of multicollinearity for any MLR model were some of the original motivators for its use (see Hoerl and Kennard 1970).

The tuning parameter κ in (7.32) controls the amount of regularization and shrinkage seen in the eventual parameter estimates. The ordinary LS estimator in (7.6) obtains as $\kappa \to 0$, while as $\kappa \to \infty$, $\hat{\boldsymbol{\beta}}_{\kappa}$ shrinks to a zero vector.

It can be shown (Exercise 7.22) that $E[\hat{\boldsymbol{\beta}}_{\kappa}] = \mathbf{A}_{\kappa}\boldsymbol{\beta}$ where

$$\mathbf{A}_{\kappa} = (\mathbf{I} + \kappa[\mathbf{Z}^{\mathrm{T}}\mathbf{Z}]^{-1})^{-1}.$$

Clearly, if $\kappa > 0$, then $\mathbf{A}_{\kappa} \neq \mathbf{I}$ and, therefore, the ridge estimator is biased! The bias vanishes as $\kappa \to 0$, but it conversely grows as $\kappa \to \infty$. The associated variance of $\hat{\boldsymbol{\beta}}_{\kappa}$ is proportional to $\mathbf{A}_{\kappa}\mathbf{Z}^{\mathrm{T}}\mathbf{Z}\mathbf{A}_{\kappa}^{\mathrm{T}}$, producing a mean squared error (MSE, as variance plus squared bias; see Section 5.2.1) that depends on κ. Hoerl and Kennard (1970) showed that there exists some $\kappa > 0$ such that the MSE of the ridge estimator in (7.32) is strictly smaller than the ordinary LS estimator in (7.6). Unfortunately, the value of this minimizing κ depends on the true value of $\boldsymbol{\beta}$, so it is usually unknown. (Selection of κ is discussed later.) The existence theorem nonetheless indicates a trade-off between increased bias and decreased MSE that can play to the ridge estimator's advantage (Hastie et al. 2009, Section 7.3): by accepting a (small amount of) bias in the point estimator, the overall precision can be enhanced and in the process can improve prediction of $E[\mathbf{U}]$. For problems where some form of regularization is indicated, this gives the ridge estimator and the larger strategy significant validity.

The ridge predicted values using (7.32) are

$$\hat{\mathbf{U}} = \mathbf{Z}\hat{\boldsymbol{\beta}}_{\kappa} = \mathbf{Z}(\mathbf{Z}^{\mathrm{T}}\mathbf{Z} + \kappa\mathbf{I})^{-1}\mathbf{Z}^{\mathrm{T}}\mathbf{U} \tag{7.33}$$

which for

$$\mathbf{H}_{\kappa} = \mathbf{Z}(\mathbf{Z}^{\mathrm{T}}\mathbf{Z} + \kappa\mathbf{I})^{-1}\mathbf{Z}^{\mathrm{T}} \tag{7.34}$$

gives $\hat{\mathbf{U}} = \mathbf{H}_{\kappa}\mathbf{U}$. Thus $\mathbf{H}_{\kappa}$ can be viewed as a form of ridge 'hat' matrix, analogous to the MLR hat matrix $\mathbf{H}$ in (7.7). Recall there that the trace of $\mathbf{H}$, $\mathrm{tr}(\mathbf{H})$, is the number of parameters, p. We often say this represents the 'parameter d.f.' being modeled by the MLR equation. Similarly, the trace of $\mathbf{H}_{\kappa}$ can be shown to equal

$$\mathrm{tr}(\mathbf{H}_{\kappa}) = \sum_{j=1}^{p} \frac{\lambda_j}{\lambda_j + \kappa}, \tag{7.35}$$

where the λ_js are the ordered, decreasing eigenvalues of $\mathbf{Z}^{\mathrm{T}}\mathbf{Z}$. (Eigenvalues are reviewed in Section A.5.) Borrowing the 'trace as d.f.' concept for the ridge regression setting, (7.35) is called the *effective d.f.* of the ridge model fit. Notice that this expression is monotone deceasing in κ and, in particular, as $\kappa \to \infty$ the effective d.f. drop to zero.

In practice, selection of κ is conducted in an exploratory manner. Where possible, one 'trains' the choice of κ with a set of training data – using methods discussed later – and then applies this choice to a separate target or test data set. Separate training data are not always available, however. For choosing κ from a single set of data, a variety of algorithms exist. A popular, standalone estimate was given by Hoerl et al. (1975), who suggested a summary value that divides the residual mean square (MSE) by the harmonic mean of the regression coefficients:

$$\kappa_{\mathrm{HKB}} = p\,\frac{\mathrm{MSE}}{\hat{\boldsymbol{\beta}}^{\mathrm{T}}\hat{\boldsymbol{\beta}}}. \tag{7.36}$$

One can modify this by iterating over successive values of κ, if desired. Also see Cule and De Iorio (2013).

A more-computational, subjective and also more-participatory approach was given by Hoerl and Kennard (1970): they recommended graphing each individual $\hat{\beta}_{j\kappa}$ as a function of $\kappa \geq 0$ on the same plot. Called a *ridge trace*, the plot will show the point estimates shrinking toward each other and toward zero as κ grows. For small κ, the divergence and disparities among the estimated recession coefficients will appear clear, but typically a value of κ will present where the various traces begin to converge. Hoerl and Kennard recommended stabilizing value of κ for use in (7.32). Some authors alternatively plot the traces against the effective d.f. in (7.35), which produces a reversed graphic but otherwise provides similar information. Friendly (2013) gives an extension of the ridge trace using ellipsoids to better visualize the bias-variance trade off.

Perhaps most common in current practice is a form of *cross-validation* (CV) to select κ (Golub et al. 1979). CV removes an observation from the original data set and then uses the remaining data to estimate the value of that excised observation under the proffered model. The concept is similar to the case-deletion strategy with the PRESS statistic in Section 7.3.1. As there, the squared difference between the actual observation and its cross-validated prediction gives a measure of prediction error. Reapplying the single-deletion effort across all n data points and averaging produces a CV error, which can be minimized to select a value of κ. In practice, a number of convenient simplifications and approximations allow the analyst to avoid repeated model fitting to find the case-deleted prediction errors. The result is a generalized cross-validation (GCV) error:

$$\mathrm{GCV}(\kappa) = \frac{1}{n}\sum_{i=1}^{n}\left\{\frac{U_i - \hat{U}_i(\kappa)}{1 - n^{-1}\mathrm{tr}(\mathbf{H}_\kappa)}\right\}^2, \tag{7.37}$$

where for a given κ, $\hat{U}_i(\kappa)$ is the ith element of (7.33) and $\mathrm{tr}(\mathbf{H}_\kappa)$ are the effective d.f. from (7.35). Select the GCV estimator $\kappa_{\mathrm{GCV}} \geq 0$ to minimize (7.37).

In **R**, a number of packages provide functions that conduct ridge regression. The most basic is the `lm.ridge()` function in the *MASS* package. Others include the `ridge()` function in the external *genridge* package, the `linearRidge()` function in the external *ridge* package, and the `glmnet()` function in the external *glmnet* package.

Example 7.4.3 Ridge regression with genetic SNP data. Regularization methods such as ridge regression can be useful for analyzing large data sets in modern genetics. The response variable might be a phenotypic outcome that is regressed against a variety of genotypic input variables. The goal is to identify potential associations between the genotypic variants and the phenotypic response. A popular genetic regressor variable involves different single nucleotide polymorphisms (SNPs) identified along a species' genome (Austin et al. 2013; Cule et al. 2011). A typical SNP has three levels coded as 0, 1, or 2 to represent copies of the minor allele.

It is not uncommon for a set of p SNP predictors to exhibit high multicollinearity and/or represent many regressor variables relative to the number of subjects n (the so-called 'large-p/small-n' problem, which can nonetheless involve a not-so-small n). Standard MLR analysis is contraindicated. The ridge penalty in (7.32) can still produce a viable regression estimator, however, endowing the methodology with substantive applicability for this estimation problem.

For a moderate-scale illustration, take the sample `GenCont` data provided in **R**'s external *ridge* package. The observations are given as $n = 500$ outcomes for a continuous phenotypic response, Y_i, generated via the Fregene software program (Chadeau-Hyam et al. 2008). A set of $p = 11$ SNPs are reported as the regressor variables. A selection of the data appear in Table 7.5. (The complete set is available by loading the *ridge* package, or at http://www.wiley.com/go/piegorsch/data_analytics.)

Table 7.5 Selected phenotypic responses (Y_i) and SNP genetic predictors from a larger set of $n = 500$ observations generated from the Fregene software program (Chadeau-Hyam et al. 2008).

Y_i	SNP_{i1}	SNP_{i2}	SNP_{i4}	$\cdots$	$\text{SNP}_{i,11}$	$\text{SNP}_{i,12}$
1.4356316	1	1	0	$\cdots$	1	0
2.9226960	1	0	0	$\cdots$	1	0
0.5669319	0	0	0	$\cdots$	2	0
4.8515051	1	0	0	$\cdots$	1	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
0.4047265	0	0	0	$\cdots$	2	0
-0.1720107	0	0	0	$\cdots$	1	0

Note: The SNP_3 variable replicates existing information in the data set and is not listed. Source: `data(GenCont)` from **R** *ridge* package.

Application of a standard MLR model is possible with these data, although one quickly sees that multicollinearity is present. The sample **R** code

```
> require( car )   #load external car package for vif() function
> vif( lm(Phenotypes ~ SNP1 + SNP2 + SNP4 + SNP5 + SNP6 + SNP7
                       + SNP8 + SNP9 + SNP10 + SNP11 + SNP12) )
```

produces the following VIFs from (7.20) for the individual SNP predictors:

```
      SNP1       SNP2       SNP4       SNP5       SNP6       SNP7
 19.113512 45.075246  1.024255 12.613304 33.850875  1.010278
      SNP8       SNP9     SNP10      SNP11      SNP12
  1.311907  1.007684 1.244746  1.814737  1.011572
```

A number of these VIFs exceed the action limit of 10, some substantially so, and indeed, even their mean exceeds 10: $\overline{\text{VIF}} = 10.825$.

To address this obvious multicollinearity, a regularization method such as ridge regression could be applied. For instance, the `lm.ridge()` function in the *MASS* package provides a basic ridge analysis. Start by centering the response variable and standardizing the predictors. (To correspond with the default scaling in `lm.ridge()`, the following sample **R** code sets the predictors to $z_{ij} = \sqrt{n/(n-1)}(x_{ij} - \bar{x}_j)/s_j$.)

```
> U <- Phenotypes - mean(Phenotypes)
> z1 <- sqrt(n/(n-1)) * scale( SNP1 )
> z2 <- sqrt(n/(n-1)) * scale( SNP2 )

    ⋮

> z11 <- sqrt(n/(n-1)) * scale( SNP12 )
```

Next, set a range of values for κ (here, $0 < \kappa \leq 50$), and call the `lm.ridge()` function. (The `lambda=` option calculates the regularized fit over each of the values for κ.)

```
> kappa = seq(.01,50,.01)
> GenCont11.ridgelm = lm.ridge( U ~ z1 + z2 + z4 + z5 + z6
                  + z7 + z8 + z9 + z10 + z11 + z12, lambda=kappa )
```

The `GenCont11.ridgelm` object now contains the various summary statistics used for selecting an operative value for κ. In particular, a quick summary is available via `select(GenCont11.ridgelm)`, producing (edited) output

```
modified HKB estimator is 9.40075
smallest value of GCV  at 22.05
```

That is, the HKB estimate from (7.36) ('HKB' stands for Hoerl, Kennard, and Baldwin, the authors of the originating 1975 article) is roughly $\kappa_{\text{HKB}} = 9.4$, while the minimum-GCV error occurs at $\kappa_{\text{GCV}} = 22.05$. A ridge trace plot helps to visualize the former suggestion: simply use `plot(GenCont11.ridgelm)` for a stock graphic or construct the plot directly by overlaying the individual $\hat{\beta}_{j\kappa}$ values available in the **R** vectors

```
GenCont11.ridgelm$coef[j,]
```

($j = 1, \ldots, p$). The latter strategy produces the trace plot in Figure 7.8. Notice how the trace curves flatten and stabilize as κ grows large, with a clear effect occurring after about $\kappa_{\text{HKB}} = 9.4$. Similarly, a graph of the GCV error created by plotting

```
GenCont11.ridgelm$GCV
```

against κ clearly shows the minimum at $\kappa_{\text{GCV}} = 22.05$ (see Figure 7.9).

From this analysis, either $\kappa_{\text{HKB}} = 9.4$ or $\kappa_{\text{GCV}} = 22.05$ could be taken as reasonable operating values for κ. Using the latter, reimplement `lm.ridge()`:

```
> GenCont.ridgelm <- lm.ridge( U ~ z1 + z2 + z4 + z5 + z6
                  + z7 + z8 + z9 + z10 + z11 + z12, lambda=22.05 )
> GenCont.ridgelm$coef
```

The resulting output gives the estimated regression coefficients:

```
       z1       z2       z4       z5       z6       z7
   0.3651 -0.1016   0.0273   0.7736 -0.1706 -0.0179
       z8       z9      z10      z11      z12
  -0.0707   0.0169   0.0280   0.0414 -0.0367
```

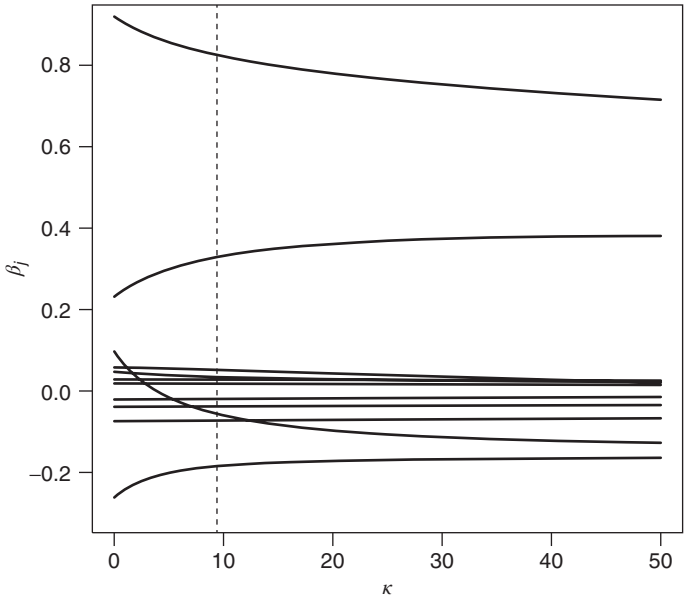

Figure 7.8 Ridge trace plot over values of tuning parameter κ for the Genetic SNP data in Example 7.4.3. Dashed horizontal line marks location of HKB estimator $\kappa_{\text{HKB}} = 9.4$. Source: Data from `data(GenCont)` in **R** *ridge* package.

From these, the minimum-GCV predicted values $\hat{Y}_i(\kappa_{\text{GCV}})$ can be constructed using (7.33):

```
> Zmtx <- as.matrix( cbind(z1,z2,z4,z5,z6,z7,z8,z9,z10,z11,z12) )
> Yhat <- Zmtx %*% GenCont.ridgelm$coef + mean(Phenotypes)
```

(The sample **R** code here adds back $\overline{Y}$ to recover the original scale.) Study of the $\hat{Y}_i(\kappa_{\text{GCV}})$ values may identify potential associations with the SNP predictors. For example, Figure 7.10 gives single-variable prediction plots of $\hat{Y}_i(\kappa_{\text{GCV}})$ against z_1 (Figure 7.10a) and z_5 (Figure 7.10b). Positive associations are evidenced between the predicted phenotypic outcome and the individual SNP predictors, suggesting the need for further investigation.

Exercise 7.24 explores other features of the ridge regression analysis with these data. □

Ridge regression is employed primarily to improve prediction of the mean response in the face of high multicollinearity and/or when the number of predictor variables is very large. In the latter case, it becomes a useful strategy to help avoid overfitting. At its core, however, it is primarily a prediction/estimation methodology. Construction of standard errors or inferences such as confidence limits is less common, due to the bias in $\hat{\boldsymbol{\beta}}_\kappa$. (Standard errors for biased point estimators can give an indication of the estimation uncertainty, but they may also give a distorted impression of that uncertainty because they may not take the bias into account.)

If confidence intervals/statements on, say, the regression coefficients are of absolute necessity, suggestions have been made to apply bootstrap methods with the ridge fit (Crivelli et al. 1995), as in Section 5.3.6. Obenchain (1977) discussed some other possibilities. In general, caution is advised: any confidence region method must carefully incorporate all sources of

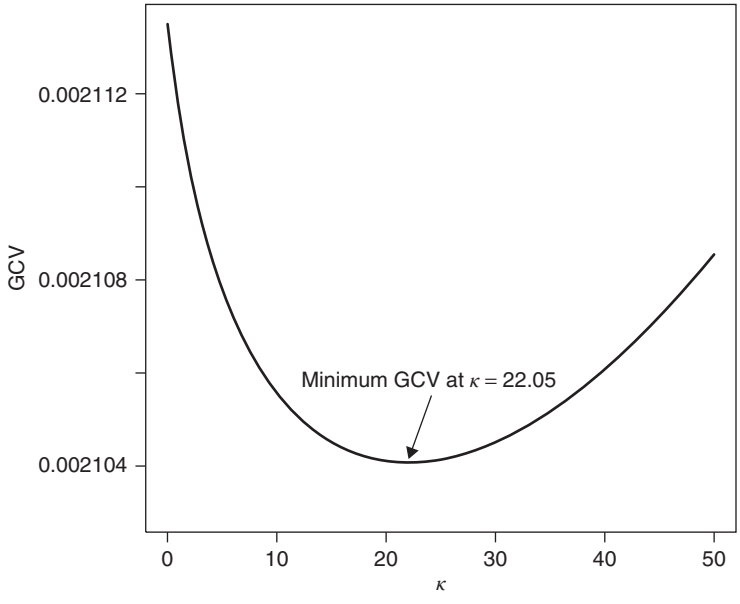

Figure 7.9 Generalized cross validation (GCV) error over values of tuning parameter κ for the genetic SNP data in Example 7.4.3. Minimum GCV identified at $\kappa_{\text{GCV}} = 22.05$. Source: Data from `data(GenCont)` in **R** *ridge* package.

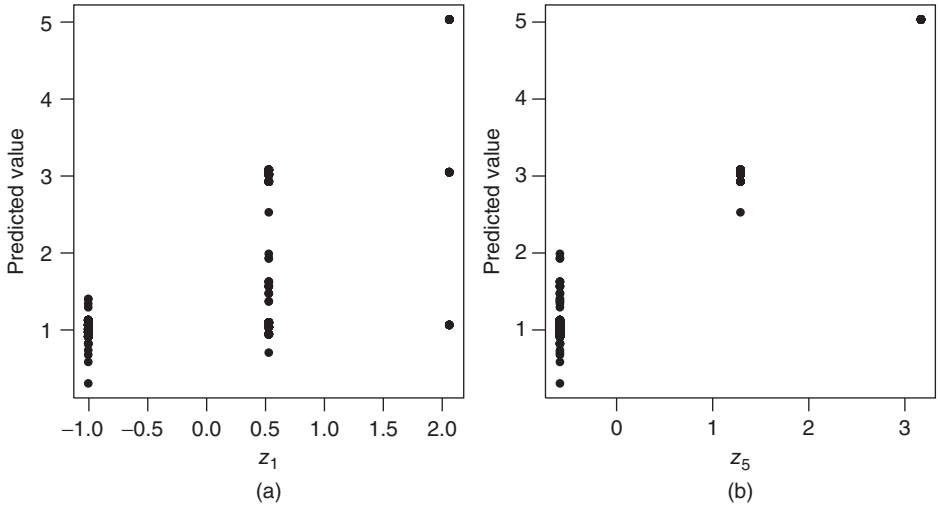

Figure 7.10 Individual predicted value plots against selected SNP predictors (a) z_1 and (b) z_5 for the genetic SNP data in Example 7.4.3. Source: Data from `data(GenCont)` in **R** *ridge* package.

uncertainty when constructing statements about the model components in (biased) shrinkage regression.

7.4.3 Regularization and variable selection: the Lasso

Another popular penalty term for use in the regularization objective function (7.31) is the (absolute) first-order penalty

$$\mathcal{Q}(\beta_j) = |\beta_j|,$$

often called an 'L_1' penalty because it mimics a first-order length measure (or 'norm') in mathematics (Vidaurre et al. 2013). (With its appeal to sums of squared terms, ridge regression corresponds to an L_2 penalty.) Thus the new objective function to be minimized is

$$\mathcal{D}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left\{ Y_i - \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) \right\}^2 + \kappa \sum_{j=1}^{p} |\beta_j|. \tag{7.38}$$

Tibshirani (1996) proposed (7.38) for use with MLR data when a large number of predictors leads to problems with unstable or overdetermined structure. Similar to the L_2 ridge setting, minimizing (7.38) is equivalent to a constrained optimization:

$$\text{minimize} \sum_{i=1}^{n} \left\{ Y_i - \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) \right\}^2 \text{ subject to } \sum_{j=1}^{p} |\beta_j| \le t_\kappa, \tag{7.39}$$

where again, the positive bound t_κ will be related to the tuning parameter κ.

The penalized optimization operates somewhat differently here than in ridge regression. There is no closed-form expression for the estimator $\hat{\boldsymbol{\beta}}$, and a numerical solution is required. The result is a more-abrupt form of shrinkage, with some $\hat{\beta}_j$ coefficients driven exactly to zero as κ grows. Thus the L_1-penalized fit is more *sparse*, that is, it contains only a subset, and sometimes a substantially reduced subset, of the original p predictors. The regularization, therefore, serves both as a form of shrinkage regression and as a sort of variable selector, because it literally zeros out the contribution of low-impact or low-consequence predictor variables. In effect, the variable selection is conducted by penalizing on the regression coefficients rather than on the number of parameters (Chen et al. 2014). Tibshirani called this procedure the *least absolute shrinkage and selection operator* or 'Lasso' for short.

When substantial multicollinearity is present among the predictors, ridge regression will usually be more effective in terms of its predictive capability (Tibshirani 1996); however, the ability to combine shrinkage regression with *de facto* variable selection makes the Lasso a useful addition to the MLR toolkit. As with ridge regression, the data are usually assumed to have been centered, $U_i = Y_i - \overline{Y}$, and the predictor variables to have been standardized into z-scores $z_{ij} = (x_{ij} - \overline{x}_j)/s_j$. As a result, the intercept term β_0 estimates exactly as zero.

Selection of κ in (7.38) – or equivalently t_κ in (7.39) – is usually conducted via appeal to CV, similar to the approach taken with the ridge regression tuning parameter in Section 7.4.2. In the simplest case, one removes each observation from the original data set and then uses the remaining data to estimate the value of that excised observation under a series of candidate values for κ. With very large n, this becomes computationally costly, so an efficient modification breaks the data into K equal subsamples. Over $k = 1, \ldots, K$, the kth subsample is sequestered for validation purposes and the remaining $K - 1$ subsamples are collected into a training set

to estimate the responses in that kth subsample. The squared differences between the actual observations and their cross-validated predictions measure the prediction error. Reapplying this K-fold deletion across all K subsamples and averaging produces a CV error. Select the κ that minimizes this error estimate. This method is known as *K-fold cross-validation*. If $K = n$, it collapses to single-case deletion ('leave-one-out') CV. For further introductions to CV in statistical learning, see, for example, Clarke et al. (2009, Section 1.3.2) or James et al. (2013, Section 5.1).

As one might imagine, application of the Lasso when n and/or p is large is not trivial. Strategies for numerical implementation vary, although ongoing advances have drastically decreased the computational burden. A preferred algorithm is known as 'coordinate descent' (Friedman et al. 2007; Wu and Lange 2008). In **R**, this is available via the `glmnet()` function in the external *glmnet* package.

Example 7.4.4 Ridge regression with genetic SNP data (Example 7.4.3, continued). Continuing with the genetic SNP data from Example 7.4.3, consider application of the Lasso to the regression with all $p = 11$ SNP predictors. The nontrivial multicollinearity here would normally call for regularization via ridge regression, but it is instructive to examine how the Lasso fit operates with these data. Indeed, there is value here in viewing the exercise from a variable-selection perspective.

To allow for comparison with Example 7.4.3, the phenotypic outcome is centered into the response variable U_i and the various predictors are centered and scaled via $z_{ij} = \sqrt{n/(n-1)}(x_{ij} - \overline{x}_j)/s_j$. To apply the Lasso in **R**, invoke `glmnet()`. Sample code is

```
> require( glmnet )
> Zmtx <- cbind( z1,z2,z4,z5,z6,z7,z8,z9,z10,z11,z12 )
> GenCont11.glmnet <- glmnet( x=Zmtx, y=U,
                              family='gaussian', alpha=1 )
```

where `U` and the `z`-vectors are as given in Example 7.4.3. The call to `glmnet()` requires direct input of the design matrix **Z**, here as `x=Zmtx`. Notice that no intercept is included in the matrix. The specification `y=U` identifies the (centered) response variable. The `family='gaussian'` option indicates that the data are continuous measurements suitable for modeling with a normal distribution. (Other options exist for nonnormal data; see `help(glmnet)`.) The `alpha=1` option institutes the Lasso L_1 penalty.

The available outputs from `glmnet` are substantial and span a range of uses; simplest for our purposes here is a stock graphic of the $\hat{\beta}_j$ coefficient profiles:

```
> plot( GenCont11.glmnet, xvar='lambda', label=T )
> abline( h=0, lwd=3 )
```

The `xvar='lambda'` option plots the estimated coefficients against $\log(\kappa)$ (the log scale is recommended for better visualization), while the `label=T` option inserts (small) markers at the end of each trace indicating the associated predictor. The subsequent call to `abline()` overlays a thick horizontal line at $\beta = 0$ (see Figure 7.11).

The coefficient profile plot illustrates the expected shrinkage as $\log(\kappa)$ grows, along with the Lasso's characteristic contraction of each $\hat{\beta}_j$ to exactly zero with increasing penalization. Viewed from a variable-selection perspective, most of the SNP predictors drop fairly quickly, with only three variables remaining by $\log(\kappa) = -3$, that is, $\kappa \approx 0.05$. (The digits at the top of the plot help in this regard: they indicate how many nonzero predictors remain in the model at

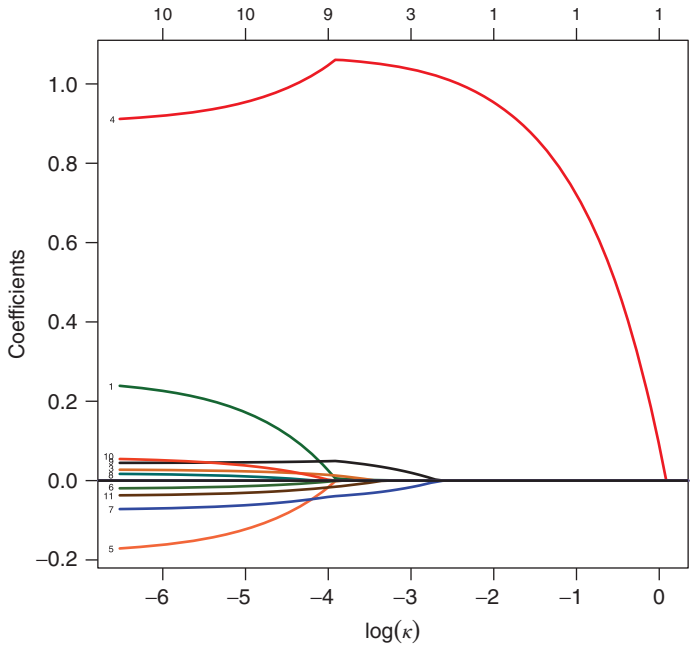

Figure 7.11 Lasso-penalized regression coefficient profiles for the genetic SNP data in Example 7.4.4, plotted as a function of $\log(\kappa)$. Solid horizontal line indicates $\beta_j = 0$. Source: Data from `data(GenCont)` in **R** *ridge* package.

the corresponding value of $\log(\kappa)$ on the horizontal axis.) These three predictors are, in order of retention, z_4 and then (essentially tied) z_7 and z_9. At, for example, $\kappa = 0.05$, the nonzero coefficients can be estimated via the **R** command `coef(GenCont11.glmnet,s=0.05)`, where the `s=0.05` option dictates the specific value of κ at which `glmnet` generates the estimates. The resulting output (edited) is

```
(Intercept)   2.984784e-16
 V4           1.036733e+00
 V7          -1.643679e-02
 V9           1.952590e-02
```

(Notice that the intercept, as expected, estimates as essentially machine zero.) Further investigation would be warranted to determine which genetic features these three predictors were marking.

Selection of κ via K-fold CV is available using *glmnet*'s `cv.glmnet()` subfunction:

```
> GenCont11.cv.glmnet <- cv.glmnet( x=Zmtx, y=U, nfolds=20 )
```

The CV is conducted over an internally selected range for $\kappa > 0$; the `nfolds=20` option here sets $K = 20$. Upon output, the CV error is contained in the vector `GenCont11.cv.glmnet$cvm`; the corresponding values for κ are in `GenCont11.cv.glmnet$lambda`. A plot of the error against $\log(\kappa)$ appears in Figure 7.12. (One can alternatively generate a pre-supplied stock graphic via `plot(GenCont11.cv.glmnet)`. See `help(plot.cv.glmnet)`.)

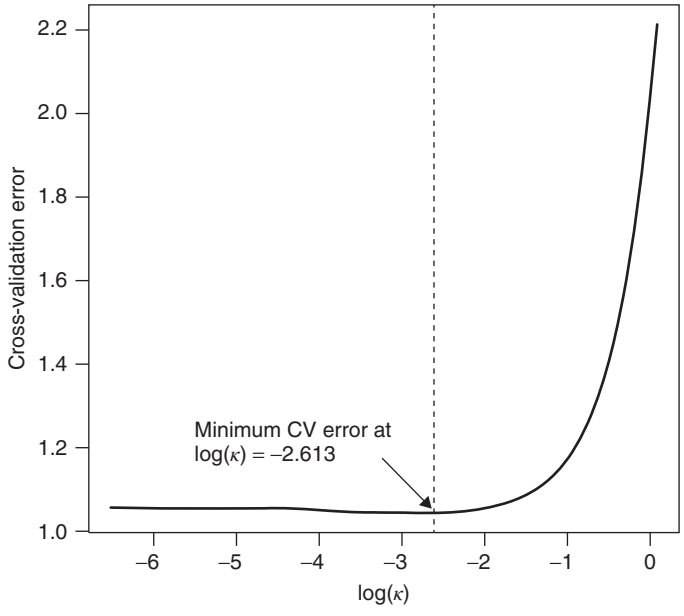

Figure 7.12 K-fold cross-validation error at $K = 20$ for the genetic SNP data in Example 7.4.4, plotted as a function of $\log(\kappa)$. Dashed vertical line indicates minimum error at $\log(\kappa) = -2.6130$. Source: Data from `data(GenCont)` in **R** *ridge* package.

As indicated in the figure, the CV error is minimized with these data at $\log(\kappa) = -2.6130$, that is, $\kappa = 0.0733$. (This is also calculated via `GenCont11.cv.glmnet$lambda.min`.) Referring back to Figure 7.11, this corresponds to shrinkage of all but one coefficient to zero: the only SNP predictor selected by the Lasso at the minimum (20-fold) CV is z_4. Its estimated regression coefficient can be conveniently accessed via

```
> coef( GenCont11.cv.glmnet, s='lambda.min' )
```

producing (output edited)

```
 (Intercept)  2.909530e-16
  V4          1.015418e+00
```

□

A variety of extensions on the regularization concept have appeared in the statistical learning literature. For example, explicitly incorporating a power parameter $q \geq 0$ into the penalization term produces the objective quantity

$$\mathcal{D}_q(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left\{ Y_i - \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) \right\}^2 + \kappa \sum_{j=1}^{p} |\beta_j|^q$$

(Frank and Friedman 1993), now referred to as *bridge regression* (Fu 1998). Or, an expanded class of shrinkage estimators that includes both the Lasso and ridge regression is known as *elastic net regularization* (Zou and Hastie 2005). This extends the penalty function in (7.38)

into a convex combination of L_1 and L_2 terms:

$$\mathcal{D}_\alpha(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left\{ Y_i - \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) \right\}^2 + \kappa \left\{ \alpha \sum_{j=1}^{p} |\beta_j| + (1-\alpha) \sum_{j=1}^{p} \beta_j^2 \right\}$$

for some selected $\alpha \in [0, 1]$. At the extremes, $\alpha = 0$ produces ridge regression while $\alpha = 1$ yields the Lasso. The `glmnet()` function in the external *glmnet* package performs elastic net regularization via its `alpha=` option.

For more on regularization/shrinkage in linear regression problems, see Hastie et al. (2009, Section 3.4) or Sundberg (2012).

7.5 Qualitative predictors: ANOVA models

When the predictors thought to affect the mean response are qualitative rather than quantitative – for example, sex or ethnic status in a marketing survey, or patient condition when categorizing medical records – modifications are required in the MLR analysis. A natural approach is to assign a set of codes to the different levels of the qualitative variable(s) and build from these a series of quantitative predictor variables. For instance, set $x_{i1} = 1$ if the ith observation corresponds to the first level of the qualitative predictor (zero otherwise), $x_{i2} = 1$ if the ith observation corresponds to the second level (zero otherwise), and so on. In this manner, multiple linear predictors can be constructed to account for qualitative as well as quantitative predictor variables.

With qualitative predictors, however, interpretation of the regression coefficients as slopes/changes-in-effect becomes ambiguous. As a result, the model is usually written with a more traditional parameterization, called an ANOVA structure. (The name comes from use of an *analysis of variance* to assess differences between the levels of the qualitative factor. How variation compares among these levels is analyzed to identify the differences; see Table 7.6.) To each qualitative factor, the model assigns certain effect parameters. Thus if a single factor 'A' has $a > 1$ categories or levels, write the effect parameters as α_i over the $i = 1, \ldots, a$ levels. A second index, j, is included to account for replicate observations at each combination of this single factor. Similar to (7.1), the statistical model becomes $Y_{ij} \sim$ indep. $\mathrm{N}(\mu_i, \sigma^2)$, where now the mean response $\mu_i = E[Y_{ij}]$ is taken as

$$\mu_i = \theta + \alpha_i, \tag{7.40}$$

over $i = 1, \ldots, a$ levels of the single qualitative factor and $j = 1, \ldots, n_i$ independent replicates per factor level. The standalone parameter θ may be interpreted as the *grand mean* of the model, while the α_i parameters are viewed as deviations from θ due to the effects of factor A. This is a *one-factor ANOVA model*. The total sample size is $n_+ = \sum_{i=1}^{a} n_i$. If n_i is constant at each level of i, that is, $n_i = n$, then the ANOVA is *balanced*. Deviations from a balanced design are called *unbalanced*.

Unfortunately, this factor-effect parameterization does not provide sufficient information to estimate every parameter. As currently written, (7.40) describes $a + 1$ parameters, $\theta, \alpha_1, \ldots, \alpha_a$; however, only a different factor levels are sampled from which to estimate these values. For instance, one could estimate all a values of α_i, but not θ. A simple solution to overcome this imposes an *estimability constraint* on the α_is. Many possible constraints

exists; two popular versions are the *zero-sum constraint* $\sum_{i=1}^{a} \alpha_i = 0$ and the *corner-point constraint* $\alpha_1 = 0$.

If applied properly, constraints on the α_is do not affect tests of the effects among levels of the factor; however, they do affect interpretation of the parameter estimates. For example, under the zero-sum constraint, θ is the mean of all the a group means and α_i is the difference between the ith factor level mean and the overall mean. By contrast, under the corner-point constraint, θ is the mean of the first factor level, while α_i is the difference between the ith level's mean and this first level's mean.

The predicted response within each ith factor level is estimated as the per-level mean:

$$\hat{Y}_{ij} = \overline{Y}_i = \frac{1}{n_i} \sum_{i=1}^{a} Y_{ij}.$$

Differential variation between levels of factor A is quantified by summing the squared differences between these $\overline{Y}_i$s and the overall mean $\overline{Y} = \sum_{i=1}^{a} \sum_{j=1}^{n_i} Y_{ij}/n_i$. This is

$$\text{SSA} = \sum_{i=1}^{a} \sum_{j=1}^{n_i} (\overline{Y}_i - \overline{Y})^2.$$

As it is a sum of squares, SSA has an associated number of d.f.; here, $\text{df}_\text{A} = a - 1$. As such, we can divide SSA by its d.f. to produce a mean square for between-factor effects: $\text{MSA} = \text{SSA}/(a-1)$.

Variation within factor A may be similarly quantified as the sum of squared differences between Y_{ij} and its predicted value $\hat{Y}_{ij} = \overline{Y}_i$. Pooled over all levels of A, this is essentially a residual sum of squares, as in (6.10), so we write

$$\text{SSE} = \sum_{i=1}^{a} \sum_{j=1}^{n_i} (Y_{ij} - \overline{Y}_i)^2.$$

The associated d.f. here are $\text{df}_E = n_+ - a$, with $\text{MSE} = \text{SSE}/(n_+ - a)$. As previously, the MSE is unbiased for estimating the unknown variance σ^2 (Exercise 7.6).

It can be shown (Kutner et al. 2005, Section 16.5) that the two sums of squares SSA and SSE will themselves add to the total sum of squares

$$\text{SSA} + \text{SSE} = \text{SSTo} = \sum_{i=1}^{a} \sum_{j=1}^{n_i} (Y_{ij} - \overline{Y})^2.$$

This decomposition of observed variation is collected together for convenient calculation. The result is called an *analysis of variance (ANOVA) table*, as given in Table 7.6. Notice the inclusion of an F-statistic in the table: $F = \text{MSA}/\text{MSE}$. This is essentially the discrepancy measure in (7.18), which is used here to test the RM H_o: $\alpha_1 = \alpha_2 = \cdots = \alpha_a$, that is, factor A has no effect on the mean response. Under this null hypothesis, $F \sim F(a-1, n_+ - a)$. Reject H_o in favor of any departure at false positive level α when the calculated statistic F_calc exceeds the critical point $F_\alpha(a-1, n_+ - a)$. The corresponding P-value is $P[F(a-1, n_+ - a) \geq F_\text{calc}]$.

In the simplest case of $a = 2$ levels with a single factor, this construction collapses to the two-sample t-test from section 5.4.2. The F-statistic in Table 7.6 equates exactly to the square

Table 7.6 Schematic for single-factor analysis of variance (ANOVA) table.

Source	d.f.	SS	MS	F
Factor A	$a-1$	$\text{SSA} = \sum_{i=1}^{a}\sum_{j=1}^{n_i}(\overline{Y}_i - \overline{Y})^2$	$\text{MSA} = \dfrac{\text{SSA}}{a-1}$	MSA/MSE
Residual	$n_+ - a$	$\text{SSE} = \sum_{i=1}^{a}\sum_{j=1}^{n_i}(Y_{ij} - \overline{Y}_i)^2$	$\text{MSE} = \dfrac{\text{SSE}}{n_+ - a}$	
Total	$n_+ - 1$	$\text{SSTo} = \sum_{i=1}^{a}\sum_{j=1}^{n_i}(Y_{ij} - \overline{Y})^2$		

of the t-statistic in Equation (5.44), and a similar relationship holds between their reference distributions. Thus the (two-sided) inferences from both tests will always coincide.

A second, qualitative factor 'B' can be added to the study design, expanding the notation for the observations to Y_{ijk}, with indices $i = 1, \ldots, a$ for factor A, $j = 1, \ldots, b$ for factor B, and $k = 1, \ldots, n_{ij}$ for replicate observations at each combination of the two factors. Equation (7.40) becomes

$$\mu_{ij} = \theta + \alpha_i + \beta_j.$$

This is a *two-factor, main-effects ANOVA model*, so named because it contains effects due to only the two main factors. Allowing for the possibility that the two factors may interact leads to addition of cross-classified *interaction parameters*

$$\mu_{ij} = \theta + \alpha_i + \beta_j + \gamma_{ij}. \tag{7.41}$$

In either case, the total sample size is $n_{++} = \sum_{i=1}^{a}\sum_{j=1}^{b} n_{ij}$.

As in (7.40), estimability constraints are required under this factor-effects parameterization. The zero-sum constraints are $\sum_{i=1}^{a}\alpha_i = \sum_{j=1}^{b}\beta_j = 0$, $\sum_{j=1}^{b}\gamma_{ij} = 0$ for all i, and $\sum_{i=1}^{a}\gamma_{ij} = 0$ for all j. The alternative corner-point constraints are $\alpha_1 = \beta_1 = 0$, $\gamma_{i1} = 0$ for all i, and $\gamma_{1j} = 0$ for all j.

Table 7.7 displays an archetypal two-factor ANOVA table. The table gives 'sequential' sums of squares (SS) for testing the significance of model components using the sequential order in which they are fit. Similar to the partial t-tests in Section 7.1.3, the last sequential is known as the 'partial' SS; it tests the significance of a model component when fitted last in sequential order. Ratios of the specific mean squares to the MSE produce F-statistics to assess the significance of any factor or any multifactor interaction, using the general approach embodied in (7.18). Extensions to other multifactor models are also possible (Kutner et al. 2005, Chapter 24).

Table 7.7 Schematic for two-factor analysis of variance (ANOVA) table.

Source	d.f.	SS	MS	F
Factor A	$a-1$	SSA	MSA	MSA/MSE
Factor B	$b-1$	SSB	MSB	MSB/MSE
A × B interaction	$(a-1)(b-1)$	SSAB	MSAB	MSAB/MSE
Residual	$n_+ - ab$	SSE	MSE	
Total	$n_+ - 1$	SSTo		

Similar to earlier comments on model construction with higher-order quantitative predictors, it does not usually make sense to include a higher-order qualitative term in a model without also including all associated lower-order terms. Thus, for example, inclusion in (7.41) of a two-factor interaction using the γ_{ij}s also requires inclusion of both main effects terms via the α_is and β_js. This suggests a natural ordering for hypothesis testing: test the interaction first as H_o: $\gamma_{ij} = 0$ for all i, j. If the interaction is insignificant, follow back up the sequential order with a separate test of the next main effect via, for example, H_o: $\beta_j = 0$ for all j.

Example 7.5.1 **Per capita income in US counties.** (Kutner et al. 2005, Appendix C) present data on per capita income (in $) across the $n = 440$ largest counties in the United States, as related to a variety of demographic measures. Here, consider the two factors A = {Geographic region (Northeast, North central, South, or West) of the county} and B = {% college (baccalaureate) degrees (Low = less than 15%, Middle = between 15% and 30%, or High = greater than 30%) among county residents}.

Table 7.8 gives cell mean summaries of the untransformed data. (The complete data set is available at http://www.wiley.com/go/piegorsch/data_analytics. Notice the lack of balance in the design.) As might be expected, per capita income rises with increasing college experience, but it also shows some disparities among the different geographic regions. The differential pattern among the four regions is a question of interest here, which translates to testing (i) whether a significant interaction exists in these data and then (ii) testing for differences among the regions. (To find the means in **R**, use

```
> aggregate( PerCapInc ~ A+B, FUN='mean' )
```

where `PerCapInc` is the original income variable, and where `A` and `B` are the two indicators, each of **R** class `'factor'`.)

As per capita incomes often exhibit a large skew, apply a logarithmic transform and take $Y = \log\{\text{income}\}$ as the response variable. With this, assume the full two-factor model from (7.41). In **R**, this is fit by appealing to the `lm()` function, but with the special introduction of 'factor' variables for building the linear predictor. To do so, the `factor()` function can create qualitative factors from any appropriately structured variable. Sample **R** code is

```
> Ex751.lm <- lm( Y ~ factor(B) + factor(A) + factor(B):factor(A) )
```

Table 7.8 Summary cell means (and sample sizes) of per capita income (standardized to 1990 $) among $n = 440$ US counties.

		Factor B: % college degrees		
		Low: <15%	Middle: 15%–30%	High: >30%
Factor A: Region	Northeast	15 916.05 ($n_{11} = 19$)	20 585.29 ($n_{12} = 70$)	27 021.29 ($n_{13} = 14$)
	North central	16 578.25 ($n_{21} = 36$)	18 600.18 ($n_{22} = 60$)	21 974.17 ($n_{23} = 12$)
	South	14 332.97 ($n_{31} = 31$)	17 581.20 ($n_{32} = 101$)	21 899.95 ($n_{33} = 20$)
	West	15 045.75 ($n_{41} = 16$)	17 978.56 ($n_{42} = 50$)	24 652.64 ($n_{43} = 11$)

The linear predictor here calls for sequential fit of first the factor B main effect, then the factor A main effect, and finally the B × A interaction. (The ordering places B first because the college factor here is an expected source of variation – called a 'blocking variable' in the experimental design literature – and, hence, should have its effects accounted for first in the sequential order.) The interaction is indicated in **R** via the colon (`:`) operator. A shorthand syntax for the combination of all three terms is simply

```
> formula = Y ~ factor(B)*factor(A)
```

The `*` operator, when employed in an `lm` linear predictor, is *not* to be confused with simple multiplication. Here, it produces the cross-classification necessary to fit (7.41), which is something quite different.

In constructing the consequent ANOVA, **R** operates under corner-point constraints. To display the ANOVA table, use `anova( Ex751.lm )`, which produces the following (edited) output

```
Analysis of Variance Table
Response: Y
                     Df   Sum Sq  Mean Sq   F value     Pr(>F)
factor(B)             2   6.0454  3.02272  120.9328  < 2.2e-16
factor(A)             3   1.6253  0.54177   21.6750  4.366e-13
factor(B):factor(A)   6   0.3934  0.06557    2.6232    0.01652
Residuals           428  10.6979  0.02500
```

(**R** does not display SSTo in its standard ANOVA tables.) The P-values in the final column correspond to sequential F-tests of that factor's or interaction's effect.

From the **R** output, begin with the test for no A × B interaction via H_o: $\gamma_{ij} = 0$ for all i, j. The pertinent F-statistic is the partial (i.e., last sequential) $F_{calc} = 2.6232$. Referred to $F(6, 428)$, this is significant at, say, the 10% level: $P = 0.0165 < 0.10$. For these data, the pattern of response among different regions changes significantly as the rate of college education varies.

In the presence of a significant interaction, it is inappropriate to test for a main effect, because the main effect pools over levels of its factor. This can mask true differences across each individual factor level when interaction is present (Kutner et al. 2005, Section 19.7). For these data, this requires us to examine the 'simple effects' among the factor A cell means within each level of factor B. (As factor B is a known source of variation here, there is no need to test its effects. If we were to do so, however, we would also study 'simple effects' among the factor B cell means within each level of factor A.)

The comparisons can be performed in a variety of ways (Kutner et al. 2005, Section 23.3). For example, we can build hypothesis tests of the simple pairwise comparisons $H_{ii'j}$: $\mu_{ij} = \mu_{i'j}$ $(i < i')$ at each individual $j = 1, 2, 3$. The test statistics have the form

$$T_{ii'j} = \frac{\overline{Y}_{ij} - \overline{Y}_{i'j}}{\sqrt{\left(\frac{1}{n_{ij}} + \frac{1}{n_{i'j}}\right) \text{MSE}}}, \tag{7.42}$$

where $\overline{Y}_{ij} = \sum_{k=1}^{n_{ij}} Y_{ijk}/n_{ij}$. Under $H_{ii'j}$, (7.42) is referenced to a t-distribution: $T_{ii'j} \sim t(\text{df}_E)$ where $\text{df}_E = n_+ - ab$. To correct for multiplicity, compare each of the resulting $3 \times \binom{4}{2} =$

Table 7.9 Test statistics (7.42) and Bonferroni-corrected P-values (7.43) from simple effects/pairwise comparison analysis of factor A (Geographic region) in Example 7.5.1.

	Factor B: % college degrees		
Comparison	Low: <15%	Middle: 15–30%	High: >30%
Northeast vs North central	$T_{121} = -0.9017$ $P_{121} = 1$	$T_{122} = 3.3927$ $P_{122} = 0.0136$	$T_{123} = 3.3113$ $P_{123} = 0.0181$
Northeast vs South	$T_{131} = 2.5503$ $P_{131} = 0.2000$	$T_{132} = 6.4345$ $P_{132} = 2.1 \times 10^{-7}$	$T_{133} = 3.8586$ $P_{133} = 0.0024$
Northeast vs West	$T_{141} = 1.1409$ $P_{141} = 1$	$T_{142} = 4.6611$ $P_{142} = 0.0001$	$T_{143} = 1.3797$ $P_{143} = 1$
North central vs South	$T_{231} = 4.0760$ $P_{231} = 0.0010$	$T_{232} = 2.4774$ $P_{232} = 0.2451$	$T_{233} = 0.1148$ $P_{233} = 1$
North central vs West	$T_{241} = 2.1394$ $P_{241} = 0.5934$	$T_{242} = 1.3901$ $P_{242} = 1$	$T_{243} = -1.7890$ $P_{243} = 1$
South vs West	$T_{341} = -1.1562$ $P_{341} = 1$	$T_{342} = -0.7959$ $P_{342} = 1$	$T_{343} = -2.1010$ $P_{343} = 0.6520$

Corrected P-values above 10% FWE rate are deemphasized via gray tone.

18 statistics to a Bonferroni-adjusted critical point: reject $H_{ii'j}$ in favor of any (two-sided) departure when $|T_{ii'j}| \geq t_{\alpha/(2\times 18)}(428)$. At an FWE rate of 10%, the Bonferroni critical point is $t_{0.10/36}(428) = t_{0.0038}(428) = 2.7871$. Equivalently, reject $H_{ii'j}$ when its Bonferroni-corrected P-value

$$P_{ii'j} = \min\{(18)(2)P[t(428) \geq |T_{ii'j}|], 1\} \tag{7.43}$$

exceeds $\alpha = 0.10$.

Table 7.9 gives the t-statistics from (7.42) and the associated adjusted P-values. As expected, the pattern of pairwise differences among levels of factor A (geographic region) differs across the three levels of factor B (college degrees). Regions at the low percentage of college degrees only differ significantly between the North central and South regions (and, this is the only instance where those two regions differ significantly).

By contrast, Regions at the middle and high percentages of college experience show consistent, significant differences for both the Northeast versus North central and Northeast versus South comparisons. The Northeast versus West comparison at the middle level of college experience is also significant. (None of the North central versus West or South versus West comparisons is significant.) Further investigation into how these patterns relate per capita income and college experience could lead to intriguing knowledge discovery.

A residual plot of $Y_{ij} - \hat{Y}_{ij} = Y_{ij} - \overline{Y}_{ij}$ against $\hat{Y}_{ij} = \overline{Y}_{ij}$ is presented in Figure 7.13. No serious outliers are detected, although an indication of nontrivial variance heterogeneity appears. Perhaps a transformation of the original per capita incomes other than the natural logarithm may be more appropriate; this is explored in Exercise 7.30. □

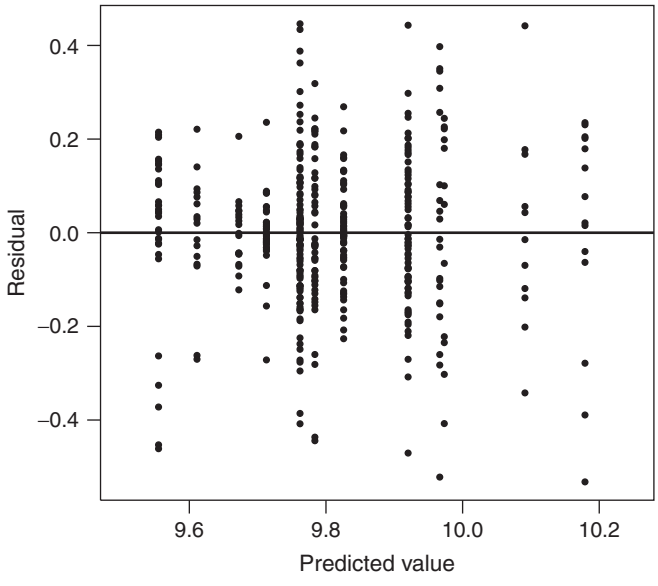

Figure 7.13 Raw residual plot from two-factor ANOVA fit in Example 7.5.1. Source: Data from Kutner et al. (2005, Appendix C).

Exercises

7.1 Return to the MLR analysis of the cancer mortality data in Example 7.1.1. Corroborate the indication in the example that the two predictor variables $x_4 = \{\text{Civic activity}\}$ and $x_6 = \{\text{Emergency bed-days}\}$ are insignificant by applying a single 2 d.f. test via the F-statistic discrepancy measure in (7.18). Operate at $\alpha = 0.05$.

7.2 Yeh (1998) described data on the compressive strength (in MPa) of high-performance concrete, as related to a variety of component predictors. A selection of the $n = 1030$ observations follows. (Download the complete data set at http://www.wiley.com/go/piegorsch/data_analytics.) All predictor variables are measured as kilogram per cubic meter unless otherwise specified.

	Replicate index, i					
Outcome variable	$i = 1$	$i = 2$	$i = 3$	$\cdots$	$i = 1029$	$i = 1030$
Y = Strength (MPa)	79.99	61.89	40.27	$\cdots$	32.77	32.40
x_1 = Age (days)	28	28	270	$\cdots$	28	28
x_2 = Cement	540.00	540.00	332.50	$\cdots$	159.10	260.90
x_3 = Furnace slag	0.00	0.00	142.50	$\cdots$	186.70	100.50
x_4 = Superplasticizer	2.50	2.50	0.00	$\cdots$	11.30	8.60
x_5 = Water	162.00	162.00	228.00	$\cdots$	175.60	200.60
x_6 = Fly ash	0.00	0.00	0.00	$\cdots$	0.00	78.30
x_7 = Coarse aggregate	1040.00	1055.00	932.00	$\cdots$	989.60	864.50
x_8 = Fine aggregate	676.00	676.00	594.00	$\cdots$	788.90	761.50

(a) It was recognized that the compressive strength outcome variable was skewed and would require a transformation, as in Section 3.4.3. Here, use $Y = \sqrt{\text{Strength}}$ as the response. Fit the MLR model to these data with this transformed response variable and the $p = 8$ predictors given above. Use partial t- or F-tests to determine if any of the predictors significantly affects the response. (Include a Bonferroni correction to adjust for multiplicity.) Operate at $\alpha = 0.05$.

(b) If you found any of the predictors in Exercise 7.2a to be insignificant, remove them in a RM to fine-tune the fit. Corroborate your choice by testing the RM against the eight-predictor FM via an F-statistic as in (7.18). Operate at $\alpha = 0.05$.

(c) Find the raw residuals from your RM in Exercise 7.2b and plot them against the fitted values. Also construct a normal quantile plot of these residuals. Comment on the quality of the diagnostic plots.

(d) Find the Studentized deleted residuals from your RM in Exercise 7.2b and plot them against the fitted values. Determine if any points are potential outliers by assessing these residuals against the exceedance limits $\pm t_{\alpha/(2n)}(n - p - 2)$. (Set $\alpha = 0.05$. Can you see any possible problems with these Bonferroni-adjusted limits here?) Do any points appear to be potential outliers?

(e) Expand on the transformation in Exercise 7.2a and investigate whether a different transformation may further stabilize the fit. Do so via the Box–Cox power transformation from (3.13). To implement a Box–Cox search in **R**, apply the `boxcox()` function to the `lm` object containing the eight-variable MLR fit of the data. The function will search for a value of the power parameter, λ, that maximizes the log-likelihood along the λ direction. Here, restrict λ to the range $-2 \leq \lambda \leq 2$. Transform Y according the recommended value of λ (to the nearest reasonable number; for example, if the function recommends $\lambda = 0.71$, set $\lambda = \frac{3}{4}$). Repeat the analyses above using the transformed response variable.

7.3 The study on admissions data from Example 6.5.3 also included the response variable $Y = \{\text{Student GPA (Grade Point Average)}\}$ in college. An additional question of interest with these data was the predictive ability of the the two preliminary scores $x_1 = \{\text{ACT}\}$ and $x_2 = \{\text{Class rank}\}$. Conditioning the analysis on the observed pattern of response in both ACT and Class rank scores, fit the MLR model with $E[Y] = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ to these data.

(a) Find the residuals $e_i = Y_i - \hat{Y}_i$ from the LS fit and plot them against the fitted values $\hat{Y}_i$. What patterns appear that bring into question the quality of the fit?

(b) To identify a possible transformation that may stabilize the fit for these data, consider the Box–Cox power transformation from (3.13). To implement a Box–Cox search in **R**, apply the `boxcox()` function to the `lm` object containing your two-variable MLR fit of the data. The function will search for a value of the power parameter, λ, that maximizes the log-likelihood along the λ direction. Here, restrict λ to the range $-3 \leq \lambda \leq 3$. Transform Y according to the recommended value of λ (to the nearest reasonable number; e.g., if the function recommends $\lambda = 2.03$, set $\lambda = 2$). Fit the appropriate MLR model using the transformed response variable.

(c) Plot the residuals from the transformed fit. Does the pattern appear more stable?

(d) Find the LS estimators for β_1 and β_2 from the data-transformed fit. What interpretation does each have?

(e) Test if either predictor variable significantly affects the transformed college GPA value. Use partial t- or F-tests. Operate at an FWE rate of $\alpha = 0.01$ and adjust your inferences for multiplicity. (Technically, this is also conditional on the selected value of λ.) What do you conclude?

(f) Calculate the hat matrix diagonals, h_{ii}, for the (x_{i1}, x_{i2}) pairs and determine if any points exhibit high leverage. (*Hint*: For visualization purposes, plot x_2 versus x_1 and mark the high-leverage points, if any, on the plot.)

(g) Add a cross-product/interaction predictor, x_1x_2, to the (transformed) MLR. Use a partial t- or F-test to determine if it significantly improves the fit. Operate at $\alpha = 0.01$. If the term is significant, also plot the residuals to check for any improvement or deterioration in their pattern.

7.4 A cadre of modern recording artists had their Twitter activity examined to determine if the data could relate to first week sales of a new album. $p = 3$ predictor variables were taken: $x_1 = \{\text{Number of Twitter followers (thousands)}\}$, $x_2 = \{\text{Average tweets per day}\}$, and $x_3 = \log\{\text{Previous album's first week sales}\}$. The response was $Y = \log\{\text{New album's first week sales}\}$. The data were

Artist	Y	x_1	x_2	x_3
Asher Roth	8.7160	118.5	2.1	11.0880
Ciara	11.3022	202.3	10.0	12.7321
Fabolous	11.6440	228.0	12.1	11.9767
Jordin Sparks	10.7579	350.8	17.4	11.6869
Maxwell	12.6635	70.0	1.1	12.5994
Trey Songz	11.9250	352.0	14.4	11.1982

(a) Fit an MLR model to these data. Test if the overall three-predictor model is significant. Operate at a false positive rate of 10%.

(b) Use partial t-tests to assess whether each individual predictor variable significantly affects mean (log-)new-album sales. Adjust for multiplicity at a false positive FWE of 10%.

(c) What concerns might exist about the quality of the model fit with this data set?

7.5 Return to the automobile fuel economy in Example 7.4.2. Verify the indication there that multicollinearity is elevated, but not serious for these data by calculating VIFs for the two (scaled) predictor variables x_1 and x_2.

7.6 Use (7.8) to prove that the MSE under the MLR model is an unbiased estimator of the population variance σ^2.

7.7 Verify the indication in Section 7.1.1 that the SSE under the MLR model can be written as $\text{SSE} = \mathbf{e}^{\mathrm{T}}\mathbf{e} = \mathbf{Y}^{\mathrm{T}}(\mathbf{I} - \mathbf{H})\mathbf{Y}$. (*Hint*: What is $(\mathbf{I} - \mathbf{H})^{\mathrm{T}}(\mathbf{I} - \mathbf{H})$?) Is this also true for the SLR model?

7.8 Apply the matrix formulation in Section 7.1.1 to the SLR model in Section 6.2.1. In particular,

(a) Find the $\mathbf{X}$ matrix and from this the $(\mathbf{X}^T\mathbf{X})$ and $(\mathbf{X}^T\mathbf{Y})$ matrices. From (7.5), verify the normal equations in (6.3).

(b) Find the $(\mathbf{X}^T\mathbf{X})^{-1}$ matrix and from (7.6) verify the equations for the LS estimators in (6.4).

(c) Verify the expression for the hat matrix diagonal elements h_{ii} in (6.24).

7.9 Similar to the study discussed in Exercise 6.11, the online site payscale.com released 2013 data on median, annual, full-time earnings (in \$, to the nearest hundred) among $n = 452$ public/state colleges and universities in the United States; see http://www.payscale.com/college-salary-report-2013. For each college, reported were $x = \{\text{Starting salaries}\}$ and $Y = \{\text{Mid-career salaries}\}$. The data are available online at http://www.wiley.com/go/piegorsch/data_analytics; a sample is given as follows:

x = Starting salary:	35 400	33 800	44 900	$\cdots$	76 000	72 200
Y = Mid-career salary:	43 600	44 400	81 800	$\cdots$	120 000	122 000

(a) Plot Y versus x. Does a curvilinear pattern appear?

(b) Condition the analysis on the observed pattern of starting salaries and fit a quadratic regression model to these data via LS. Report the estimated mean response $\hat{\mu}(x)$ and overlay it on the scatterplot of the data. Comment on the quality of the fit.

(c) Construct a 2 d.f. test to determine if a significant effect exists due to the predictor variable, x. Operate at $\alpha = 0.01$.

(d) Test if the quadratic term significantly improves on the model fit, above and beyond the linear term. Operate at $\alpha = 0.01$.

(e) Recall that in Exercise 6.11, a concern was raised over possible variance heterogeneity. Assess if this is also a concern with these data.

7.10 Moore (2010, Compan. Chapter 27) listed data on how the price (in \$) of a diamond relates to the weight of the stone (in carats). Of interest is estimating the mean price for differently sized diamonds. The observations comprise $n = 351$ data pairs and are available online at http://www.wiley.com/go/piegorsch/data_analytics; a sample is given as follows:

x = Carats:	3.35	3.17	3.01	$\cdots$	0.31	0.31
Y = Total price:	56454.40	54884.60	53191.20	$\cdots$	979.30	544.10

(a) Plot Y = Price against x = Carats. Is the pattern for the mean response linear or curvilinear?

(b) Assume a quadratic regression model as in Section 7.2 for these data: set $\mu(x_i) = \beta_0 + \beta_1(x_i - \overline{x}) + \beta_2(x_i - \overline{x})^2$ and work under the normal (Gaussian) model in (7.1). Calculate LS estimates for the β-parameters and also report the LS estimate for $\mu(x)$.

(c) Find the raw residuals $Y_i - \hat{Y}_i$ from your LS fit and plot them against the fitted values $\hat{Y}_i$. What pattern emerges? (*Hint*: Focus away from the small handful of diamonds with very large fitted values.) Does this call into question any of the assumptions made during this analysis?

7.11 To illustrate the effect centering the x variable can have when performing polynomial regression, suppose $n = 1000$ observations are to be taken at equally spaced values of $x_i = 1, 2, \ldots, n$. As $\sum_{x=1}^{n} x = n(n+1)/2$, we know $\bar{x} = (n+1)/2$.

(a) Calculate the correlation between x_i and x_i^2, and compare it to the correlation between $x_i - \bar{x}$ and $(x_i - \bar{x})^2$.

(b) Find the VIFs for x_i and x_i^2 in the linear model $E[Y_i] = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$. (If you appeal to the computer, recall that VIF calculations do not depend on the Y_is. Thus if your program requires Y_is, just generate any arbitrary set of n values.) Does the maximum VIF exceed the recommended threshold of 10?

(c) Now find the VIFs for $x_i - \bar{x}$ and $(x_i - \bar{x})^2$ in the linear model $E[Y_i] = \beta_0 + \beta_1(x_i - \bar{x}) + \beta_2(x_i - \bar{x})^2$. How do they compare?

(d) The use of integer-valued predictors is irrelevant here. Illustrate that the same VIF effects appear if you divide x_i by some positive constant.

7.12 For the diamond data in Exercise 7.10 recognize that the residual pattern indicates clear departure from homogeneous variance, so apply a WLS fit (Section 7.1.2) under a quadratic model. For your weights, take $w_i = 1/x_i$, where x_i is the carat variable. Find the WLS estimates for the β parameters, their standard errors, and the WLS estimate for $\mu(x_i) = \beta_0 + \beta_1(x_i - \bar{x}) + \beta_2(x_i - \bar{x})^2$. What changes do you find versus the LS fit?

7.13 Return to the feature selection analysis with the UK cancer mortality data in Examples 7.3.1–7.3.4. Examine the data further, as follows. Throughout, operate at $\alpha = 0.05$.

(a) Calculate an MLR fit using the seven predictors suggested in Example 7.3.1: the four x-variables along with $(x_2 - \bar{x}_2)^2, x_1x_2$, and x_1x_4'. Test to see if any of the three new predictors are significant. (Remember to include a Bonferroni adjustment if you apply simple 1 d.f. partial t-tests.) Also examine the residuals from your fit to ascertain if any unusual patterns emerge.

(b) Replace the simple $R^2_{[p]}$ measure in Example 7.3.1 with the adjusted $R^2_{A[p]}$. Plot $R^2_{A[p]}$ against p and determine if a point of diminishing increase in $R^2_{A[p]}$ suggests a possible collection of predictor variables for further study. Do your results differ from those achieved in the example?

(c) Replace the simple $R^2_{[p]}$ measure with Mallows' C_p. Plot C_p against p; you should see a general decrease toward $C_p = p$ as p increases. Determine the smallest p such that $C_p \approx p$ (but not substantially greater), and find the associated predictor configuration for further study. Do your results differ from those achieved in the example?

(d) In the `regsubsets` output from Example 7.3.2, what other interesting patterns emerge among the 14 potential predictor variables?

(e) In Example 7.3.4, replace the AIC with the BIC as the information-based optimality criterion. Do the results change qualitatively?

7.14 Return to the compressive strength data in Exercise 7.2 and apply a feature selection search to identify a possible RM among the eight predictor variables with this data set. Use the square-root-transformed response variable $Y = \sqrt{\text{Strength}}$. Employ backward elimination and take minimum-BIC as your selection criterion. How does the recommended set of variables compare to the RM you chose in Exercise 7.2b?

7.15 Return to the following data sets to experiment with loess smoothing. Begin by (re)constructing scatterplots of the data. Then, apply a robust quadratic loess smooth at $q = \frac{1}{2}$. Overlay the loess predicted curve on your scatterplot. Also apply other smoothing parameters in the range $0.2 \leq q \leq 0.8$ to visualize how changing the span affects the loess smooth in each case.

(a) The financial moving average data on the Dow Jones Industrial index in Example 3.5.2. Overlay the original moving average smooth in Figure 3.1 and comment on any differences.

(b) The Rocky Mountain Rainfall data in Example 4.2.10. Take $x = \{\text{Year}\}$ as the predictor variable and view each station's rainfall as a replicated observation Y at each x.

(c) The (random sample of) airline on-time performance data in Exercise 6.9a. Overlay the SLR fitted line from the exercise and comment on any differences.

(d) The public college salary data in Exercise 7.9. Overlay the fitted parabola from the exercise and comment on any differences.

7.16 Cleveland (1979) suggested an intriguing use of loess smoothing for enhancing residual diagnostics. The approach can be used to verify, or perhaps call into question, indications of variance heterogeneity in a residual plot. From a regression fit (of any sort: SLR, MLR, loess, etc.), find the absolute residuals $|e_i|$, $i = 1, \ldots, n$. To these, apply a loess smooth against the fitted values $\hat{Y}_i$. If the loess curve for the $|e_i|$s exhibits departure from a horizontal line, variance heterogeneity is indicated/validated. If the smooth appears relatively flat, however, the loess diagnostic suggests that variation is not heterogeneous. To illustrate, apply this strategy to the following data sets. In each case, indicate whether or not the loess smooth substantiates the earlier indication of heterogeneous variances.

(a) The baseball batting average data in Example 6.4.1.

(b) The private college salary data in Exercise 6.11.

(c) The diamond data in Exercise 7.12.

7.17 Explore the loess fit for the supernovae data in Example 7.4.1 as follows:

(a) Calculate the final raw residuals $\check{e} = Y_i - \check{Y}_i$ and plot them against the fitted values $\check{Y}_i$. Does any untoward pattern appear? Do some points appear to be possible outliers? If so, identify the supernovae and research them for knowledge discovery purposes to see if they have any features in common.

(b) Plot the absolute residuals $|\check{e}_i|$ from Exercise 7.17a against $\check{Y}_i$ and overlay a loess smooth of the $|\check{e}|$s as in Exercise 7.16. Use a smoothing parameter of $q = 0.75$. How does the loess smooth inform your residual diagnostic(s)?

(c) Recalculate the local quadratic loess fit in the example by varying the smoothing parameter over $q = 0.2, 0.4, 0.6, 0.8$. Indicate whether or not these changes affect the smoothed predicted values in a substantive manner.

(d) Recalculate the local quadratic loess fit in the example by switching to single-iteration (in **R**, use the `family='gaussian'` in `loess()`) for the local quadratic smoother, retaining $q = \frac{1}{2}$. Does this affect the final loess curve in a substantive manner?

(e) In Exercise 7.17d, apply instead a local linear smoother (with the robust extension) and vary the smoothing parameter over $q = 0.25, 0.50, 0.75$. Indicate whether or not these changes affect the final loess curve in a substantive manner.

(f) Fit a quadratic polynomial without loess smoothing, as in Section 7.2. Overlay the predicted quadratic curve on the original scatterplot in Figure 7.5, and comment on the visual quality of the fit.

7.18 Gammon (2009) reported a study on body mass index (BMI) of US photographer's models from the 1950s to the late 2000s. (BMI is a standardized measure that combines a person's weight in inches and height in pounds: BMI $= 703 \times$ weight/height^2.) While most Western populations have seen increases in BMI over that time span, these models show a different pattern. The data comprise $n = 609$ data pairs and are available online at http://www.wiley.com/go/piegorsch/data_analytics; a sample is given as follows:

Date:	Dec. 1953	Mar. 1954	Nov. 1954	$\cdots$	Dec. 2008	Jan. 2009
BMI:	19.6341	19.0436	20.4825	$\cdots$	17.4838	18.9492

(a) Plot $Y =$ BMI against $x =$ time. Does the pattern deviate from the rise seen in most Western populations?

(b) Calculate a robust, linear, loess fit with soothing parameter set to $q = 0.5$. Overlay the loess fit on the scatterplot. Does this improve visualization of the pattern?

(c) Calculate the final raw residuals $\check{\epsilon} = Y_i - \check{Y}_i$, and plot them against the fitted values $\check{Y}_i$. Does any pattern appear? Do some points appear to be possible outliers?

(d) Apply instead a robust quadratic smoother, again with smoothing parameter $q = 0.5$. Overlay the loess fit on the scatterplot. Is there much change from the local linear fit?

7.19 Cleveland et al. (1992, Section 8.2.4) described an astrometric data set with $n = 323$ observations on the radial velocity of the NGC7531 spiral galaxy, taken from different east/west and north/south positions. The data are available from a variety of **R** sources, for example, as the `galaxy` data frame in the external *ElemStatLearn* package associated with Hastie et al. (2009). Of interest is studying how $Y =$ Velocity is affected by the two predictor variables $x_1 =$ East/West position and $x_2 =$ North/South position, via a loess fit. Download these data and mimic Cleveland et al.'s analysis as follows.

(a) Plot the north/south versus east/west positions to visualize the measurement locations.

(b) Calculate a robust, quadratic loess fit for Y against the two x-variables. Include a cross-product term x_1x_2 in the model formula as in (7.29) to account for a possible interaction. (As the two predictors are of similar magnitude, there is no need to normalize them for this analysis.) Set the smoothing parameter to $q = 0.35$.

(c) Plot the contours of the resulting loess surface against x_1 and x_2. Comment on the features that appear.

(d) Calculate the final raw residuals $\check{\epsilon}_i = Y_i - \check{Y}_i$ and plot them against the fitted values $\check{Y}_i$. Does any pattern appear? Do some points appear to be possible outliers? Also construct a normal probability plot of the $\check{\epsilon}_i$s. Does the plot suggest that a simpler, one-step, 'Gaussian' loess fit would be appropriate?

(e) Plot the absolute residuals $|\check{\epsilon}_i|$ from Exercise 7.19d against $\check{Y}_i$ and overlay a loess smooth of the $|\check{\epsilon}|$s as in Exercise 7.16. Use a smoothing parameter of $q = 0.75$. How does the loess smooth inform your residual diagnostic(s)?

7.20 Explore the loess fit for the MPG data in Example 7.4.2, as follows.

(a) Calculate the final raw residuals $\check{\epsilon} = Y_i - \check{Y}_i$, and plot them against the fitted values $\check{Y}_i$. Does any pattern appear? Do some points appear to be possible outliers?

(b) Plot the absolute residuals $|\check{\epsilon}_i|$ from Exercise 7.20a against $\check{Y}_i$ and overlay a loess smooth of the $|\check{\epsilon}|$s as in Exercise 7.16. Use a smoothing parameter of $q = 0.75$. How does the loess smooth inform your residual diagnostic(s)?

(c) Recalculate the local quadratic loess fit in the example by varying the smoothing parameter over $q = 0.4, 0.5, 0.6$. Indicate whether or not these changes affect the smoothed predicted values in a substantive fashion.

7.21 Under the MLR model in (7.1) and (7.2), show that by centering the response variable and all the predictor variables about their means, the LS estimate for the intercept β_0 will be zero.

7.22 Under the MLR model in (7.1) and (7.2), let the ridge regression shrinkage estimator for $\boldsymbol{\beta}$ be given by (7.32). Find $E[\hat{\boldsymbol{\beta}}_\kappa]$ and $\text{Var}[\hat{\boldsymbol{\beta}}_\kappa]$. (*Hint*: Recall the matrix relationships given in Section 7.1.1.)

7.23 Return to the college admissions data studied in Exercise 7.3. Using the three predictors $x_1 = \{\text{ACT}\}$, $x_2 = \{\text{Class rank}\}$, and the interaction x_1x_2, assess if multicollinearity is a concern with the MLR fit. If so, apply a ridge regression analysis. (Remember to center the response variable and standardize the three predictors for the ridge analysis.) Find the GCV error and minimize it to determine the tuning parameter κ and use this to calculate ridge-regression predicted values. Report the final values on their original scale.

7.24 Return to the genetic SNP data in Example 7.4.3 and further examine the ridge regression fit, as follows.

(a) Verify the shrinkage effect by extending the ridge trace plot over much a larger range of κ, say, $0 \leq \kappa \leq 10^5$ or greater. Does the expected 'shrinkage' toward zero appear?

(b) Plot the predicted values $\hat{Y}_i(\kappa_{\text{GCV}})$ against the other predictors. Do any interesting (or uninteresting) associations appear?

(c) Replace κ_{GCV} with κ_{HKB} and recalculate the ridge regression coefficients. Is there a substantial change?

(d) Replace the trace plot in Figure 7.8 with the alternative trace of each $\hat{\beta}_j$ plotted as a function of the effective d.f. from (7.35). Notice how the plot now begins with the traces tightly grouped and then shows greater variation and instability as the d.f. increase. At what effective d.f. do the traces appear to stabilize? How does this relate to the values for κ_{GCV} and κ_{HKB} chosen in the example? (*Hint*: Operate over a very wide range of κ, say, $0 \le \kappa \le 10^6$ or greater, in order to capture the effects as the d.f. $\to 0$. You may find the `ridge()` function from the external *genridge* package convenient for calculation of the d.f.)

7.25 Return to the compressive strength data in Exercise 7.2 and apply the Lasso from Section 7.4.3 for purposes of variable selection. Use all eight original predictor variables and $Y = \sqrt{\text{Strength}}$ as the response. (Remember to center Y and standardize the eight predictor variables.) Compare the variables retained by the Lasso at its minimum CV κ against the models selected in Exercises 7.2b and 7.14. Comment on whether and how any differences appear between the Lasso regularization and those previous analyses.

7.26 Return to the genetic SNP data in Example 7.4.4 and, in particular, the Lasso regularization calculation from that example. Reperform the K-fold CV for a variety of other values of K, say, $K = 5, 10, 25, 50$. Do the results change substantively from those seen in the example? In particular, at each minimum-CV value for κ, how many and which predictors are retained in the model?

7.27 Kutner et al. (2005, Appendix C) reported data on health care claims (in total $ over 24 mos.) generated by an insurance company's female subscribers with ischemic heart disease, as related to a series of seven predictor variables. A selection of the $n = 608$ observations follows. (Download the complete data set at http://www.wiley.com/go/piegorsch/data_analytics.)

	Subscriber index, i				
Outcome variable	$i = 1$	$i = 2$	$\cdots$	$i = 607$	$i = 608$
Y = Claims ($)	179.10	319.00	$\cdots$	1282.20	586.00
x_1 = Age (years)	63	59	$\cdots$	58	56
x_2 = Procedures (no.)	2	2	$\cdots$	7	4
x_3 = Prescribed drugs (no.)	1	0	$\cdots$	2	4
x_4 = Emerg. Room visits (no.)	4	6	$\cdots$	2	6
x_5 = Complications (no.)	0	0	$\cdots$	0	0
x_6 = Comorbidities (no.)	3	0	$\cdots$	7	3
x_7 = Duration (days)	300	120	$\cdots$	244	336

Claim costs are notoriously skewed, so begin by transforming the response variable to $Y = \log\{\text{Claims}\}$ and calculate the centered response $U = Y - \overline{Y}$. Also center the

seven predictor variables and scale them to have unit variance. Apply the Lasso from Section 7.4.3 for purposes of variable selection with these data by proceeding as follows.

(a) Begin with an assessment of multicollinearity among the (standardized) predictors. Does the maximum VIF exceed 10?

(b) Fit the seven predictors to the centered response via the Lasso. Construct an estimated coefficient profile plot. Comment on the features in the plot. Does the typical Lasso-shrinkage pattern appear?

(c) Apply K-fold CV at $K = 10$ to select a single value for κ. What value is selected? Which predictors are retained in the model and what are their estimated $\hat{\beta}_j$ coefficients?

7.28 Blæsild and Granfeldt (2002, Exercise 4.6) presented data on egg development of sand lizards (*Lacerta agilis*). Across a series of $a = 20$ different incubation treatment regimes, incubation times of lizard's eggs were recorded in days for $n_+ = 119$ hatchlings. The data are available online at http://www.wiley.com/go/piegorsch/data_analytics; a sample is given follows:

Treatment code:	1	2	$\cdots$	20
Incubation times:	92.9, 93.9, 83.9	59.2, 58.0, … , 55.9	$\cdots$	26.0, 26.4, … , 26.7

Temporal data often exhibit skewness and/or variance heterogeneity, so operate with $Y = \log\{\text{Incubation time}\}$. Apply a one-factor ANOVA model to these data and assess whether any difference in log-incubation was seen across the different treatments. Include pertinent diagnostic assessments. What do you conclude at $\alpha = 0.01$?

7.29 Illustrate the claim in Section 7.5 that when a single factor has only $a = 2$ levels, the ANOVA F-test and the pooled, two-sample t-test are equivalent, as follows. Return to the lung function data in Exercise 5.27c and fit a one-factor ANOVA model. Test for equality between the two means against a two-sided alternative. As there, operate at $\alpha = 0.10$. Show that the two test statistics equate via $t^2 = F$ and that the corresponding P-values are identical.

7.30 Return to the per capita income data from Example 7.5.1 and investigate whether a different transformation may further stabilize the fit. Do so via the Box–Cox power transformation from (3.13). To implement a Box–Cox search in **R**, apply the `boxcox()` function to the `lm` object containing the full two-factor fit to the data. The function will search for a value of the power parameter, λ, that maximizes the log-likelihood along the λ direction. Here, restrict λ to the range $-2 \leq \lambda \leq 2$. Transform Y according the recommended value of λ (to the nearest reasonable number; e.g., if the function recommends $\lambda = -0.44$, set $\lambda = -\frac{1}{2}$). Repeat the two-factor ANOVA in the example using this newly transformed response variable.

8

Supervised learning: generalized linear models

A core assumption underlying the regression analyses in Chapters 6 and 7 was that the independent observations Y_i were continuous with constant variances. Normality via $Y_i \sim \mathrm{N}(\mu_i, \sigma^2)$ was also often included, possibly after a stabilizing transformation (as in Section 3.4.3). While applicable to a wide variety of settings, these assumptions need not hold for every regression data set. Observations may have variances $\mathrm{Var}[Y_i]$ that change in relation to the mean response, $\mu_i = E[Y_i]$. They may be restricted to only positive outcomes, be bounded over a fixed interval, or be discrete such as counts and proportions. In such cases, the normal linear regression model will not hold. It can be generalized, however, to include nonnormal parent distributions for the data, allow for heterogeneous variances among the Y_is, and incorporate nonlinear relationships between the mean response and the linear predictor. This chapter presents a class of *generalized linear models* (GLiMs) that achieves these goals.

8.1 Extending the linear regression model

8.1.1 Nonnormal data and the exponential family

As in Chapters 6 and 7, assume the data Y_i $(i = 1, \ldots, n)$ are recorded along with a series of concomitant predictor variables x_{ij} $(j = 1, \ldots, p)$. GLiMs continue to view the mean response $\mu_i = E[Y_i]$ as a function of these p predictors but accept that the distribution of Y_i may differ from normal. To do so, they appeal to the larger *exponential family* of distributions from Section 2.3.11. Recall that the exponential family is a rich collection that can accommodate both discrete and continuous probability functions, $f_Y(y)$, via the general expression

$$f_Y(y) = \exp\left\{\frac{y\theta - b(\theta)}{a(\varphi)} + c(y, \varphi)\right\}, \tag{8.1}$$

Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery,
First Edition. Walter W. Piegorsch.

www.wiley.com/go/piegorsch/data_analytics

where $a(\varphi)$, $b(\theta)$, and $c(y, \varphi)$ are functions of known form. The unknown parameter θ is called the *natural parameter* of the distribution and is related to the population mean via $E[Y] = \mu = b'(\theta)$. The *dispersion* or *scale parameter* $\varphi > 0$ relates to the variance via the relationship

$$\text{Var}[Y] = a(\varphi) \frac{\partial^2 b(\theta)}{\partial \theta^2}.$$

When the dispersion parameter φ is known, the quantity $\partial^2 b(\theta)/\partial\theta^2$ is called the *variance function* of Y, because it incorporates all the unknown aspects of the variance term. Because $\partial^2 b(\theta)/\partial\theta^2 = \partial\mu/\partial\theta$ is usually a function of μ, the variance function is denoted by $V(\mu)$.

Example 2.3.9 showed that the normal distribution probability density function (p.d.f.) does satisfy (8.1) and, therefore, is a member of the larger exponential family. Other well-known examples include the binomial (Example 2.3.10) and Poisson (Exercise 2.26) models. As discussed in Section 2.3.11, however, a notable exception is the uniform distribution. As seen there, the uniform distribution, or for that matter any distribution whose support depends on unknown parameters, cannot be included in the exponential family.

8.1.2 Link functions

A second extension to the classical linear model made in a GLiM expands the link between the mean of the ith response, μ_i, and the linear predictor. For notational convenience, write the latter as

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}. \tag{8.2}$$

Under a classical simple linear regression (SLR) or multiple linear regression (MLR) model, the mean and linear predictor are identically related: as modeled in (7.2), $\mu_i = \eta_i$. GLiMs extend this into a functional relationship, say, $g(\mu_i) = \eta_i$. The function $g(\cdot)$ is assumed monotone and is called the *link function* or *link* between μ and η. The link function defines the scale over which the systematic effects represented by η are modeled as additive.

In some cases, the link is trivial; for example, $\mu_i = \eta_i$ represents an *identity link*, $g(\mu_i) = \mu_i$. In other cases, the link can be used to model a necessary relationship for $\mu = E[Y]$. For instance, in Exercise 2.26, the Poisson mean is seen to be positive, so the link should relate the strictly positive quantity $\mu_i > 0$ to a linear predictor of any sign. A common choice then is the natural logarithm, $g(\mu_i) = \log(\mu_i)$.

As the link function is assumed monotone, it has an inverse link function, $g^{-1}(\eta_i)$, which characterizes the mean as a function of the linear predictor. For simplicity, one often sees the notation $h(\eta_i) = g^{-1}(\eta_i)$, so that $\mu_i = h(\eta_i)$. For example, with the Poisson distribution, the inverse link to $g(\mu_i) = \log(\mu_i) = \eta_i$ is $\mu_i = h(\eta_i) = e^{\eta_i}$.

The exponential family based on (8.1) and the link function $g(\cdot)$ relating μ to η are the two key generalizations that make up a GLiM. Within this context, a number of technical, theoretical developments that allow for regression modeling from any member of the parent class in (8.1) are possible. These are discussed in the next section.

8.2 Technical details for GLiMs*

From a broad perspective, estimation and inference for GLiMs are similar to that for the SLR and MLR models in Chapters 6 and 7, respectively. The extended nature of the model class

does require some technical manipulation, however. In what follows, advanced familiarity with differential calculus and matrix algebra will be necessary. Introductory readers may wish to skip ahead to Section 8.3 and study applications of the models before taking up the details that follow here.

8.2.1 Estimation

By relaxing the homogeneous variance assumption on Y_i, GLiMs suppose the observations contribute heterogeneous information on the regression parameters in (8.2). As in Section 5.2.3, this then calls for estimation of $\boldsymbol{\beta} = [\beta_0 \ \beta_1 \ \cdots \ \beta_p]^{\mathrm{T}}$ via weighted least squares. The weights will typically depend on μ_i and thus, through the link, on $\boldsymbol{\beta}$. Iteration, therefore, becomes necessary, and the estimation proceeds by iterating through the weighted least squares equations. This iteratively (re)weighted least squares (IWLS) approach can be shown to correspond to a maximum likelihood (ML) solution (Nelder and Wedderburn, 1972). Thus the estimation process may be viewed as maximizing the log-likelihood

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \exp\left\{ \frac{Y_i \theta_i(\boldsymbol{\beta}) - b\left(\theta_i(\boldsymbol{\beta})\right)}{a(\varphi)} + c(Y_i, \varphi) \right\}, \tag{8.3}$$

where the notation $\theta_i(\boldsymbol{\beta})$ emphasizes that the natural parameters are modeled as functions of the regression parameters in $\boldsymbol{\beta}$. Note that this assumes, for the moment, that the dispersion parameter φ is known.

The IWLS/ML algorithm is not trivial (Myers et al. 2012, Section A.6) and has no closed-form solution; however, it can be shown to produce a set of iterative ML estimating equations for the β_js (Piegorsch and Bailer 2005, Section 3.2.1):

$$\sum_{i=1}^{n} \frac{x_{ij} h'(\eta_i)}{a(\varphi) V(\mu_i)} (Y_i - \mu_i), \quad \text{for } j = 0, \ \ldots \ , p,$$

with $\mu_i = h(\eta_i)$ and η_i dependent on $\boldsymbol{\beta}$ through (8.2). From these, the iterative solution is easily programmed. For instance, **R**'s `glm()` function can perform the IWLS/ML fit for most GLiMs seen in practice. This is illustrated in the examples throughout Section 8.3.

The IWLS/ML estimator $\hat{\boldsymbol{\beta}}$ exhibits typical ML optimality properties, as discussed in Section 5.2.4. Thus $\hat{\boldsymbol{\beta}}$ possesses an approximate $(p+1)$-variate normal distribution with mean vector $\boldsymbol{\beta}$ and covariance matrix found by inverting the Fisher information matrix, $\mathbf{F}(\boldsymbol{\beta})$, from (5.5). For shorthand notation, write $\mathrm{Var}[\hat{\boldsymbol{\beta}}] \approx \mathbf{F}^{-1}(\boldsymbol{\beta})$. (The approximation improves as $n \to \infty$.) The individual variances $\mathrm{Var}[\hat{\beta}_j]$ are taken from the diagonal elements of $\mathrm{Var}[\hat{\boldsymbol{\beta}}]$. These often depend on $\boldsymbol{\beta}$ and so are estimated by replacing β_j with its estimator $\hat{\beta}_j$ wherever it appears. Denote the estimated variances as $\hat{\sigma}_j^2$, with corresponding standard errors $\mathrm{se}[\hat{\beta}_j] = \hat{\sigma}_j$. The estimated off-diagonal covariances $\hat{\sigma}_{jk}$ $(j \neq k)$ are calculated similarly by replacing β_j with $\hat{\beta}_j$.

Note that while $E[\hat{\boldsymbol{\beta}}] \approx \boldsymbol{\beta}$, in small samples, the approximation's quality can vary. Thus for small n, analysts may wish to apply a bias correction. This will depend on the nature of the linear predictor, the link function, and the underlying parent distribution (Cordeiro and McCullagh 1991; Firth 1993).

If the dispersion parameter φ is unknown, it may also be estimated. One can apply ML or the method of moments (Section 5.2.2). In the latter case, one finds a quantity whose first

moment is a function of φ and manipulates the relationship to produce the estimator $\hat{\varphi}$ (see the following text).

8.2.2 The deviance function

A fundamental quantity useful for testing hypotheses, assessing model adequacy, and estimating dispersion in a GLiM is known as the *deviance function*. It is constructed to measure the discrepancy of a posited model when fit to a data vector $\mathbf{Y} = [Y_1 \;\; Y_2 \;\; \cdots \;\; Y_n]^{\mathrm{T}}$. The deviance uses the log-likelihood $\ell(\boldsymbol{\beta}) = \log\{L(\boldsymbol{\beta}; \mathbf{Y})\}$ from (8.3) to quantify discrepancy in that model's fit. This is defined as the deviation in $\ell(\cdot)$ from the fullest possible model that can be fit to the data. The latter quantity is found by fitting a separate parameter to each observation, giving a log-likelihood of the form $\log\{L(\mathbf{Y}; \mathbf{Y})\}$.

The notation here is avowedly awkward. Technically, one estimates each value of μ_i by its corresponding Y_i wherever μ_i appears in the log-likelihood. But because μ_i is related to $\theta_i(\boldsymbol{\beta})$ via $\mu_i = b'(\theta_i)$ and because $\theta_i = \theta_i(\boldsymbol{\beta})$ is a function of $\boldsymbol{\beta}$, one can in the extreme case view $\boldsymbol{\beta}$ as being comprised of n 'parameters,' each corresponding to a value of Y_i. The term for such an effect is *saturation*. The model and, hence, the log-likelihood has been saturated by n parameters, and we write $\hat{\theta}_i(\mathbf{Y})$ to indicate that $\theta_i(\boldsymbol{\beta})$ is estimated under this saturation. The corresponding value of $\ell(\boldsymbol{\beta})$ is denoted by $\ell(\mathbf{Y}) = \log\{L(\mathbf{Y}; \mathbf{Y})\}$.

Compared with the saturated model, a reduced model (RM) allocating only $p + 1 < n$ parameters to $\boldsymbol{\beta}$ will have a smaller maximized log-likelihood. Evaluated at the maximum likelihood estimate (MLE) of $\boldsymbol{\beta}$, this is $\ell(\hat{\boldsymbol{\beta}}) = \log\{L(\hat{\boldsymbol{\beta}}; \mathbf{Y})\}$. The quality of the model fit can be quantified by twice the difference of the two values: $2\{\ell(\mathbf{Y}) - \ell(\hat{\boldsymbol{\beta}})\}$. (Multiplying by 2 has some useful consequences; see the following text.) The closer this difference is to zero, the more the RM using only $p + 1$ parameters mimics that of a saturated model.

Applied to the exponential family in (8.1), suppose $a(\varphi)$ is known. Then,

$$\ell(\mathbf{Y}) = \sum_{i=1}^{n} \left\{ \frac{Y_i \hat{\theta}_i(\mathbf{Y}) - b\left(\hat{\theta}_i(\mathbf{Y})\right)}{a(\varphi)} + c(Y_i, \varphi) \right\},$$

while

$$\ell(\hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \left\{ \frac{Y_i \theta_i(\hat{\boldsymbol{\beta}}) - b\left(\theta_i(\hat{\boldsymbol{\beta}})\right)}{a(\varphi)} + c(Y_i, \varphi) \right\}.$$

Twice their difference is $2\{\ell(\mathbf{Y}) - \ell(\hat{\boldsymbol{\beta}})\}$. Write this as

$$D^*(\hat{\boldsymbol{\beta}}) = \frac{2\sum_{i=1}^{n} \left[Y_i \left\{ \hat{\theta}_i(\mathbf{Y}) - \theta_i(\hat{\boldsymbol{\beta}}) \right\} - \{ b(\hat{\theta}_i(\mathbf{Y})) - b(\theta_i(\hat{\boldsymbol{\beta}})) \} \right]}{a(\varphi)}. \tag{8.4}$$

The numerator of (8.4) is traditionally denoted as $D(\hat{\boldsymbol{\beta}})$ and called the *deviance function*. The entire quotient, $D^*(\hat{\boldsymbol{\beta}}) = D(\hat{\boldsymbol{\beta}})/a(\varphi)$, is called the *scaled deviance*.

One special case of $D(\hat{\boldsymbol{\beta}})$ is notable. When $Y_i \sim \mathrm{N}(\mu_i, \sigma^2)$, the deviance simplifies to the residual sum of squares: $D(\hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} (Y_i - \hat{\mu}_i)^2$, where $\hat{\mu}_i = b'[\theta_i(\hat{\boldsymbol{\beta}})]$ is the estimated mean response at the ith observation (Myers et al. 2012, Section 5.7).

The deviance is useful for testing discrepancies between models, as in (7.18). That is, suppose the 'full' model (FM) is represented by the $(p + 1)$-vector $\boldsymbol{\beta} = [\beta_0 \cdots \beta_{q-1} \;\; \beta_q \;\; \cdots \beta_p]^{\mathrm{T}}$ and a putative 'reduced' model (RM) contains only the first $q + 1 < p + 1$

elements of $\boldsymbol{\beta}$. In terms of a hypothesis test, the null hypothesis corresponding to RM constrains $p - q$ of the β_js to zero. The alternative hypothesis corresponds to FM, where no constraints are imposed on the β_js. Denote the reduced vector as $\boldsymbol{\beta}_{\text{RM}} = [\beta_0 \ \cdots \ \beta_q \ 0 \ \cdots \ 0]^{\text{T}}$. For consistency, also write the FM vector as $\boldsymbol{\beta}_{\text{FM}}$. Fit both models to the data and calculate their scaled deviances, $D^*(\hat{\boldsymbol{\beta}}_{\text{RM}})$ and $D^*(\hat{\boldsymbol{\beta}}_{\text{FM}})$, respectively. The difference between these is simply

$$D^*(\hat{\boldsymbol{\beta}}_{\text{RM}}) - D^*(\hat{\boldsymbol{\beta}}_{\text{FM}}) = 2\left\{\ell(\hat{\boldsymbol{\beta}}_{\text{FM}}) - \ell(\hat{\boldsymbol{\beta}}_{\text{RM}})\right\}. \tag{8.5}$$

Notice that this is twice the log of the ratio of the likelihoods. Recall from Section 5.4.3, however, that a ratio of likelihoods between a reduced (or null) model and the FM forms the likelihood ratio (LR) test of the RM. That is, for known φ, to test if the $p - q$ parameters set to zero under RM are significant, the LR statistic is

$$\begin{aligned} G^2 &= -2\log\left\{\frac{L(\hat{\boldsymbol{\beta}}_{\text{RM}}; \mathbf{Y})}{L(\hat{\boldsymbol{\beta}}_{\text{FM}}; \mathbf{Y})}\right\} = -2\left[\log\left\{L(\hat{\boldsymbol{\beta}}_{\text{RM}}); \mathbf{Y})\right\} - \log\left\{L(\hat{\boldsymbol{\beta}}_{\text{FM}}); \mathbf{Y})\right\}\right] \\ &= -2\left\{\ell(\hat{\boldsymbol{\beta}}_{\text{RM}}) - \ell(\hat{\boldsymbol{\beta}}_{\text{FM}})\right\}, \end{aligned}$$

which is simply (8.5). Thus the difference in scaled deviances is an LR statistic for assessing the RM. Recall that under appropriate regularity conditions, $G^2 \dot{\sim} \chi^2(p - q)$, where the symbol $\dot{\sim}$ is read 'is approximately distributed as.' The χ^2 approximation here improves as $n \to \infty$. This result is useful for testing the significance of RMs in a nested GLiM hierarchy (see Section 8.2.4).

A χ^2 approximation also holds for the scaled deviances themselves: $D^*(\hat{\boldsymbol{\beta}}) \dot{\sim} \chi^2(n - p - 1)$. Unfortunately, in small samples, this approximation can be poor near the tails of the distribution and it is not recommended for unregulated use (see McCullagh and Nelder 1989, Section 4.4.3). One case where it is useful, however, is when the dispersion parameter φ is unknown. Then, equating the expected value of $D^*(\hat{\boldsymbol{\beta}})$ under the $\chi^2(n - p - 1)$ approximation produces the estimating equation

$$E[D^*(\hat{\boldsymbol{\beta}})] = E[D(\hat{\boldsymbol{\beta}})/a(\varphi)] \approx n - p - 1.$$

Solving for φ yields, in effect, a method of moments estimator for φ. In the special case where $a(\varphi) = \varphi$, this is just

$$\hat{\varphi}_D = \frac{D(\hat{\boldsymbol{\beta}})}{n - p - 1},$$

a *deviance-based moment estimator* for φ.

8.2.3 Residuals

Assessing model adequacy is as important for GLiMs as it is for classical SLRs and MLRs. Residual analysis is a natural component of this process. With GLiMs, however, a variety of possible formulations exists for how to define a 'residual.' Given predicted values $\hat{Y}_i$, $i = 1, \ldots, n$, the *raw residual* is the usual quantity $e_i = Y_i - \hat{Y}_i$. Plotting the e_is against $\hat{Y}_i$ can provide useful a diagnostic, as seen in Section 6.3. For many distributional models in the exponential family, however, the variance of Y_i varies with the mean μ_i. When this occurs, plots of the raw residuals can create spurious patterns and/or mask truly important relationships in the data.

To adjust for differential variation, it is common to scale the e_is by the (estimated) standard deviation of Y_i:

$$\frac{Y_i - \hat{Y}_i}{\sqrt{\text{Var}[Y_i]}} = \frac{Y_i - \hat{Y}_i}{\sqrt{a(\varphi)V(\hat{\mu}_i)}},$$

where $V(\hat{\mu}_i)$ is the variance function evaluated at the predicted value for μ_i.

If, as is common, $a(\varphi) = \varphi/w_i$ for some known weights $w_i > 0$, this scaled residual becomes $\varphi^{-1/2}(Y_i - \hat{Y}_i)/\sqrt{V(\hat{\mu}_i)/w_i}$. For use as a diagnostic, we can ignore the constant dispersion parameter φ, producing

$$c_i = \frac{Y_i - \hat{Y}_i}{\sqrt{V(\hat{\mu}_i)/w_i}}.$$

These quantities are known as *Pearson residuals*, honoring the statistician Karl Pearson who proposed squaring and summing the c_is for use in analyzing tabular count data (Pearson 1900). The resulting summary measure of residual variation is the *Pearson* χ^2 *statistic*:

$$X^2 = \sum_{i=1}^{n} c_i^2 = \sum_{i=1}^{n} \frac{(Y_i - \hat{Y}_i)^2}{V(\hat{\mu}_i)/w_i}.$$

Scaling X^2 by φ leads to $X^2/\varphi \dot{\sim} \chi^2(n-p-1)$. Thus $E[X^2/\varphi] \approx n-p-1$. When φ is unknown, simple appeal to the method of moments, therefore produces $\hat{\varphi}_P = X^2/(n-p-1)$, a χ^2*-based moment estimator* for φ. This is preferred over $\hat{\varphi}_D$ when $V(\mu)$ is not constant with respect to μ, because the latter can be unstable in such cases.

An adjusted residual similar to c_i is based on the deviance function $D(\hat{\boldsymbol{\beta}})$. From (8.4), the ith contribution to the deviance is $2[Y_i\{\hat{\theta}_i(\mathbf{Y}) - \theta_i(\hat{\boldsymbol{\beta}})\} - \{b(\hat{\theta}_i(\mathbf{Y})) - b(\theta_i(\hat{\boldsymbol{\beta}}))\}]$. Then, the *deviance residual* is

$$d_i = \text{sgn}(e_i)\sqrt{2\left[Y_i\left\{\hat{\theta}_i(\mathbf{Y}) - \theta_i(\hat{\boldsymbol{\beta}})\right\} - \left\{b\left(\hat{\theta}_i(\mathbf{Y})\right) - b\left(\theta_i(\hat{\boldsymbol{\beta}})\right)\right\}\right]}, \tag{8.6}$$

where $\text{sgn}(e) = -I_{(-\infty,0)}(e) + I_{(0,\infty)}(e)$ is the *signum function* that reports the sign of its argument. Deviance residuals tend to be somewhat more stable than Pearson residuals (McCullagh and Nelder 1989, Section 2.4), although they do possesses a nonzero bias in small samples. Jørgensen (2012) suggests some modifications to d_i that can help to correct for this.

GLiM residuals can be standardized, similar to the strategies discussed in Section 6.3. As there, one begins with the design matrix $\mathbf{X}$, whose columns are the individual predictor variables. Unless otherwise specified, an additional column of ones for the 'intercept' is included as in (7.3). Next, define the matrix $\mathbf{Q} = \text{diag}\{q_{11}, \ldots, q_{nn}\}$ with elements

$$q_{ii} = \frac{\sqrt{w_i/V(\hat{\mu}_i)}}{|g'(\hat{\mu}_i)|},$$

where $g'(\hat{\mu}_i)$ is $\partial g(\mu)/\partial\mu$ evaluated at $\hat{\mu}_i$. These are used in constructing the hat matrix for the GLiM as

$$\mathbf{H} = \mathbf{Q}^{\mathrm{T}}\mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Q}$$

To achieve a standardized GLiM residual whose variance is approximately constant, divide d_i or c_i by $\sqrt{1-h_{ii}}$, where h_{ii} is the ith the diagonal element of $\mathbf{H}$.

8.2.4 Inference and model assessment

Inferences in a GLiM generally enlist the large-sample features of the ML estimators (Section 5.2.4). For example, if $\text{se}[\hat{\beta}_j]$ is the standard error of $\hat{\beta}_j$ (from Section 8.2.1), then the Wald ratio $W_j = (\hat{\beta}_j - \beta_j)/\text{se}[\hat{\beta}_j]$ will be approximately standard normal: $W_j \dot{\sim} \text{N}(0, 1)$, $j = 0, \ldots, p$. From this, large-sample confidence intervals and hypothesis tests may be constructed. A (pointwise) $1 - \alpha$ Wald confidence interval for β_j is $\hat{\beta}_j \pm z_{\alpha/2}\text{se}[\hat{\beta}_j]$, where $z_{\alpha/2}$ is the upper-$\frac{\alpha}{2}$ standard normal critical point.

Similarly, a Wald test of H_o: $\beta_j = 0$ versus H_a: $\beta_j \neq 0$ employs the statistic

$$W_j = \frac{\hat{\beta}_j}{\text{se}[\hat{\beta}_j]}$$

and rejects H_o when $|W_j| \geq z_{\alpha/2}$ or, equivalently, when $W_j^2 \geq \chi_\alpha^2(1)$. The approximate P-value is $P \approx 1 - 2\Phi(|W_j|)$, where $\Phi(\cdot)$ is the standard normal cumulative distribution function (c.d.f.) from (2.35). Notice that this is equivalent to $P \approx P[\chi^2(1) \geq W_j^2]$. For a one-sided test against, say, H_a: $\beta_j > 0$ reject H_o when $W_j \geq z_\alpha$. The approximate P-value becomes $1 - \Phi(W_j)$.

Stability of Wald tests/intervals varies greatly among members of the exponential family. For example, Wald tests for many Poisson-based GLiMs are usually quite stable, while those for some binomial-based GLiMs can be so unstable as to be expressly contraindicated (see Section 8.3.2).

Hypotheses involving multiple predictors can be assessed via matrix-based extensions of the Wald test (Cox 1988) or by LR methods. The latter strategy is especially attractive with GLiMs, because it corresponds to the use of the deviance function: recall that the difference in scaled deviances between a FM and an RM nested within FM corresponds to an LR statistic. This concept carries forward into a nested hierarchy of, say, $r \leq p + 1$ models from FM down to the simplest RM, where a series of LR statistics can be constructed for each of the scaled deviance differences. (The degrees of freedom (d.f.) for the corresponding test will equal the number of parameters that are constrained to reduce FM to RM. Generally, this is simply the difference in d.f. between the two scaled deviances.) Arranged in tabular form, this produces an *analysis of deviance table*, as in Table 8.1. In the table, RM_k refers to the kth (sub)model being fit, while 'Resid. d.f.' (residual d.f.) is n minus the number of nonzero parameters fit for that (sub)model and all previous RMs.

In an analysis of deviance, one sequentially assesses the significance of each nested RM by referring the corresponding LR statistic G^2 from Table 8.1 to an appropriate χ^2 distribution.

Table 8.1 Schematic for an analysis of deviance table.

Source	Resid. d.f.	$D^*(\cdot)$	Δd.f.	G^2
RM_1	df_1	$D^*(\text{RM}_1)$	–	–
RM_2	df_2	$D^*(\text{RM}_2)$	$\text{df}_1 - \text{df}_2$	$D^*(\text{RM}_1) - D^*(\text{RM}_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
RM_{r-1}	df_{r-1}	$D^*(\text{RM}_{r-1})$	$\text{df}_{r-2} - \text{df}_{r-1}$	$D^*(\text{RM}_{r-2}) - D^*(\text{RM}_{r-1})$
FM	df_FM	$D^*(\text{FM})$	$\text{df}_{r-1} - \text{df}_\text{FM}$	$D^*(\text{RM}_{r-1}) - D^*(\text{FM})$

Reject the null hypothesis that the additional components in a larger nested model are unimportant at false positive rate α if $G^2 \geq \chi^2_\alpha(\Delta\text{d.f.})$. The corresponding approximate P-value is $P[\chi^2(\Delta\text{d.f.}) \geq G^2]$.

As with classical linear modeling, the assessments in an analysis of deviance are made sequentially. Inferences on any nested RM may depend on the terms that precede it in the model and on the sequential order in which they are fit.

If φ is unknown in an analysis of deviance, adjustments are required. For example, suppose $a(\varphi) = \varphi$ and employ a moment-based estimator such as $\hat{\varphi}_P$. Then, the difference in estimated scaled deviances between two nested models, RM_2 and RM_1 (with corresponding residual d.f. $\text{df}_2 < \text{df}_1$), is calculated as $\{D(\text{RM}_1) - D(\text{RM}_2)\}/\hat{\varphi}_P$. Unfortunately, using an estimate of φ degrades the χ^2 approximation for G^2. To adjust for this, divide G^2 by the corresponding $\Delta\text{d.f.} = \text{df}_1 - \text{df}_2$. This results in an F-statistic:

$$F = \frac{D(\text{RM}_1) - D(\text{RM}_2)}{(\text{df}_1 - \text{df}_2)\hat{\varphi}_P}.$$

Under certain regularity conditions on $\hat{\varphi}_P$, $F \dot{\sim} F(\text{df}_1 - \text{df}_2, \text{df}_2)$, and we reject the null hypothesis that the additional components in RM_2 are insignificant at false positive rate α when the calculated F-statistic F_{calc} exceeds the upper-α critical point $F_\alpha(\text{df}_1 - \text{df}_2, \text{df}_2)$. The corresponding approximate P-value is $P[F(\text{df}_1 - \text{df}_2, \text{df}_2) \geq F_{\text{calc}}]$.

An important question here is, under which model – RM_1, RM_2, FM, etc. – should $\hat{\varphi}$ be determined? (As regression variables are added to or deleted from the model, $\hat{\varphi}$ will usually change.) Many strategies are possible, each varying in quality based on the underlying parent distribution, linear predictor, link function, etc., and it is difficult to make an omnibus recommendation. The default used in **R**'s `glm()` function is, however, a useful standard: when employing an estimator of φ, **R** fits the maximal model (FM) to determine $\hat{\varphi}$ and uses this same value for all its sequential P-values.

Note that these differences in deviances can also be used to construct approximate (pointwise) $1 - \alpha$ confidence intervals on any β_j by inverting the corresponding LR statistic from (5.35). These are called *profile likelihood confidence intervals*. The intervals are not usually available as closed-form expressions, but they can be acquired from a GLiM computer output. For instance, the **R** function `confint(obj.glm)` can compute profile likelihood intervals from the GLiM object `obj.glm` constructed via the `glm()` function. (Be sure to load the *MASS* package before applying `confint()` to a `glm` object.)

8.3 Selected forms of GLiMs

This section illustrates GLiMs for a few of the more popular members from the exponential family. In some cases, allied methods of analysis are included, depending on the nature of the parent distribution and the link function of interest. For a wider introduction to different forms of GLiMs, see, for example, Faraway (2006, Chapter 7), Myers et al. (2012, Chapter 5), or the classic text by McCullagh and Nelder (1989).

8.3.1 Logistic regression and binary-data GLiMs

When discrete data are observed as binary observations (generically: 'success' vs 'failure') leading to proportion responses, it is natural to consider use of the binomial probability mass

function (p.m.f.) from Section 2.3.1 as the parent distribution. Take Y as the number of successes recorded from M independent, binary 'trials' for the outcome. Then $Y \sim \text{Bin}(M, \pi)$, where $\pi = P[\text{success}]$. (The proportion response is Y/M.)

As seen in Example 2.3.10, this is a member of the exponential family in (8.1), and hence, GLiMs may be applied. The natural parameter for the binomial is the log-odds or 'logit' of success: $\theta = \text{logit}(\pi) = \log\{\pi/(1-\pi)\}$. Also, the dispersion parameter is a constant, $\varphi = 1$, along with the function $a(\varphi) = 1$. Thus there is no unknown dispersion parameter to estimate with a binary-data GLiM.

While the p.m.f. can be written as a function of the binomial mean $E[Y] = \mu = M\pi$, it is more common in practice to model π, rather than μ, directly. Thus a minor adjustment with the GLiM notation for binary data is to use π instead of μ as the argument of the link function, that is, write and model $g(\pi)$ rather than $g(\mu)$.

The link function most often employed with binomial data is the logit link: $g(\pi) = \text{logit}(\pi) = \log\{\pi/(1-\pi)\}$, connecting with the natural parameter θ. The corresponding inverse link is $h(\eta) = 1/(1+\text{e}^{-\eta})$, which transforms a linear predictor over $-\infty < \eta < \infty$ to a probability $0 < \pi < 1$. This feature is particular to binary data: because π is a probability, every inverse link should return a value between 0 and 1. And since $g(\pi)$ is monotone, so is its inverse $h(\eta)$. Notice then that if $h(\eta)$ is nondecreasing in η, it exhibits the properties of a c.d.f.: it is between 0 and 1, and it is a monotone, nondecreasing function. For example, with the logit link, the inverse is the c.d.f. from the standard logistic distribution (Johnson et al. 1995, Chapter 23). The corresponding GLiM is often called a *logistic regression*.

Connecting the inverse link to c.d.f.s opens a broad variety of possibilities for modeling $h(\eta)$ and, hence, $g(\pi)$. For example, if we appeal to the standard normal c.d.f. in (2.35), we find $h(\eta) = \Phi(\eta)$ and thus $g(\pi) = \Phi^{-1}(\pi)$. This is known as a *probit link*. The corresponding GLiM is called a *probit regression model*. Another useful c.d.f. is $h(\eta) = 1 - \exp\{-e^{\eta}\}$, which produces the *complementary log–log link* $g(\pi) = \log\{-\log(1-\pi)\}$.

When the subject matter does not provide an obvious choice for the link function, many analysts default to the logit. Among other features, the logit link has appeal due to the interpretability of its regression parameters: note that $\log\{\pi/(1-\pi)\}$ represents the log-odds that a 'success' occurs. Modeling π via the logit and fitting, say, the multiple linear predictor $\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ allows β_2 to represent the unit change in log-odds – a *log-odds ratio* – for a unit change in x_2 when holding x_1 constant.

One can also choose the link function empirically by applying a series of candidate links to the data and selecting that link associated with the most favorable model fit. Useful in this regard is the scaled deviance function, $D^*(\hat{\boldsymbol{\beta}})$ from Section 8.2.2. Recall from the χ^2 approximation for $D^*(\hat{\boldsymbol{\beta}})$ that $E[D^*(\hat{\boldsymbol{\beta}})] \approx n - p - 1$, so we expect $E[D^*(\hat{\boldsymbol{\beta}})/(n-p-1)] \approx 1$. Reasonable model fits will, therefore, display values of $D^*(\hat{\boldsymbol{\beta}})/(n-p-1)$ near 1. When $D^*(\hat{\boldsymbol{\beta}})/(n-p-1)$ is much larger than 1, however, potential model inadequacy is indicated. A common diagnostic rule-of-thumb (whenever $\varphi = 1$) is to view a fit as questionable if

$$\frac{D^*(\hat{\boldsymbol{\beta}})}{n-p-1} > 1 + \frac{2.8}{\sqrt{n-p-1}} \tag{8.7}$$

(McCullagh and Nelder 1989, Section 4.4.3).

Example 8.3.1 Logistic regression in joint-action toxicology. Modern cell-screening techniques provide toxicologists with rapid-throughput screening of hazardous chemicals. In a single toxicity study, tens of thousands of cells can be examined. For example, Shi et al. (2010)

Table 8.2 Proportions of human hepatic cells exhibiting micronuclei after exposure to DDT (in μmol/L) and nano-TiO_2 (in μg/L).

		Nano-TiO_2 concentration			
		0	0.01	0.1	1.0
DDT concentration	0	59/3000	65/3000	70/3000	67/3000
	0.001	67/3000	75/3000	83/3000	84/3000
	0.01	76/3000	87/3000	96/3000	83/3000
	0.1	94/3000	107/3000	110/3000	117/3000

Source: Deutsch and Piegorsch (2012, Table 1).

studied the joint action of two potential toxins, dichlorodiphenyltrichloroethane (DDT) and titanium dioxide nanoparticles (nano-TiO_2) in human hepatic cells. The first agent, DDT, was a heavily used pesticide and is a known environmental carcinogen. The second, nano-TiO_2, is a strongly reacting oxidizer in particulate form. Combination of the two may produce enhanced toxicity (called 'synergy'), prompting study of their joint action.

For this experiment, the outcome variable was a binary indicator of cellular damage, as represented by formation of extranuclear, damaged chromosome fragments known as 'micronuclei.' Specifically, $Y = 1$ if a cell exhibited chromosomal damage and $Y = 0$ if it did not. The data in Table 8.2 present the observed proportions of damaged cells across a variety of single- and joint-exposure combinations.

Interest with these data centers on assessing whether and how the two agents significantly impact the rate of cellular damage. As the data are proportions, it is natural to apply the binomial distribution: $Y_i \sim$ indep. $\text{Bin}(M_i, \pi_i)$, $i = 1, 2, \ldots, 16$, where $\pi_i = \pi(x_{i1}, x_{i2})$ is modeled as a function of the two exposures. Notice that M_i is held constant at 3000 cells/exposure combination.

For the predictor variables, the geometric spacing of the two exposure regimes argues for use of logarithmic transforms. From Table 8.2, base-10 logs are indicated, so set $x_{i1} = \log_{10}(\text{DDT}_i) + 4$ and $x_{i2} = \log_{10}(\text{TiO}_{2i}) + 3$. (The added constants ensure that the final exposure levels are nonnegative.) For the control levels, apply *consecutive-dose average spacing*: from a set of roughly equispaced nonzero doses $x_2 < x_3 < \cdots < x_n$, set the 'control' dose to

$$x_1 = x_2 - \frac{x_n - x_2}{n - 1} \tag{8.8}$$

(Margolin et al. 1986). Here, applied to the $\log_{10}$-transformed doses, this conveniently produces $x_{11} = 0$ at the DDT control and $x_{12} = 0$ at the nano-TiO_2 control.

For the linear predictor, include an intercept β_0, terms for each single log-exposure, and also a cross-product term, $x_{i1}x_{i2}$ to allow for possible synergism via the interaction of the two agents. This gives

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i1} x_{i2}.$$

Consider use of the logit link function for this analysis. (This is a reasonable default, although the probit or complementary log–log links could alternatively be applied; see Exercise 8.6.) The following sample **R** code uses the `glm()` function to conduct the analysis. (The variables are `Y` for Y_i, `Trials` for M_i, `x1` for the log-transformed DDT exposure, and `x2` for the log-transformed TiO_2 exposure.)

```
> Ex831logit.glm <- glm( cbind(Y,Trials-Y) ~ x1 + x2 + I(x1*x2),
                                          family=binomial('logit') )
> anova( Ex831logit.glm, test='Chisq' )
```

In the **R** code, some features require explanation:

- To indicate that the response is a proportion, Y_i/M_i, **R** requires separate presentation of the two component vectors, as Y_i and $Y_i - M_i$. This is performed by binding the two columns together via `cbind()`. Do *not* present **R** with the calculated proportions Y_i/M_i.
- In the linear predictor, the 'Identity' function `I()` protects the direct multiplication of x_1 and x_2. Without it, **R** would interpret the call to `x1*x2` as a more-complex series of terms not germane to the model being fit here, as in Example 7.5.1. See Dalgaard (2008, Section 12.5).
- The `family=binomial('logit')` option instructs **R** to fit a binomial likelihood with the logit link.
- The call to `anova(Ex831logit.glm,test='Chisq')` produces the analysis of deviance table. The `test='Chisq'` option conveniently adds a final column with the LR P-values

The resulting **R** output (edited) is just the analysis of deviance table:

```
Analysis of Deviance Table
Model: binomial, link: logit
Response: cbind(Y, Trials - Y)
Terms added sequentially (first to last)
          Df  Deviance   Resid.Df   Resid.Dev   Pr(>Chi)
NULL                         15      53.723
x1         1    43.944         14       9.779  3.379e-11
x2         1     5.550         13       4.229    0.01848
I(x1*x2)   1     0.022         12       4.207    0.88211
```

R employs a slightly different ordering than that seen in Table 8.1, with the 'residual' quantities `Resid.Df` and $D^*(\hat{\beta})$ (as `Resid.Dev`) appearing after the other two columns. The analysis reports information for model terms when they are '`added sequentially (first to last)`' as per their order in the call to `glm()`.

From the analysis of deviance, one finds that when fitted last, the interaction term `I(x1*x2)` is insignificant ($P = 0.8821$). Alternatively, a 95% profile likelihood confidence interval for β_3 can be produced using the following sample **R** command

```
> confint( Ex831logit.glm, level=0.95, parm='x1x2' )
```

which gives

```
       2.5 %       97.5 %
-0.04066755  0.04730930
```

The interval contains $\beta_3 = 0$, indicating that no significant synergy between the two exposures is evidenced. As it provides no additional explanatory value in the model, the $x_{i1}x_{i2}$ interaction term can be removed. The resulting RM is sometimes referred to as an 'additive model,' because its linear predictor only contains additive terms in x_1 and x_2. Sample **R** code is

```
> Ex831RM.glm <- glm( cbind(Y,Trials-Y) ~ x1 + x2,
                                        family=binomial('logit') )
> anova( Ex831RM.glm, test='Chisq' )
> summary( Ex831RM.glm )
```

The code is similar to that for the FM seen earlier, with a few additions. The call to `summary(Ex831RM.glm)` will display the IWLS/ML point estimates, their standard errors, and the corresponding single-d.f. Wald tests. The output (edited) for the analysis of deviance becomes

```
Analysis of Deviance Table
Model: binomial, link: logit
Response: cbind(Y, Trials - Y)
Terms added sequentially (first to last)
     Df  Deviance   Resid.Df   Resid.Dev   Pr(>Chi)
NULL                    15      53.723
x1    1    43.944         14       9.779  3.379e-11
x2    1     5.550         13       4.229    0.01848
```

The output (edited) from the call to `summary()` is

```
Coefficients:
            Estimate   Std.Error  z value  Pr(>|z|)
(Intercept) -3.90377     0.06394  -61.050   < 2e-16
 x1          0.16521     0.02506    6.593  4.31e-11
 x2          0.05845     0.02483    2.354    0.0186
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 53.723  on 15  degrees of freedom
Residual deviance:  4.229  on 13  degrees of freedom
AIC: 109.75
```

From the RM analysis of deviance, the remaining two terms representing each log-exposure variable are significant: when fitted last, the TiO_2 term gives a partial LR test P-value for H_o: $\beta_2 = 0$ of $P = 0.018$. Reversing the order – try it! – to place `x1` last gives a partial LR P-value for H_o: $\beta_1 = 0$ of $P \approx 3 \times 10^{-11}$. Even when adjusting for multiplicity via, for example, a Bonferroni correction, both terms are significant at a familywise false positive error (FWE) rate of $\alpha = 0.05$.

A more coherent, consolidated, 2 d.f. test of H_o: $\beta_1 = \beta_2 = 0$ can be performed by relating the deviance under the `NULL` model (with just β_0) to the current RM fit:

```
> anova( glm(cbind(Y,Trials-Y) ~ 1, family=binomial('logit')),
                                        Ex831RM.glm, test='Chisq' )
```

which yields a P-value of 1.788×10^{-11}; again, highly significant.

One might also consider including additional quadratic terms, x_{i1}^2 and x_{i2}^2, in the linear predictor, but these do not significantly improve the model fit (see Exercise 8.6).

Other features of the output include the point estimates and their standard errors (under `Estimate` and `Std.Error`, respectively), along with a reminder that the dispersion parameter φ was '`taken to be 1`,' as desired for the binomial model. The eventual predicted response probability as a function of x_1 and x_2 is then

$$\hat{\pi}(x_1, x_2) = \frac{1}{1 + \exp\{3.9038 - 0.1652x_1 - 0.0585x_2\}}. \tag{8.9}$$

A few final calculations can generate the ratio $D^*(\hat{\boldsymbol{\beta}})/(n-p)$ and the rule-of-thumb in (8.7), respectively, for informal model adequacy assessment:

```
> residDF <- Ex831RM.glm$df.residual
> Ex831RM.glm$deviance/residDF       # stability measure
> 1 + ( 2.8/sqrt(residDF) )          # rule-of-thumb
```

These give

```
 [1] 0.3253062
 [1] 1.77658
```

We find $D^*(\hat{\boldsymbol{\beta}})/(n-p-1) = 0.3253$, while $1 + 2.8/\sqrt{n-p-1} = 1.7766$. As the former is much smaller than the later, no concerns with the model fit are indicated.

Further diagnostics may be conducted by examining the deviance residuals, d_i, from (8.6). For example, a residual plot of d_i versus $\hat{\pi}(x_{i1}, x_{i2})$ from (8.9) is available via the sample **R** code

```
> di <- residuals( Ex831RM.glm, type='deviance' )
> pihat <- predict( Ex831RM.glm, type='response' )
> plot( di ~ pihat, pch=19 ); abline( h=0 )
```

The residual plot appears in Figure 8.1. The pattern is generally stable, except for one large negative residual at bottom, corresponding to the observed proportion 83/3000 = 2.767% at exposure combination DDT = 0.01, $TiO_2 = 1.0$. (A similar pattern emerges when plotting the standardized residuals against the predicted probabilities; see Exercise 8.6.) Referring back to the data in Table 8.2, this value clearly dips lower than the trend in its neighboring responses. Further investigation is necessary to determine if this is just a naturally low response or if some form of contamination or other disturbance has led to an outlying data point.

The overall conclusion to take from this analysis is that the logit link appears to fit the data reasonably well. In doing so, it indicates that both exposure variables do produce significant genotoxic effects in these cell systems. □

As the logistic (and probit, etc.) GLiM represents a form of supervised regression, it is amenable to the various diagnostics and exploratory techniques from Chapter 7. Beyond the adequacy rule-of-thumb and residual analysis illustrated in the previous example, full-scale regression diagnostics can be conducted (Hosmer and Lemeshow 2013, Section 5.3) and adjustments are possible when the suite of predictor variables is highly multicollinear (Schaefer 1986).

In high-dimensional problems with large p, it may be of interest to conduct logistic regression on a reduced subset of predictors. If so, regularized logistic regression via the least absolute shrinkage and selection operator (Lasso) from Section 7.4.3 can be performed (Hastie et al. 2009, Section 4.4.4), producing a form of *sparse logistic regression*. Like the Lasso,

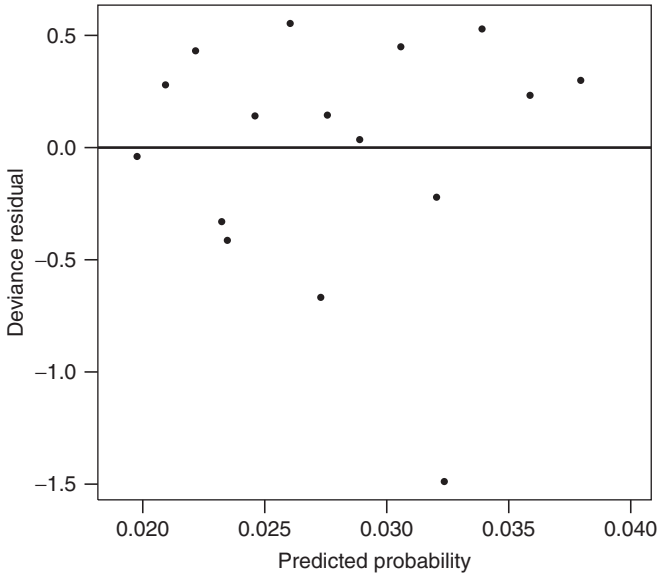

Figure 8.1 Deviance residual plot for reduced model fit in Example 8.3.1.

regularization again produces a form of regression a variable selector. For application in **R**, the `glmnet()` function from the external *glmnet* package includes an option for regularized/sparse logistic regression. (In fact, `glmnet` has capabilities that handle any logistic regression with extremely large data sets.)

8.3.2 Trend testing with proportion data

For some binary-data regressions, only a sole predictor variable, x_i, is recorded. Inferences can then be distilled down to a single question: do increases in x_i lead to significant increases in the corresponding response probability, π_i, when the latter is viewed as a function of x_i? This is often called the *quantal response problem*. The no-trend, null hypothesis here is H_o: $\pi_1 = \cdots = \pi_n$.

A powerful test procedure that detects departures from H_o is known as the *Cochran–Armitage* (CA) *trend test* (for proportions); its test statistic can be written as

$$Z_{CA} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})Y_i}{\sqrt{\overline{p}(1-\overline{p})\sum_{i=1}^{n} M_i(x_i - \overline{x})^2}}, \tag{8.10}$$

where $\overline{p} = \sum_{i=1}^{n} Y_i / \sum_{i=1}^{n} M_i$ is a pooled estimator of the common π under H_o and $\overline{x} = \sum_{i=1}^{n} M_i x_i / \sum_{i=1}^{n} M_i$.

In large samples, the null reference distribution for the CA statistic is approximately standard normal. (The approximation is valid when every M_i is at least 10 and $\sum_{i=1}^{n} M_i \geq 50$.) This allows for straightforward construction of the trend test: reject H_o in favor of an increasing trend when $Z_{CA} \geq z_\alpha$. The corresponding, approximate, one-sided P-value is $P \approx 1 - \Phi(Z_{CA})$. To test against a decreasing trend, reject if $Z_{CA} \leq -z_\alpha$. The corresponding,

approximate, one-sided P-value is $P \approx \Phi(Z_{CA})$. To test against any (two-sided) departure from H_o, reject if $|Z_{CA}| \geq z_{\alpha/2}$.

The test based on (8.10) was developed by Cochran (1954) and Armitage (1955) and also given in a more general form by Yates (1948). It possesses many favorable properties. Z_{CA} is invariant to linear transformations of x_i, and it has a form of omnibus optimality: it has locally highest power against any form of twice-differentiable, monotone function for $\pi(x_i)$ (Tarone and Gart 1980). If the true regression relation is logistic, so that $\pi(x_i) = 1/(1 + \exp\{-\beta_0 - \beta_1 x_i\})$, the CA trend test is even more powerful. The no-effect, null hypothesis now simplifies to H_o: $\beta_1 = 0$, versus the increasing-effect alternative of H_a: $\beta_1 > 0$. In this setting, the CA trend statistic has uniformly highest power for testing H_o against H_a. The same is true if we test H_o: $\beta_1 = 0$ against H_a: $\beta_1 < 0$ (Cox 1958). In **R**, the basic CA trend statistic (actually, its square) can be found via the `prop.trend.test()` function.

A caveat: computation of (8.10) assumes that the x_is are reasonably symmetric about $\bar{x}$. When this is not the case, and if a simple transformation of the x_is does not achieve approximate symmetry, a skewness correction is advocated (Tarone 1986): calculate the skewness measure

$$\hat{\gamma} = \frac{m_3(1 - 2\overline{p})\sqrt{M_+ - 1}}{(M_+ - 2)\sqrt{m_2^3\, \overline{p}\, (1 - \overline{p})}}\,, \tag{8.11}$$

where $M_+ = \sum_{i=1}^{n} M_i$ and $m_k = \sum_{i=1}^{n} M_i (x_i - \bar{x})^k / M_+$ for $k = 2, 3$. If $\hat{\gamma}$ deviates appreciably from 0, reject H_o in favor of an increasing trend when $Z_{CA} \geq z_\alpha + \hat{\gamma}(z_\alpha^2 - 1)/6$.

To illustrate, consider the following example from Piegorsch and Bailer (2005, Example 3.6), using data originally discussed in Huang and Smith (1999).

Example 8.3.2 **Trend test for ozone exceedance.** Ozone (O_3) is an important constituent of the Earth's upper atmosphere. At ground level, however, it is an irritant and can become a health hazard. Public health officials typically base ozone air quality standards on the number of exceedances over some safety level of O_3 concentration. For example, the data in Table 8.3 give $Y = \{$Number of surface O_3 concentration exceedances above an air quality limit of 120 ppb$\}$ across a series of 2349 recordings at monitoring stations near Chicago, IL. The table also lists the associated $x = \{$Average ozone concentration (in ppb)$\}$.

A plot of these data (not shown) indicates an increasing, sigmoidal trend in exceedance rate with increasing average surface ozone. To verify this indication statistically, one can apply the CA trend test via (8.10). (One data point had only a handful of recordings, where $M_i < 10$ and was removed from the presentation.)

These surface O_3 data provide a weighted mean of $\bar{x} = 71.7080$, with no large deviation from symmetry in the design: $\hat{\gamma} = 0.0543$ from (8.11). For the sake of illustration, however, consider inclusion of Tarone's skewness adjustment here. The calculations are not difficult to complete by hand, although a simple **R** function can be written to find the various statistics. For example:

```
>  CAtrendSkew.stat <- function(x , Trials , Y) }
     sumM <- sum(Trials);  pbar <- sum( Y )/sumM
     crrctx <-  x - (sum( x*Trials )/sumM)
     m2 <- sum( Trials*(crrctx^2) )/sumM
     m3 <- sum( Trials*(crrctx^3) )/sumM
     Zdenom <- sqrt( pbar*(1-pbar)*sumM*m2 )
     gtop <- (1 - 2*pbar)*sqrt( sumM-1 )*m3
```

```
  gbot <- (sumM-2)*Zdenom*m2/sqrt(sumM)
  gamma = gtop/gbot
  ZCA <- sum(crrctx*Y) / Zdenom
return ( list( statistic=ZCA, gamma=gamma) )
                                    }   # end function
```

Applying this to the data in Table 8.3 gives the following output:

```
$statistic
[1] 20.3481
$gamma
[1] 0.054322
```

Notice the verification that $\hat{\gamma} = 0.0543$. From these, we reject the null hypothesis of no trend against an alternative of increasing trend at, say, $\alpha = 0.01$ if the Z_{CA} statistic exceeds $z_{0.01} + \hat{\gamma}(z_{0.01}^2 - 1)/6 = 2.326 + 0.0399 = 2.366$. This clearly occurs, so a significant, increasing trend in exceedance is evidenced. □

Table 8.3 Exceedance rates over an air quality threshold of 120 ppb for surface ozone (O_3) as related to average O_3 surface concentrations (in ppb).

x = average O_3 concentration	39.31	49.42	56.89	64.2	65.1	74.67	80.46	85.05
Y = exceedances	0	2	0	2	2	8	11	3
M = recordings	184	230	315	285	191	292	309	95
x = average O_3 concentration	87.62	99.43	106.3	114	125.4	143.6		
Y = exceedances	7	27	15	28	18	16		
M = recordings	110	148	62	77	33	18		

Owing to its many favorable properties, the CA trend test is the method of choice for identifying an increasing (or decreasing) trend in a set of quantal response data. It is particularly important in one special case: testing individual predictor variables in logistic regression. For example, in a logistic regression with a single quantitative predictor variable – that is, $\pi(x_i) = 1/(1 + \exp -\beta_0 - \beta_1 x_i)$ – the Wald test of H_o: $\beta_1 = 0$ against either a one- or two-sided alternative hypothesis is known to behave aberrantly, even in large samples. The test statistic $W_1 = \hat{\beta}_1/\text{se}[\hat{\beta}_1]$ can decrease to zero as $\hat{\beta}_1$ grows far from zero, giving weak indications of departure from H_o (Hauck and Donner 1977). Væth (1985) gave general conditions for the Wald test to exhibit this poor performance. The CA trend statistic does not suffer from this instability and is recommend for use in this setting.

8.3.3 Contingency tables and log-linear models

When discrete data are observed as unbounded counts, it is natural to consider a Poisson distribution as the parent p.m.f. for the probability model. Take Y_i as the ith observed count in a random sample, with mean rate of occurrence $E[Y_i] = \mu_i$, $i = 1, \ldots, n$. As recognized in Exercise 2.26, the Poisson is member of the exponential family in (8.1) and, hence, may be employed in a GLiM. The natural parameter is $\theta = \log(\mu)$. As with binomial-based GLiMs in Section 8.3.1, the dispersion parameter for a Poisson GLiM is set to $\varphi = 1$, along with the function $a(\varphi) = 1$.

The link function most often employed with a Poisson GLiM is the natural logarithm $g(\mu) = \log(\mu)$. The corresponding inverse link is $h(\eta) = e^{\eta}$. This is known as a (Poisson) *log-linear regression model*, because the log of the mean is being modeled as linear. Notice here that because $\mu > 0$, any choice for the inverse link, including e^{η}, must be chosen to return a value greater than 0.

Example 8.3.3 Contingency table analysis of hardwood tree associations. Digby and Kempton (1987, Chapter 6) presented a classic ecological data set on spatial associations between 2076 different trees in a hardwood forest. The association is defined as whether or not any tree's nearest neighbor is of the same species. If not, no association is evidenced regarding how different hardwoods distribute themselves spatially in this forest.

Five specific types of tree were studied: hickory, maple, black oak, red oak, and white oak. The data were taken as $Y = \{$Number of times a given type of tree was a nearest neighbor to itself or another type$\}$; they conveniently collect into a cross-classified table of counts, as in Table 8.4.

The presentation in Table 8.4 represents the predictor variables as two separate qualitative factors that contribute to the observed pattern of counts: factor A = {Marker tree} and factor B = {Neighbor tree}. To assess the association between nearest neighbors, the table in effect asks whether the two factors act as independent contributors. If not, counts identified with one factor are contingent on the other factor. Thus this tabular display is often called an $R \times C$ *contingency table* of count data. In Table 8.4, $R = C = 5$. The analytic question is one of association versus independence between the R levels of the row factor and the C levels of the column factor.

Many approaches exist for assessing association between two factors in an $R \times C$ contingency table. A full survey exceeds the scope here; more details are available in sources such as Agresti (2013). For analytic purposes, it is sufficient to recognize that associations in a contingency table may be modeled via a Poisson GLiM by making use of the table's $R \times C$ factorial structure. That is, view the row and column factors as qualitative predictors as in Section 7.5, such that $i = 1, \ldots, R$ indexes the row factor's levels and $j = 1, \ldots, C$ indexes the column factor's levels. Assuming each count is $Y_{ij} \sim$ indep. Poisson(μ_{ij}), the mean μ_{ij} is then related to a two-factor ANOVA-type linear predictor

$$\eta_{ij} = \theta + \alpha_i + \beta_j + \gamma_{ij},$$

where the α_is represent the row factor main effect, the β_js represent the column factor main effect, and the γ_{ij}s represent the row × column interaction. No association between row and

Table 8.4 Cross-classified counts of nearest-neighbor trees in a hardwood forest.

		Neighbor tree				
		Hickory	Maple	Black oak	Red oak	White oak
Marker tree	Hickory	355	71	48	105	108
	Maple	79	242	21	74	70
	Black oak	51	25	27	12	20
	Red oak	95	64	20	104	59
	White oak	117	95	14	62	138

Source: Digby and Kempton (1987, Chapter 6).

column factors is indicated when $\gamma_{ij} = 0$ for all combinations of i and j. Since μ_{ij} must be positive under the Poisson assumption, apply a logarithmic link in the log-linear model:

$$\log(\mu_{ij}) = \theta + \alpha_i + \beta_j + \gamma_{ij}. \tag{8.12}$$

Problematically, the linear predictor in (8.12) involves more parameters, $1 + R + C + RC$, than the RC observations available to fit them. As with the ANOVA-type models in Section 7.5, an estimability constraint must be imposed. Examples include the corner-point constraints, $\alpha_1 = \beta_1 = \gamma_{i1} = \gamma_{1j} = 0$, or the zero-sum constraints $\sum_{i=1}^{a} \alpha_i = \sum_{j=1}^{b} \beta_j = \sum_{j=1}^{b} \gamma_{ij} = \sum_{i=1}^{a} \gamma_{ij} = 0$. The choice between these is usually arbitrary. For the full two-factor model in (8.12), the constraints reduce the set of parameters down to a set of $1 + (R - 1) + (C - 1) + (R - 1)(C - 1) = RC$ parameters. Notice, however, that for the RC independent observations in the $R \times C$ table, even under the estimability constraints each independent count is essentially fit to one parameter; this *saturates* the model fit as described in Section 8.2.2. Nonetheless, this still allows for unambiguous estimation of all pertinent parameters in η_{ij}.

Some important caveats are required when modeling contingency table data via a Poisson model. In essence, different sampling constraints on how the counts are obtained can affect the Poisson assumption, although in most cases the resulting test statistics – including those discussed here – all have the same form and interpretation. For the technical details, interested readers are referred, for example, to Agresti (2013, Chapter 9) or the discussion in Piegorsch and Bailer (2005, Section 3.3.4).

To test for association, fit the saturated 'FM' model (8.12) and also an RM under the no-association null hypothesis $\mathrm{H_o}$: $\gamma_{ij} = 0$ for all i, j. The difference in scaled deviances between the two models produces an LR test statistic for testing $\mathrm{H_o}$. Using an analysis of deviance from Table 8.1, the difference in deviances will be approximately distributed as χ^2 with $(R - 1)(C - 1)$ d.f.

Unfortunately, this LR statistic exhibits poor small-sample stability and can incur false-positive errors too often. It is not recommended for standard use. Instead, an asymptotically equivalent approach for $R \times C$ contingency tables call on the Pearson χ^2 statistic mentioned in Section 8.2.3. Start with the predicted values under the reduced, $\mathrm{H_o}$-restricted model, which can be found to be

$$\hat{Y}_{\mathrm{o}ij} = \frac{\left(\sum_{k=1}^{R} Y_{kj}\right)\left(\sum_{\ell=1}^{C} Y_{i\ell}\right)}{\sum_{k=1}^{R}\sum_{\ell=1}^{C} Y_{k\ell}}. \tag{8.13}$$

Next, calculate the Pearson statistic, which simplifies to

$$X^2 = \sum_{i=1}^{R}\sum_{j=1}^{C} \frac{\left(Y_{ij} - \hat{Y}_{\mathrm{o}ij}\right)^2}{\hat{Y}_{\mathrm{o}ij}}. \tag{8.14}$$

In large samples, $X^2 \mathrel{\dot\sim} \chi^2[(R - 1)(C - 1)]$. This is asymptotically equivalent to the LR statistic, but far more stable if every expected cell frequency, $\hat{Y}_{\mathrm{o}ij}$, is at least 1 and the overall average expected cell frequency is at least 5, that is,

$$\frac{1}{RC}\sum_{i=1}^{R}\sum_{j=1}^{C} \hat{Y}_{\mathrm{o}ij} \geq 5.$$

If the table satisfies these rules-of-thumb, reject H_o in favor of some association between row and column factors when the calculated statistic X^2_{calc} exceeds the critical point $\chi^2_\alpha[(R-1)(C-1)]$; the corresponding approximate P-value is $P\{\chi^2[(R-1)(C-1)] \geq X^2_{calc}\}$. ($R \times C$ tables that do not satisfy the large-sample rules-of-thumb are called *sparse*. Piegorsch and Bailer (1997, Section 9.3.4) gave some guidance on how to analyze sparse contingency tables.)

For the nearest-neighbor data in Table 8.4, rejection of the no-association hypothesis

$$H_o: \gamma_{ij} = 0, \text{ for all } i, j,$$

implies that some ecological effect causes the trees to associate with or disassociate from one another. To perform the fit in **R**, input the data as $5 \times 5 = 25$ rows of per-cell information, for example, as the respective variables `Marker`, `Neighbor`, and `Y` in

```
hickory   hickory   355
hickory     maple    71

    ⋮  ⋮  ⋮

whiteoak   redoak    62
whiteoak whiteoak   138
```

Then, fit the saturated model via the sample **R** code

```
> A = factor(Marker)
> B = factor(Neighbor)
> neighbor.glm <- glm( Y ~ A*B, family=poisson('log') )
> anova( neighbor.glm, test='Chisq' )
```

This produces an analysis of deviance table in the (edited) output

```
Analysis of Deviance Table
Model: poisson, link: log
Response: Y
Terms added sequentially (first to last)
          Df  Deviance  Resid.Df  Resid.Dev  Pr(>Chi)
NULL                        24    1301.65
A          4    430.83        20     870.82  <2.2e-16
B          4    451.73        16     419.10  <2.2e-16
A:B       16    419.10         0       0.00  <2.2e-16
```

Notice the value of zero under `Resid.Df` for the interaction term, indicating that the model fit has been saturated, as expected. The LR statistic for testing H_o here is given as the difference $G^2 = 419.10 - 0 = 419.10$, although as mentioned earlier this is not recommended for general use.

To find the Pearson X^2 statistic, simply fit the RM without the interaction terms, recover the Pearson residuals, and sum their squares:

```
> neighborRM.glm <- glm( Y ~ A+B, family=poisson('log') )

> ci <- residuals( neighborRM.glm, type='pearson' )
> sum( ci^2 )
> pchisq( sum(ci^2), df=neighborRM.glm$df.residual, lower=F )
```

The final two calculations give the X^2 statistic and its approximate P-value:

```
[1] 460.1468
[1] 8.401886e-88
```

A check shows that the rules-of-thumb for the χ^2 approximation are satisfied with this 5×5 table (left to reader). Thus the χ^2 P-value, which reports as essentially zero, provides a valid large-sample inference. We see that a significant association exists among nearest-neighbor species with these trees.

It is worth noting that **R** has a self-contained function (in fact, many) that calculates Pearson's X^2: `chisq.test()`. The data must be structured or manipulated into an $R \times C$ table or matrix, as in the following sample **R** command:

```
> chisq.test( xtabs( Y ~ factor(Marker)+factor(Neighbor) ) )
```

This gives (edited)

```
        Pearson's Chi-squared test
X-squared = 460.1468, df = 16, p-value < 2.2e-16
```

which corroborates the above `glm`-based calculations.

Further discovery is available here by asking in what direction(s) any significant association in the table lies. For instance, recall that the Pearson (and deviance) residuals may be standardized to give them roughly equal variances and thus greater comparability. In **R**, use

```
> chisq.test( xtabs( Y ~ factor(Marker)+factor(Neighbor) ) )$stdres
```

or equivalently (with a call to `round` the reported values)

```
> zci = rstandard( neighborRM.glm, type='pearson' )
> round( xtabs( zci ~ factor(Marker)+factor(Neighbor) ), digit=2 )
```

The (edited) output from the latter is

```
               factor(Neighbor)
factor(Marker)  blackoak hickory   maple  redoak whiteoak
      blackoak      6.81    1.07   -1.53   -2.65    -1.29
      hickory       0.96   12.28  -10.22   -1.62    -2.70
      maple        -2.02   -9.24   15.26   -1.32    -2.97
      redoak       -0.35   -2.48   -2.48    7.09    -0.92
      whiteoak     -2.84   -3.00   -0.89   -1.62     7.88
```

R defaults to order the variables alphabetically, differing from that seen in Table 8.4 (if desired, this can be easily changed). Nonetheless, the pattern here is revealing. Strong positive residuals are seen along the diagonal, while the off-diagnol residuals are almost always negative. (Only hickory and black oak combine to show positive off-diagonal values, and these are of relatively low magnitude for standardized residuals.) This indicates that many more trees were observed on the diagonals than would be expected under the null hypothesis of no association, with the reverse predominantly true for off-diagonals. It appears that trees of similar type tend to cluster near each other, at least for those studied in this hardwood forest.

Digby and Kempton (1987, Section 6.1) gave suggestions for other advanced methods to tease out subtleties in the nearest-neighbor associations with these data. □

A special case of the general $R \times C$ table occurs when $R = C = 2$. This gives a 2×2 contingency table, as per the schematic in Table 8.5.

For testing the null hypothesis of no association between Factors A and B in a 2×2 contingency table, (8.13) and (8.14) remain valid. These can be applied if the large-sample

Table 8.5 Schematic for 2 × 2 contingency table.

		Factor B		
		Level B_1	Level B_2	Row total
Factor A	Level A_1	Y_{11}	Y_{12}	Y_{1+}
	Level A_2	Y_{21}	Y_{22}	Y_{2+}
	Column total	Y_{+1}	Y_{+2}	Y_{++}

rules-of-thumb hold: all four $\hat{Y}_{oij}$s are at least 1 and $\sum_{i=1}^{2}\sum_{j=1}^{2}\hat{Y}_{oij}/4 \geq 5$. If so, reject the null hypothesis of no association when $X^2 \geq \chi^2_\alpha(1)$.

The 2 × 2 table can be interpreted in a number of ways, however. In particular, suppose the study design has *fixed* the column totals Y_{+1} and Y_{+2}, for example, if the numbers of sample elements at levels B_1 and B_2 were chosen before determining their Factor A status. The sampling then reverts to binomial, with $Y_{1j} \sim$ indep. $\text{Bin}(Y_{+j}, \pi_j)$ for $j = 1, 2$. Focus is now on Factor B and whether differences exist between its two levels when 'success' is defined (generically) by achieving Factor A's level A_1, versus 'failure' at level A_2. (If, alternatively, the row totals are fixed, the interpretation is similar: just transpose the perspective for which variates take on the binomial parent.)

The no-effect null hypothesis can, therefore, be written as H_o: $\pi_1 = \pi_2$, and this will in fact be equivalent to the no-association null hypothesis from the broader 2 × 2 perspective. Now, however, one-sided (along with two-sided) alternatives can conveniently be studied; for example, H_a: $\pi_1 < \pi_2$ asks whether moving from level B_1 to level B_2 increases the probability of A_1 success.

A different sampling constraint may instead force the table total Y_{++} to be fixed but allow all cells to otherwise vary. This induces what is known as a *multinomial distribution* structure – an extension of the binomial – on the table. When properly constructed, however, the various hypotheses and statistics for testing association in the table will all coincide.

Inferences for the 2 × 2 table can be developed using exact calculations, that is, without call to a large-sample reference distribution such as χ^2. This is called *exact testing*, as discussed in Section 5.4.3, and dates back to the theories of Fisher (1935). To calculate the exact *P*-value, Fisher fixed the row and column margins in the 2 × 2 table. He then recognized that the *P*-value associated with the 1 d.f. in a 2 × 2 table is the conditional probability of recovering a tabular configuration as extreme or more extreme – in the direction of H_a – than that actually observed. If this *P*-value drops below the nominal, target, α level, reject H_o in favor of H_a: $\pi_1 < \pi_2$.

The conditional probabilities that make up the exact *P*-value may be indexed by Y_{11}. Conditioning on the margin totals drives Y_{11} to take a *hypergeometric distribution*, with conditional p.m.f.

$$P[Y_{11} = y \mid Y_{1+}, Y_{2+}, Y_{+1}, Y_{+2}] = \frac{\binom{Y_{+1}}{y}\binom{Y_{+2}}{Y_{1+}-y}}{\binom{Y_{++}}{Y_{1+}}} \tag{8.15}$$

for $y = 0, \ldots, v_1$ and where $v_1 = \min\{Y_{1+}, Y_{+1}\}$. The one-sided P-value is then

$$P = \sum_{i=y_o}^{v_1} P[Y_{11} = i \mid Y_{1+}, Y_{2+}, Y_{+1}, Y_{+2}],$$

where y_o is the observed value of Y_{11}. Similar calculations are available if the one-sided alternative is reversed.

This construction with hypergeometric probabilities is known as the *Fisher exact test* (FET), or the *Fisher–Irwin exact test* to acknowledge the contributions by Irwin (1935). In some circles, it is also referred to as a 'hypergeometric test.'

The FET is available for any sample size and is 'exact' in the sense that the false positive rate of the test, say, α_e is never larger than the prespecified nominal rate of α. (Hence, the FET is also 'conservative.' This results from the discrete nature of the hypergeometric p.m.f.). The test can suffer loss of power, however, when the design is highly unbalanced, that is, when Y_{+1} is far larger (or smaller) than Y_{+2}, say, $Y_{+k}/Y_{+\ell} > 20$ ($k \neq \ell$). Kang and Ahn (2008) offered some alternatives for this extreme case.

For computing two-sided P-values, the calculations grow complex, because departure from H_o now involves both directions. Strategies for doing so vary. One possibility, used in the **R** function `fisher.test()`, is to find those tables whose hypergeometric probabilities (8.15) rest less than or equal to the observed table and sum these values. Or, one could modify the core test by identifying some two-sided measure of departure such as X^2 and computing that measure for all possible 2×2 tables with the same row and column totals as those observed. The modified P-value is then the sum of those hypergeometric probabilities corresponding to tables whose measures are larger than that of the observed table. Hirji (2005, Section 7.6) and Lydersen et al. (2009) discussed the FET and exact testing in depth; also see Berger (1996).

Note that, in principle, nothing prevents us from extending the 2×2 FET to any $R \times C$ table, except perhaps that the underlying computations can grow burdensome. For moderate tables/sample sizes, this is a shrinking concern, given ongoing advancements in computing technology. (**R**'s `fisher.test` usually handles moderately sized tables. Very large tables may benefit by specifying its `simulate.p.value=T` option.) For large sample sizes, the χ^2 approximation to Pearson's X^2 exhibits sufficient accuracy and, at least with two-sided alternatives, can be reasonably employed instead.

Example 8.3.4 Asthma/allergy association. Halonen et al. (1997) explored occurrence of respiratory ailments in 755 children (aged 6) who were sensitized to specific allergens. Among the data they reported was a 2×2 comparison between positive allergic reactions via skin-prick tests to the common mold *Alternaria alternata* and prevalence of asthma, as given in Table 8.6.

Here the focus is on differences in asthma response between the two *Alternaria* categories in Table 8.6. In particular, are children who respond positively to the *Alternaria* skin test more likely to be asthmatic? This translates to testing whether H_o: $\pi_1 = \pi_2$ versus H_a: $\pi_1 < \pi_2$, where

$$\pi_1 = P[\text{Asthma pos.} \mid \textit{Alternaria} \text{ neg.}] \quad \text{and} \quad \pi_2 = P[\text{Asthma pos.} \mid \textit{Alternaria} \text{ pos.}].$$

The sample proportions $\hat{\pi}_1 = Y_{11}/Y_{+1} = 42/624 = 6.73\%$ and $\hat{\pi}_2 = Y_{12}/Y_{+2} = 41/131 = 31.30\%$ are certainly suggestive, but a formal inference is still needed.

Table 8.6 A 2 × 2 contingency table data for association between *Alternaria alternata* allergic response and asthma status for 6-year old children.

		Alternaria status		
		Allergic negative	Allergic positive	Row total
Asthma status	Positive	42	41	83
	Negative	582	90	672
	Column total	624	131	755

Source: Halonen et al. (1997).

Sample code to enter the data in **R** is

```
> table2x2.mtx <- matrix( c(42, 582, 41, 90), nrow=2, byrow=F )
> colnames( table2x2.mtx ) <- c( 'Alt neg', 'Alt pos' )
> rownames( table2x2.mtx ) <- c( 'Asthma pos', 'Asthma neg' )
```

The `table2x2.mtx` object is then

```
> table2x2.mtx
             Alt neg  Alt pos
 Asthma pos       42       41
 Asthma neg      582       90
```

mimicking Table 8.6. The FET is implemented via the standalone command

```
> fisher.test( table2x2.mtx, alternative='less' )
```

Note the use of the `alternative='less'` option to perform the one-sided test in the appropriate direction. The consequent FET output contains a variety of components; for our use, the pertinent information is

```
        Fisher's Exact Test for Count Data
 data:  table2x2.mtx
 p-value = 5.014e-13
 alternative hypothesis: true odds ratio is less than 1
```

(The output highlights that technically `fisher.test` tests against the alternative that the *odds ratio*

$$\frac{\pi_1/(1-\pi_1)}{\pi_2/(1-\pi_2)}$$

is less than 1. This is equivalent to testing $\pi_1 < \pi_2$, as desired.) The reported, one-sided P-value is $P = 5.014 \times 10^{-13}$, which is highly significant at any reasonable false positive rate. We find that a positive *Alternaria* skin test does indeed suggest a significantly higher prevalence of asthma in these children. □

Of course, not every study with Poisson count data is as complex as the $R \times C$ table illustrated in Example 8.3.3. Some analyses assess the effect of just a single quantitative predictor, x_i, via a simple log-linear regression: $\log(\mu_i) = \beta_0 + \beta_1 x_i$. Interest then centers on inferences for the log-linear slope parameter β_1, including its MLE, $1 - \alpha$ confidence limits, and/or tests

of the null hypothesis H_o: $\beta_1 = 0$. Both the Wald and LR tests (and their related confidence intervals) perform adequately in this case. (All these operations are available via **R**'s `glm()` function.) A CA trend test for Poisson counts is also available, as described in Exercise 8.15.

8.3.4 Gamma regression models

A common form of data in large-scale analytics involves positively valued outcomes, $Y > 0$, with nonconstant variances. Data in this form may be modeled with the gamma distribution from Section 2.3.8. There, the p.d.f. was given in (2.32) via its traditional form:

$$f_Y(y) = \frac{1}{\Gamma(\alpha)\beta^\alpha}\, y^{\alpha-1}\, e^{-y/\beta}\, I_{(0,\infty)}(y).$$

Exercise 8.16 shows, however, that by reparameterizing α and β in terms of the population mean $E[Y] = \mu = \alpha\beta$ and of the dispersion parameter $\varphi = 1/\alpha$, the p.d.f. satisfies the exponential family criterion in (8.1). The exercise further shows that $\text{Var}[Y] = \mu^2\varphi$ (so the variance is quadratic in μ) and that the coefficient of variation, $\sqrt{\text{Var}[Y]}/E[Y]$ from Section 2.3.7, is, therefore, constant with respect to μ. Thus positive-valued data with constant coefficients of variation can be analyzed on their original scale of measurement via a gamma GLiM (also called a 'gamma regression.')

A caveat: some confusion exists on what to call φ with this model. Many sources refer to φ as the exponential family *scale* parameter. In the general literature, however, most authors refer to β (or sometimes $1/\beta$) as the 'scale parameter' under a gamma model, with α as the 'shape parameter.' Analysts must ensure they know precisely which parameter is being used for which purpose when fitting gamma GLiMs to data.

As $Y > 0$ and, therefore, $E[Y] = \mu > 0$, a useful link function to employ with gamma GLiMs is the natural logarithm: $\eta = g(\mu) = \log(\mu)$, with strictly positive inverse link $\mu = h(\eta) = e^\eta$. This is a form of log-linear model for continuous measurements, because it relates a linear predictor to the mean response via a logarithmic link.

Given a set of predictor variables for the linear predictor η_i, estimation proceeds via the IWLS/ML approach in Section 8.2.1. To estimate the dispersion parameter, the Pearson-based χ^2 moment estimator $\hat{\varphi}_P$ is recommended. One could also extend the ML calculations by assuming φ is unknown in (8.3) if desired – see the `gamma.dispersion()` function from the *MASS* package – although both $\hat{\varphi}_{\text{ML}}$ and the deviance-based moment estimator $\hat{\varphi}_D$ are known to exhibit instabilities with the gamma model (Faraway 2006, Section 7.1).

Example 8.3.5 **US emergency room expenditures.** The US Department of Health and Human Services studies patterns in US health care costs through its Medical Expenditure Panel Survey (MEPS); see http://meps.ahrq.gov/mepsweb/. For instance, Table 8.7 gives data on total expenditures (in $) from a 2009 sample of the US population. The sample contains nonzero expenditures from $n = 6438$ emergency room (ER) visits. (As previously, the table gives only a portion of the data. The complete data is available online at http://www.wiley.com/go/piegorsch/data_analytics.) The data are stratified by whether the ER visit involved an actual 'emergency' (accident or injury) versus any nonemergency event.

Of interest with these data is whether total expenditures differ significantly between the two types of ER visits. This is a basic two-sample comparison, as seen in Section 5.4.2. The methods in that section require, however, that the parent distribution for Y be normal (Gaussian). Monetary expenditure data are notoriously skewed – as is true here; see

Table 8.7 Total nonzero expenditures (in $) for $n = 6438$ ER visits sampled throughout the United States in 2009.

Visit category	Total expenditure
Nonemergency	0.92, 1.23, 1.68, 2.68, 3.00, … , 25 413.91, 25 413.91, 25 413.91, 25 683.96, 34 053.42
Emergency	0.18, 3.00, 4.20, 6.89, 7.68, … , 25 413.91, 25 413.91, 31 057.14, 37 165.86, 37 492.08

Abbreviation: ER, emergency room.
Source: http://meps.ahrq.gov/mepsweb/.

Exercise 8.17 – and not directly amenable to normal-based analyses. One could consider a stabilizing transformation, as in Section 3.4.3, although it is instructive to first examine the data more closely.

In particular, if the variable `Y` contains the data and the variable `visit` indicates the visit category (`0` for nonemergency; `1` for emergency), then sample **R** code to calculate the coefficient of variation for each subsample is

```
> sd(Y[visit==0])/mean(Y[visit==0])    #nonemergency
> sd(Y[visit==1])/mean(Y[visit==1])    #emergency
```

which produces

```
[1] 2.0956
[1] 2.0723
```

respectively. The two coefficients are almost identical, suggesting appeal to a gamma parent distribution. The two-sample comparison can then be performed by employing a gamma GLiM, using the single *qualitative* predictor variable `visit` to delineate the visit categories. Sample **R** code is

```
> expER.glm = glm( Y ~ factor(visit), family=Gamma('log') )
> anova( expER.glm, test='F' )
```

Notice the use of `factor()` to produce the qualitative predictor. Because the gamma dispersion parameter must be estimated, the call to `anova()` includes the `test='F'` option. This employs an F-statistic to calculate the sequential P-values, as described in Section 8.2.4.

The use of a single qualitative predictor generates predicted values $\hat{Y}_i$ from the `glm` fit which on the response scale are simply the corresponding subsample means. Find these via

```
> Yhat = predict( expER.glm, type='response' )
> round( unique(Yhat), digit=2 )
```

This gives average expenditures of $914.61 for non-ER visits and $977.34 for ER visits.

Output (edited) from the call to `anova()` for the analysis of deviance table is

```
Analysis of Deviance Table
Model: Gamma, link: log
Response: Y
Terms added sequentially (first to last)
              Df  Deviance  Resid.Df  Resid.Dev      F  Pr(>F)
NULL                           6437      10710
factor(visit)  1    7.0261      6436      10703  1.616  0.2037
```

The approximate P-value to test for differences between the two levels of the `visit` factor is seen to be $P = 0.2037$. This is insignificant at any reasonable false positive rate. We conclude that total expenditures do not appear to vary significantly between these two types of ER visits.

In passing, recall that the Pearson estimate for φ is found by summing the squares of the Pearson residuals and dividing by the residual d.f. Sample **R** code is

```
> ci = resid( expER.glm, type='pearson' )
> phihat.P = sum( ci^2 )/expER.glm$df.residual
```

from which we find $\hat{\varphi}_P = 4.3480$. This estimate is also found as a component of the output (not shown) from `summary(expER.glm)`. □

Exercises

8.1 As noted in Section 8.1.1, the normal (Gaussian) distribution is a member of the exponential family. Thus, for example, the SLR model from Section 6.2.1 qualifies as a GLiM. For illustration, fit this SLR as a GLiM to the data on UK mortality in Table 6.1.

(a) Appeal to **R**'s `glm()` function using the `gaussian` family. Produce a `summary` output of the resulting `glm` object and compare the results to the output from `summary(lm(Y ~ x))` as displayed in Example 6.2.2. Comment on any differences.

(b) Also construct an analysis of deviance table via a call to `anova()`. As the dispersion parameter $\varphi = \sigma^2$ here is estimated from the data, use the `test='F'` option as in Section 8.2.4. What inferences can you make using this table? What is the estimate of φ?

(c) One can construct an ANOVA table to study variation in any SLR (or MLR) model, mimicking the format in Table 7.6. Do so with these data, via the **R** command `anova(lm(Y ~ x))`. Compare this to the output in Exercise 8.1b.

8.2 Upadhyay (2014) presented proportion data, Y_i/M_i, describing a large bank's loan defaults as a function of the borrower's age group. The predictor values, x_i, are taken as the mid-points of the group ranges. The complete data set is available online at http://www.wiley.com/go/piegorsch/data_analytics; a portion of it is given as follows:

Age group (years)	21–24	24–27	27–30	30–33	33–36	···	57–60
Midpoint (years)	22.5	25.5	28.5	31.5	34.5	···	58.5
Y_i = Defaults	14	20	172	169	188	···	9
M_i = Loans	310	511	4000	4568	5698	···	788

(a) Plot the proportions Y_i/M_i against x_i and comment on the result.

(b) View load default as a 'success' and analyze these data via a binomial GLiM. Use a simple linear predictor, $\eta_i = \beta_0 + \beta_1 x_i$. Apply each of (i) the logit link, (ii) the probit link, and (iii) the complementary log-log link. (In **R**, use `family=binomial('probit')` and `family=binomial('cloglog')`, as needed, in the call to `glm()`.) Identify which of the three links (logit, probit, or complementary log–log) gives the best fit, as gauged by smallest stability measure $D^*(\hat{\boldsymbol{\beta}})/(n-p-1)$.

(c) For the best-fitting link found in Exercise 8.2b, find a 90% profile likelihood confidence interval on the regression parameter, β_1. (*Hint*: Recall the `confint()` function in **R**. Be sure to load the *MASS* package first.) Interpret your finding; in particular, does the interval contain $\beta_1 = 0$ and if so/if not, what does this suggest about loan default rates at this bank?

(d) For the best-fitting link found in Exercise 8.1b, find the standardized deviance residuals $d_i/\sqrt{1-h_{ii}}$ from the model fit and plot these against the predicted probabilities. (*Hint*: In **R**, the `hatvalues()` and/or `rstandard()` functions may prove useful.) Comment on any patterns in the plot.

8.3 Return to the rainbow trout carcinogenesis data in Exercise 5.34. Of additional interest here is assessing whether or not a significantly increasing trend in cancer occurrence is observed over the exposure dose x_j. Consider the following.

(a) Plot the observed proportions Y_j/n_j against the dose x_j. What pattern appears?

(b) Apply the CA trend test for proportions to these data, using the Z_{CA} statistic from (8.10). As the doses are geometrically spaced, include possible adjustment for skew in the predictor variable via (8.11). Operate at $\alpha = 0.01$. What do you conclude? How does this compare to the inferences achieved in Exercise 5.34?

(c) Another way to adjust for skew in the dose/exposure regime here is to apply a logarithmic transform to x_j. As adjacent doses are essentially doubled, calculate $u_j = \log_2(x_j)$. Replace x_j with u_j in your calculations from Exercise 8.3b. Does the skewness adjustment appear necessary? (What is $\hat{\gamma}$?) Do the conclusions change in any meaningful way?

8.4 The following data, modified from Silvapulle (1981), give proportions of male psychiatric patients' responses to a general health questionnaire (GHQ). The predictor variable, x, is an integer-valued score quantifying decreasing health status as x increases. The response Y is number of patients ('cases') exhibiting psychiatric disorders, out of a total M at that value of x.

GHQ score	0	1	2	4	5	7	10
Y_i = 'Cases'	0	0	1	1	3	2	1
M_i = Total tested	18	8	2	1	3	2	1

(a) Plot the proportions Y_i/M_i against x_i. Is there a visible increase in the response as x grows?

(b) Fit a logistic regression model with a simple linear predictor $\eta_i = \beta_0 + \beta_1 x_i$ to these data. Find the MLE of β_1. Also find its standard error and construct the Wald statistic for testing H_o: $\beta_1 = 0$ against H_o: $\beta_1 > 0$. What do you conclude at $\alpha = 0.05$? Does this seem unusual?

(c) The strange result in Exercise 8.4b results from an instability in the MLE with this sort of response pattern. Return to your data plot and notice that no overlap

exists across the predictor scale between proportions exhibiting only nonresponses ($Y/M = 0$) and proportions exhibiting complete responses ($Y/M = 1$). When this occurs in logistic regression the data are *separated*, and the likelihood will not have a finite maximum for any value of β_1 (Hosmer and Lemeshow 2013, Section 4.4). The MLE of β_1 is, therefore, infinite or, more properly, does not exist. (**R**'s attempt to report a finite MLE is a consequence of this instability. This effect is not limited to completely separated data as given here; Silvapulle (1981) discussed a form of quasi-complete separation that can lead to infinite parameter estimates as well.) From your output, calculate the log odds ratio for a unit change in x and comment on the result.

(d) Use the CA statistic to test for an increasing trend. Does this test exhibit any instabilities? If not, what do you conclude at $\alpha = 0.05$?

8.5 Show that the CA trend statistic for proportions is invariant to linear transformations of the predictor variable; that is, show algebraically that replacing x_i with $u_i = a + bx_i$ in (8.10), for known constants a and b, does not change the value of the test statistic.

8.6 Return to the GLiM analysis of the joint-action toxicology data in Example 8.3.1.

(a) Under the logit link, verify that by adding quadratic terms in x_1 and x_2 to the liner predictor provides no significant improvement in the model fit. Operate at a false positive rate of 5%.

(b) Find the standardized deviance residuals $d_i/\sqrt{1-h_{ii}}$ from the RM fit using only x_1 and x_2, and plot these against the predicted probabilities. (*Hint*: In **R**, the `hatvalues()` and/or `rstandard()` functions may prove useful.) Verify that the pattern does not differ drastically from that in Figure 8.1.

(c) Repeat the analysis given in the example, now using (i) a probit link and (ii) a complementary log–log link. (In **R**, use the `family=binomial('probit')` and `family=binomial('cloglog')` options, respectively.) Identify which of the three links (logit, probit, or complementary log–log) gives the best fit, as gauged by smallest stability measure $D^*(\hat{\boldsymbol{\beta}})/(n-p)$. (One could alternatively employ an information criterion such as the Akaike information criterion (AIC) as in Section 7.3.2, although here either measure will give the same qualitative result.) For the best model, do the final results change drastically from those in the example?

8.7 Lindsay and Liu (2009) present a contingency table illustrating an archetypal biological association: hair color and eye color in humans. The data, over 592 subjects, form a 4×4 table:

		Eye color			
		Brown	Blue	Hazel	Green
Hair color	Black	68	20	15	5
	Brunette	119	84	54	29
	Red	26	17	14	14
	Blonde	7	94	10	16

(a) Check the large-sample rules-of-thumb for application of Pearson's X^2 statistic (8.14) to this table. If valid, use X^2 to assess whether hair color and eye color exhibit a significant association. Operate at a false positive rate of 10%.

(b) Calculate the standardized Pearson residuals from the χ^2 analysis and determine if patterns exist that help to describe any association in the table.

8.8 Similar to the Altmetric data in Exercise 5.18, a study was conducted of social media posts for a sample of 13 023 scientific articles appearing in a database collected by researchers at altmetric.com. The articles were grouped by general subject area and then the number of Facebook, Twitter, and general blog entries associated with each article were counted. The data, kindly provided by Mr. Euan Adie of altmetric.com, form a 6×3 contingency table, as follows:

	Altmetric source		
	Facebook	Twitter	Blogs
Biology and Medicine	2476	25192	1036
General	873	11053	620
Humanities	34	514	23
Mathematics and Information Science	49	800	51
Physical Science and Engineering	336	3104	336
Social Science and Business	239	4140	242

(a) One might ask whether the pattern of social media posts differs between the different subject areas. This translates to an interaction between article subject area and Altmetric source. To assess this, check the large-sample rules-of-thumb for the application of Pearson's X^2 statistic (8.14) with this table. If valid, use X^2 to assess if a significant interaction exists. Operate at a false positive rate of 1%.

(b) Calculate the standardized Pearson residuals from the χ^2 analysis and determine if patterns exist that help to describe any interaction in the table.

8.9 Ezzikouri et al. (2008) reported on a study of liver cancers among 318 hepatitis patients, relating their cancer status (present or absent) to the human manganese superoxide dismutase *MnSOD* gene. Of interest was whether any association exists between cancer status and different genotypic polymorphisms of *MnSOD*. The data comprise a 3×2 table, as follows:

		Liver cancer status	
		Absent	Present
MnSOD polymorphism	Val/Val	21	81
	Val/Ala	45	101
	Ala/Ala	30	40

(a) Check the large-sample rules-of-thumb for the application of Pearson's X^2 statistic (8.14) to this table. If valid, use X^2 to assess if cancer status exhibits any association with the genetic factor. Operate at a false positive rate of 5%.

(b) Calculate the standardized Pearson residuals from the χ^2 analysis and determine if patterns exist that help to describe any association in the table.

8.10 Faraway (2006, Section 4.1) displayed data on semiconductor wafer manufacturing in an industrial plant, relating the presence of contaminant particles on machine dies to quality of the wafers produced by those dies. Of interest was whether any association exists between contamination status and wafer quality. A sample of 450 wafers produced the following 2 × 2 table:

		Wafer quality		
		Good	Bad	Row total
Contamination	Absent	320	80	400
	Present	14	36	50
	Column total	334	116	450

Use the FET to assess if any association is evidenced between the two factors in this table. Also apply Pearson's X^2 statistic from (8.14). (Verify first if the large-sample rules-of-thumb are valid for the application of X^2.) How do the two inferences compare at a nominal false positive rate of 5%?

8.11 Recall Exercise 5.29, where the concept of 'A–B testing' was discussed: a web site designer tests whether a new site (variant 'B') attracts more customers, compared to the current site (variant 'A'). When such testing involves two *independent* groups of Internet users, the data essentially fall into a 2 × 2 table for which the FET would be appropriate. A new design ('B') for a web site's signup page was tested on $Y_{+2} = 2000$ volunteers, along with the original design ('A') on an independent set of $Y_{+1} = 2000$ volunteers. $Y_{12} = 793$ of the 2000 volunteers clicked on the B-design's Sign Up Now link, compared with $Y_{11} = 610$ for the A-design's link. Use an FET to assess if the B-group exhibited a significant increase in sign-up activity, compared with the A-group. Operate at a nominal false positive rate of 1%.

8.12 In a US Census Bureau study of annual incomes (in 1994 $), data were collected on whether or not US residents reported incomes above $ 50 000 (Kohavi 1996). Stratified by sex of the respondent, the data showed that $Y_{11} = 1179$ out of $Y_{+1} = 10\,771$ women exceeded the $ 50K threshold, while $Y_{12} = 6662$ out of $Y_{+1} = 21\,790$ men exceeded the threshold. Do these data indicate that US women whose 1994 incomes exceeded $ 50 000 did so significantly less often than US men? Justify your answer. (Operate at a 5% false positive rate.)

8.13 Verify the indication in Example 8.3.4 that the null hypotheses H_o: $\pi_1 = \pi_2$ and

$$H_o : \frac{\pi_1/(1-\pi_1)}{\pi_2/(1-\pi_2)} = 1$$

are equivalent. Also verify that the alternative hypotheses H_a: $\pi_1 < \pi_2$ and

$$H_a : \frac{\pi_1/(1-\pi_1)}{\pi_2/(1-\pi_2)} < 1$$

are equivalent. (A similar equivalence will hold for one-sided upper alternatives.)

8.14 From a study of species biodiversity, Carroll (1998) gave data on numbers of different bird species observed throughout the Indian subcontinent, as a function of altitudinal relief (i.e., the difference between the highest and lowest elevations above sea level in a study region of fixed size). To predict $Y = \{\text{Number of species in a } 1\,\text{km}^2 \text{ region}\}$, take $x = \log\{\text{Relief}\} + 2$. (2 is added to make all log-relief values positive.) The data are available online at http://www.wiley.com/go/piegorsch/data_analytics; a sample is given as follows:

$x = \log\{\text{Relief}\} + 2$	4.050	3.452	3.662	3.917	$\cdots$	2.571	2.916
$Y =$ Number of species	261	224	171	326	$\cdots$	225	193

(a) Plot the species counts against x. Does any pattern appear?

(b) Fit a log-linear Poisson regression model to the data, using the simple linear predictor $\eta_i = \beta_0 + \beta_1 x_i$. Calculate the stability measure $D^*(\hat{\boldsymbol{\beta}})/(n-p-1)$ for this fit. Verify that it violates the rule-of-thumb in (8.7).

(c) A common reason for $D^*(\hat{\boldsymbol{\beta}})/(n-p-1)$ to violate (8.7) with counts is the presence of *overdispersion* in the data, that is, where $\text{Var}[Y_i]$ grows larger than that predicted under the parent Poisson assumption. A useful alternative p.m.f. for Y in this case is the negative binomial distribution from Section 2.3.4, using its μ, δ parameterization in (2.25). When δ is unknown, this is not technically a GLiM; however, regression estimators can still be calculated using ML via, for example, **R**'s `glm.nb()` function from the *MASS* package. Apply this to the species counts: continue to use the log link and a simple linear predictor. Does the stability measure $D^*(\hat{\boldsymbol{\beta}})/(n-p-1)$ improve under the negative binomial parent?

(d) Build an analysis of deviance table from the negative binomial log-linear regression fit in Exercise 8.14c. Is the x variable's contribution significant at $\alpha = 0.01$? Include a 99% confidence interval for β_1 in your calculations.

(e) Find the standardized deviance residuals from the fit in Exercise 8.14c and plot these against the predicted response. Do any untoward patterns appear?

8.15 A CA trend test also exists for independent count data. Suppose $Y_{ij} \sim \text{Poisson}(\mu[x_i])$, $i = 1, \ldots, n; j = 1, \ldots, J_i$. The no-trend null hypothesis is H_o:$\mu(x_1) = \cdots = \mu(x_n)$. At each x_i, let the ith sample mean be $\overline{Y}_{i+} = \sum_{j=1}^{J_i} Y_{ij}/J_i$. If H_o is true, each $\overline{Y}_{i+}$ will approximate the pooled mean $\overline{Y}_{++} = \sum_{i=1}^{n}\sum_{j=1}^{J_i} Y_{ij} / \sum_{i=1}^{n} J_i$. Using these quantities, the CA test quantifies departure from H_o via the statistic

$$Z_{CA} = \frac{\sum_{i=1}^{n} J_i(x_i - \overline{x})\overline{Y}_{i+}}{\sqrt{\overline{Y}_{++}\sum_{i=1}^{n} J_i(x_i - \overline{x})^2}},$$

where $\bar{x} = \sum_{i=1}^{n} J_i x_i / \sum_{i=1}^{n} J_i$. Reject H_o in favor of an increasing (decreasing) trend when Z_{CA} is greater (less) than or equal to z_α ($-z_\alpha$). (Two-sided testing is also possible.)

(a) Show that this CA trend statistic for counts is invariant to linear transformations of the predictor variable; that is, show that replacing x_i with $u_i = c + dx_i$, for known constants c and d, does not change the value of Z_{CA}.

(b) In the presence of overdispersion, as in Exercise 8.14c, Z_{CA} will be unstable and reject H_o too often. A adjustment given by Astuti and Yanagawa (2002) constructs the generalized CA statistic

$$Z_{GCA} = \frac{\sum_{i=1}^{n} J_i (x_i - \bar{x}) \overline{Y}_{i+}}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{j=1}^{J_i} (Y_{ij} - \overline{Y}_{++})^2}}.$$

Reject H_o in favor of an increasing (decreasing) trend when Z_{GCA} is greater (less) than or equal to z_α ($-z_\alpha$). Reject against a two-sided alternative if $|Z_{GCA}| \geq z_{\alpha/2}$. Apply this test to the data in Exercise 8.14; test against two-sided departure from H_o. How do the conclusions compare at $\alpha = 0.01$?

8.16 Let $Y \sim \text{Gamma}(\alpha, \beta)$ from (2.32).

(a) Show that the gamma p.d.f. can be written in the form of an exponential family density, as in (8.1), by reparameterizing the traditional parameters α and β in terms of the mean $\mu = \alpha\beta$ and the dispersion parameter $\varphi = 1/\alpha$.

(b) Show that the population variance is $\text{Var}[Y] = \mu^2\varphi$.

(c) Consider the population coefficient of variation, $\sqrt{\text{Var}[Y]}/\mu$, defined in Section 2.3.7. Show that this is constant with respect to μ.

(d) Can you find another distribution whose coefficient of variation is constant with respect to its mean?

8.17 Return to the ER cost data in Example 8.3.5 and verify the skew in the data: plot histograms for each subsample's total expenditures. (Use Scott's normal reference rule from Section 4.1.4 for bin selection.) Include rug plots and overlay a kernel density estimator on each histogram.

8.18 Similar to the ER cost data in Example 8.3.5, the US MEPS also sampled total nonzero expenditures (in $) with $n = 3272$ in-patient hospital stays for calendar year 2009; see http://meps.ahrq.gov/mepsweb/. Almost half (46.7%) of the stays were preceded by an ER visit which then resulted in hospital admission. The rest were direct admissions. Using this as a stratifying marker produces another two-sample data set, a portion of which is given as follows (the complete data are available online at http://www.wiley.com/go/piegorsch/data_analytics):

Route of admission	Total expenditure ($)
ER admission	12.47, 20.03, 24.74, 29.36, … , 14 1122.60, 15 3802.80, 15 4155.70, 15 4758.80, 29 4472.60
Direct admission	5.96, 13.28, 16.10, 24.26, 25.10, … , 20 0351.20, 20 7024.10, 22 4619.70, 24 0235.40, 28 9176.20

Perform a two-sample analysis analogous to that seen in Example 8.3.5. To visualize the variation, start with histograms, rug plots, and overlayed kernel density estimates for each subsample's total expenditures. Next, calculate the two subsamples' means and standard deviations, and from these, determine if the two coefficients of variation are roughly equal. If so, invoke a gamma GLiM and assess if any significant difference exists between the two admission routes. Operate at a false positive rate of 5%. What do you conclude? Also estimate the unknown dispersion parameter, φ.

8.19 The study of in-patient hospital expenditures in Exercise 8.18 also recorded $x = \{\text{Number of nights}\}$ associated with each in-patient stay in 2009. The paired data are are available online at http://www.wiley.com/go/piegorsch/data_analytics; a sample is given follows:

x = Number of nights:	3	1	$\cdots$	20	5
Y = Expenditure ($):	5041.81	12 748.45	$\cdots$	20 563.31	9556.04

(a) Plot Y against x. Comment on the observed pattern.

(b) Fit a gamma GLiM using Y as the response and x as a single quantitative predictor. Construct an analysis of deviance table and test if x has a significant impact on the mean response. Operate at a false positive rate of 1%. Also find an estimate of the unknown dispersion parameter, φ.

(c) Calculate a 99% profile likelihood confidence interval on the regression parameter β_1 associated with x. Comment on the statistical inference being produced.

(d) Determine the predicted response $\hat{\mu}(x)$ and overlay the predicted curve on your scatterplot in Exercise 8.19a. Comment further on the pattern(s) observed.

(e) Find the standardized deviance residuals $d_i/\sqrt{1-h_{ii}}$ from the model fit and plot these against x. (*Hint*: In **R**, use the `rstandard()` function with option `type='deviance'`.) Do you see a possible outlier? If so, excise it and reanalyze the remaining data with the gamma GLiM. Include a new (standardized) residual analysis. What changes, if any, are evident?

9

Supervised learning: classification

Along with the regression-based approaches seen in Chapters 6–8, another important form of supervised learning involves methods of *classification*. From a broad perspective, regression and classification are similar: both employ a set of p inputs quantified as a vector of predictor or feature variables $\mathbf{X}_i = [X_{i1}\ X_{i2}\ \cdots\ X_{ip}]^{\mathrm{T}}$ to explain the value of an outcome variable Y_i. For the classification setting, however, Y_i represents a series of discrete categories, $\{C_1, C_2, \ldots, C_Q\}$, exclusively and exhaustively defining the possible states within which the ith subject or element lies. For example, in a cancer classification study, the C_qs could be different stages or types of cancer identified in an ith patient's tissue sample and $\mathbf{X}_i$ could represent microarray expression levels from p different genes that potentially relate to cancer status. The goal is to determine an accurate classification scheme for stages of the cancer based on the various predictor inputs (cf. Exercise 9.3). It is prior known specification of the C_q categories that makes this a 'supervised' effort. (When no predefined categories exist, the process is 'unsupervised' and may appear as a form of *cluster analysis*, as described in Chapter 11.)

This chapter gives an overview of some standard methods for supervised classification. In most cases, the topics are mature enough to demand their own standalone chapters; hence, the descriptions in this single chapter are intended only as an introduction. Selected references are provided throughout for readers interested in pursuing the methods in more detail.

The n observed data pairings $(\mathbf{X}_i, Y_i)$ are often viewed as a *training set* of data from which the classification scheme is estimated. The given scheme may then be applied to a separate *test set*, where independent values of $\mathbf{X}_i$ are sequestered or collected with the aim of predicting their associated categories Y_i. In contrast to the regression setting, however, the statistical goals in classification may be more exploratory and less inferential in nature. This then follows the

Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery,
First Edition. Walter W. Piegorsch.

www.wiley.com/go/piegorsch/data_analytics

general strategy of an exploratory data analysis (EDA) often seen in statistical learning (cf. Section 1.3).

It is useful to differentiate the exercise into two slightly different forms: *discrimination* (or *discriminant analysis*) and focused *classification*. The former is more traditional, where interest centers on identifying patterns and structure in the data by discriminating between separable classes. The latter is more prevalent when the focus is on prediction of the class or category within which a new test case may lie. In many settings, the distinction between the two is ambiguous, however, and the terms are increasingly applied interchangeably.

Discriminant analyses are sometimes conducted by calculating discriminant *scores*, denoted generically as $\delta_q(\mathbf{X})$. The scores are used to quantify the association between $\mathbf{X}$ and category or class C_q. Usually, a high value of $\delta_q(\mathbf{X})$ favors inclusion in C_q. Indeed, it can be convenient for the score vector $[\delta_1(\mathbf{X}) \cdots \delta_Q(\mathbf{X})]^{\mathrm{T}}$ to itself represent the class inclusion probabilities. Then, we assign a datum $\mathbf{X}$ to the class for which it has the highest probability of inclusion (assuming the Q classes are afforded equal a priori weight or cost).

If the inputs exhibit stochastic variation, the vector $\mathbf{X}$ is assigned a distributional structure, for example, via some p-variate joint probability function $f(\mathbf{x})$. When this stochastic feature is not required, any statistical inferences are constructed conditionally on the observed pattern of the $\mathbf{X}_i$s.

9.1 Binary classification via logistic regression

In the simplest form of supervised classification, only two possible categories exist, so $Q = 2$. This is the simple binary classifier/discriminant problem. The outcome variable is taken as $Y_i = 1$ when subjects populate the target, 'success' category C_1, and as $Y_i = 0$ if they populate the remaining reference or 'background' category C_2. (If no target or background categories exist, then we arbitrarily choose either category as the 'success' and delineate it consistently so throughout the training set.)

9.1.1 Logistic discriminants

Binary outcomes lead naturally to consideration of a binomial parent distribution for Y. Assume each observation represents an independent Bernoulli trial such that $Y_i \sim$ indep. $\mathrm{Bin}(1, \pi_i)$. Here $\pi_i = \pi(\mathbf{x}_i)$ models the success probability as a function of $\mathbf{x}_i$, conditional on the observed realizations of the predictor values $\mathbf{X}_i = \mathbf{x}_i$ $(i = 1, \ldots, n)$. In effect, $\pi(\mathbf{x})$ operates as a potential score function for classification purposes. From Section 8.3.1, the ubiquitous logistic function is then a popular choice for $\pi(\cdot)$:

$$\pi(\mathbf{x}_i) = \pi(x_{i1}, \ldots, x_{ip}) = \frac{1}{1 + \mathrm{e}^{-\eta_i}},$$

where, as in Chapter 8,

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$$

denotes the linear predictor. The conditional probability of category residence is modeled via

$$P[Y_i = 1 \mid \mathbf{X}_i = \mathbf{x}_i] = \frac{1}{1 + \exp\{-\beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip}\}}, \tag{9.1}$$

a form of logistic regression for Y_i.

Given data, estimation proceeds via the iteratively (re)weighted least squares (IWLS)/maximum likelihood (ML) algorithm from Section 8.2.1, producing a maximum likelihood estimate (MLE), $\hat{\boldsymbol{\beta}} = [\hat{\beta}_0 \cdots \hat{\beta}_p]^{\mathrm{T}}$, for the vector of regression coefficients. Appealing to ML invariance (Section 5.2.4), the $\hat{\beta}_q$s are substituted into the corresponding expressions for η_i and $\pi(\mathbf{x}_i)$ to yield MLEs for the ith linear predictor and success probability, respectively.

When predictive classification is the goal, the estimator $\hat{\pi}(\mathbf{x})$ provides a sense of how common or likely the 'success' category is to be populated by a new/test observation with feature vector $\mathbf{x}$. Indeed, formal inferences are possible: a (pointwise) conditional, large-sample, $1-\alpha$ confidence interval for $\pi(\mathbf{x})$ starts with the Wald interval on η: for any given $\mathbf{x} = [x_1 \cdots x_p]^{\mathrm{T}}$, find $\hat{\eta} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$ along with its standard error se$[\hat{\eta}]$. Take $z_{\alpha/2}$ as the upper-$\frac{\alpha}{2}$ standard normal critical point, and form the $1-\alpha$ Wald confidence interval for η as $\hat{\eta} \pm z_{\alpha/2}se[\hat{\eta}]$. Then, apply this in the logistic model to produce the interval for $\pi(\mathbf{x})$:

$$\frac{1}{1 + \exp\{-\hat{\eta} \mp z_{\alpha/2}\mathrm{se}[\hat{\eta}]\}}\,. \tag{9.2}$$

Notice that se$[\hat{\eta}]$ is the square root of the (estimated) variance of $\hat{\eta}$. This involves a sum of the variances and covariances of the $\hat{\beta}_j$s, as in (2.16), which are found from the estimated covariance matrix for $\hat{\boldsymbol{\beta}}$. These are usually taken from the output of a logistic regression program. One could alternatively find se$[\hat{\pi}(\mathbf{x})]$ directly and then report the Wald interval $\hat{\pi}(\mathbf{x}) \pm z_{\alpha/2}se[\hat{\pi}(\mathbf{x})]$; however, this is not guaranteed to lie wholly within the unit interval, unlike (9.2), and is, therefore, less preferred.

For discriminant calculations, the logistic regression estimators can be used to characterize separation between the two categories. Given a feature input $\mathbf{x}$, the discriminant function favors $Y = 1$ when $\hat{\pi}(\mathbf{x}) \geq \frac{1}{2}$, and $Y = 0$ otherwise (again, assuming the two categories are assigned equal a priori weight or cost). Under the conditional logistic model, this simplifies to

$$\hat{\eta} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p \geq 0. \tag{9.3}$$

When the feature inputs are all of first order, (9.3) corresponds to a *linear discriminant function* for the boundaries between categories. Higher-order terms such as quadratic predictors or interactions – for example, x_{i1}^2 or $x_{i1}x_{i2}$, respectively – lead to curvilinear discriminants. Graphing these relationships is often a useful way to visualize the discriminant outcome.

Example 9.1.1 Wholesale distribution channel study. Cardoso (2013) described a business study of the client base for a Portuguese wholesale food distributor. The database includes information on annual client spending (given as generic monetary units, m.u.) on food products, and the distribution channel – Retail versus Service (Hotel/Restaurant/Cafe) – in which the client resides. For the purposes of illustration, we work here with a subset of the data, restricting attention to the cohort of clients outside of the two largest Portuguese cities of Lisbon and Porto. This gives a data set containing $n = 316$ client records.

Of interest is discrimination between the two classes of distribution channels the clients occupy and in particular, whether ($Y = 1$) or not ($Y = 0$) a client's channel is Retail. Focus on two key annual spending markers: (i) fresh food products and (ii) grocery products. Because spending outcomes are notoriously skewed, use the natural logarithms of these inputs: $X_1 = \log\{\text{Fresh-product spending}\}$ and $X_2 = \log\{\text{Grocery spending}\}$. A third predictor representing the interaction X_1X_2 is also calculated. Table 9.1 presents a selection of the log-transformed data. (The original set is available at http://www.wiley.com/go/piegorsch/data_analytics.)

Table 9.1 Selected data from a larger set of $n = 316$ observations recorded on distribution channel (Retail channel: $Y = 1$; Service channel: $Y = 0$) and annual spending (in generic monetary units, m.u.) for clients of a wholesale food distributor.

Y = Distribution channel:	1	1	1	$\cdots$	0	0
X_1 = log{Fresh-product spending}:	9.447	8.862	8.757	$\cdots$	9.239	7.933
X_2 = log{Grocery spending}:	8.931	9.166	8.947	$\cdots$	7.711	7.828

For these data, the conditional logistic model is

$$P[Y_i = 1 \mid X_{i1} = x_{i1}, X_{i2} = x_{i2}] = \frac{1}{1 + \exp\{-\beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \beta_3 x_{i1} x_{i2}\}} . \tag{9.4}$$

Similar to that seen in Section 8.3.1, sample **R** code for fitting this model to the data is

```
> X12 <- X1 * X2
> wholesale.glm <- glm( Y ~ X1+X2+X12, family=binomial('logit') )
> summary( wholesale.glm )
```

This produces output (edited) indicating

```
  Coefficients:
               Estimate Std. Error z value Pr(>|z|)
 (Intercept)  -109.8416    32.2516  -3.406 0.000660
  X1             8.7513     3.3309   2.627 0.008606
  X2            12.2642     3.5842   3.422 0.000622
  X12           -0.9805     0.3704  -2.647 0.008123
 (Dispersion parameter for binomial family taken to be 1)

      Null deviance: 401.81  on 315  degrees of freedom
  Residual deviance: 174.50  on 312  degrees of freedom
```

from which the estimated discriminant function is found to be

$$\hat{\eta}(x_1, x_2) = -109.8416 + 8.7513 x_1 + 12.2642 x_2 - 0.9805 x_1 x_2.$$

This is a curve in (x_1, x_2)-space providing a discriminant boundary between Retail clients (above the curve) and Service clients (below the curve). Figure 9.1 plots the actual points from this sample (solid circles indicate Retail clients, open circles Service clients), along with the curvilinear discriminant. The graph suggests that higher annual spending tends to associate with the Retail distribution channel, particularly along the grocery metric. This is useful knowledge discovery for a distributor wishing to identify client spending patterns. The discriminant curve also appears to do a reasonable job of separating the two classes: more retail clients tend to lie above the curve – as desired – and more service clients tend to lie below it. (Also see Section 9.1.2.)

Although not as crucial in this discriminant exercise, we can also estimate the probability that a new client spending x_1 log-m.u. on fresh foods and x_2 log-m.u. on groceries would classify into the Retail channel. To do so, simply substitute the estimated regression coefficients into (9.4). For example, to estimate the Retail channel probability at $x_1 = \log(13\,500)$ and $x_2 = \log(11\,000)$, use the sample **R** code

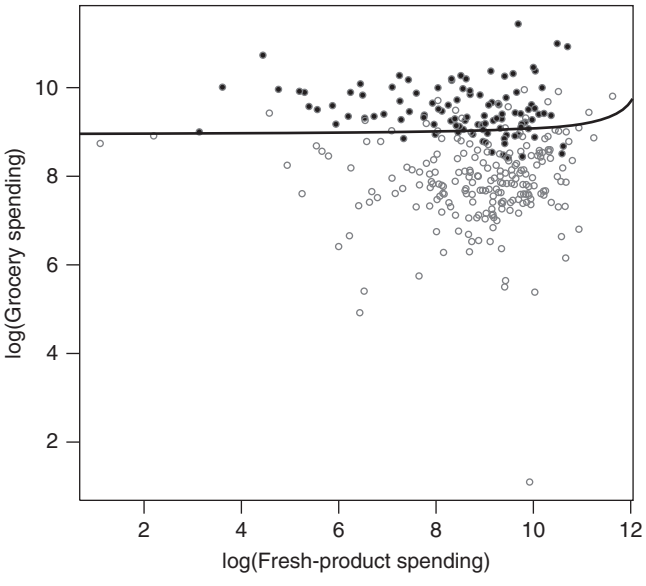

Figure 9.1 Observed $(x_1 = \log\{\text{Fresh-product spending}\}, x_2 = \log\{\text{Grocery spending}\})$ pairs for wholesale food clients in Example 9.1.1 indicating retail clients (solid circles) and service clients (open circles). Solid curve gives discriminant function from logistic regression model containing predictors x_1, x_2, and an x_1x_2 interaction. Source: Data from Cardoso (2013).

```
> new.df <- data.frame( X1=log(13500), X2=log(11000),
                                          X12=log(13500)*log(11000) )
> p.new <- predict( wholesale.glm, type='response', newdata=new.df )
> round( p.new, digits=4 )
```

This gives $P[Y = 1 \mid X_1 = \log(13\,500), X_2 = \log(11\,000)] = 0.6765$ (output not shown), or slightly more than $\frac{2}{3}$. As the estimate exceeds $\frac{1}{2}$, the putative client would be classified into the Retail channel. For a conditional 95% large-sample confidence interval, appeal to (9.2) via

```
> z025 <- qnorm( 0.025, lower=F )
> eta.new <- predict( wholesale.glm, type='link', newdata=new.df,
                                                        se.fit=T)
> p.lower <- 1/( 1 + exp(-(eta.new$fit - z025*eta.new$se.fit)) )
> p.upper <- 1/( 1 + exp(-(eta.new$fit + z025*eta.new$se.fit)) )
> round( cbind(p.lower, p.new, p.upper), digits=4 )
```

to find

```
      p.lower   p.new  p.upper
 [1]   0.5657  0.6765   0.7705
```

that is, $56.57\% < P[Y = 1 \mid X_1 = \log(13\,500), X_2 = \log(11\,000)] < 77.05\%$. □

As a discriminant/classification routine, logistic regression exhibits the typical flexibility associated with regression models. It often works well in practice; however, it can destabilize

if the categories are widely separated. Extensions of the basic logistic model to accommodate multiple categories with $Q > 2$ are also possible; see, for example, Hastie et al. (2009, Section 4.4).

9.1.2 Discriminant rule accuracy

Readers may notice in Figure 9.1 that not all the retail clients (solid circles) lie above the decision boundary and that not all the service clients (open circles) lie below it. This is a realization of *misclassification error*. One of the goals in predictive analytics is minimization of misclassification error to as propitious an extent as possible.

To formalize this, consider the two-category setting ($Q = 2$) and assume a discriminant/classification rule has been applied to a training set of n observations. Let the number of correct classifications into category C_q be ν_{qq} ($q = 1, 2$) and the number of category C_m elements misclassified into category C_q be ν_{qm} ($q \neq m$). These counts can be arranged in a 2×2 cross-classification, similar to the 2×2 structure in Table 8.5. Here, however, the table's counts represent the correct and incorrect classifications in the training data. Table 9.2 displays a schematic version, colorfully known as a *confusion matrix*.

Table 9.2 A 2×2 confusion matrix from classification rule outcomes with $Q = 2$ categories.

		True category		
		C_1	C_2	Row total
Prediction	C_1	ν_{11}	ν_{12}	ν_{1+}
	C_2	ν_{21}	ν_{22}	ν_{2+}
	Column total	ν_{+1}	ν_{+2}	n

A number of useful predictive measures can be derived from the confusion matrix. The correct classification rate, or *accuracy*, is obviously $(\nu_{11} + \nu_{22})/n$, also known in statistical parlance as the *concordance* of the 2×2 table. Conversely, the misclassification (error) rate is $(\nu_{12} + \nu_{21})/n$.

An additional set of measures often seen in biomedical screening and similar applications are the *sensitivity* $\gamma_1 = \nu_{11}/\nu_{+1}$ and *specificity* $\gamma_2 = \nu_{22}/\nu_{+2}$, that is, the correct proportion of C_1 predictions and C_2 predictions, respectively. (These are sometimes called the true positive and true negative rates, respectively, when applied in a formal screening setting. Their complements then have connections to the false positive and false negative rates, respectively, from hypothesis testing in Section 5.4.)

Example 9.1.2 Wholesale distribution channel study (Example 9.1.1, continued). For the wholesale channel study in Example 9.1.1, the confusion matrix can be computed by counting the numbers of clients that fall into each of the four categories in Table 9.2. Doing so produces the result in Table 9.3.

We see that the logistic discriminant's overall accuracy is $(84 + 193)/316 = 87.7\%$, while its overall misclassification error is $(18 + 21)/316 = 12.3\%$. Sensitivity is good: $\gamma_1 = 84/105 = 80\%$, although specificity is higher: $\gamma_2 = 193/211 = 91.5\%$. The rule picks up Service channel

Table 9.3 A 2×2 confusion matrix from logistic discriminant rule outcomes in Examples 9.1.1 and 9.1.2.

		True category		
		Retail channel	Service channel	Row total
Prediction	Retail channel	84	18	102
	Service channel	21	193	214
	Column total	105	211	316

clients slightly better than it predicts Retail channel clients. Indeed, it is not unusual for classification rules to make implicit trade-offs in sensitivity for specificity, or vice versa, depending on the structure of the observed response categories in the population. □

9.1.3 ROC curves

A popular way to visualize the interrelationship between sensitivity and specificity is known as a *receiver operating characteristic* (*ROC*) *curve* (Pepe 2003, Chapter 4). (The curious name stems from the method's origins in radio signal detection during World War II.) The curve is a plot of γ_1 against $1 - \gamma_2$ within the unit square. A 'perfect' ROC curve will rise up the γ_1 sensitivity axis from the origin to $\gamma_1 = 1$, then move horizontally across to $1 - \gamma_2 = 1$. The area under this perfect curve will be 1. In practice, however, sample ROC curves do not fill the unit square. Instead, they take a concave shape over a series of upwards steps connecting the square's lower left and upper right corners. (Some programs offer options to smooth the curve.) A 45° line connecting those reference corners is often included; below this line, the ROC curve is substandard. Satisfactory performance is indicated by an area under the curve (AUC) well above $\frac{1}{2}$ and preferably close to 1.

External **R** packages that can construct ROC curves include *pROC* and *ROCR*. Figure 9.2 gives a sample ROC curve from *pROC*, using the wholesale channel data in Example 9.1.2. The corresponding AUC is 0.708.

The ROC curve and its AUC are sometimes applied as classification rule diagnostics; however, they can suffer from selected irregularities. Concerns over this, and a pertinent alternative, are detailed by Hand (2009a). Indeed, a variety of measures are available to quantify association in a 2×2 confusion matrix (Myatt and Johnson 2009, Section 4.1.5), although the ways in which these actually represent a rule's performance can vary dramatically. Analysts must apply careful attention when employing and interpreting any measure of classification fidelity (Hand 2012). While simplistic, the overall accuracy and misclassification error given above are often useful starting points and will be the focus throughout this chapter.

9.2 Linear discriminant analysis (LDA)

9.2.1 Linear discriminant functions

Linear discriminant functions such as (9.3) can be fashioned in a variety of ways. A classic formulation is due to Fisher (1936). He developed a linear combination of the feature

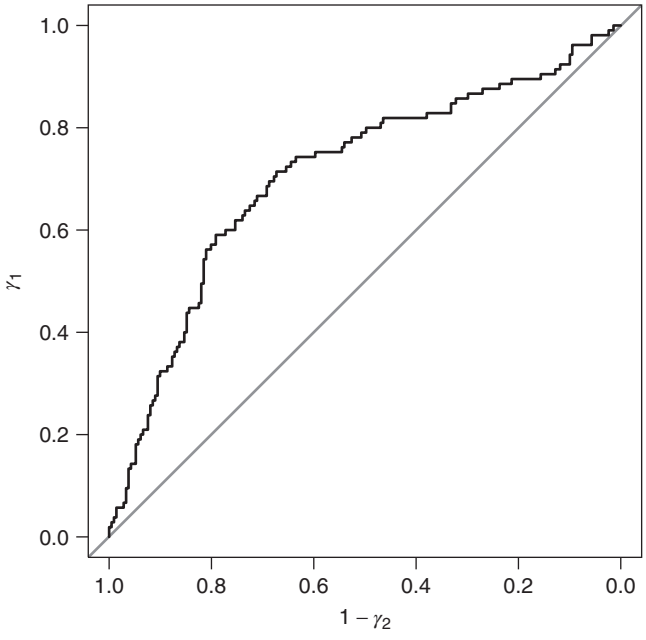

Figure 9.2 Sample ROC curve from logistic classification rule in Examples 9.1.1–9.1.2. Source: Data from Cardoso (2013).

variables, say,

$$\zeta(\mathbf{X}) = \mathbf{a}^{\mathrm{T}}\mathbf{X} = a_1X_1 + a_2X_2 + \cdots + a_pX_p, \tag{9.5}$$

where $\mathbf{a} = [a_1 \ \cdots \ a_p]^{\mathrm{T}}$. The a_js are chosen to maximize the ratio of between-category variance to within-category variance. This is a natural objective: maximize 'signal' relative to 'noise' across the differential categories. Fisher showed that the target quantity becomes a ratio of quadratic forms $\mathbf{a}^{\mathrm{T}}\mathbf{B}\mathbf{a}/\mathbf{a}^{\mathrm{T}}\mathbf{W}\mathbf{a}$, where $\mathbf{B}$ is the between-category covariance matrix and $\mathbf{W}$ is the within-category covariance matrix.

In particular, one can calculate $\mathbf{B}$ and $\mathbf{W}$ from their component vectors and matrices. Let $\mathbf{G}$ be a category indicator matrix with elements $g_{iq} = I_{\{q\}}(Y_i)$ identifying whether the ith subject resides in category C_q. Here $I_S(Y)$ is the indicator function from (2.20). Also, let $\mathbf{M}$ be the matrix of category means with elements

$$\hat{\mu}_{qj} = \frac{1}{N_q}\sum_{i=1}^{n} g_{iq}X_{ij} \ ,$$

where $N_q = \sum_{i=1}^{n} g_{iq}$ is the qth category's sample size ($q = 1, \ldots, Q$). Lastly, let $\overline{\mathbf{X}} = [\overline{X}_{+1} \ \overline{X}_{+2} \ \cdots \overline{X}_{+p}]^{\mathrm{T}}$ be the vector of per-variable means $\overline{X}_{+j} = \sum_{i=1}^{n} X_{ij}/n$. With these, write the full predictor matrix as

$$\boldsymbol{\Xi} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1^{\mathrm{T}} \\ \mathbf{X}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{X}_n^{\mathrm{T}} \end{bmatrix}.$$

Then

$$\mathbf{W} = \frac{(\boldsymbol{\Xi} - \mathbf{GM})^{\mathrm{T}}(\boldsymbol{\Xi} - \mathbf{GM})}{n - Q}$$

and

$$\mathbf{B} = \frac{(\mathbf{GM} - \mathbf{h}\overline{\mathbf{X}}^{\mathrm{T}})^{\mathrm{T}}(\mathbf{GM} - \mathbf{h}\overline{\mathbf{X}}^{\mathrm{T}})}{Q - 1},$$

where $\mathbf{h} = [1\ 1\ \cdots\ 1]^{\mathrm{T}}$ is an $n \times 1$ column vector made up entirely of ones.

One can show, analogous to the sums of squares in an analysis of variance (ANOVA) table (Section 7.5), that the sum $(n - Q)\mathbf{W} + (Q - 1)\mathbf{B}$ adds to a scaled matrix of 'total' variation which is independent of Q. But because this sum is constant with respect to Q, maximizing $\mathbf{a}^{\mathrm{T}}\mathbf{Ba}/\mathbf{a}^{\mathrm{T}}\mathbf{Wa}$ becomes, essentially, a constrained optimization problem. Appeal is then made to an optimization method known as *Lagrange multipliers* (Hughes-Hallett et al. 2013, Section 15.3). The solution employs the eigenvalues and eigenvectors of the sample covariance matrices and is built into various discriminant-rule computer programs; see the following text. (The technical details exceed the scope here; interested readers may refer, for example. to Clarke et al. (2009, Section 5.2.1). For a refresher on eigenanalysis, see Section A.5.)

For the simple two-category case with $Q = 2$, the optimal solution sets $\mathbf{a} = \mathbf{W}^{-1}(\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2)$, where $\hat{\boldsymbol{\mu}}_q$ is the mean vector for the qth group (the qth row of $\mathbf{M}$ taken as a column vector). Fisher's linear discriminant function $\zeta(\mathbf{X}) = \mathbf{a}^{\mathrm{T}}\mathbf{X}$ is then designed to classify an observation with discriminant score $\zeta(\mathbf{X})$ to category $q = 1$ if $\zeta(\mathbf{X}) > \frac{1}{2}(\overline{\zeta}_1 + \overline{\zeta}_2)$, where $\overline{\zeta}_q$ is the mean discriminant score for category q and the categories have been organized such that $\overline{\zeta}_1 > \overline{\zeta}_2$, assuming equal prior probability of allocation to both groups (Everitt 2005, Section 7.2.2).

Example 9.2.1 **Bank depositor marketing study.** Moro et al. (2011) described a southern European bank's marketing campaign for subscribing customers to a new term deposit. Their database includes information on customer characteristics (education, age, loan status, etc.) as related to likelihood of subscribing to the new deposit vehicle. For purposes of illustration, we work here with a subset of the data, focusing on the cohort of married bank customers who had completed secondary school and had not been contacted by the bank regarding any previous deposit campaigns. This gives a data set containing $n = 1172$ customer records.

Of interest is characterization of the eventual category decisions, $Y = q$, made by these customers, regarding whether ($q = 1$) or not ($q = 2$) each subscribed to the new deposit vehicle. Two predictor variables were found to be important in predicting subscriber success for this cohort, $X_1 = \{\text{Age of depositor}\}$ and $X_2 = \log\{\text{Duration of marketing phone call (in s)}\}$. A selection of the data appear in Table 9.4. (The complete set is available at http://www.wiley.com/go/piegorsch/data_analytics.)

Table 9.4 Selected data from a larger set of $n = 1172$ observations on bank customer subscription classifications (subscribed: $Y = 1$; unsubscribed: $Y = 2$) after a marketing campaign.

Y = Subscription class:	2	2	2	$\cdots$	2	1
X_1 = Age (years):	59	39	39	$\cdots$	33	56
$X_2 = \log\{\text{Duration (seconds)}\}$:	5.4205	5.0173	5.6095	$\cdots$	5.7236	7.1412

Source: Moro et al. (2011).

For purposes of discriminating between subscribers and nonsubscribers, consider here application of Fisher's linear discriminant analysis (LDA). To find the linear discriminant function in (9.5), we can appeal to **R**'s `lda()` function from the external *MASS* package. Assume equal prior weights for both categories. Sample code is

```
> Bank.lda <- lda( Y ~ X1+X2, prior=c(0.5,0.5) )
> print( Bank.lda )
```

Notice the specification of equal prior weights via the `prior=` option. This produces the summarizing output (edited)

```
Call:
lda(Y ~ X1 + X2, prior = c(0.5, 0.5))

Prior probabilities of groups:
  1   2
0.5 0.5

Group means:
       X1       X2
1 44.4800 6.223444
2 42.1103 5.113879

Coefficients of linear discriminants:
          LD1
X1 -0.02172443
X2 -1.12114596
```

The listed `Group means` are the per-variable category means $\hat{\mu}_{qj}$ ($q = 1, 2$; $j = 1, 2$). The linear discriminant function is given by $\zeta(X_1, X_2) = a_1 X_1 + a_2 X_2$, where the a_js are found under '`Coefficients of linear discriminants`.' They are also available for direct computation as `Bank.lda$scaling`. Here, $\zeta(X_1, X_2) = -0.0217 X_1 - 1.1211 X_2$.

To construct the classification rule, find the category indicator matrix **G** via, for example,

```
> G <- as.matrix( cbind(2-Y, Y-1) )
```

and from this, calculate the discriminant means $\overline{\zeta}_q$:

```
> a <- Bank.lda$scaling
> zeta <- a[1]*X1 + a[2]*X2
> zeta1bar <- sum( zeta*G[,1] )/sum( G[,1] )
> zeta2bar <- sum( zeta*G[,2] )/sum( G[,2] )
```

This yields $\overline{\zeta}_1 = -7.9437$ and $\overline{\zeta}_2 = -6.6482$ (output not shown). As $\overline{\zeta}_1 < \overline{\zeta}_2$, the linear discriminant rule predicts a depositor will subscribe (into category $q = 1$) when

$$\zeta(X_1, X_2) = -0.0217 X_1 - 1.1211 X_2 < \frac{1}{2}(\overline{\zeta}_1 + \overline{\zeta}_2) = \frac{-6.6482 - 7.9437}{2} = -7.2960.$$

A plot of the data coded by observed subscription outcomes (solid circles indicate new subscribers, open circles nonsubscribers) helps to visualize the complexity of the intertwined categories here (see Figure 9.3). In the figure, the linear discriminant decision boundary, corresponding to the line $-0.0217 X_1 - 1.1211 X_2 = \frac{1}{2}(\overline{\zeta}_1 + \overline{\zeta}_2)$, is superimposed. Depositors lying

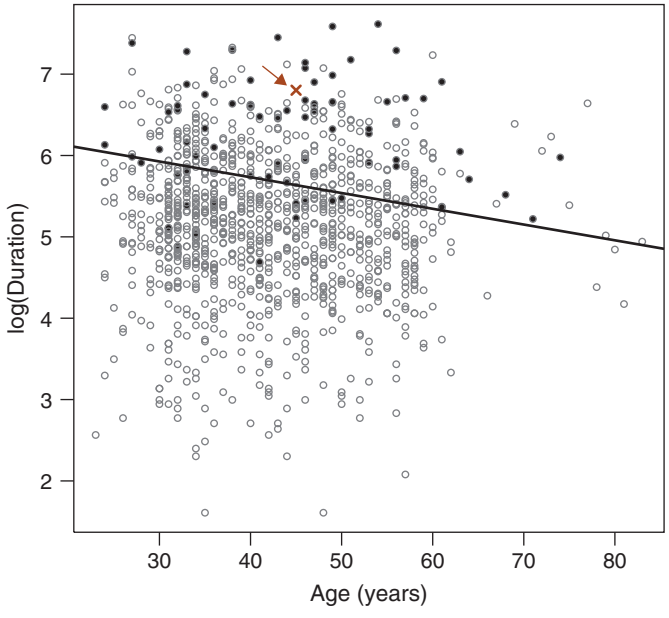

Figure 9.3 Observed (X_1 = Age, X_2 = log {Duration}) pairs for bank depositors in Example 9.2.1 indicating subscribers (solid circles) and nonsubscribers (open circles) to new deposit vehicle. Solid line gives linear discriminant decision boundary. New depositor with $X_1 = 45$ and $X_2 = \log(900) = 6.8024$ marked by an × (see arrow). Source: Data from Moro et al. (2011).

above the boundary are predicted into category C_1 (subscribers); those below are predicted into category C_2 (nonsubscribers).

The pattern in Figure 9.3 provides useful knowledge discovery. Apparently, longer-duration marketing calls are required to predict successful new subscriptions with younger depositors: calls of duration shorter than about 7 min (i.e., $\log\{x_2\} < \log\{420\} = 6.04$) produce few successes. Subscriptions appear slightly more likely with older depositors over shorter durations, although with very senior customers, the discriminant function may have limited predictive value due to the small number of observations in that upper age range.

In any case, a substantial portion of the feature space in Figure 9.3 is below the discriminant line and, therefore, allocated to nonsubscribers. Indeed, most depositors did not respond successfully to the marketing campaign, illustrating the limited success rate of the marketing effort.

The confusion matrix for this linear discriminant rule is given in Table 9.5. We see that the discriminant's overall accuracy is $(59 + 811)/1172 = 74.2\%$, while its misclassification error is $(286 + 16)/1172 = 25.8\%$. Sensitivity is moderate at $\gamma_1 = 59/75 = 78.7\%$, as is specificity at $\gamma_2 = 811/1097 = 73.9\%$.

To predict the category of, say, a new 45-year old depositor contacted for 15 min (900 s), calculate the linear discriminant as $\zeta(X_1, X_2) = -(0.0217)(45) - (1.121)(\log\{900\}) = -8.604$. As this is below -7.296, predict that this customer will indeed subscribe to the new deposit product.

Table 9.5 The 2×2 confusion matrix from linear discriminant rule outcomes in Example 9.2.1.

		True category		
		Subscriber	Nonsubscriber	Row total
Prediction	Subscriber	59	286	345
	Nonsubscriber	16	811	827
	Column total	75	1097	1172

In **R**, the `lda()` function can perform these various calculations internally and report the category prediction via the `predict()` function. For instance, the sample command

```
> predict( Bank.lda, newdata=data.frame(X1=45, X2=log(900)) )$class
```

gives the class prediction as

```
[1] 1
Levels: 1 2
```

That is, the predicted category membership for a depositor with $X_1 = 45$ and $X_2 = \log(900)$ has label `1`, corresponding to subscription category C_1, as expected. In Figure 9.3, the point (marked by an $\times$) lies clearly above the decision boundary. □

In high-dimensional problems with large p, it may be of interest to conduct the supervised classification on a reduced subset of predictors. Similar to the sparse logistic regression approach mentioned in Section 8.3.1, a type of *sparse discriminant analysis* may be applied to address this issue (Clemmensen et al. 2011). The method embeds a regularizing L_1 penalty into a set of discriminant scoring criteria, harkening back to the Lasso approach of Section 7.4.3. As a result, the discrimination effort is combined with a form of variable selection/feature selection; this can greatly improve the discriminant outcomes in high-dimensional classification.

For more on Fisher's linear discriminant function, see Everitt (2005, Section 7.2), Izenman (2008, Chapter 8), or the larger treatment in Huberty and Olejnik (2006).

9.2.2 Bayes discriminant/classification rules

If the analyst is able or willing to assign probability densities to the input variables in $\mathbf{X}$, discriminant functions can be built from the probability models that ensue. For simplicity, it was assumed above that any prior interest in the categories was constant. Now, suppose that the qth category is assigned a formal *prior probability*, π_q, containing an observed predictor vector $\mathbf{X} = [X_1 \; X_2 \; \cdots \; X_p]^{\mathrm{T}}$, where $\sum_{q=1}^{Q} \pi_q = 1$. Suppose further that a probability density function (p.d.f.) $f(\mathbf{x} \mid Y = q)$ is modeled to describe the probability that $\mathbf{X}$ classifies in category C_q for each $q = 1, \ldots, Q$. Then from Bayes' rule in (2.4), the *posterior probability* of observing an element in category C_q, given the vector $\mathbf{X} = \mathbf{x}$, is

$$P[Y = q \mid \mathbf{X} = \mathbf{x}] = \frac{f(\mathbf{x} \mid Y = q)\, \pi_q}{f(\mathbf{x})}, \tag{9.6}$$

where $f(\mathbf{x}) = \sum_{q=1}^{Q} f(\mathbf{x} \mid Y = q)\ \pi_q$ is the marginal p.d.f. of $\mathbf{X}$. Clearly, $f(\mathbf{x})$ is independent of q and so (9.6) is often expressed simply as the proportional relationship $P[Y = q \mid \mathbf{X} = \mathbf{x}] \propto f(\mathbf{x} \mid Y = q)\ \pi_q$.

With this, the *Bayes classification rule* predicts a future or test-case observation $\mathbf{X}$ will lie in that category $C_{q'}$ whose $P[Y = q' \mid \mathbf{X} = \mathbf{x}]$ is largest among all Q posterior probabilities. In this form, Bayes discriminant rules can be shown to minimize the expected number of misclassifications they induce (Clarke et al. 2009, Section 5.2.2) and, hence, serve as a form of optimal classifier in supervised learning.

In effect, the posterior probabilities from (9.6) partition the feature space into Q disjoint regions, each with highest classification probability in that region. For example, when comparing two categories C_q and C_m, $\mathbf{X}$ is classified into category C_q when $P[Y = q \mid \mathbf{X} = \mathbf{x}] > P[Y = m \mid \mathbf{X} = \mathbf{x}]$. That is, when

$$\log \frac{f(\mathbf{x} \mid Y = q)}{f(\mathbf{x} \mid Y = m)} + \log \frac{\pi_q}{\pi_m} \ > \ 0. \tag{9.7}$$

Of course, this assumes that the cost of misclassification is essentially equal between the two categories. If this is not the case – say, the domain expert indicates that misclassifying into C_q is twice as costly as misclassifying into C_m – one can incorporate a cost parameter and modify the decision rule accordingly; cf. James et al. (2013, Section 4.4.3).

For many specifications of $f(\mathbf{x} \mid Y = q)$, implementation of the Bayes rule can be problematic in practice, especially with high-dimensional/multicategory classifications. One case where it simplifies into a convenient form, however, is under a normal (Gaussian) assumption. This is discussed in the next section.

9.2.3 Bayesian classification with normal data

Suppose the analyst is willing to adopt a normal (Gaussian) assumption on the input variables. For instance, with the multivariate normal p.d.f. from (2.40), we have

$$f(\mathbf{x} \mid Y = q) = \frac{1}{(2\pi)^{n/2}|\mathbf{V}|^{1/2}}\ \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_q)^{\mathrm{T}}\mathbf{V}^{-1}(\mathbf{x} - \boldsymbol{\mu}_q)\right\},$$

where $\boldsymbol{\mu}_q = E[\mathbf{X} \mid Y = q]$ and $\mathbf{V} = \mathrm{Var}[\mathbf{X}]$ is the covariance matrix of $\mathbf{X}$. Notice here that $\mathbf{V}$ is assumed constant across all categories and is, therefore, independent of q. Under this multivariate normal assumption, the first term in (9.7) simplifies to

$$\log \frac{f(\mathbf{x} \mid Y = q)}{f(\mathbf{x} \mid Y = m)} = (\boldsymbol{\mu}_q - \boldsymbol{\mu}_m)^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{x} - \frac{\boldsymbol{\mu}_q^{\mathrm{T}}\mathbf{V}^{-1}\boldsymbol{\mu}_q - \boldsymbol{\mu}_m^{\mathrm{T}}\mathbf{V}^{-1}\boldsymbol{\mu}_m}{2}\ . \tag{9.8}$$

This has the form $\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{x} - \theta$, where the constant θ is independent of $\mathbf{x}$. Thus this Gaussian discriminant rule is, once again, a linear (in $\mathbf{x}$) decision boundary for discriminating between C_q and C_m.

In practice, the unknown quantities in (9.8) are estimated from the training data: use $\hat{\boldsymbol{\mu}}_q$ for $\boldsymbol{\mu}_q$, the within-group covariance matrix $\mathbf{W}$ for $\mathbf{V}$, and unless some known prior specification is available, $\hat{\pi}_q = N_q/n$ for π_q. Then, the estimator of the coefficient vector for $\mathbf{x}$ in the Gaussian classifier will have the same form as Fisher's LDA counterpart from Section 9.2.1: $\mathbf{a} = \mathbf{W}^{-1}(\hat{\boldsymbol{\mu}}_q - \hat{\boldsymbol{\mu}}_m)$.

This convergence is worth emphasizing: in developing Fisher's LDA, we made no parametric assumptions on the distribution of $\mathbf{X}$. By appealing formally to the normal model for $f(\mathbf{x} \mid Y = q)$ and applying the theoretical constructs from Bayes' rule, however, we recover essentially the *same* linear discriminator.

Applied to all Q categories, the Gaussian classifier allocates an observed $\mathbf{X}$ to category C_q when

$$\delta_q(\mathbf{X}) = \log f(\mathbf{X} \mid Y = q) + \log(\pi_q)$$

is a maximum across all $q = 1, \ldots, Q$. In the special case with $\mathbf{X} \mid \{Y = q\} \sim \mathrm{N}_p(\boldsymbol{\mu}_q, \mathbf{V})$, this is

$$\delta_q(\mathbf{X}) = \theta' + \boldsymbol{\mu}_q^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{X} - \frac{1}{2}\boldsymbol{\mu}_q^{\mathrm{T}}\mathbf{V}^{-1}\boldsymbol{\mu}_q + \log(\pi_q), \tag{9.9}$$

where the initial constant θ' is comprised of terms independent of q and may be ignored when maximizing across q. Replacing unknown quantities with their sample estimators produces the *Gaussian classification function*

$$d_q(\mathbf{X}) = \hat{\boldsymbol{\mu}}_q^{\mathrm{T}}\mathbf{W}^{-1}\mathbf{X} - \frac{1}{2}\hat{\boldsymbol{\mu}}_q^{\mathrm{T}}\mathbf{W}^{-1}\hat{\boldsymbol{\mu}}_q + \log\left(\frac{N_q}{n}\right), \tag{9.10}$$

a hyperplane in the feature space. (A *hyperplane* is a multidimensional extension of a flat plane from three-dimensional space. It represents a flat $(p-1)$-dimensional subspace of the p-dimensional space in which it resides, separating the larger space into two half-spaces. Allowing for a nonzero intercept, a hyperplane takes the general formula $a_0 + a_1X_1 + \cdots + a_pX_p = 0$. A special case is a line in (X_1, X_2)-space: $a_0 + a_1X_1 + a_2X_2 = 0$.)

Under this model, allocate $\mathbf{X}$ to category C_q when $d_q(\mathbf{X})$ is a maximum for all q. The decision borders between any two categories C_q and C_m are the intersecting points for which $d_q(\mathbf{X}) = d_m(\mathbf{X})$. As the classification functions $d_q(\mathbf{X})$ produce hyperplanes in $\mathbf{X}$-space, these decision boundaries will themselves be linear.

Useful graphics for visualizing the category predictions and decision boundaries are available in the `partimat()` and `drawparti()` functions from the external *klaR* package.

Example 9.2.2 Urban terrorism vulnerability. A study was conducted to examine vulnerability to terrorist attacks in $n = 132$ of the largest cities in the United States. Data were collected on whether each city had experienced a terrorism event over the 35-year period 1970–2004, and if so, whether any human casualties (injuries or deaths) were recorded (Piegorsch et al. 2007). The classification scheme was defined as follows:

Category	Category outcome	Event status
C_1	$Y = 1$	No events
C_2	$Y = 2$	One or more noncasualty events
C_3	$Y = 3$	One or more events with casualties

This produced $Q = 3$ categories in which C_1 contained cities where no events were observed, C_2 contained cities where an event occurred with no casualties, and C_3 contained cities where an event occurred with casualties.

A pair of predictor variables was used to index (i) the frequency and diversity of natural hazards – tornados, floods, etc. – the cities had experienced and (ii) the nature and diversity of each city's infrastructure. The former was used as a surrogate for community experience in responding to extreme events, while the latter quantified vulnerable characteristics of the built urban environment such as transportation infrastructure and age of housing. Higher index values indicated greater vulnerability to adverse outcomes. Taken on a logarithmic scale, the index data, along with the terrorism outcome classifications Y, appear in Table 9.6. (The X_1 log-hazard scores were seen previously in Exercise 3.3. As previously, only a selection of the data is given in the table; the complete set is available at http://www.wiley.com/go/piegorsch/data_analytics.)

Table 9.6 Selected data on terrorism events from $n = 132$ US cities (no event: $Y = 1$; noncasualty event(s): $Y = 2$; event(s) with casualties: $Y = 3$) and vulnerability log-indices.

Urban area	Y	$X_1 = \log\{\text{Natural hazard}\}$	$X_2 = \log\{\text{Infrastructure}\}$
Albany, NY	3	1.854	1.713
Albuquerque, NM	2	1.078	0.739
Allentown/Bethlehem, PA	1	1.577	1.773
⋮	⋮	⋮	⋮
Worcester.MA	1	1.221	1.540
Youngstown, OH	1	1.641	1.871

Source: Piegorsch et al. (2007).

Assuming a bivariate normal distribution with equal category covariance matrices for the (X_1, X_2) pairs, Gaussian classification functions for these data can be constructed as per (9.10). In **R**, appeal again to the `lda()` function from the *MASS* package. For example, the sample commands

```
> cities132.lda <- lda( Y ~ X1+X2 )
> print( cities132.lda$prior )
```

list the calculated prior probabilities, $\hat{\pi}_q = N_q/n$, as

```
        1         2         3
0.4924242 0.2348485 0.2727273
```

We see that almost half (49.2%) of the cities in this study escaped any terrorism events, while the remainder split about evenly between casualty and noncasualty events.

For visualization, the **R** command `plot(X2 ~ X1, pch=as.character(Y))` produces a scatterplot of the (X_1, X_2) pairs with their category labels. More useful, however, may be the `partimat()` function from the external *klaR* package. For instance, the sample commands

```
> require( klaR )
> partimat( factor(Y) ~ X2+X1, imageplot=F, col.wrong='gray' )
```

plot the (X_1, X_2) pairs with their category labels, overlay the linear decision boundaries (the intersections of the classification functions), mark in gray those points misclassified in each partitioned category, and if desired, plot the category means. Other options are also available;

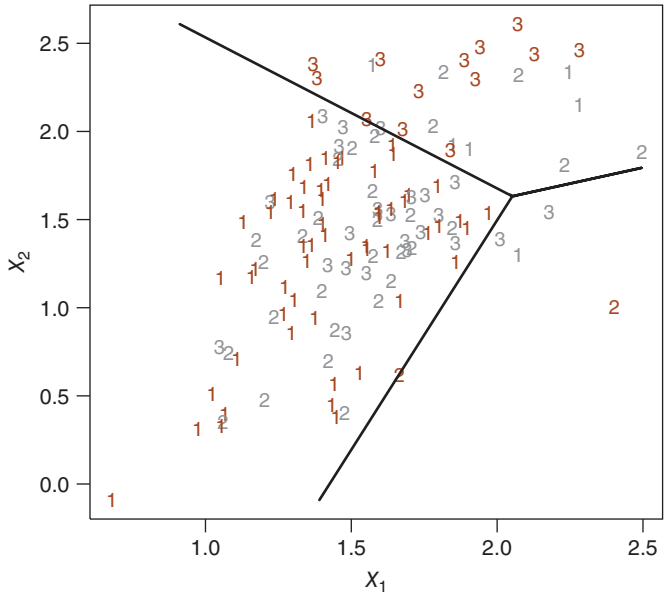

Figure 9.4 Scatterplot of $X_2 = \log\{\text{Infrastructure index}\}$ versus $X_1 = \log\{\text{Natural hazard index}\}$ for 132 US cities in Example 9.2.2. The plot labels indicate category membership (1 for no terrorism event, 2 for noncasualty event, 3 for event with casualties). Grayed labels indicate misclassifications. Solid lines give linear discriminant decision boundaries and delineate the category structure: C_1 to lower left, C_2 to lower right; C_3 to top. Source: Data from Piegorsch et al. (2007).

see `help(partimat)`. Figure 9.4 gives a version (stylized for presentation) of the result. In the figure, the no-event category C_1 locates down and to the left, corresponding to low values of both X_1 and X_2. Viewing C_1 as the least vulnerable category suggests, as expected, that low values on both log-indices imply lower urban vulnerability to detrimental outcomes from terrorism events.

By contrast, the high-vulnerability category C_3 locates at the top and somewhat to the right in the graphic. This suggests that high values of both log-indices predict this extreme classification and also that large log-infrastructure values X_2 drive the effect somewhat more than large log-hazard values X_1. The mid-vulnerability, no-casualty event category, C_2, lies primarily to the lower right in the figure and is sparsely populated: the classification schema, at least under the normal model for these data, does not emphasize the C_2 outcomes to as great an extent.

The **R** command

```
> Yhat <- predict(cities132.lda)$class
```

generates (and stores in `Yhat`) the predicted classifications for the 132 cities in the database, from which some intriguing knowledge discovery is possible. For instance, the location names for the 132 metropolitan areas reside in the database variable `urbName`. Calling `urbName[Yhat==3]` displays 23 metropolitan areas assigned to category C_3; these are predicted as high-vulnerability locations. Table 9.7 displays the predicted C_3 category

Table 9.7 US metropolitan areas in Example 9.2.2 assigned to high-vulnerability category C_3 by Gaussian classification rule, ordered by posterior probability $P[Y = 3 \mid X_1, X_2]$.

Urban area	$P[Y = 3\|X_1, X_2]$	Urban area	$P[Y = 3\|X_1, X_2]$
New York, NY/Newark, NJ	0.699		
New Orleans, LA	0.667	St. Louis, MO	0.523
Washington, DC	0.646	Trenton, NJ	0.511
Philadelphia, PA	0.640	Mission Viejo, CA	0.487
Norfolk/Chesapeake, VA*	0.619	Milwaukee, WI†	0.458
Chicago, IL	0.608	Detroit/Warren, MI	0.438
Richmond, VA†	0.598	Baton Rouge, LA†	0.436
Cleveland/Akron, OH	0.575	Tampa/St. Petersburg, FL	0.435
Baltimore/Annapolis, MD†	0.574	Columbia, SC*	0.424
Boston, MA	0.566	Charlotte, NC*	0.422
Toledo, OH*	0.549	San Francisco/Oakland, CA	0.413
Charleston, SC*	0.546	Houston, TX†	0.406

All areas observed as $Y = 3$ unless otherwise indicated.
* Observed with $Y = 1$.
† Observed with $Y = 2$.

members by order of their posterior probabilities $P[Y = 3 \mid X_1, X_2]$ from (9.6) found in **R** using `predict(cities132.lda)$posterior`.

Most locations in Table 9.7 experienced previous C_3 events, although a number reported $Y = 2$ (noncasualty events) or even $Y = 1$ (no events), as identified in the table. Other interesting patterns exist as well (see Exercise 9.8). Despite past outcome(s), however, a predicted, high-vulnerability classification under this model could (or should) encourage a C_3 community to study more closely its vulnerability status. For example, emergency managers in coastal cities such as Norfolk, VA, or Charleston, SC, might consider new or updated forms of shoreline antiterrorist protection, because their C_3 classification indicates high predicted vulnerability to terrorist events at the community level. Similar sorts of targeted, place-based efforts would be pertinent for the other locations in Table 9.7 as well. □

Gaussian classification as introduced here can be effective when the data satisfy the normal parent supposition, at least to a good approximation. It also extends easily to multiclass settings where $Q > 2$, as seen in Example 9.2.2. The method is often criticized, however, for its relatively inflexible linear decision boundaries, a consequence of the stringent model assumptions. It can also be detrimentally affected by large outliers. When there are only $Q = 2$ categories under study and for nonnormal X-variables with sufficiently large sample sizes, the logistic approach in Section 9.1 often proves more adaptable (Press and Wilson 1978).

When normality is valid, some flexibility can be achieved by relaxing the supposition of equal covariance matrices in the multivariate normal p.d.f.s. The resulting classification functions will involve quantities of the form $\frac{1}{2}(\mathbf{X} - \hat{\boldsymbol{\mu}}_q)^{\mathrm{T}}\mathbf{W}_q^{-1}(\mathbf{X} - \hat{\boldsymbol{\mu}}_q)$. These now contain quadratic terms in $\mathbf{X}$ that vary with q, creating curvilinear decision boundaries. The result is a type of *quadratic discriminant analysis* (QDA), which can add a level of pliancy to the classification process (James et al. 2013, Section 4.4.4). QDA comes with a concomitant increase in complexity, however, as each category's covariance matrix must now be estimated. The `qda()` function in the *MASS* package proves useful in this case.

9.2.4 Naïve Bayes classifiers

When modeling the posterior probability via (9.6), the analyst might assume that the individual X_j variables are independent. If so, the joint p.d.f. of $\mathbf{X}$ simplifies using the Multiplication Rule as in (2.11). This gives

$$P[Y = q \mid \mathbf{X} = \mathbf{x}] = \frac{\pi_q \prod_{j=1}^{p} f_j(x_j \mid Y = q)}{f(\mathbf{x})},$$

where the $f_j(x_j \mid Y = q)$ terms are the individual, independent p.d.f.s and $f(\mathbf{x})$ is again the marginal p.d.f. of $\mathbf{X}$.

The independence assumption further simplifies the model: estimation of unknown parameters in the posterior probabilities now involves a series of univariate operations, rather than a more complicated, multivariate calculation. And, there is no need to estimate any of the covariances because these are all assumed to be zero. The method can also be easier to apply when some of the X_j variables are discrete.

Whether or not the independence assumption is valid in practice is usually open to question, however. Perhaps as a result, this approach is known as *Naïve Bayes classification* or sometimes more descriptively *Idiot's Bayes classification*. Despite the obvious caution such nomenclature implies, naïve Bayes classifiers can prove useful. Their simplicity makes them easier to implement and interpret, particularly if the dimensionality of the problem is very large. Indeed, selected studies have shown that the 'naïve' independence model can produce competitive classification procedures in certain cases (Hand and Yu 2001); a popular application is with spam filtering for Internet communications (Seewald 2007). For an instructive introduction, see Hand (2009b).

9.3 *k*-Nearest neighbor classifiers

The methods in the previous sections often rely on parametric model assumptions to undertake the discrimination/classification exercise. When such reliance is felt to be a concern, nonparametric techniques can be applied instead. Perhaps the simplest of these involves the *k-nearest neighbor* (*k*-NN) approach, a concept reaching back to the work of Fix and Hodges (1951). Suppose a set of training data contains n feature vectors, $\mathbf{X}_i$, each unambiguously identified with one of the Q categories under study. Given a new or test feature vector $\mathbf{X}_o$, the k-NN method finds those $k > 1$ observations in the training set that lie nearest to $\mathbf{X}_o$ in the p-dimensional feature space. It then classifies $\mathbf{X}_o$ into that category represented most often among the k neighbors (a *majority voting scheme*).

To avoid ties, k is usually assumed odd; if any ties still occur, these are broken at random. Increasing k can also help to avoid ties; a possible suggestion along this vein is to set k as the smallest odd integer greater than Q (why?). This comes with subsequent compromises, however. For instance, large values of k can increase bias in the classification rule and also heighten the computing burden. On the other hand, the increased bias often associates with decreased variance; a 'bias-variance trade-off' that may be to the analyst's advantage (Hastie et al. 2009, Section 7.3). In a certain sense, k acts as a tuning parameter that modulates the amount of this trade-off.

The method's simplicity belies its computational needs; very large training sets can push the calculations to their proverbial limits. So too can high-dimensional feature spaces, that is, including many X-variables. The issue is similar to the 'curse of dimensionality' mentioned in Section 7.4.1: in high dimensions, distances between feature vectors can become sparse, making detection of classifying relationships much more difficult.

Along with selection of k, the analyst must also specify a metric to define 'distance' in the feature space. Most common is Euclidean distance, that is, the usual 'as-the-crow-flies' measure. Table 9.8 lists this along with a variety of other popular choices. Notice that the Minkowski form is actually a general class of metrics, characterized by its positive parameter γ. Special cases include Euclidian ($\gamma = 2$), Manhattan ($\gamma = 1$, sometimes called 'Hamming' distance), and Maximum ($\gamma \to \infty$) distances. For Canberra distance, terms with zero numerator and denominator are omitted from the sum; Canberra distance is often intended for use with nonnegative counts.

Table 9.8 Selected metrics for defining 'distance' between two vectors, $\mathbf{X}_o = [X_{o1} \quad X_{o2} \quad \cdots \quad X_{op}]^T$ and $\mathbf{X}_i = [X_{i1} \quad X_{i2} \quad \cdots \quad X_{ip}]^T$, in a p-dimensional feature space.

Name	Distance
Euclidean	$\sqrt{\sum_{j=1}^{p} (X_{oj} - X_{ij})^2}$
Manhattan ('city block')	$\sum_{j=1}^{p} \lvert X_{oj} - X_{ij} \rvert$
Maximum/Tchebychev	$\max_{j=1,\ldots,p} \{\lvert X_{oj} - X_{ij} \rvert\}$
Minkowski	$\left\{ \sum_{j=1}^{p} \lvert X_{oj} - X_{ij} \rvert^{\gamma} \right\}^{1/\gamma} \quad (\gamma > 0)$
Canberra	$\sum_{j=1}^{p} \lvert X_{oj} - X_{ij} \rvert / \lvert X_{oj} + X_{ij} \rvert$

If the feature variables differ greatly in their scales of measurement and/or variances, they are usually, if somewhat arbitrarily, divided by their standard deviations before any distance calculations. This may help assuage problems with wildly differential scales or arbitrary units of measurement.

Given k and the chosen distance metric, the calculations for k-NN classification are fairly straightforward, if tedious. In **R**, the `knn()` function from the external *class* package is a stalwart choice when using Euclidean distance. Also available are the `knnVCN()` function from the external *knnGarden* package and the `gknn()` function from the external *scrime* package, either of which can institute any of the distance metrics in Table 9.8.

Example 9.3.1 Remote sensing of tree disease. In a study to classify arboreal disease progression, Johnson et al. (2013) reported data on diseased oak trees in a Japanese forest. Recorded were the status of the trees in a high-resolution satellite image segment ($Y = 1$ for diseased trees, $Y = 2$ for other land cover), along with $p = 5$ predictor variables representing features of each image. A selection of the data, comprising $n = 4339$ training observations and $n_o = 500$ test observations, appears in Table 9.9 (download both data sets at http://www.wiley.com/go/piegorsch/data_analytics).

Table 9.9 Selected data from a training set of $n = 4339$ satellite observations and a test set of $n_o = 500$ satellite observations on oak tree disease (diseased trees: $Y = 1$, other land cover: $Y = 2$) in a Japanese forest.

Training set				
Y = Disease status	1	1	···	2
X_1 = Mean gray-level (pansharp. band)	120.362	124.740	···	125.172
X_2 = Mean green band	205.500	202.800	···	559.048
X_3 = Mean red band	119.395	115.333	···	365.968
X_4 = Mean near-infrared band	416.581	354.333	···	439.272
X_5 = Std. deviation (pansharp. band)	20.676	16.707	···	15.392
Test set				
Y_0 = Disease status	1	2	···	2
X_{01} = Mean gray-level (pansharp. band)	121.383	109.829	···	119.732
X_{02} = Mean green band	218.357	183.700	···	182.238
X_{03} = Mean red band	112.018	82.950	···	74.286
X_{04} = Mean near infrared band	426.607	251.750	···	301.690
X_{05} = Std. deviation (pansharp. band)	19.083	16.079	···	22.944

Source: Johnson et al. (2013).

To construct a classifier for predicting disease status from a satellite image, consider the use of the k-NN method. We train the learning algorithm on the larger set of $n = 4339$ observations, then apply it to the test set of $n_o = 500$ observations, and examine the various accuracy and misclassification rates. The training variables are coded as $X_1, \ldots, X_5, Y$ and the test variables as $X_{01}, \ldots, X_{05}, Y_0$. (The feature variables will be scaled by their standard deviations as part of the analysis.)

Set $k = 7$ and employ Euclidean distance as the separation metric. With this, sample **R** code for the k-NN analysis employs functions from the external *class* package:

```
> trainset <- as.matrix( cbind(X1,X2,X3,X4,X5) )
> testset <- as.matrix( cbind(X01,X02,X03,X04,X05) )
> require( class )
> Yhat.knn <- knn( train=scale(trainset, center=F),
                   test=scale(trainset, center=F), cl=Y, k=7 )
> classify.knn <- knn( train=scale(trainset, center=F),
                       test=scale(testset, center=F), cl=Y, k=7 )
```

In the code, the `knn()` function is the workhorse: its `train=` and `test=` subcommands define the feature variables from the (scaled) training and test sets, respectively. The `cl=` subcommand indicates the category labels in the training set that correspond to the feature variables defined by `train=`. All these terms are required. Lastly, the `k=` option sets the number of elements in the nearest-neighbor window. (The default is `k=1`.) Notice that by listing the training set in both the `train=` and `test=` subcommands, the function will give predicted values for the training responses; this is assigned to the `Yhat.knn` object. The consequent confusion matrix for the training data can be determined simply as

```
> table( Yhat.knn, Y )
```

Using similar code, the confusion matrix for the test data is available via

```
> table( classify.knn, Y0 )
```

The results appear in Table 9.10, where the confusion matrices provide some intriguing insights. First, the overall accuracy of the seven-neighbor rule in the training set is (34 + 4265)/4339 = 99.1%; the complementary misclassification rate is 40/4339, or less than 1%. While striking, these values only provide validation that the rule has accurately learned the extant patterns in the training data. The challenge comes with the test data: accuracy there is a more modest (6 +313)/500 = 63.8%, while misclassification error grows to 181/500 = 36.2%. It is not uncommon to see drops in accuracy and consequent rises in misclassification rates as the 'test' of the rule is conducted. Here the shrinkage is substantial, but it nonetheless affords some useful information: notice that for both data sets the specificity is perfect at 100%. The seven-neighbor rule apparently has strong ability to accurately identify nondiseased remote images. It might be anticipated, however, that accurate identification of diseased images is the more important goal here; the sensitivity of 34/74 = 45.9% in the training data confusion matrix is far smaller. (Test data sensitivity is even worse, barely over 3%.)

Table 9.10 A 2 × 2 confusion matrices from *k*-nearest neighbor classifications in Example 9.3.1.

		Training set			Test set		
		Observed			Observed		
		Diseased	Other	Row total	Diseased	Other	Row total
Predicted	Diseased	34	0	34	6	0	6
	Other	40	4265	4305	181	313	494
	Column total	74	4265	4339	187	313	500

Notice from Table 9.10 that the bulk of the training data, 4265/4339 = 98.3%, are nondiseased images. The rule appears to do well on nondiseased images with this substantial amount of information but has clear difficulty learning from the far fewer diseased images. Whether changes to the *k*-neighbor window (Exercise 9.12) or other classification strategies can improve the error rates here is a question explored further in the following. □

If the choice of the tuning parameter *k* is in doubt, many analysts select it adaptively from the data using leave-one-out cross-validation. As mentioned in Section 7.4.2, cross-validation removes an observation from the data set and then uses the remaining data to estimate the value of that excised observation under the proffered model. The `knn.cv()` function in the external *class* package can perform leave-one-out cross-validation on a set of training data.

A *k*-NN classifier can only classify a test vector $\mathbf{X}_0$ from among the categories presented to it by the *k*-NN, thus it does not actively search out other possible classification options. This is often referred to as 'lazy learning.' Also, as with many simple nonparametric techniques, simplicity trumps capability for this very basic classification methodology. Extensions are available that can enhance the nearest-neighbor strategy but still retain the nonparametric flavor; see, for example, Hastie et al. (2009, Section 13.4).

9.4 Tree-based methods

9.4.1 Classification trees

An alternative and somewhat more advanced nonparametric method for constructing supervised classification rules is based on application of *decision trees*: simple, branching flowcharts that are both straightforward to plot and easy to intuit. (They are part of a larger class of tree-structured diagrams known as *dendrograms*.) This makes them very popular in decision analysis; de Ville (2013) provided an engaging review.

The decision tree approach apportions the feature space into a series of disjoint hyper-rectangles by recursively partitioning larger rectangles into smaller ones. This grows the tree, $\mathcal{T}$, hierarchically from an originating 'root' collection $\mathcal{T}_1$ – traditionally located at the top of the display – down to the final classification of $M > 1$ terminal 'nodes.' (The arboresque terminology here becomes rather imaginative: the terminal nodes are the 'leaves,' while bifurcations from existing, intermediate nodes are 'branches' of the tree.) Each node is examined to determine the quality of its classification regime: as more and more elements or fewer and fewer elements from a single category populate the node, we say it has greater 'purity' for classifying the category. Statistical operations are usually conducted on the complementary 'impurity' scale, so the goal becomes one of finding rules that decrease impurity as the tree grows.

In its simplest form, this recursive partitioning evolves by applying a series of dichotomous splitting criteria to the feature variables. These take the form $X_j \geq \tau$ versus $X_j < \tau$ for some threshold τ, and they create a pair of new child nodes branching out from that parent node. The routine searches through the set of feature variables to determine a splitting rule that provides greatest decrease in impurity at each new branch; so-called greedy search algorithms are commonly applied, focusing on local improvement to rapidly progress the optimization (Kantardzic 2003, Section 7.1). The tree continues growing until a predetermined stopping rule is attained, say, when every node reaches a minimum size or when no further decrease in impurity is possible from any existing node. The leaves are then defined as the terminal nodes at which the tree stops.

The result in the feature space is a series of recursively embedded (hyper)rectangles that define the terminal nodes and that contain the predicted classifications. Figure 9.5 gives an idealized example with two input variables X_1 and X_2 under a dichotomous classification scheme ($Y = 1$ for 'Positive' and $Y = 2$ for 'Negative').

To define the impurity measure at any mth node ($m = 1, \ldots, M$) denote by N_{qm} the number of observations classified into the qth category ($q = 1, \ldots, Q$). Let $N_{+m} = \sum_{q=1}^{Q} N_{qm}$ be the sum over all Q categories. Consider first the simplest case of $Q = 2$. Conditional on the X_js and on the previous branchings, $N_{qm} \sim \text{Bin}(N_{+m}, \pi_{qm})$ where π_{qm} is the probability of classifying into category q at node m. Clearly, as $\pi_{1m} \to 0$ or $\pi_{1m} \to 1$, the node's purity for classifying category $q = 1$ improves. (Similarly for category $q = 2$.) Away from these extremes, the impurity grows.

A possible measure to quantify impurity here is (proportional to) the variance of N_{qm}, $\pi_{qm}(1 - \pi_{qm})$, because binomial variation also increases as π_{qm} departs from 0 or 1. Summing over q yields the *Gini impurity*

$$\mathcal{G}_m = \pi_{1m}(1 - \pi_{1m}) + \pi_{2m}(1 - \pi_{2m}) = 2\pi_{1m}(1 - \pi_{1m}),$$

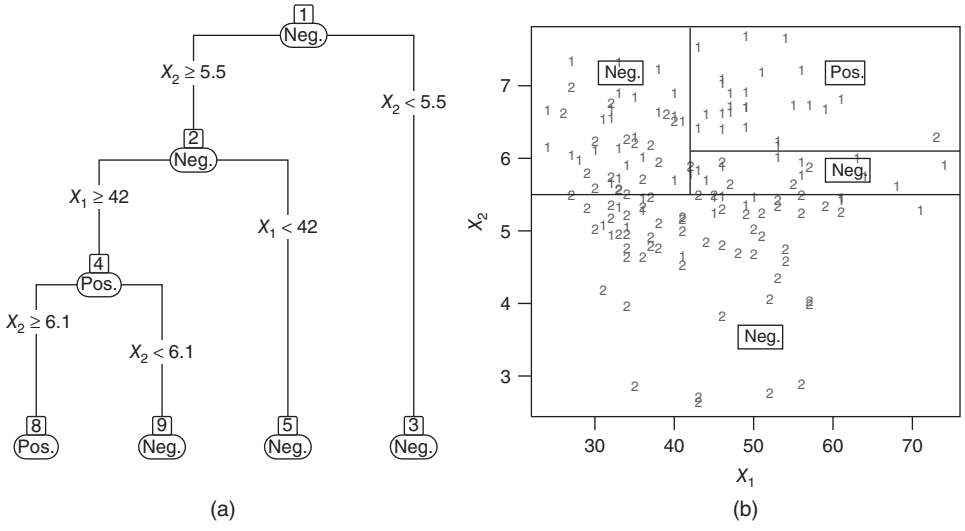

Figure 9.5 Idealized four-leaf classification tree (a) with $Q = 2$ outcome categories ($Y = 1$ for 'Pos.' and $Y = 2$ for 'Neg.') and (b) corresponding partitions in the (X_1, X_2) feature space.

where the latter equality recognizes that $\pi_{2m} = 1 - \pi_{1m}$ in this binomial case. Alternatively, another measure for variation/disorder is based on the information-theoretic entropy from (2.7). For a discrete binomial variable, this produces

$$\mathcal{E}_m = -\pi_{1m} \log_2(\pi_{1m}) - \pi_{2m} \log_2(\pi_{2m}) = -\pi_{1m} \log_2 \left(\frac{\pi_{1m}}{1 - \pi_{1m}} \right) - \log_2(1 - \pi_{1m}).$$

(Many authors use the more-popular natural log instead of the base-2 log in $\mathcal{E}_m$, because the result differs by only a constant multiple.)

A third alternative involves the node's misclassification error. Let successful classification in the mth node be defined as achieving maximal containment for the observations in the qth category (James et al. 2013, Section 8.1.2). Then the *misclassification error* is

$$\mathcal{M}_m = 1 - \max\{\pi_{1m}, \pi_{2m}\} = 1 - \max\{\pi_{1m}, 1 - \pi_{1m}\}.$$

In practice, we estimate π_{qm} by the per-node proportion $p_{qm} = N_{qm}/N_{+m}$. With this, and extending to $Q > 2$ categories, the estimated Gini impurity becomes

$$\mathcal{G}_m = \sum_{q=1}^{Q} p_{qm}(1 - p_{qm}) = 1 - \sum_{q=1}^{Q} p_{qm}^2,$$

while the estimated entropy (also called *cross-entropy*) is

$$\mathcal{E}_m = -\sum_{q=1}^{Q} p_{qm} \log_2(p_{qm}). \tag{9.11}$$

If one replaces the base-2 log with the natural log when computing (9.11), the impurity becomes

$$\mathcal{E}_m = -\sum_{q=1}^{Q} p_{qm} \log(p_{qm}) = -\frac{1}{N_{+m}} \sum_{q=1}^{Q} N_{qm} \log(p_{qm}), \tag{9.12}$$

where (0)log(0) is defined as 0. Now, the extension from the binomial case with $Q = 2$ to the general case with $Q > 2$ induces a multinomial probability structure in the N_{qm}s. (The multinomial distribution is a multiple-category generalization of the binomial model, also mentioned in Section 8.3.3.) In this case, Venables and Ripley (2002, Section 9.1) showed, in effect, that for fixed m the impurity in (9.12) differs by only a constant from the *deviance* (defined in Section 8.2.2) under the multinomial model:

$$-2 \sum_{q=1}^{Q} N_{qm} \log(p_{qm}).$$

As a result, many authors refer to the entropy and the deviance interchangeably.

Lastly, the estimated misclassification error is

$$\mathcal{M}_m = 1 - \max_{q=1,\ldots,Q} \{p_{qm}\}.$$

Among the three impurities $\mathcal{G}_m$, $\mathcal{E}_m$, and $\mathcal{M}_m$, misclassification error is generally the least sensitive, while the Gini and entropy impurities operate in a roughly similar manner. A variety of additional measures can also be constructed (Linoff and Berry 2011, Chapter 7), although the three impurities presented above are common in statistical learning.

This bifurcating, two-branch splitting strategy can be broadened to allow for multiway branching from each node and/or to use linear combinations of the X_js in the splitting rules. Some of these features are embodied in a famous series of tree-based algorithms known progressively as the ID3, C4.5, and C5.0 routines; see Quinlan (1996). For more on these and other extensions of the basic classification tree paradigm, Loh (2010) and Linoff and Berry (2011, Chapter 7), among many others, provided instructive expositions.

9.4.2 Pruning

Simple stopping rules for tree-based classifiers can vary, leading to potentially wide differences in how a tree is actually used for classification. One common problem is *overfitting*, the inclusion when a tree grows out too far of classifications that overaccommodate the particular features in the training data. This decreases bias, but typically at a cost of increased variance and poor predictive ability. (As in Sections 7.4 and 9.3, another bias-variance trade-off.)

A strategy for combatting overfitting involves *pruning* a tree $\mathcal{T}$ as or after it is formed. View this as the culling of possibly superfluous nodes from $\mathcal{T}$, not unlike pruning overgrown branches from a real tree. To do so, denote by $|\mathcal{T}|$ the number of terminal nodes of the full tree and index these nodes via $m' = m_1, m_2, \ldots, m_{|\mathcal{T}|}$. Define the *cost-complexity risk* as

$$R_\alpha(\mathcal{T}) = R(\mathcal{T}) + \alpha|\mathcal{T}|, \tag{9.13}$$

where α is a complexity parameter that tunes the risk and $R(\mathcal{T})$ is some function describing the total cost or risk associated with the full tree. A common choice for $R(\mathcal{T})$ is the terminal misclassification error $\sum_{m'} \mathcal{M}_{m'}$, although other impurity measures such as terminal entropy are also possible.

Notice the similarity with the objective functions in regression regularization from Section 7.4: α acts as a form of penalty parameter that controls the complexity of overfitted models (here, trees). As $\alpha \to 0$, the pruned tree approaches the full tree, while as $\alpha \to \infty$, the pruning drives towards the original root node. It can be shown that as α increases, the full tree can be cut back to some optimal subtree that minimizes (9.13). This is known as 'weakest-link cutting' or 'weakest-link pruning' (Breiman et al. 1984, Section 3.3).

In **R**, the external *rpart* package provides functions to perform the recursive partitioning algorithm, including weakest-link pruning, as does the external *tree* package. The latter is slightly simpler to use, although the former is popular for construction of classification trees. A variety of other packages also offer enhanced output graphics for classification trees, including *rpart.plot* and *maptree*.

The complexity parameter α is often chosen in a data-dependent manner from the training set. As in Section 9.3, cross-validation is a favored approach, here applied in an L-fold manner. (At $L = n$, we recover the leave-one-out form of cross-validation mentioned earlier, although for very large training sets, this can become computationally prohibitive.) When applied to classification trees, typical default choices for L lie in the range $5 \le L \le 10$.

To apply L-fold cross-validation, randomly separate the training data into L disjoint subsets of roughly equal size. Then, remove the ℓth subset and use the remaining data to grow a full tree ($\ell = 1, \ldots, L$). Apply the predicted classifications to the removed ℓth subset and record the misclassification error. Repeat across all ℓ folds and average the result. (Also include the standard error of the mean for comparison purposes.) Finally, perform these operations over a range of possible values for α, and choose that α with minimum average error. Return to the full training set and apply cost-complexity pruning with that chosen α.

The cross-validation error often drops sharply from α at ∞ and then may flatten as $\alpha \to 0$. Thus there may be a number of possible candidates for α with essentially equal errors. We then turn to the colloquially named *one-standard-error rule*: prune to the highest α (i.e., the most parsimonious tree) whose average cross-validation error is no larger than the minimum average error plus its standard error. In **R**, the external *rpart* package provides a number of functions to assist in this effort.

Example 9.4.1 Remote sensing of tree disease (Example 9.3.1, continued). Return to the oak tree disease study in Example 9.3.1 and consider growing a classification tree from that example's training data set. Recursive partitioning as implemented in the **R** *rpart* package is featured here. Given the classification variable Y ($Y = 1$ for diseased trees, $Y = 2$ for other land cover) and the predictor variables X_1, X_2, X_3, X_4, X_5 from Table 9.9, sample **R** code for constructing the full tree is

```
> Yfac <- factor( Y, levels=1:2, labels=c('Diseased','Other') )
> require( rpart )
> control.set <- rpart.control( cp=0, xval=10, minbucket=5 )
> set.seed( 941 )
> classify.rpart <- rpart( Yfac ~ X1+X2+X3+X4+X5, method='class',
                parms=list(split='gini'), control=control.set )
```

In the code, notice the following:

- The response variable is rebuilt into the factor variable `Yfac` so that the `rpart()` function will properly construct a classification tree (also see `method=` in the following).

- The preliminary call to `rpart.control()` sets various control parameters for constructing the tree:
  - to build a full tree, `cp=0` initially sets the complexity parameter α to 0;
  - `xval=10` calls for 10-fold cross-validation (in order to later select a value for α); and
  - `minbucket=5` sets the stopping rule to allow no fewer than five observations in a terminal node/leaf.
- Because 10-fold cross-validation populates the folds randomly, it is good practice to set the seed for the random number generator explicitly – here this is `set.seed(941)`.
- In the call to `rpart()`,
  - the leading formula has a familiar structure, similar to that seen in earlier regression-type functions such as `lm()` and `glm()`;
  - `method='class'` instructs **R** to build a classification tree (because the response variable is a factor, the function would do so by default; still, it is instructive to present it here);
  - `parms=list(split='gini')` calls for use of Gini impurity (this is also the default; `split='information'` would call for entropy impurity); and
  - `control=control.set` draws in the control settings from the earlier use of `rpart.control()`.

The resulting object `classify.rpart` contains all the information necessary to build the tree. For instance, simply calling the object prints out the splitting criteria and node containment details. The corresponding full tree can be visualized by applying the commands `plot(classify.rpart)` and `text(classify.rpart)`. These plot the tree and add informative labels, respectively. Or, `summary(classify.rpart)` provides a highly detailed output (all results left to reader).

As described earlier, however, pruning the tree is a logical next step. Useful in this regard is *rpart*'s `printcp(classify.rpart)` command. This constructs a so-called CP table for choosing the complexity parameter α. Here, the table lists the average 10-fold cross-validation errors along with their standard errors over a range of α values. The output (edited) is

```
Classification tree:
rpart(formula = Yfac ~ X1 + X2 + X3 + X4 + X5, method = "class",
    parms = list(split = "gini"), control = control.set)

Variables actually used in tree construction:
[1] X2 X3 X4
Root node error: 74/4339 = 0.017055
n = 4339
          CP  nsplit  rel error   xerror      xstd
1  0.2432432       0    1.00000  1.00000  0.115252
2  0.0405405       2    0.51351  0.54054  0.085072
3  0.0270270       3    0.47297  0.50000  0.081848
4  0.0225225       5    0.41892  0.48649  0.080744
5  0.0202703       8    0.35135  0.48649  0.080744
6  0.0067568      10    0.31081  0.44595  0.077333
7  0.0000000      12    0.29730  0.41892  0.074971
```

The output reveals a number of important features. Perhaps the most interesting thing we see under '`Variables actually used`' is that only X_2, X_3, and X_4 were employed in forming the full tree. (This would have been evident when displaying the full tree graphically as well.) Thus the two pansharpening band variables from Table 9.9, X_1 ={Mean gray-level} and X_5 ={Std. deviation}, were not seen to add value in constructing the tree.

Next is the data in the displayed CP table: minimum cross-validation error (`xerror` column) occurs at $\alpha = 0$ (`CP` column). Applying the one-standard error (s.e.) rule, however, shows that the cross-validation error at $\alpha = 0.0225$ is within one standard error of this minimum ($0.48649 < 0.41892 + 0.07497 = 0.49389$), so choose a marginally larger value, say $\alpha = 0.025$, as the complexity parameter for pruning the tree. (Similar determinations can be viewed graphically via *rpart*'s `plotcp(classify.rpart)` command.) The pruned tree is then computed using

```
> prune.rpart <- prune( classify.rpart, cp=0.025 )
```

(One could return to the `rpart` command and first fit the reduced model using only the three variables, X_2 = {Mean green band}, X_3 ={Mean red band}, and X_4 ={Mean near-infrared band}, that is, with `formula = Yfac ~ X2 + X3 + X4`. Pruning to the same $\alpha = 0.025$ would not change the final decision tree, however.)

Visualization is possible via a number of functions and routines. For instance, by restricting attention to the three variables X_2, X_3, and X_4, an intriguing graphic can be produced: Figure 9.6 is a three-dimensional scatterplot of the data using only these three feature variables. 'Diseased' image points are colored differently than the 'Other' image points. (The figure was created using routines from the external *scatterplot3d* package.) A clear pattern emerges: the diseased images locate in a restricted swath near smaller values of the X_2 and X_3 inputs. Clearly, strong potential exists for developing a useful classification rule from these training data.

To view the pruned tree, `print(prune.rpart)` details the splitting rules and pertinent per-node statistics (output edited):

```
node), split, n, loss, yval, (yprob)
      * denotes terminal node
 1) root 4339 74 Other (0.01705462 0.98294538)
   2) X3>=114.6723 1221 51 Other (0.04176904 0.95823096)
     4) X2< 222.4286 48  6 Diseased (0.87500000 0.12500000)
       8) X4>=317.4018 43  2 Diseased (0.95348837 0.04651163) *
       9) X4< 317.4018 5  1 Other (0.20000000 0.80000000) *
     5) X2>=222.4286 1173  9 Other (0.00767263 0.99232737) *
   3) X3< 114.6723 3118 23 Other (0.00737652 0.99262348)
     6) X2< 184.8764 155 14 Other (0.09032258 0.90967742)
      12) X3>=87.83333 14  5 Diseased (0.64285714 0.35714286) *
      13) X3< 87.83333 141  5 Other (0.03546099 0.96453901) *
     7) X2>=184.8764 2963  9 Other (0.00303746 0.99696254) *
```

A more visually digestible representation, however, displays the tree graphically. As noted above, `plot(prune.rpart)` and `text(prune.rpart)` are available, although these are somewhat rudimentary. The external *rpart.plot* package and its powerful `prp()` function can build more engaging tree graphics; for example, the sample **R** code

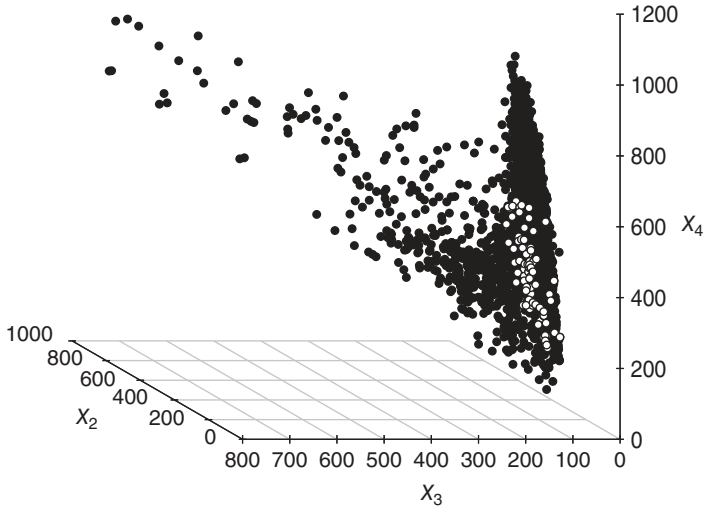

Figure 9.6 Three-dimensional scatterplot of oak tree disease training data in Example 9.4.1, using feature variables X_2, X_3, and X_4 from Table 9.9. Points are distinguished by observed classification: open circles are 'Diseased' tree images and filled circles are 'Other' images. Data from Johnson et al. (2013).

```
> require( rpart.plot )
> prp( prune.rpart, type=4, extra=104, clip.right.labs=F, nn=T,
                                                    fallen.leaves=T )
```

generates a tree with a variety of enhanced features; see `help(prp)` for details. Figure 9.7 displays the pruned tree with the root node (no. 1) at the top and branches eventually ending in six leaves (terminal nodes' numbers 8, 9, 5, 12, 13, 7, as per the numbers in the small boxes above each node). The graphic gives the following information:

- The splitting criteria used to form each new branch; for example, the first split was created by dichotomizing between $X_3 \geq 115$ (left branch) and $X_3 < 115$ (right branch).
- The predicted classifications in each node. Focus on the terminal leaves; for example, proceeding down the far left branches to terminal node/leaf no. 8 predicts an observed image's trees into the 'Diseased' category when (i) $X_3 \geq 115$, then (ii) $X_2 < 222$, and finally (iii) $X_4 \geq 317$.
- Each node's containment statistic. For example, terminal node/leaf no. 8 contains only $43/4339 = 1\%$ of the training set observations, the percentage given at the bottom of the leaf box. A few leaves list 0% containment, but this is due to rounding. For instance, leaf no. 9 contains $5/4339 = 0.1\%$ of the training observations. (Find these values in the `print(prune.rpart)` output.)
- Labeling that gives each node's summary classification probabilities, also called 'posterior probabilities.' In terminal node/leaf no. 8, for example, 95% of the observations

lie in the 'Diseased' category. As this exceeds 50%, the fitted/predicted category is 'Diseased' as labeled in the leaf box. Obviously, the remaining 5% is 'Other.' Find these values via `unique(predict(object=prune.rpart, type='prob'))` Corresponding statistics are displayed in the other leaf boxes.

The pruned tree can also be used for predictive analytics. In **R**, if `train.df` is the original training set data frame, the confusion matrix for the training data can be calculated via

```
> Yhat.train = predict( object=prune.rpart, newdata=train.df,
                                                    type='class' )
> table( Yhat.train, Yfac )
```

Similarly, suppose `test.df` contains the test set data frame with a column `Y0` giving the observed categories for each observation (`Y0` = 1 for diseased trees; `Y0` = 2 for other land

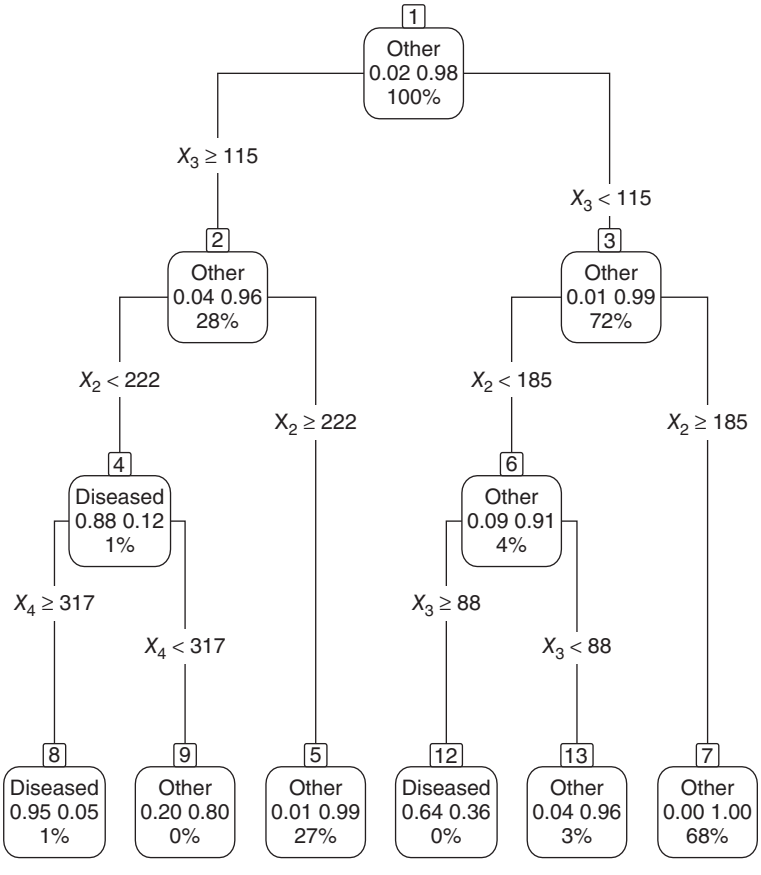

Figure 9.7 Classification tree for training data in oak tree disease study from Example 9.4.1, using feature variables listed in Table 9.9. Displayed tree gives six terminal modes, pruned from the full classification tree using a complexity parameter ('cp') of $\alpha = 0.025$. See text for descriptions of the individual node labels. Source: Data from Johnson et al. (2013).

cover) and with five additional columns giving the corresponding feature variables X_{01}, X_{02}, X_{03}, X_{04}, and X_{05}. Then, the confusion matrix for the test data is available via

```
> testYfac = factor( Y0, levels=1:2, labels=c('Diseased','Other') )
> Yhat.test = predict( object=prune.rpart, newdata=test.df,
                                                  type='class' )
> table( Yhat.test, testYfac )
```

The results appear in Table 9.11.

Table 9.11 2×2 confusion matrices from pruned classification tree in Example 9.4.1.

		Training set			**Test set**		
		Observed			Observed		
		Diseased	Other	Row total	Diseased	Other	Row total
Predicted	Diseased	50	7	57	74	3	77
	Other	24	4258	4282	113	310	423
	Column total	74	4265	4339	187	313	500

From the confusion matrices in Table 9.11, we see the pruned tree's overall accuracy for the training set is high: `mean(Yhat.train==Yfac)` gives $(50 + 4258)/4339 = 99.3\%$. The corresponding misclassification error is only $31/4339 = 0.7\%$. Compare these to their counterparts from the k-NN fit in Example 9.3.1: there the accuracy was incrementally lower at 99.1% while misclassification was higher at 0.9%. These values are arguably indistinguishable, suggesting roughly similar capacities between the two methods for classifying the training data. (The pruned tree does classify more training images as diseased, both correctly and incorrectly.)

For the test data, however, more substantive improvement is seen: the test accuracy in Table 9.11 is $(74 + 310)/500 = 76.8\%$, much larger than the 63.8% seen with the k-NN analysis. Similarly, the test misclassification error of pruned tree is only $116/500 = 23.2\%$, well below the k-NN error of 36.2%.

Recall that sensitivity – that is, the true positive rate – was felt to be a pertinent target quantity with these data. From Table 9.11, the sensitivity for the training data is $50/74 = 67.6\%$, while for the test data, it is $74/187 = 39.6\%$. Both values are increased over the k-NN results of 45.9% and 3.2%, respectively. At least for these remote sensing data, the pruned decision tree analysis has clearly improved on these various summary measures. Exercise 9.16 explores this analysis further. □

Tree-based methods are often favored in classification analytics, due to their relative simplicity, natural graphic output, and fairly intuitive interpretation(s). Their nonparametric flavor, not requiring a formal parametric model for implementation, also drives much of their popularity. Indeed, they can operate with both continuous and categorical predictor/feature variables, and nothing in their construction stymies them by missing values in the data. They do possess some disadvantages, however, not the least of which is their tendency to overfit, producing high variances. Associated with this is a tendency to propagate early classification errors down the tree, due to their fundamental, hierarchical nature. An adjustment known as

bagging (Breiman 1996) employs a form of bootstrap resampling (Section 5.3.6) to average over many different trees and help to alleviate these concerns. Such averaging is often called a form of *ensemble learning*. See Hastie et al. (2009, Section 8.7).

For further discussions on recursive partitioning and classification trees, see the presentations in Hand et al. (2001, Section 10.5) and Clarke et al. (2009, Section 5.3).

9.4.3 Boosting

Another technique useful for improving the classification capacity of a decision tree is known as *boosting* (Schapire 1989). Often discussed in association with bagging (above) – but based on a different core technology – the method attempts to 'boost' the performance of weak classifiers. (A 'weak' classifier exhibits predictive ability only marginally better than random guessing.) In boosting, one constructs an averaged classification rule by combining sequentially updated classifiers from the same training data. This is sometimes referred to as 'classification by committee' and is another form of ensemble learning.

A highly effective algorithm to perform boosting is AdaBoost (Freund and Schapire 1996, 1997). AdaBoost starts by equally weighting each training observation and constructing an associated classification tree. It then updates the weights to increase the influence of points that were misclassified. The new weights determine the next classifier. The process repeats and the final classification rule is itself based on a weighted average of the updated classifiers. The algorithm can drastically improve the performance of weak classification trees. In **R**, boosting is available in a number of external packages, including *ada*, *adabag*, *gbm*, and *mboost*.

As a concept, boosting is naturally extensible and can be employed with more than just classification trees. It has been applied across a rich panoply of data mining contexts. For more details, see Hastie et al. (2009, Chapter 10).

9.4.4 Regression trees

Tree-based methods may also be applied in supervised learning when the response variable Y is quantitative. This technically places us back in the regression setting from Chapters 6 to 8, although it is useful to briefly discuss the approach here.

If one applies basic recursive partitioning to a continuous response variable Y, the result is, in effect, a form of nonparametric regression. The splitting criteria continue to take the form $X_j \geq \tau$ versus $X_j < \tau$ for some threshold τ. The predicted response will then be a series of flat (hyper)rectangles lying above the rectangular partitions of the feature space produced by each m'th terminal node. The height of the predicted response will be equal to the mean of the Y_is contained in that node, say, $\overline{Y}_{i(m')}$. In this sense, the regression tree is often referred to as a *local model*: it captures local characteristics within each terminal node but does not attempt to make global statements across the entire feature space.

The splitting criterion is once again based on a quantitative impurity measure. Most natural with a continuous outcome is reduction in the residual sum of squares when splitting the mth node into two child nodes (Hastie et al. 2009, Section 9.2.2). As overfitting remains a problem, pruning is also conducted using the cost-complexity approach and incorporating L-fold cross-validation where necessary.

Taken together, regression trees and classification trees comprise a larger class of tree-based methods known as *classification and regression trees* (*CART*), developed by Breiman et al. (1984). A modern introduction is available, for example, in James et al. (2013,

Section 8.1). In **R**, *rpart* fits regression trees using the same `rpart()` function: simply invoke the `method='anova'` option. See `help(rpart)`.

As they are sister methodologies, regression trees share both the strengths and the weaknesses of classification trees from Section 9.4.1. Bagging (bootstrapped aggregation) can again be applied to improve predictive accuracy. A further enhancement, known as *random forests*, extends the bootstrapping by randomly sampling from the feature variables at each split (Breiman 2001). The result, which may seem counterintuitive, can actually reduce correlation in the eventual regression tree. In **R**, the external *randomForest* package can conduct both bagging and random forests for CARTs (see James et al. 2013, Sections 8.2–8.3).

9.5 Support vector machines*

A powerful, modern, supervised learning technology is based on *support vector* (SV) methods (Vapnik 1995, 1998). SV classifiers are increasingly applied with high-dimensional data in, for example, genetic research with microarray or single nucleotide polymorphism data (O'Fallon et al. 2013; Zhang et al. 2006), or in large-scale signal processing studies (Salcedo-Sanz et al. 2014). The calculations involve fairly sophisticated statistical and mathematical techniques, so this section provides only a brief introduction; Mammone et al. (2009) gave a larger, contemporary review. Focus herein is on two classes ($Q = 2$). It will be convenient to reformulate the categorical response as now $Y = 1$ for category $q = 1$ and $Y = -1$ for category $q = 2$.

SV methods are best conceptualized by distinguishing between two disjoint cases: data that are completely separable in the input-variable space and those that exhibit some nonseparable overlap. The next section begins with the former case.

9.5.1 Separable data

Separable data occur when the observations lie in clearly disjoint regions of the input space and when no overlap is evident between them. In particular, assume that the separation is linear such that a roughly straight band cleaves the two classes. While uncommon in practice, this sort of compete linear separation is useful for laying the foundations of the SV approach. As in previous sections, the goal is to determine a linear decision boundary that efficiently discriminates between the two categories. When the data are linearly separated, however, there will be infinitely many such boundaries. (Why?) To find an optimal separator when $Q = 2$, Vapnik in his 1995 textbook showed that a unique separating line exists that maximizes the distance between itself and the closest points in both classes; see Vapnik (2000, Section 5.4) in the book's second edition. The separating distance between the two classes is called the *margin*; this *optimal separating line* lies in the middle of the margin and acts to maximize its width. (It is, therefore, often described as a *maximum margin classifier*.) In higher dimensions, the line becomes a hyperplane – cf. Section 9.2.3 – and we refer to the decision boundary generically as the *optimal separating hyperplane*. For consistency, the generic terminology is used here. Denote this hyperplane by $\mathcal{H}_0$.

The two borders at the edge of the margin are called *supporting hyperplanes*, and any points that lie directly on these borders – there must be at least one for each hyperplane – are called *support vectors*. (The terminology harkens to the geometric interpretation where a point in space defines a vector to the origin.) Denote the supporting hyperplane for category $q = 1$

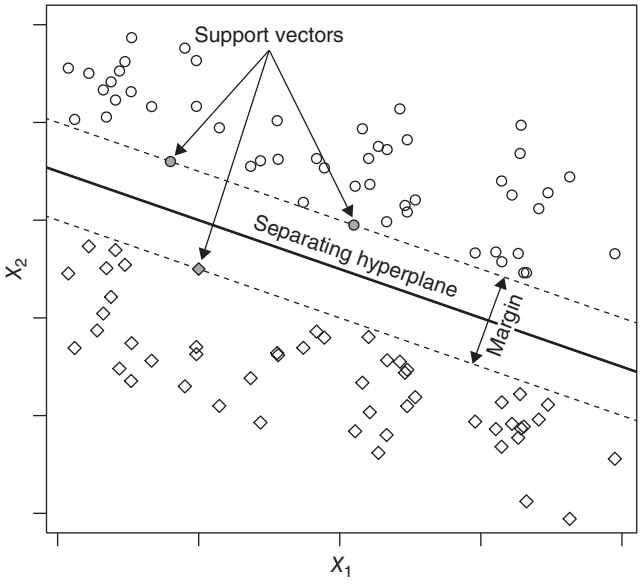

Figure 9.8 Fully separable data: idealized display of supporting hyperplanes $\mathcal{H}_1$ (upper dashed line) and $\mathcal{H}_2$ (lower dashed line) and optimal separating hyperplane $\mathcal{H}_0$ (solid line) in (X_1, X_2) plane with $Q = 2$ categories ($Y = 1$ for upper circles; $Y = -1$ for lower diamonds). Support vectors are highlighted in gray.

by $\mathcal{H}_1$ and that for category $q = 2$ by $\mathcal{H}_2$. Figure 9.8 gives an idealized illustration. (Therein, and in what follows, assume without loss of generality that the 'positive' category at $q = 1$ is defined to lie above $\mathcal{H}_0$.)

Mathematically, the separating hyperplane $\mathcal{H}_0$ in Figure 9.8 satisfies

$$\mathcal{H}_0 = \{\beta_0, \boldsymbol{\beta}\colon\ \beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0\}$$

for some coefficient vector $\boldsymbol{\beta} = [\beta_1 \quad \beta_2]^{\mathrm{T}}$ and some constant β_0. This gives a rule for classification: assign Y_i into category $q = 1$ if $\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} > 0$ and into category $q = 2$ if $\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} < 0$. Since $Y_i = \pm 1$, this gives

$$Y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}) > 0, \quad \text{for all } i = 1, \ldots, n. \tag{9.14}$$

Notice that the farther a point is from the optimal separating hyperplane, the more certain we are of its classification. This gives distances from $\mathcal{H}_0$ intrinsic interpretation. Indeed, one can show that the distance from $\mathcal{H}_0$ to either supporting hyperplane is $1/\parallel \boldsymbol{\beta} \parallel$ (Clarke et al. 2009, Section 5.4.1), where the notation $\parallel \boldsymbol{\beta} \parallel$ represents the L_2 norm of $\boldsymbol{\beta}$:

$$\parallel \boldsymbol{\beta} \parallel = \sqrt{\beta_1^2 + \beta_2^2}\,. \tag{9.15}$$

To find the β_js, given the data, we appeal to optimization arguments. The goal is to identify that separating hyperplane whose margin is pushed as far apart as possible but still supports

the separated classes. That is, we wish to maximize the width of the margin subject to (9.14). It will be convenient to work with just the margin half-width, which we write as some positive value $\mathscr{M} > 0$. To contain the separated classes, this then requires

$$Y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}) \geq \mathscr{M}, \quad \text{for all } i = 1, \ldots, n. \tag{9.16}$$

In effect, we aim to maximize $\mathscr{M}$ subject to (9.16). Unfortunately, in its present form, this problem cannot be solved uniquely: if the quantities β_j satisfy (9.16), so will $\psi\beta_j$ for any $\psi \geq 1$. To compensate, impose the additional constraint $\| \boldsymbol{\beta} \| = 1$. Rearranging terms and recognizing that $\mathscr{M} = 1/\| \boldsymbol{\beta} \|$, the problem then simplifies into the following constrained minimization: find the separating hyperplane $\mathcal{H}_0$ of the form $\beta_0 + \beta_1 X_1 + \beta_2 X_2$ to

(a) minimize the objective quantity $\frac{1}{2} \| \boldsymbol{\beta} \|^2$, subject to

(b) $Y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}) \geq 1$ for all $i = 1, \ldots, n$.

This inequality-constrained minimization falls into the class of convex optimization problems (Lange 2013, Chapter 14). The specifics exceed the scope of discussion here, although the result is fairly simple to express. The coefficients for the optimal separating hyperplane $\mathcal{H}_0$ are

$$\beta_j = \sum_{i=1}^{n} a_i Y_i X_{ij} \tag{9.17}$$

($j = 1, 2$), where the a_i terms satisfy

$a_i = 0$ if X_{ij} lies strictly within its classifying region, or
$a_i > 0$ if X_{ij} lies *on* its supporting hyperplane.

Notice here that only the SVs affect construction of the separating hyperplane: observations wholly within the classification region contribute no weight ($a_i = 0$) to $\mathcal{H}_0$. This gives the SV approach a certain robustness to extreme, outlying observations.

To find the a_i coefficients in (9.17), dualities in the convex optimization can be exploited to write the problem equivalently as a second-order optimization. Maximize the objective quantity

$$\mathcal{D}(a_1, \ldots, a_n) = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{h=1}^{n} a_i a_h Y_i Y_h \mathbf{X}_i^{\mathrm{T}} \mathbf{X}_h \tag{9.18}$$

subject to

$$\sum_{i=1}^{n} a_i Y_i = 0 \tag{9.19}$$

and $a_i \geq 0$ for all $i = 1, \ldots, n$. Standard quadratic optimization algorithms then complete the calculations (Karatzoglou et al. 2006).

With the β_js given by (9.17), β_0 is found by solving $Y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}) = 1$ for all training tuples $(\mathbf{X}_i, Y_i)$ satisfying $a_i > 0$. As there will likely be many solutions, one can average the results or, following Clarke et al. (2009, Section 5.4.4), take

$$\beta_0 = -\frac{1}{2} \min_{\{i: Y_i = 1\}} \left\{ \beta_1 X_{i1} + \beta_2 X_{i2} \right\} - \frac{1}{2} \max_{\{i: Y_i = -1\}} \left\{ \beta_1 X_{i1} + \beta_2 X_{i2} \right\}.$$

Using these quantities, the SV classifier can be written as a simple function of any putative or test feature vector $\mathbf{X} = [X_1 \quad X_2]^{\mathrm{T}}$:

$$\delta_{\mathrm{SV}}(\mathbf{X}) = \mathrm{sgn}\left(\beta_0 + \sum_{i=1}^{n} a_i Y_i \mathbf{X}_i^{\mathrm{T}} \mathbf{X}\right), \tag{9.20}$$

where $\mathrm{sgn}(x) = -I_{(-\infty,0)}(x) + I_{(0,\infty)}(x)$ is the signum function that reports the sign of its argument. Notice that a majority of the terms in (9.20) will likely correspond to $a_i = 0$; this sparseness can reduce much of the computational burden in the calculation.

Generalization of the SV classifier to $p > 2$ feature variables is conceptually straightforward: simply extend $[X_{i1} \; X_{i2}]^{\mathrm{T}}$ to $\mathbf{X}_i = [X_{i1} \; X_{i2} \; \cdots \; X_{ip}]^{\mathrm{T}}$ and $[\beta_1 \; \beta_2]^{\mathrm{T}}$ to $\boldsymbol{\beta} = [\beta_1 \; \cdots \; \beta_p]^{\mathrm{T}}$, and write the separating hyperplane in the general form $\beta_0 + \mathbf{X}_i^{\mathrm{T}} \boldsymbol{\beta}$. Make corresponding changes to (9.17), (9.18), and (9.20).

9.5.2 Nonseparable data

When the data fail to separate, that is, when some observations with $Y_i = 1$ and $Y_h = -1$ cross in the (X_1, X_2) input space, we say that they are *nonseparable*. This creates what is known as a 'soft margin' problem, because imposition of the 'hard' margin constraints as in Section 9.5.1 no longer admits an optimal solution. The linear separator is now called a *soft margin line*, or more generally a *soft margin hyperplane*.

To accommodate nonseparable data, we admit the existence of a *slack variable*, $\xi_i \geq 0$, that incorporates location information for each $\mathbf{X}_i = [X_{i1} \; X_{i2}]^{\mathrm{T}}$. If $\mathbf{X}_i$ is correctly contained within its classification margin (above it if $Y_i = 1$ and below if $Y_i = -1$), set $\xi_i = 0$. If $\mathbf{X}_i$ is misclassified such that it incorrectly lies beyond its opposing margin, set $\xi_i > 1$. In between, set $0 < \xi_i < 1$. The misclassification error is then conveniently written as $\sum_{i=1}^{n} I_{(1,\infty)}(\xi_i)/n$, where $I_{\mathcal{A}}(\cdot)$ is the indicator function from (2.20).

We continue to write the separating hyperplane $\mathcal{H}_0$ as

$$\mathcal{H}_0 = \{\beta_0, \boldsymbol{\beta} \colon \beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0\}.$$

By employing the slack variables, the classification rule now assigns Y_i into category $q = 1$ if $\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} > 1 - \xi_i$ and into category $q = 2$ if $\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} < -(1 - \xi_i)$. As $Y_i = \pm 1$, a unifying relationship again obtains

$$Y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}) > 1 - \xi_i, \quad \text{for all } i = 1, \ldots, n.$$

The constrained optimization problem evolves into the following form:

(a) minimize the objective quantity $\frac{1}{2} \parallel \boldsymbol{\beta} \parallel^2 + C \sum_{i=1}^{n} \xi_i$, subject to

(b) $Y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}) \geq 1 - \xi_i$ for all $i = 1, \ldots, n$, and

(c) $\xi_i \geq 0$ for all $i = 1, \ldots, n$,

where $C \geq 0$ is a tuning parameter that incorporates the cost of misclassification. Notice that $\sum_{i=1}^{n} \xi_i / n$ is an upper bound on the misclassification error (Exercise 9.20). In effect, this 'budgets' for misclassification/training errors: as $C \to \infty$, the margin narrows and misclassification violations are viewed as more 'expensive.' Conversely, as $C \to 0$, the margin widens, $\parallel \boldsymbol{\beta} \parallel$

shrinks, and violations are viewed as more tolerable. This is another example of bias-variance trade-off – small C drives bias down but increases variance – similar to that seen with regularized regression in Section 7.4. Indeed, the cost reciprocal $\lambda = 1/C$ is often called the SV *regularization parameter*.

If no prior choice or external validation data are available for choosing C, some authors choose a large default value such as $C = 10^2$ or higher; this tightens the margin and minimizes overlap but can sometimes lead to overfitting. One could instead compare results across a selection of values for C and subjectively select that which provides the best predictive performance. Or, Hastie et al. (2009, Section 12.3.5) described an algorithm to compute a piecewise-linear *regularization path* of the *support vector machine* (SVM) across all values of λ. Cross-validation can also be employed, using the current training set.

The corresponding solution leads to similar relationships as those found for the separable case in Section 9.5.1. In particular, β_j takes the same form as in (9.17) where the a_i coefficients now satisfy

$$\begin{array}{lll} a_i = 0 & \text{for } Y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}) \geq 1 & \text{and } \xi_i = 0, \text{ or} \\ 0 < a_i < C & \text{for } Y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}) = 1 & \text{and } \xi_i = 0, \text{ or} \\ a_i = C & \text{for } Y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}) \leq 1 & \text{and } \xi_i \geq 0 \end{array}$$

(Clarke et al. 2009, Section 5.4.5). Once again, observations wholly within their (correct) classification region contribute no weight ($a_i = 0$) to the separating hyperplane $\mathcal{H}_0$. Those that do affect $\mathcal{H}_0$ now either lie on the (correct) margin or lie within the margin on the correct side of $\mathcal{H}_0$. To distinguish these, we say the former are *margin support vectors*, while the latter are *nonmargin support vectors*. Points lying in the incorrect classification region – which also affect construction of $\mathcal{H}_0$ – are of course viewed as misclassifications. Figure 9.9 presents an idealized illustration.

With β_j given by (9.17), β_0 is found by solving $Y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}) = 1$ over all training tuples $(\mathbf{X}_i, Y_i)$ satisfying $a_i > 0$ at $\xi_i = 0$ and averaging the results (Hastie et al. 2009, Section 12.2.1).

The dual optimization criteria (9.18) and (9.19) also continue to hold, now with $0 \leq a_i \leq C$ for all $i = 1, \ldots, n$. Computer routines are used to conduct the calculations (Karatzoglou et al. 2006).

9.5.3 Kernel transformations

When classification data do not present linear separations the effort to find effective classification rules becomes more challenging. Classes that can appear difficult to separate in the initial $\mathbf{X}$-input space may be easily distinguished in some higher-dimensional space, however, after appropriate transformation of the input variables. It is within such settings that SV technology becomes particularly useful. Figure 9.10 gives an abstract representation, motivated by an example from Schölkopf and Smola (2002, Section 2.1). The (X_1, X_2) points in Figure 9.10a illustrate a classic 'bull's-eye' pattern where the two groups are clearly distinct, but where no linear classifier can completely capture the separation. As seen in Figure 9.10b, however, transforming (X_1, X_2) to the three-dimensional vector $(X_1^2, X_2^2, \sqrt{2}X_1X_2)$ now easily distinguishes the two classes.

We can take advantage of this potential for effective separation by moving to transformed spaces. Return to the SV objective quantity in (9.18) and notice that the input vectors $\mathbf{X}_i$ only

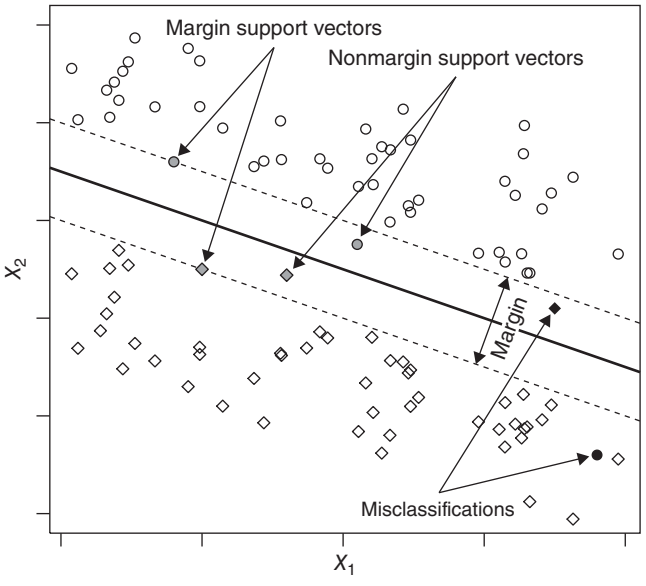

Figure 9.9 Nonseparable data: idealized display of separating hyperplane $\mathcal{H}_0$ (solid line) and associated margin (bounded by dashed lines) in (X_1, X_2) plane with $Q = 2$ categories ($Y = 1$ for upper circles; $Y = -1$ for lower diamonds). Margin and nonmargin support vectors are indicated in gray. Misclassified points are indicated in black.

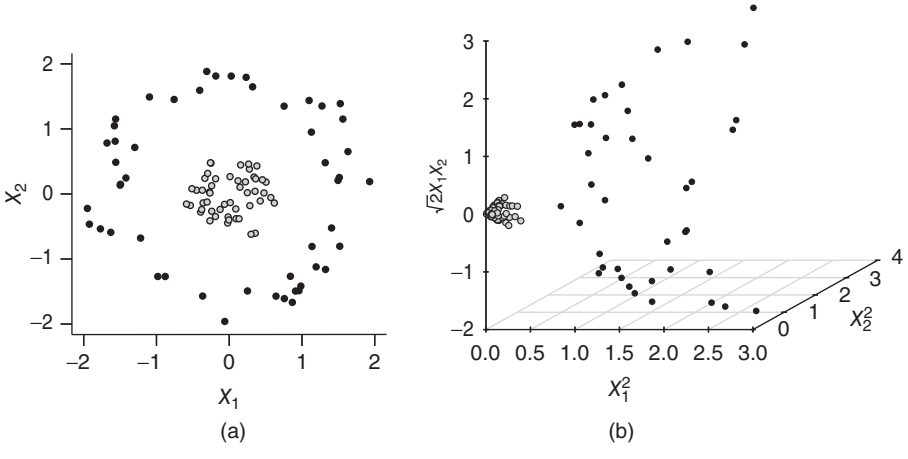

Figure 9.10 Idealized effect of nonlinear separation and kernel transforms. (a) Nonseparable (X_1, X_2) data ($Y = 1$ as gray dots versus $Y = -1$ as black dots) in classic 'bull's-eye' pattern. (b) Transformation into three dimensions via $(X_1^2, X_2^2, \sqrt{2}X_1X_2)$ produces clear separation.

enter into the expression via the Euclidean inner product $\mathbf{X}_i^{\mathrm{T}}\mathbf{X}_h$. A similar effect is evident with the SV classification rule in (9.20): $\delta_{\mathrm{SV}}(\mathbf{X})$ depends on the inputs only through the inner product $\mathbf{X}_i^{\mathrm{T}}\mathbf{X}$. Thus we could consider some vector-valued transformation of $\mathbf{X}$ to

$$\mathbf{U} = \mathbf{U}(\mathbf{X}) = \begin{bmatrix} U_1(\mathbf{X}) \\ U_2(\mathbf{X}) \\ \vdots \\ U_r(\mathbf{X}) \end{bmatrix},$$

where $r > p$ is the dimension of the transformed feature space. Suppose $\mathbf{U}$ successfully separates the two classes in this transformed space, and we apply it in place of $\mathbf{X}$ in (9.18). That is, optimize the objective quantity $\mathcal{D}(a_1, \ldots, a_n)$ with $\mathbf{X}_i^{\mathrm{T}}\mathbf{X}_h$ replaced by $\mathbf{U}_i^{\mathrm{T}}\mathbf{U}_h$. This calculation will then yield a valid, r-dimensional, optimal separating hyperplane $\mathcal{H}_0$.

For example, the three-dimensional transformation in Figure 9.10 takes $\mathbf{X}_i^{\mathrm{T}}\mathbf{X}_h = X_{i1}X_{h1} + X_{i2}X_{h2}$ to

$$\begin{aligned} \mathbf{U}_i^{\mathrm{T}}\mathbf{U}_h &= X_{i1}^2X_{h1}^2 + 2X_{i1}X_{i2}X_{h1}X_{h2} + X_{i2}^2X_{h2}^2 \\ &= (X_{i1}X_{h1} + X_{i2}X_{h2})^2 = (\mathbf{X}_i^{\mathrm{T}}\mathbf{X}_h)^2. \end{aligned}$$

Notice that we can write this transformation as a function of the two input vectors $\mathbf{X}_i$ and $\mathbf{X}_h$, say, $K(\mathbf{X}_i, \mathbf{X}_h) = (\mathbf{X}_i^{\mathrm{T}}\mathbf{X}_h)^2$. The function $K(\cdot, \cdot)$ is called a *kernel*. In the general case, when a kernel satisfies appropriate regulatory conditions (Clarke et al. 2009, Section 5.4.6), it can be used as a separation-inducing, dimension-transforming replacement in the SV constructions. For instance, the optimization problem generalizes to maximizing the objective quantity

$$\mathcal{D}_{\mathrm{K}}(a_1, \ldots, a_n) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{h=1}^{n} a_i a_h Y_i Y_h K(\mathbf{X}_i, \mathbf{X}_h) \tag{9.21}$$

subject to

$$\sum_{i=1}^{n} a_i Y_i = 0$$

and $0 \le a_i \le C$ for all $i = 1, \ldots, n$. The a_is are used in the classifier

$$\delta_{\mathrm{SV}}(\mathbf{X}) = \mathrm{sgn}\left\{\beta_0 + \sum_{i=1}^{n} a_i Y_i K(\mathbf{X}_i, \mathbf{X})\right\},$$

for any putative or test feature vector $\mathbf{X}$. With these, the constant β_0 is found as described in Section 9.5.2. The resulting construction is known formally as a *support vector machine*, although the term is sometimes used more generally to describe the general methods described throughout in this section.

A wide variety of kernel functions can be employed in (9.21). Table 9.12 lists some of the more popular forms. The linear kernel at top serves as an 'identity' transform, while the succeeding polynomial kernels of order d are popular in practice; indeed, the homogeneous, second-order kernel was employed in Figure 9.10. Another common choice is the Gaussian radial basis function (RBF), due to its obvious connections with the normal (Gaussian) p.d.f. from Section 2.3.9.

Table 9.12 Selected kernel functions $K(\mathbf{A},\mathbf{B})$ of two p-dimensional vector inputs $\mathbf{A}$ and $\mathbf{B}$, for use with support vector machines in Section 9.5.3.

Name	$K(\mathbf{A},\mathbf{B})$	
Linear	$\mathbf{A}^T\mathbf{B}$	
Homogeneous polynomial	$(\gamma\mathbf{A}^T\mathbf{B})^d$	
Inhomogeneous polynomial	$(\phi_o + \gamma\mathbf{A}^T\mathbf{B})^d$	
Gaussian RBF	$\exp\{-\gamma \parallel \mathbf{A}-\mathbf{B} \parallel^2\}$	$(\gamma > 0)$
Laplace RBF	$\exp\{-\gamma \parallel \mathbf{A}-\mathbf{B} \parallel\}$	$(\gamma > 0)$
Cauchy	$\frac{1}{\pi}(1 + \parallel \mathbf{A}-\mathbf{B} \parallel^2)^{-1}$	
Sigmoid	$\tanh(\phi_o + \gamma\mathbf{A}^T\mathbf{B})$	
Thin-plate spline	$\parallel \mathbf{A}-\mathbf{B} \parallel \log(\parallel \mathbf{A}-\mathbf{B} \parallel)$	

γ and ϕ_o are kernel-specific tuning parameters.
Abbreviation: RBF, radial basis function.
Note: $\parallel \mathbf{B} \parallel = (B_1^2 + B_2^2 + \cdots + B_p^2)^{1/2}$ is the L_2 norm, extending (9.15).

Notice that some kernels contain additional parameters. For many, a quantity γ is included as an additional tuning parameter. If no prior choice or external validation data are available for determining the kernel parameter(s), cross-validation can be employed using the current training set.

It is worth remarking that most analysts normally *avoid* adding dimensions to a problem, because this can have enigmatic consequences with, for example, the curse of dimensionality mentioned in Section 7.4.1. The 'kernel trick' employed with SVMs often overcomes this, however, due to computational advantages implicit in application of the kernel transform (Schölkopf and Smola 2002, Section 2.2).

In **R**, a number of external packages provide SVMs in one form or another. These include the `svm()` function in the *e1071* package, the `ksvm()` function in the *kernlab* package, and for regularization path analysis, the `svmpath()` function in the *svmpath* package. Karatzoglou et al. (2006) give a useful review, include training time benchmarks.

Example 9.5.1 Remote sensing of tree disease (Example 9.3.1, continued). Return to the oak tree disease data in Example 9.3.1 and now apply SVMs to the classification exercise. As per the results in Example 9.4.1, restrict attention to the three input variables X_2, X_3, and X_4 from Table 9.9. Also, recode the output to $Y = 1$ for 'Diseased' tree images and $Y = -1$ for 'Other' images.

Consider use of the Gaussian RBF kernel from Table 9.12. Application here is via the `ksvm()` function in the external *kernlab* package. Given the ± 1 category outcomes in the vector `Y`, the function requires construction of a corresponding category factor, along with a matrix of the input variables. Sample **R** code is

```
> Yfac <- factor( Y, levels=c(1,-1), labels=c('Diseased','Other') )
> train.mtx <- as.matrix( cbind(X2,X3,X4) )
```

To set the cost parameter, a search over the range $C = 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2$ with the kernel parameter γ initially fixed at 1 (not shown) indicates that training set misclassification error can be driven to zero with C as low as 1. Note that a training error of zero need not be optimal: the underlying classification rule may be overfitting the training set and may not perform well on future test data (as seen later in this example). Nonetheless, it serves here

as a useful starting point, so set $C = 1$. With this, to select γ for the Gaussian RBF, the *kernlab* package offers an internal estimation scheme based on the package's `sigest` function. This estimates the kernel parameter from the empirical quantiles of $\| \mathbf{X}_i - \mathbf{X}_j \|$ $(i \neq j)$ and is implemented via the `kpar='automatic'` option. Employing these settings in the full `ksvm` command gives

```
> require( kernlab )
> train.ksvm <- ksvm( x=train.mtx, y=Yfac, scaled=F, type='C-svc',
                      kernel='rbfdot', kpar='automatic', C=1 )
```

where the **R** object `train.ksvm` is assigned the SVM classifier information based on the training data. The various options operate as follows:

- The matrix of input variables is specified via the `x=` option and the corresponding vector of category factors is specified via the `y=` option.
- `scaled=F` instructs **R** to refrain from centering and scaling the various data vectors/matrices. (The default is `scaled=T`.) Such scaling can be useful, but for simplicity, it is overridden here.
- `type='C-svc'` institutes the basic SVM classification scheme described in this section. (The function offers a variety of SV methods for more complex classification and regression settings.)
- The Gaussian RBF kernel is chosen via `kernel='rbfdot'`, with `kpar='automatic'` calling for automated section of γ, as noted above. To specify a predetermined value for γ, say, $\gamma = 1$, use `kpar=list(sigma=1)`. (The function's notation for γ in the Gaussian RBF is `sigma`.)
- `C=1` sets the value for the cost parameter at $C = 1$. (This is also the default.)

The resulting classification results can be determined from components of the `train.ksvm` object. Note that `ksvm` objects are rendered using **R**'s S4 object system, as opposed to the older S3 objects primarily seen herein. By default, S4 components are extracted with the `@` operator and do not exclusively use the `$` operator; see `help(ksvm)` and `help('@')`. For instance, the internally chosen value for the kernel parameter when $C = 1$ is available in

```
> train.ksvm@kernelf@kpar$sigma
```

which is found here to be $\gamma = 2.4657 \times 10^{-4}$. The corresponding misclassification rate is found from

```
> train.ksvm@error
```

as 0.004148.

The confusion matrices for both the training and the test data can be calculated via the following sample commands:

```
> YhatSVM.train <- predict( object=train.ksvm, newdata=train.mtx )
> table( YhatSVM.train, Yfac )
> testY <- factor( Y0, levels=c(1,-1), labels=c('Diseased','Other') )
> test.mtx <- as.matrix( cbind(X02,X03,X04) )
> YhatSVM.test <- predict( object=train.ksvm, newdata=test.mtx )
> table( YhatSVM.test, testY )
```

where the notation for the test data variables is taken from Example 9.4.1. The results appear in Table 9.13. We see that further improvement in overall training accuracy (4321/4339 = 99.6%) and misclassification error (0.4%, above) are evidenced, relative to their counterparts in Examples 9.3.1 and 9.4.1. Improvements are also seen with the rates for the test data; in particular, 'Diseased' image classification is clearly strengthened.

Table 9.13 Confusion matrices from SVM classification using Gaussian RBF with $C = 1$ and $\gamma = 2.4657 \times 10^{-4}$ in Example 9.5.1.

		Training set			Test set		
		Observed			Observed		
		Diseased	Other	Row total	Diseased	Other	Row total
Predicted	Diseased	56	0	56	107	3	110
	Other	18	4265	4283	80	310	390
	Column total	74	4265	4339	187	313	500

Table 9.14 summarizes the results from these various analyses, including pertinent comparisons for the sensitivity γ_1, that is, the true positive rate. Recall that γ_1 was felt to be an important target measure with this application. For these remote sensing data, at least, improvements are seen with all measures as the classification strategy moves from the k-NN to SVM methods.

Table 9.14 Comparison of summary analytic measures for Oak Tree Disease data.

	Training set		
Classification approach	Accuracy	Misclassification error	Sensitivity
k-NN (Example 9.3.1)	0.991	0.009	0.459
Classification tree* (Example 9.4.1)	0.993	0.007	0.676
SVM* (Example 9.5.1)	0.996	0.004	0.757
	Test set		
Classification approach	Accuracy	Misclassification error	Sensitivity
k-NN (Example 9.3.1)	0.638	0.362	0.032
Classification tree* (Example 9.4.1)	0.768	0.232	0.396
SVM* (Example 9.5.1)	0.834	0.166	0.572

* Pruned to employ only X_2, X_3, and X_4 input variables from Table 9.9.

Finally, it is worth returning to the initial SVM fit with $C = \gamma = 1$ to illustrate issues with overfitting. Recall that when applying a Gaussian RBF kernel at these settings, the misclassification rate was exactly 0%. This can be seen in the corresponding confusion matrices for both the training and test data given in Table 9.15. (The table was constructed using similar **R** commands as those that produced Table 9.13.) Therein, the training data are fit perfectly:

Table 9.15 Confusion matrices from SVM classification using Gaussian RBF kernel with $C = \gamma = 1$ in Example 9.5.1.

		Training set			Test set		
		Observed			Observed		
	Diseased	Other	Row total		Diseased	Other	Row total
Predicted	Diseased	74	0	74	1	0	1
	Other	0	4265	4265	186	313	499
	Column total	74	4265	4339	187	313	500

overall accuracy is 100% and no misclassifications occur. The results for the test data are less encouraging, however. The accuracy drops precipitously to 314/500 = 62.8%, with complementary increase in misclassification error to 37.2%. Perhaps more daunting, sensitivity plummets from 100% to $\frac{1}{187}$, barely one-half of 1%! This is classic overfitting: the SVM classifier has overtrained and has locked in patterns from the training data. It fails miserably, however, to predict forward from these on to the test data. The diseased-tree images are particularly misevaluated. By moving, in this case, to a smaller kernel parameter γ, the classifier generates a few more training misclassifications and also vastly improves its ability to predict the separate test data outcomes.

Exercise 9.19 explores further use of the SVM classifier with these data. □

Although the discussion here has focused on binary classification with $Q = 2$, strategies do exist for applying SVMs in the multiclass setting with any $Q \geq 2$. Popular are manipulations where multiple SVMs are constructed and then compared in a pairwise or compound manner. Observations are then assigned into a class based on some optimal containment measure. For more, see Clarke et al. (2009, Section 5.4.10) and James et al. (2013, Section 9.4).

SV methods can also be applied when the response Y is quantitative instead of categorical, similar to the case with classification trees and CART models in Section 9.4.4. This leads to a form of *support vector regression*. The error structure is (re)developed to view the linear regression function as a kind of decision boundary, to which SV-based optimization can be adapted (see Hastie et al. 2009, Section 12.3.6).

Modern applications of SVMs in classification and regression are broad and evolving; they cannot be captured easily by the short introduction given here. Besides the various sources referenced throughout this section, interested readers may benefit from the further expositions given in Moguerza and Munoz (2006) or Izenman (2008, Chapter 11). Also see Dixon and Brereton (2009) and Kruppa et al. (2014) for comparative studies of various classification methods, including many of those discussed in this chapter.

Exercises

9.1 The bank marketing study in Example 9.2.1 also queried $n = 361$ never-married customers with a college ('tertiary') education who had not been contacted in any previous marketing campaign. For this cohort, only $X = \log\{\text{Duration}\}$ of the call was seen to be important for predicting whether ($Y = 1$) or not ($Y = 0$) a depositor subscribed in the

new campaign. A selection of the (X_i, Y_i) data pairs follows. (Download the complete set at http://www.wiley.com/go/piegorsch/data_analytics.)

Y = Subscribed:	1	0	0	$\cdots$	1	1
X = log{Duration}:	6.8648	2.7726	4.6634	$\cdots$	6.7991	5.8833

(a) Plot Y against X. What does the pattern reveal?

(b) Fit a simple linear, conditional, logistic model to the data: $P[Y = 1 \mid X = x] = \pi(x) = 1/(1 + e^{-\beta_0 - \beta_1 x})$. Find the estimated success probability $\hat{\pi}(x)$ and overlay it on your scatterplot.

(c) Construct the confusion matrix and from it calculate the overall accuracy, misclassification error, sensitivity, and specificity for these data under the logistic discriminant rule.

(d) Estimate the probability $\pi(\log\{420\})$ that a depositor in this cohort will subscribe after a call lasting 7 min (420 s). Also find a 95% confidence interval for this value.

(e) At what call duration do the bank's marketers 'break even;' that is, how long a call is needed to meet or exceed a 50% probability of garnering a subscription? (This is known generically as a *median effective stimulus*; here it would be a 'median effective duration.') Interpret this quantity within the context of two-class discrimination.

9.2 Suppose a conditional logistic regression based on $p = 2$ predictors is used for two-category discrimination/classification with estimated linear predictor $\hat{\eta} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2$.

(a) Verify algebraically that classifying $Y = 1$ when $\hat{\pi}(\mathbf{x}) > \frac{1}{2}$ leads to (9.3).

(b) Derive an expression for the perpendicular distance from any observed point (x_{10}, x_{20}) to the discrimination boundary.

(c) Can you extend this to the general case with $p > 2$ predictors in $\hat{\eta} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$?

9.3 From a bioinformatic study of gene expression profiles in carcinogenesis, Dagliyan et al. (2011) provided prostate cancer outcomes in $n = 102$ male patients, along with their expression data from a variety of potential genetic markers. The response was binary: $Y = 1$ if a patient's prostate tissue was identified as positive for the tumor or $Y = 0$ if it was not. Consider here $p = 3$ genes to represent important predictors of the tumor outcome as listed in the following selection of data. (Download the complete set at http://www.wiley.com/go/piegorsch/data_analytics)

Y = tumor positive:	1	1	1	$\cdots$	0	0	0
x_1 = serum protease hepsin (X07732):	203	182	117	$\cdots$	14	46	38
x_2 = ao89h09.x1 (AI207842):	165	109	215	$\cdots$	881	809	240
x_3 = GSTM4 (M96233):	54	34	29	$\cdots$	96	236	138

(a) Construct a 3×3 scatterplot matrix that pairs all three predictors. Code the plotted points as to whether or not the patient had a positive tumor outcome. Comment on any observed patterns.

(b) Fit a conditional multiple logistic model to these data with all three predictor variables. Use this to build a linear discriminant function for classifying cancer in this population. In particular, give the equation of the estimated discriminant function. Also find the confusion matrix and from it calculate the overall accuracy, misclassification error, sensitivity, and specificity of the discriminant rule.

(c) Find the predicted probabilities $\hat{\pi}(x_1, x_2, x_3)$ for each patient and plot them against the observed value of Y. Comment on the pattern in the plot.

(d) From the fitted logistic model, estimate the probability that a new patient will be classified with a positive tumor outcome if he/she presents the following predictor values: $x_1 = 100$, $x_2 = 300$, $x_3 = 50$. Also include a 95% confidence interval for this value.

9.4 A well-known database for illustrating classification with a binary outcome involves diabetes diagnoses among the Native America Pima (Akimel O'odham) people in central-southern Arizona (Smith et al. 1988). The response was $Y = 1$ if diabetes was diagnosed or $Y = 0$ if it was not, among $n = 724$ female Pimas at least 21 years old. Following Myatt and Johnson (2009, Section 4.5), consider $p - 1 = 5$ predictor variables from the larger database as listed in the following (x_4 = Diabetes Pedigree is a score incorporating family history of diabetes; higher values suggest higher risk), along with a selection of the data after curation to remove missing or ambiguous observations. Download this complete set of data at http://www.wiley.com/go/piegorsch/data_analytics.

Y = Diabetes:	0	0	0	$\cdots$	1	0	1
x_1 = Age (years):	21	32	22	$\cdots$	33	31	25
x_2 = Glucose (mg/dL):	102	87	90	$\cdots$	137	197	180
x_3 = Body mass index:	25.1	23.2	27.3	$\cdots$	43.1	36.7	59.4
x_4 = Diabetes pedigree:	0.078	0.084	0.085	$\cdots$	2.288	2.329	2.42
x_5 = Blood pressure (diastolic):	52	80	70	$\cdots$	40	70	78

(a) Construct a 5×5 scatterplot matrix that pairs all five predictors. Code the plotted points as to whether or not the subject had a diabetes diagnosis. Comment on any observed patterns.

(b) Fit a conditional multiple logistic model to these data with all five predictor variables. Use this to build a linear discriminant function for classifying diabetes in this population. In particular, give the equation of the estimated discriminant function. Also find the confusion matrix and from it calculate the overall accuracy, misclassification error, sensitivity, and specificity of the discriminant rule.

(c) Find the predicted probabilities $\hat{\pi}(x_1, x_2, x_3, x_4, x_5)$ for each subject and plot them against the observed value of Y. Comment on the pattern in the plot.

(d) From the fitted logistic model, estimate the probability that a Pima female will be classified with diabetes if she presents the following predictor values: $x_1 = 30$,

$x_2 = 110, x_3 = 44.4, x_4 = 0.500$, and $x_5 = 85$. Also include a 95% confidence interval for this value.

9.5 Suppose a logistic regression model using (9.1) is employed for two-category classification with estimated linear predictor $\hat{\eta} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$. Verify algebraically that the Wald confidence interval for η satisfying $P(\hat{\eta} - z_{\alpha/2}\text{se}[\hat{\eta}] \le \eta \le \hat{\eta} + z_{\alpha/2}\text{se}[\hat{\eta}]) = 1 - \alpha$ can be manipulated into the confidence statement

$$P\left[\frac{1}{1+\exp\{-\hat{\eta} + z_{\alpha/2}\text{se}[\hat{\eta}]\}} \le \pi(\mathbf{x}) \le \frac{1}{1+\exp\{-\hat{\eta} - z_{\alpha/2}\text{se}[\hat{\eta}]\}}\right] = 1 - \alpha.$$

9.6 If the logistic model in (9.1) is replaced by a probit regression, as in Section 8.3.1, what eventual changes occur in (9.2) and (9.3)?

9.7 The Bank Marketing Study in Example 9.2.1 also queried $n = 449$ married customers who had not completed any secondary school education and who had not been contacted in any previous marketing campaign. For this cohort, the variables $X_1 = \{\text{Age of depositor}\}$ and $X_2 = \log\{\text{Duration of marketing phone call (in s)}\}$ were again felt to be important for predicting whether ($Y = 1$) or not ($Y = 2$) a depositor subscribed in the new campaign. A selection of the data is given as follows. (Download the complete set of data at http://www.wiley.com/go/piegorsch/data_analytics.)

Y = Subscribed:	2	2	2	$\cdots$	2	1
X_1 = Age (years):	30	55	26	$\cdots$	37	60
$X_2 = \log\{\text{Duration (s)}\}$:	4.3694	5.5094	6.0521	$\cdots$	6.2422	6.7044

(a) Plot the two predictor variables and code the plotted points as to whether or not the customer subscribed. Comment on any observed patterns.

(b) Apply a LDA to these data, as in Section 9.2.1. Use this to build a linear discriminant function for classifying subscribers in this cohort. In particular, give the equation of the estimated discriminant function. Overlay this function on your plot in Exercise 9.7a and remark on any pattern.

(c) Find the corresponding confusion matrix and from it calculate the overall accuracy, misclassification error, sensitivity, and specificity of the discriminant rule.

(d) From the LDA, predict the subscription status of a future, 40-year-old customer ($X_1 = 40$) if contacted for the following durations: 3 min (180 s), 4 min (240 s), 5 min (300 s), 6 min (360 s), 7 min (420 s), and 8 min (480 s). Display or plot these as a function of the duration time. Comment on the result.

9.8 Return to the urban vulnerability data in Example 9.2.2.

(a) Construct a confusion matrix for the classifications given in the example. (Extend the 2×2 matrix from Table 9.2 into a 3×3 matrix for these data.) From this calculate the overall accuracy and the misclassification error rate. Comment on the results.

(b) The supplied database contains the longitude and latitude for each city. If you have access to geoinformatic mapping software – the external *map*

package in **R** is a rudimentary version – plot the 132 cities on a US map and overlay the predicted class assignments from the example (found, e.g., in `predict(cities132.lda)$class`). Use different colors or symbols to distinguish the three predicted classes. Comment on the patterns that are observed.

9.9 Under the multivariate normal model, verify the equalities in (a) (9.8) and (b) (9.9). (*Hint*: Recall that the linear combination $\mathbf{a}^{\mathrm{T}}\mathbf{b}$ is a scalar, and therefore, $\mathbf{a}^{\mathrm{T}}\mathbf{b} = (\mathbf{a}^{\mathrm{T}}\mathbf{b})^{\mathrm{T}} = \mathbf{b}^{\mathrm{T}}\mathbf{a}$.)

9.10 Kohavi (1996) reported on a US Census Bureau database of income (in 1994 $) among US residents. Studied was propensity to exceed an annual income of $ 50 000. Focus here is on a cohort of 17 936 males employed in private industry or self-employed. Following suggestions in the study, a training set comprising two-thirds of these data was randomly created, with sample size $n = 11\,957$. The categorical response was set to $Y = 1$ if a subject's income exceeded $ 50 000 and $Y = 2$ if not. Quantitative input variables for consideration are $X_1 = \{\text{Age}\}$, $X_2 = \{\text{Hours worked/week}\}$, $X_3 = \{\text{Years of education}\}$, and $X_4 = \{\text{Capital gains (or losses) from previous year}\}$. A sample of the training data follows:

Training set							
Y = Exceed $ 50K:	2	2	2	···	1	2	2
X_1 = Age (years):	35	30	31	···	27	54	46
X_2 = Hours worked:	48	40	45	···	42	40	40
X_3 = Education (years):	13	9	9	···	7	9	13
X_4 = Capital gain/loss ($):	0	0	0	···	0	0	−1590

This left a test set of the remaining $n_o = 5979$ data points, a sample of which follows (download both complete data sets at http://www.wiley.com/go/piegorsch/data_analytics):

Test set							
Y_0 = Exceed $ 50K:	2	1	2	···	2	1	2
X_{01} = Age (years):	39	54	25	···	49	81	57
X_{02} = Hours worked:	80	40	45	···	55	45	25
X_{03} = Education (years):	9	9	9	···	9	2	10
X_{04} = Capital gain/loss ($):	0	0	3325	···	0	0	0

(a) Apply a k-NN analysis to these data. Set $k = 5$ and employ Euclidean distance. Scale the input variables in $\mathbf{X}$ by their standard deviations before any calculations. Construct the confusion matrices for both the training data and the test data and from these report the associated accuracy and misclassification error rates.

(b) Repeat your analysis with $k = 3$ and comment on any differences.

9.11 If applying a k-NN analysis with $k = 1$, what form would you expect the confusion matrix to take when the training data is applied to itself (i.e., when you 'predict' the

categories of the training observations using only the same observations to train the learning process)? Can you imagine a setting where this might not occur?

9.12 Return to the k-NN classification for the oak tree disease data in Example 9.3.1 and explore how the (a) misclassification error and (b) sensitivity change for the test data as k varies over a range of values, say, $k = 3, 5, 15, 25$. Comment on your results.

9.13 One can explore differences among the various impurity measures used to build classification trees in Section 9.4.1 using graphical comparisons (Hastie et al. 2009, Section 9.2.3). For simplicity, let $Q = 2$. As a function of π_1 over the range $0 < \pi_1 < 1$, plot the Gini measure $2\pi_1(1 - \pi_1)$, the entropy $-\frac{1}{2}\pi_1 \log_2(\pi_1) - \frac{1}{2}(1 - \pi_1)\log_2(1 - \pi_1)$, and the misclassification error

$$\pi_1 I_{[0,\frac{1}{2}]}(\pi_1) + (1 - \pi_1) I_{(\frac{1}{2},1]}(\pi_1),$$

where $I_{\mathcal{A}}(\cdot)$ is the indicator function from (2.20). (The entropy here is scaled to reach the same maximum as the others.) Comment on the patterns you observe.

9.14 Return to the following data sets and view each as a standalone set of training data. Construct a classification tree from the associated collection of input variables. Operate with the Gini impurity measure. Start with a full tree and use L-fold cross-validation to help to identify a complexity parameter for pruning the tree. To select the complexity parameter α, appeal to a plot of the cross-validation error versus α (available, e.g., via the `plotcp()` function in the external *rpart* package). Employ the one-standard error rule: select the largest α whose cross-validation error does not exceed the minimum error plus its standard deviation. Plot the pruned tree. Also find the confusion matrix for these training data, and calculate the consequent accuracy and misclassification rates.

(a) The bank marketing data in Example 9.2.1. Set $L = 8$.

(b) The Pima diabetes data in Exercise 9.4. Set $L = 5$.

9.15 Mansouri et al. (2013) reported a study of quantitative structure–activity relationships (QSARs) among 1055 chemical molecules, in order to classify whether or not the chemicals were readily biodegradable. The category response was set to $Y = 1$ if a chemical's biodegradation status was viewed as 'ready' and to $Y = 0$ if not. An associated set of quantitative input variables is given here as $X_1 = \{$Number of heavy atoms in the molecule$\}$, $X_2 = \{$Number of substituted benzene carbons$\}$, $X_3 = \{$Number of ring tertiary carbons$\}$, and $X_4 = \{$Intrinsic state pseudoconnectivity (ISP) index$\}$. A training set comprising 705 chemicals from the full data set was randomly generated, a sample of which follows:

Training set							
Y = Ready status	0	1	0	$\cdots$	0	0	1
X_1 = Heavy atoms	0	0	0	$\cdots$	2	0	0
X_2 = Substituted benzene C	2	0	0	$\cdots$	0	18	0
X_3 = Ring tertiary C	0	0	2	$\cdots$	0	0	1
X_4 = ISP index	0.004	0.425	−0.007	$\cdots$	−0.022	0.000	0.004

This left a test set of the remaining $n_o = 350$ data points, a sample of which follows (download both complete data sets at http://www.wiley.com/go/piegorsch/data_analytics):

Test set							
Y_0 = Ready status	1	1	1	$\cdots$	0	0	0
X_{01} = Heavy atoms	0	2	1	$\cdots$	1	0	2
X_{02} = Substituted benzene C	0	0	0	$\cdots$	3	5	9
X_{03} = Ring tertiary C	0	0	0	$\cdots$	0	0	0
X_{04} = ISP index	0.011	−0.271	0.000	$\cdots$	−0.025	0.000	0.000

(a) Construct a classification tree from the training data, as per the methods in Section 9.4. Operate with the Gini impurity measure. Start with a full tree and use 10-fold cross-validation to help to identify a complexity parameter for pruning the tree. Plot the resulting tree. Also find the confusion matrix for both the training data and the test data, and calculate the consequent accuracy and misclassification rates.

(b) Repeat your analysis with five-fold cross-validation and comment on any differences.

9.16 Return to the pruned classification tree for the oak tree disease data in Example 9.4.1.

(a) Verify the indication in the example that fitting only the three input variables X_2, X_3, and X_4 produces the same pruned classification tree (at complexity parameter $\alpha = 0.025$) as given in Figure 9.7.

(b) Reproduce the three-dimensional scatterplot in Figure 9.6 and study the pattern by changing the viewing/perspective angle. If your 3D plotter has interactive capabilities (as, e.g., in the external **R** package *rgl*), rotate the display in real time to better visualize the data cloud.

(c) Repeat the 3D visualization in Figure 9.6 now with the test data. Comment on any similarities or differences in the 3D plot. In particular, do the test data lie in a similar range of X_2, X_3, and X_4 values as the training data? If not, and based on the patterns you observe in both plots, how might this affect the classification analysis?

(d) One might be tempted to use hypothesis tests to compare the test data sensitivity of the pruned classification tree (74/187) to that from the corresponding k-NN analysis (6/187) from Example 9.3.1. For practice, conduct such a test; that is, assess the equality of the two proportions against any difference. Employ the Fisher exact test from Section 8.3.3. While perhaps informally useful for gauging differences between these two proportions, the formal inferences here are invalid. Why?

9.17 Return to the urban vulnerability study from Example 9.2.2 and view the collection as a standalone set of training data.

(a) Construct a classification tree from the two feature variables X_1 and X_2. (Notice that this is a multiclass problem with more than two categories.) Operate with the Gini

impurity measure and use 10-fold cross-validation to help to identify a complexity parameter α for pruning. Plot the resulting tree.

(b) Find the confusion matrix associated with the pruned tree, and calculate the consequent accuracy and misclassification rates with these data. Compare these to the results in Exercise 9.8.

(c) Imitate the display in Figure 9.4 by finding the splits given by the pruned tree in the (X_1, X_2) plane and overlaying on them the original data. Label the data points by their observed categories (1, 2, or 3). Include a label in each separate classification region demarcating its predicted category. Comment on the differences between the two graphics.

(d) Identify which of the 132 cities are classified by the tree into the high-vulnerability category. How does this list compare with Table 9.7?

9.18 Return to the following data sets and apply SVMs to perform the classification exercise with the associated collection of input variables. Employ a Gaussian RBF kernel; if you use the `ksvm()` function in **R**, find the kernel parameter γ via its automatic section routine. Set the cost parameter C as instructed in the following.

(a) The bank marketing data in Example 9.2.1, viewed as a standalone set of training data. Start with a selection of cost parameters: $C = 10^{-1}, 10^0, 10^1, \ldots, 10^5$. Choose that C for which the associated misclassification error is a minimum; report the associated value for γ. Find the resulting confusion matrix for the full data set and compare it, along with the accuracy and misclassification error, to that from the classification tree in Exercise 9.14a.

(b) The Pima diabetes data in Exercise 9.4, viewed as a standalone set of training data. Start with a selection of cost parameters: $C = 10^{-1}, 10^0, 10^1, \ldots, 10^5$. Choose that C for which the associated misclassification error is a minimum; report the associated value for γ. Find the resulting confusion matrix for the full data set and compare it, along with the accuracy and misclassification error, to that from the classification tree in Exercise 9.14b.

(c) The chemical structure–activity data in Exercise 9.15. Start with the training data and vary the cost parameter over $C = 10^{-2}, 10^{-1}, 10^0, 10^1, \ldots, 10^5$. Choose that C for which the associated misclassification error is a minimum; report the associated value for γ. Find the resulting confusion matrices for the training set and the test set. Compare these, along with the associated accuracies and misclassification errors, to those from the classification tree results in Exercise 9.15.

9.19 Return to the oak tree disease data in Example 9.5.1, and for illustrative purposes, apply the following kernels from Table 9.12. Use the suggested cost and kernel parameters as indicated. Find the confusion matrices for both the training set and the test set, along with the associated accuracies, misclassification rates, and sensitivities. Comment on any differences from the results seen in the example.

(a) The homogeneous polynomial kernel with $\gamma = 1$ and degree $d = 2$. Set $C = 100$.

(b) The inhomogeneous polynomial kernel with $\phi_o = \gamma = 1$ and degree $d = 2$. Set $C = 10^{-4}$.

(c) The Laplace RBF kernel. In **R**, use automated estimation for γ via the `kpar='automatic'` option within the `ksvm()` function. Set $C = 10^3$ and report the estimated value of γ.

(d) The sigmoid kernel with $\phi_o = \gamma = 1$. Set $C = 1$.

9.20 Prove the assertion in Section 9.5.2 that $\sum_{i=1}^{n} \xi_i/n$ is an upper bound on the misclassification error $\sum_{i=1}^{n} I_{(1,\infty)}(\xi_i)/n$.

10

Techniques for unsupervised learning: dimension reduction

10.1 Unsupervised versus supervised learning

In contrast to the supervised learning procedures presented in Chapters 6–9, the *unsupervised learning* paradigm applies when only input/feature data are available; no formal 'output' variables are derived from the inputs. In effect, this is 'learning without a teacher' where the discovery process is self-organized, and learning occurs wholly from the input information (Kantardzic 2003, Section 4.3). In effect, the goals shift to the discovery of hidden or latent structure among unlabeled groupings, classes, or clusters.

For unsupervised learning, the inputs are themselves treated as the observations. Often they are multivariate in nature, with data recorded on $p > 1$ different variables of interest, $x_1, x_2, \ldots, x_p$. When p is very large, the terminology often refers to 'high-dimensional data analytics.' However, the methods described in the following will apply to any $p > 1$.

Suppose each variable is measured on a sample or training set of n subjects, producing individual column vectors $\mathbf{x}_j = [x_{1j}\ x_{2j}\ \ldots\ x_{nj}]^{\mathrm{T}}$ $(j = 1, \ldots, p)$, and collected into the data matrix $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_p]$. (Unsupervised learning calculations rely heavily on matrix operations; where needed, see the vector and matrix refresher in Appendix A.) It will be useful to center the feature/input variables by their means $\overline{x}_j = \sum_{i=1}^{n} x_{ij}/n$, producing the corrected vectors $\mathbf{x}_j - \overline{x}_j\mathbf{h}$, where $\mathbf{h} = [1\ 1 \cdots 1]^{\mathrm{T}}$ is a vector of ones. Next, arrange these into the column matrix of (mean-corrected) values

$$\mathbf{X}^* = [\mathbf{x}_1 - \overline{x}_1\mathbf{h}\ \cdots\ \mathbf{x}_p - \overline{x}_p\mathbf{h}] = \begin{bmatrix} x_{11} - \overline{x}_1 & \cdots & x_{1p} - \overline{x}_p \\ x_{21} - \overline{x}_1 & \cdots & x_{2p} - \overline{x}_p \\ \vdots & \ddots & \vdots \\ x_{n1} - \overline{x}_1 & \cdots & x_{np} - \overline{x}_p \end{bmatrix} \tag{10.1}$$

as in (A.11).

Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery, First Edition. Walter W. Piegorsch.

www.wiley.com/go/piegorsch/data_analytics

Despite the unsupervised aspect of the analysis, statistical models are still required for describing the characteristics of $\mathbf{X}$ or $\mathbf{X}^*$. It is under these models that the original observations are analyzed for their underlying structure, and features are extracted. In this chapter, analytic procedures are described for the problem of dimension reduction. (Chapter 11 presents methods of unsupervised learning for the alternative issues of clustering and association.) Dimension reduction aims to abridge the p observed variables into a smaller, more manageable group from which latent structures may be more apparent. As a first step, this can be accomplished simply by studying each set of variables and applying domain-specific knowledge. For example, if a study includes the variables $x_1 =$ weight (lbs) and $x_2 =$ height (in.) among the recorded variables, removal of one or the other may be useful because these are often highly correlated/multicollinear, as in Section 7.1.4. (Or perhaps, replacement with a single integrating variable such as the body mass index (BMI), $x_3 = \text{BMI} = 703.07x_1/x_2^2$, would be appropriate instead.) Once this level of domain knowledge is exhausted, however, statistical procedures can be applied to possibly reduce the dimension of the problem further and advance the discovery process.

10.2 Principal component analysis

In practice, one often encounters substantial intercorrelation and multicollinearity among an original set of inputs $x_1, x_2, \ldots, x_p$, producing overlapping information (as in the height-vs-weight example above). The focus of the dimension-reduction effort involves reduction of the variables under consideration to a smaller number, $q < p$, of representative indices or consolidating measures. We wish to retain as much of the original information in the x_js as possible, but with no intercorrelation among the new variables.

One of the oldest and most popular methods of statistical dimension reduction based on correlations among the data is known as *principal component analysis* (PCA). The technique was presented by Pearson (1901) and also given independently by Hotelling (1933), who developed the implementation in greater detail and from whom the name apparently derives. For a short historical review, see the introduction in Abdi and Williams (2010).

10.2.1 Principal components

As a device for reducing dimension, PCA transforms the original x_js, or more typically their mean-centered versions in $\mathbf{x}^* = [x_1 - \bar{x}_1 \cdots x_p - \bar{x}_p]^{\mathrm{T}}$, to a reduced set of variables $y_1, y_2, \ldots, y_q$. The strategy is to calculate y_j as a linear combination, $y_j = \mathbf{u}_j^{\mathrm{T}}\mathbf{x}^*$, for some suitably chosen coefficient vector $\mathbf{u}_j = [u_{1j}\, u_{2j} \cdots u_{pj}]^{\mathrm{T}}$, $j = 1, \ldots, q$. The y_js are formulated to provide the maximum amount of explainable variation for describing information in the x_j^*s, while also exhibiting zero correlation. Note, however, that one can always increase variation in y_j simply by multiplying $\mathbf{u}_j$ by a constant greater than 1. Thus some additional constraint on the maximization is required. A standard choice is to normalize the coefficients such that their sum of squares is 1: $\mathbf{u}_j^{\mathrm{T}}\mathbf{u}_j = 1$.

The transformed linear combinations, $y_j = \mathbf{u}_j^{\mathrm{T}}\mathbf{x}^*$, produced in a PCA are known as the *principal components* (PCs). These are constructed in a stepwise manner:

- The first PC, $y_1 = \mathbf{u}_1^{\mathrm{T}}\mathbf{x}^*$, is chosen to maximize explainable variation under the constraint that $\mathbf{u}_1^{\mathrm{T}}\mathbf{u}_1 = 1$.

- Next, the second PC, $y_2 = \mathbf{u}_2^{\mathrm{T}}\mathbf{x}^*$, is taken to maximize the remaining variation such that $\mathbf{u}_2^{\mathrm{T}}\mathbf{u}_2 = 1$ *and* such that y_1 and y_2 are uncorrelated. This is achieved by imposing the additional constraint that $\mathbf{u}_1^{\mathrm{T}}\mathbf{u}_2 = 0$ (an 'orthogonality' restriction).
- The third PC is $y_3 = \mathbf{u}_3^{\mathrm{T}}\mathbf{x}^*$ where $\mathbf{u}_3^{\mathrm{T}}\mathbf{u}_3 = 1$ and $\mathbf{u}_1^{\mathrm{T}}\mathbf{u}_3 = \mathbf{u}_2^{\mathrm{T}}\mathbf{u}_3 = 0$.
- The process continues with $y_4 = \mathbf{u}_4^{\mathrm{T}}\mathbf{x}^*$, and so on, until a sufficient number, q, of y_js have been constructed. Each y_j explains a decreasing amount of variation, while being mutually uncorrelated with all that came before.

To visualize the matter, consider the following illustration with $p = 2$.

Example 10.2.1 PCs in two dimensions: husbands' and wives' ages. Similar to the observations on husbands' and wives' heights in Exercise 3.12, additional data in that study were recorded on the couples' ages in years. (Ages were not recorded for some wives, so the data are restricted to $n = 170$ couples.) Let x_1 be the wives' ages and x_2 be the husbands' ages. The data appear in Table 10.1. (As previously, a selection of only the smallest and largest measurements is given in the table. The complete data are available at http://www.wiley.com/go/piegorsch/data_analytics.) For a study such as this with $p = 2$, one would not normally consider a PCA to reduce two variables to a single, summary, linear combination. (Although, it is not unheard-of.) The example can help with illustrating the concepts, however.

Figure 10.1 plots the husbands' ages against the wives' ages on the original, uncentered scale. The plot indicates a tight, increasing trend and a high, positive correlation: $r = 0.9386$ from (3.9). As might be expected, younger women tend to marry younger men, and older women tend to marry older men. Superimposed on the scatterplot are the two PCs: the largest (first) is the linear combination of x_1 and x_2 that maximizes the variance of its projection on to the (x_1, x_2) plane, while the smallest (here, the second) 'sweeps' up the remaining variation. Notice that the two PC lines are perpendicular, that is, orthogonal in the plane. This is expected, because the two variables are constructed to be uncorrelated.

In the figure, the lengths of the PC lines are scaled relative to the component standard deviations to represent the information they provide: the first PC line is far longer and stretches across a more substantial portion of (x_1, x_2) space. The limited amount of information in the second PC direction indicates that little additional, uncorrelated information is being supplied once the first PC has been constructed. Were the goal to find a single measure that captured maximal information as represented by variation in the data cloud, the first PC would be the solution. □

For most applications of PCA, the original data typically represent a variety of different outcomes which, for that matter, often exist across different, arbitrary measurement scales (e.g., length as inches or centimeters). Such differences will affect the nature of the variances and covaraince/correlations in the original data and thus impact the results of the PCA.

Table 10.1 Selected data pairs with $x_1 = \{\text{Wife's age}\}$ and $x_2 = \{\text{Husband's age}\}$, from a larger set of $n = 170$ married couples.

x_1 = Wife's age (years)	18	21	21	21	22	23	···	64	64
x_2 = Husband's age (years)	20	20	22	27	20	31	···	63	64

Source: Hand et al. (1994, Section 231).

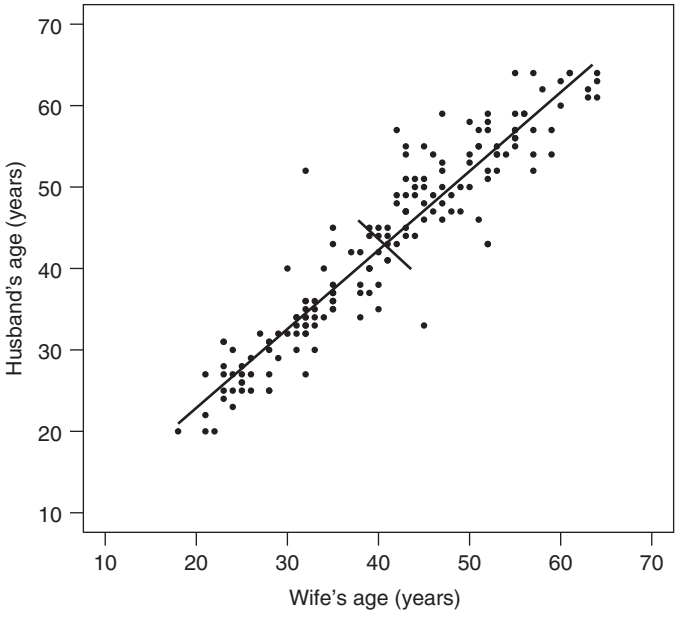

Figure 10.1 Visualization of principal component lines from Example 10.2.1 for husband-wifes age data in Table 10.1. The longer line represents the first principal component and the shorter line represents the second (and last) principal component. Source: Data from Hand et al. (1994).

Indeed, if some of the x_j^*s have particularly large variances, these will tend to dominate the initial PCs (Everitt 2005, Section 3.2). This may or may not lead to difficulties when interpreting results of the final PCA, but it nonetheless warrants attention. To avoid vagaries with wildly differential variances and problems with arbitrary units of measurement, a modification usually applied in practice scales each $\mathbf{x}_j^*$ to have unit variance before constructing the PCs, similar to previous suggestions in Section 7.4. That is, the original x_{ij}s are standardized to z-scores as in (3.7): $z_{ij} = (x_{ij} - \overline{x}_j)/s_j = x_{ij}^*/s_j,\ i = 1, \ldots, n; j = 1, \ldots, p$. (Some authors warn that the standardization treats every original variable with a kind of equal 'attention,' which is itself an arbitrary determination. If the scaling robs certain variables of a stature that is important to the analysis, then the decision to standardize the observations should be reconsidered.) Unless otherwise indicated, the default here is to operate with the standardized data z_{ij}. These then make up a target data matrix $\mathbf{Z}$ with elements z_{ij}.

10.2.2 Implementing a PCA

To calculate the PCs as described earlier, the method essentially implements sequenced, constrained optimization. The constraints – sums of squares equals 1, mutual orthogonality – apply to the coefficient vector $\mathbf{u}$. This can be achieved with an optimization method known as *Lagrange multipliers* (Hughes-Hallett et al., 2013, Section 15.3), similar to the construction in Section 9.2. As there, the technical details exceed our scope; interested readers may refer, for example, to Clarke et al. (2009, Section 9.1). The solution employs the eigenvalues and

eigenvectors of the sample correlation or covariance matrices. (For a refresher on eigenanalysis, see Section A.5.) Note that because $\mathbf{Z}$ contains the standardized data, its covariance matrix is identical to its correlation matrix and simplifies to $\mathbf{R} = \mathbf{Z}^{\mathrm{T}}\mathbf{Z}/(n-1)$ (see Section A.7).

The first PC is built using $\mathbf{u}_1$ as the orthonormal eigenvector corresponding to the first (largest) eigenvalue, λ_1, of $\mathbf{R}$. (If working instead with the centered data in $\mathbf{X}^*$, use the covariance matrix $(\mathbf{X}^*)^{\mathrm{T}}\mathbf{X}^*/(n-1)$.) The next PC is built with $\mathbf{u}_2$ as the eigenvector corresponding to the next-largest eigenvalue, λ_2, of $\mathbf{R}$. The process continues: build the jth PC with $\mathbf{u}_j$ as the eigenvector corresponding to the jth eigenvalue λ_j of $\mathbf{R}$, and so on.

Since the correlation matrix for $\mathbf{Z}$ is real valued and symmetric, its eigenvalues and eigenvectors can be found via appeal to the singular value decomposition (SVD) from Section A.6.4 or, equivalently here, the spectral decomposition from Section A.6.2. These return the (ordered) eigenvalues in a diagonal matrix $\mathbf{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_p\}$, along with their corresponding eigenvectors in a $p \times p$ orthogonal matrix $\mathbf{U} = [\mathbf{u}_1\mathbf{u}_2 \cdots \mathbf{u}_p]$. The PCs are then calculated as linear combinations of the standardized variables: $y_1 = \mathbf{u}_1^{\mathrm{T}}\mathbf{z}$, $y_2 = \mathbf{u}_2^{\mathrm{T}}\mathbf{z}$, $\ldots$, $y_p = \mathbf{u}_p^{\mathrm{T}}\mathbf{z}$. This is sometimes called the Hotelling transformation, or also the Karhunen–Loéve transformation, of $\mathbf{z}$. For computing purposes, R can conduct a PCA via its `prcomp()` function (see Example 10.2.2). The alternative `princomp()` function may also be used, although it can be numerically less accurate because it does not take advantage of the SVD.

Beyond supplying constituent calculations for each PC, this eigendecomposition possesses other useful features. In particular, the variance of the jth PC is equal to the jth eigenvalue λ_j. With this, the total variation in the sample is equal to the sum of all p eigenvalues: $\lambda_+ = \sum_{j=1}^{p} \lambda_j = p$. (As the data are standardized, the sample variance of each variable is always 1. Thus the sum of all p variances is simply p. If working instead with the centered data in $\mathbf{X}^*$, the sum of the corresponding eigenvalues will equal $s_1^2 + s_2^2 + \cdots + s_p^2$.) Thus the percentage variance explained by the jth PC is λ_j/λ_+. This can be helpful in selecting the final PCs for dimension reduction.

Another feature of the eigendecomposition is that the eigenvectors $\mathbf{u}_j$ are forced to satisfy $\mathbf{u}_j^{\mathrm{T}}\mathbf{u}_j = 1$. Thus their individual coefficients must lie between -1 and 1. This allows the kth coefficient u_{kj} in the jth PC y_j to quantify the amount *and direction* the kth original, standardized variate z_k contributes to y_j. Sometimes called *loadings* of each PC (although this invites confusion with similar terms in factor analysis from Section 10.3), these coefficients are useful for interpreting how the original data enter into the PCs. Coefficients near zero indicate limited contributions by the corresponding z_ks, while coefficients near ± 1 indicate strong contributions. Differences in sign also provide contrast: variables with different signs contribute with opposite effect to the overall component.

Notice that as constructed, the decomposition returns a full complement of p PCs y_j. For the purposes of dimension reduction, only the first q PCs of this collection are retained. The available reduction can be substantial: obviously, some information will be lost for $q < p$, but hopefully a large percentage of the explainable variation from the original data will be contained in the few retained PCs.

A number of different strategies have evolved to formalize selection of q. One can

- Choose only those PCs whose λ_j exceeds the average $\bar{\lambda} = \lambda_+/p$. When using the standardized data in $\mathbf{Z}$ to perform the PCA, $\lambda_+ = p$ so $\bar{\lambda}$ is just 1. Thus, one would retain components whose eigenvalue exceeds 1 (Kaiser 1958), because they represent transformations whose variances exceed those of any original (standardized) variables.

For more-accurate retention in practice, however, Jolliffe (1972) recommended dropping the cutoff to about 0.7.

- Choose q such that the total retained percentage variation, $\sum_{j=1}^{q} \lambda_j / \lambda_+$ exceeds a large, pre-set value. The pre-set is typically in the range 70–90%, with 80% a popular midpoint.
- Plot the λ_js against j to visualize the decrease in explainable variation. If, as is common, the plot flattens after a certain index j, a diminishing return in accumulated variation occurs at that point, often called an 'elbow' or 'knee,' in the plot. The remaining components beyond that j are discarded. The graphic is known as a *scree plot* – associating the flat portion of the plot to flattened rubble at the bottom of a sharply sloping mountain – after a suggestion by Cattell (1966).

No single one of these criteria is uniformly favored in the literature, and analysts often combine them informally for final selection of the retained PCs.

Applied to the data, the PCA produces PC scores for each observation when calculated from the retained q components, as linear combinations of the rows of $\mathbf{Z}$:

$$\mathbf{Y} = [\mathbf{y}_1\ \mathbf{y}_2 \cdots \mathbf{y}_q] = \mathbf{Z}[\mathbf{u}_1\ \mathbf{u}_2 \cdots \mathbf{u}_q].$$

To visualize the extant relationships between the PCs, one can plot the scores pairwise via, for example, a scatterplot matrix (see Section 4.2.3) or perhaps build 3D plots of various triplets. Indeed, another aim of the PCA is to build informative graphics of the interrelationships as an aid to feature extraction; the first few PC scores often provide a useful low-dimensional mapping of how the data points relate to one another (Everitt 2005, Section 3.3).

Example 10.2.2 PCA for California natural hazards. To study the socioeconomic effects of natural hazards – floods, wildfires, and so on – social and environmental scientists collect various forms of data after such events. An important endpoint is property losses, that is, financial losses to property after a natural hazard such as a flood impacts a community.

Property loss data from a variety of natural hazard events are available at the Spatial Hazard Events and Losses Database for the US (SHEDLUS): http://www.sheldus.org (Hazards & Vulnerability Research Institute 2013). Table 10.2 presents selected SHELDUS data on economic losses (in \$100000) from 3509 hazard events over the period 2000–2012 in the US state of California, collected at the per-county level. (As previously, only a selection of the data is given in the table. The complete set of data is available at http://www.wiley.com/go/piegorsch/data_analytics.) The table lists losses from $p = 7$ variables corresponding to a variety of common hazards. (Because geographic considerations affect a county's potential for loss, the data are somewhat sparse. For example, losses due to coastal/tsunami events are not evident – or likely – for interior counties such as Alpine or Yuba. Also, earthquakes losses are not included, because they represent an extraordinary, and uncommon, form of economic impact.)

To explore potential interrelationships among the seven hazards as relates to property losses, consider a PCA analysis on the $p = 7$ variables. (Not surprisingly, the \$-losses skew heavily to the right, so we conduct the analysis on $x_j = \log_{10}\{j\text{th county's loss}\}$, which remain positive throughout the data. A zero reported loss is still treated as zero.)

As for any multiple-variable analysis, the first step is to plot the data. Figure 10.2 displays a scatterplot matrix of the log-transformed values, where a variety of patterns appear between the pairs of variables. To help to quantify this, Table 10.3 gives the corresponding correlation

Table 10.2 Selected data on property losses (in $100 000) from specific natural hazards in California during the period 2000–2012, from a larger set across all 58 California counties.

	County						
Hazard variable	Alameda	Alpine	···	Los Angeles	···	Yolo	Yuba
Flood	176.0	10.0	···	13.10	···	0.04	98.06
Landslide	63.97	0.005	···	–	···	–	–
Wind	24.22	13.71	···	0.50	···	0.10	0.09
Wildfire	–	1250.0	···	350.03	···	–	–
Severe storm	0.65	–	···	250.0	···	0.22	0.16
Coastal	0.13	–	···	–	···	–	–
Winter weather	0.29	2.40	···	–	···	0.57	0.03

Landslide includes avalanche losses; wind includes tornado losses; severe storm includes hail losses; coastal events include tsunami losses. Dashes indicate no losses reported.
Source: http://www.sheldus.org.

Table 10.3 Correlation matrix, **R**, for $\log_{10}$ hazard loss data in Table 10.2; upper triangular portion only (lower triangle is transpose of upper triangle).

	Flood	Landslide	Wind	Wildfire	Storm	Coastal	Winter
Flood	1	0.466	0.283	−0.043	0.327	0.370	0.397
Landslide		1	0.582	0.045	0.119	0.665	0.245
Wind			1	−0.031	0.099	0.342	0.266
Wildfire				1	0.232	−0.119	0.230
Storm					1	0.169	0.253
Coastal						1	0.144
Winter							1

matrix found in **R** using the `cor()` function. Moderate correlations are seen throughout most of the table, the highest of which is between Landslide and Coastal (log) losses at $r_{26} = 0.665$. Some correlations are close to zero and, interestingly, only a few negative correlations appear (all with the Wildfire variable; see the following text).

To conduct the PCA, we standardize the log-transformed values into z-scores. In **R**, this can be managed automatically in the `prcomp()` function by including its `scale.=TRUE` option:

```
> prcomp( Xmatrix, center=TRUE, scale.=TRUE )
```

where `Xmatrix` is the original log-transformed data matrix. The consequent output is summarized in Table 10.4. For purposes of dimension reduction, the upper portion of the table gives the ordered eigenvalues, λ_j. The first two exceed 1.0, which by Kaiser's criterion qualifies their corresponding PCs, PC_1 and PC_2, for retention. Jolliffe's adjustment drops this cut-off to 0.7, which would then retain PC_3 and PC_4 as well. Indeed, the next two eigenvalues, $\lambda_3 = 0.88$ and $\lambda_4 = 0.78$, bring the cumulative explainable variation up past 80%, also arguing for retention. The scree plot, given in Figure 10.3, is less generous, however. A clear 'elbow' in the plot is evidenced at $j = 3$, suggesting that any PCs past the first three may be of limited value.

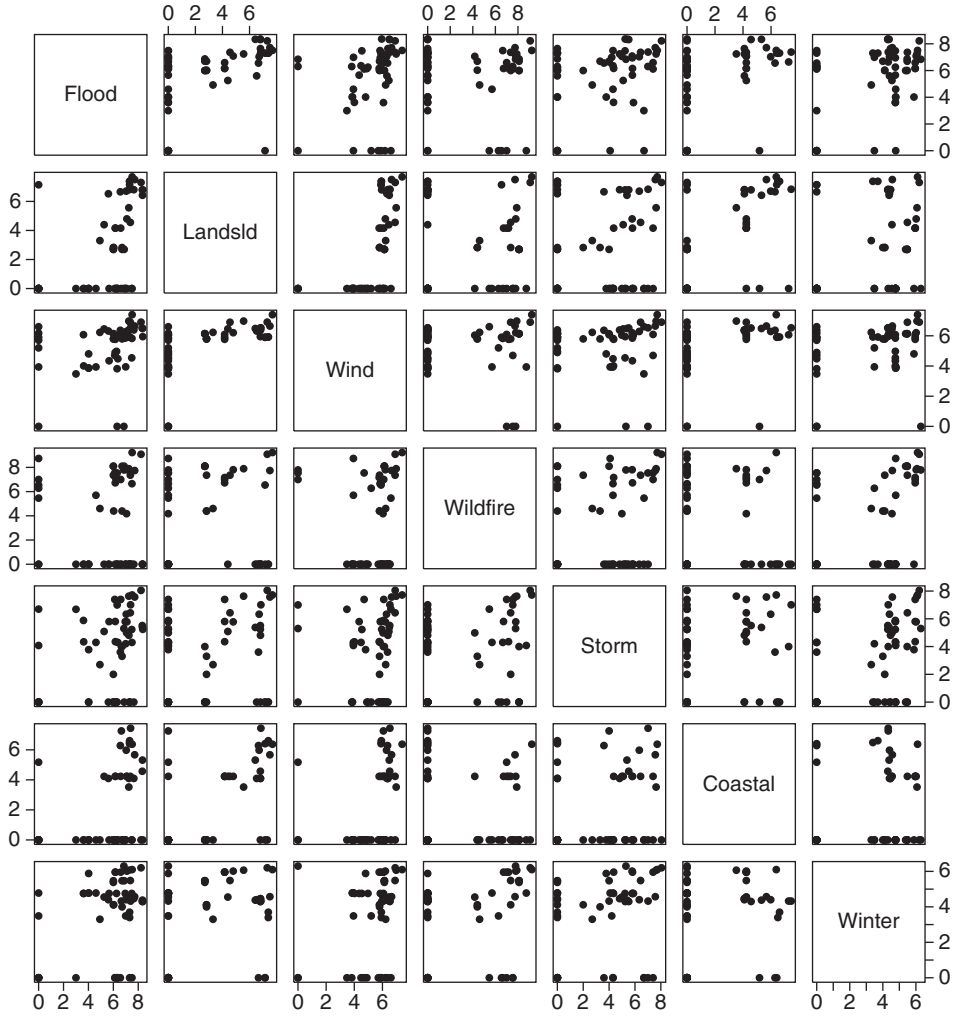

Figure 10.2 Scatterplot matrix for full $\log_{10}$ hazard loss data from Table 10.2. Originating variable names listed along diagonal. Source: Data from http://www.sheldus.org.

Suppose we center our attention on the first three PCs. (Exercise 10.1 explores further graphical visualizations with this PCA output.) In particular, Figure 10.4 gives a 3D drop plot of the three PC scores (constructed in **R** using the `plot3d()` function from the external *rgl* package). One highlight in the plot is the point for Trinity county, which 'distances' itself above the rest on the third component scale. Trinity is a rural, mountainous county in northern California, set inland to the east of coastal Humboldt county. It has no incorporated cities, few major roads, and, if informal sources are to be believed, no traffic lights! Its 2000–2012 property losses were confined to only five separate events (out of the 3509 for all of California), the most severe of which was a 2005 landslide that led to almost $14M in losses. (Next was a 2001 wildfire that incurred $3.5M in losses. From Table 10.4, we see that these two hazards are positively weighted by the third PC.) Otherwise, the county was one of the least affected by

Table 10.4 Results from principal component analysis (PCA) on $\log_{10}$ hazard loss data based on Table 10.2.

	PC_1	PC_2	PC_3	PC_4	PC_5	PC_6	PC_7
Eigenvalues (ordered)							
Variance, λ_j	2.66	1.37	0.88	0.78	0.61	0.48	0.22
Percentage of explainable variation	38.0	19.6	12.5	11.1	8.8	6.8	3.2
Cumulative percentage of variation	38.0	57.6	70.1	81.3	90.0	96.8	100.0
Coefficients ('loadings')							
Flood	−0.441	0.082	−0.449	0.251	0.225	0.656	−0.229
Landslide	−0.514	−0.221	0.256	−0.192	0.143	0.139	0.738
Wind	−0.414	−0.186	0.420	0.181	−0.687	0.084	−0.326
Wildfire	−0.044	0.636	0.559	−0.342	0.283	0.247	−0.216
Storm	−0.251	0.490	−0.483	−0.414	−0.484	−0.180	0.160
Coastal	−0.441	−0.303	−0.077	−0.409	0.360	−0.441	−0.465
Winter	−0.331	0.416	0.070	0.642	0.193	−0.504	0.094

natural hazards in all of California during the period under study. The potential for knowledge discovery here is intriguing: does Trinity county's geography and rural status singularly affect its susceptibility to losses from natural hazards? Further investigation may be warranted.

The coefficients from Table 10.4 can provide insights on how variables enter into the PCs. For example, the first PC incorporates essentially all the variables in roughly equal manner – note the constant sign among the eigenvector's elements – except for the Wildfire variable, which receives a smaller coefficient. Thus PC_1 appears to be a general 'non-fire' hazard indicator. The second PC picks up the Wildfire variable – with a large positive coefficient – and then includes the other variables in a mixed manner. This second PC is a good example for not overinterpreting small differences between coefficients (Everitt 2005, Section 3.3); past the strong coefficient for Wildfire, interpreting how the other variables enter here hinges on the analyst's reading. The third PC is similar: the Wildfire coefficient is large and positive, followed by Wind and Landslide. By contrast, Storm and Flood have nontrivial negative coefficients. PC_3 may be some form of land-vs-water contrast here, although domain-expert interpretations may differ. □

Besides its use for dimension reduction and summary visualization with complex multivariate data, PCA can also be employed as a springboard to more-advanced analyses. For example, in so-called *sparse PCA* certain coefficients are driven to zero – making the eigenvector matrix more 'sparse' – in order to highlight and help to interpret the contributions of higher-impact variables (Hastie et al. 2009, Section 14.5.5). (The external *PMA* package can perform sparse PCA in **R**, via its `SPC()` function.) Another application, mentioned previously in Section 7.1.4, is PC regression. This employs the retained PC scores as a reduced set of regressors in a multiple linear regression for modeling or predicting the outcomes of an associated response variable (Hastie et al. 2009, Section 3.5.1). In-depth discussions on these uses and other features of PCA are available in the various sources cited throughout this section or in the textbook by Jolliffe (2002).

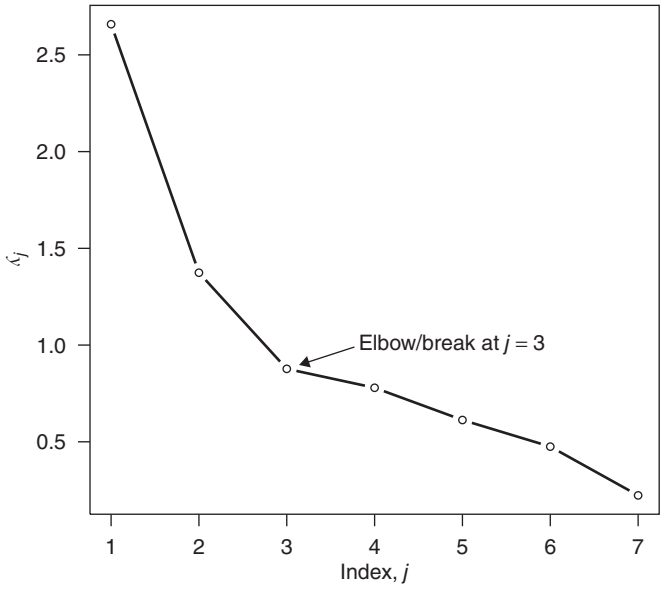

Figure 10.3 Scree plot of ordered eigenvalues, λ_j, from PCA in Table 10.4.

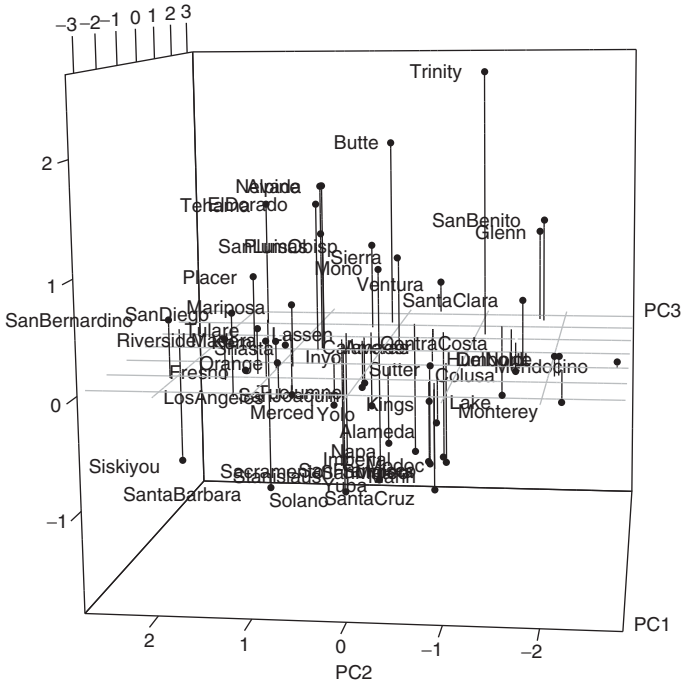

Figure 10.4 Three-dimensional visualization of first three principal component (PC) scores from PCA in Table 10.4.

10.3 Exploratory factor analysis

Similar to the analysis of PCs, and sometimes confused with it, is the method of *exploratory factor analysis* (EFA). As presented in Section 10.2, PCA is a dimension-reduction technique built essentially from algebraic and geometric principles. It is an effective way to reduce a large set of variables to a smaller grouping and, in the process, explore some of the geometric relationships extant among them. PCA makes very few statistical assumptions on that set of variables, however. By contrast, EFA employs a formal statistical *model*, the consequences of which can be exploited for knowledge discovery. As with PCA, implicit in the method is the reduction in dimension among the original variables; however, its goals and interpretations differ.

10.3.1 The factor analytic model

To construct the EFA model, begin with a set of p random variables Y_j, $j = 1, \ldots, p$, and assume that the variables have been centered about their mean so that $E[Y_j] = 0$. As with the PCA approach in Section 10.2, it is common to further standardize the data into z-scores via division by their standard deviations as in (3.7). Thus, we work with the standardized variables Z_j such that $E[Y_j] = 0$ and $\text{Var}[Z_j] = 1$ for all j. (Standardization is not required here, however, and one could operate simply with the centered variables.)

The factor analytic concept hinges on the existence of a *latent variable*: an unobservable random factor, F_k, whose effects are captured by the observed (often called 'manifest') variables Z_j. The model relates the p manifest variables to a reduced set of $q < p$ latent factors via

$$Z_j = b_{j1}F_1 + b_{j2}F_2 + \cdots + b_{jq}F_q + \epsilon_j \tag{10.2}$$

($j = 1, \ldots, p$), where F_k represents the kth latent factor ($k = 1, \ldots, q$), the b_{jk}s are coefficients that 'load' each manifest observation Z_j on to the unobserved factor, and ϵ_j is a residual term. These latter quantities serve as a source of variation specific to each Z_j after modeling the F_ks. By contrast, each kth factor F_k represents a source of variation that affects all the observations via its loading coefficients b_{jk}. Dimension reduction occurs in the use of $q < p$ factors to *model* the underlying effects among the p observed variables.

The concepts underlying factor analysis were introduced by Spearman (1904a) for use in psychometric testing with human subjects. In many psychometric applications, measurements are taken on a wide variety of outcomes, but it is often felt that a smaller set of latent factors can explain the observed variation in a more compact manner. Spearman's example involved intelligence testing with school children: test scores on mathematics, English, foreign language, music, and so on were recorded, but it was hypothesized that these all could be explained by a smaller set of factors – in fact, just a single factor representing overall 'intelligence.' Thurstone (1931, 1947) later generalized the model to multiple factors. From those roots, EFA has become a popular, if very specific technique for dimension reduction, and sees wide application in the behavioral, social, economic, and physical sciences.

In matrix terms, we write

$$\mathbf{Z} = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_p \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1q} \\ b_{21} & b_{22} & \cdots & b_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pq} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_q \end{bmatrix}, \quad \text{and} \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_p \end{bmatrix}.$$

With these, (10.2) becomes the matrix expression

$$\mathbf{Z} = \mathbf{BF} + \boldsymbol{\epsilon}. \tag{10.3}$$

Notice that this is a *linear model*, similar in form to the multiple regression models in Section 7.1. As applied here, however, its construction and interpretation are quite different. The goals of an EFA are to *detect structure* – represented by the modeled factors – within the manifest variables, while reducing the dimension of the data. The structure(s) are identified by studying patterns among the loadings in $\mathbf{B}$ (see the following text); feature extraction often results through detection of this factor structure.

Since all three constituents, $\mathbf{B}$, $\mathbf{F}$, and $\boldsymbol{\epsilon}$, in (10.3) are unknown and unobserved, the model is indeterminate and requires further specification to be useful. Typically, both $\mathbf{F}$ and $\boldsymbol{\epsilon}$ are assumed random with zero means; that is, $E[F_k] = E[\epsilon_j] = 0$ for all k and j. The factors are assigned unit variances, $\text{Var}[F_k] = 1$, while the residuals are given variable-specific variances

$$\text{Var}[\epsilon_j] = \psi_j \geq 0.$$

Further, all variables are taken to be uncorrelated such that their covariances are zero: $\text{Cov}[F_k, F_m] = \text{Cov}[\epsilon_j, \epsilon_i] = 0$ for all $k \neq m, j \neq i$, and $\text{Cov}[F_k, \epsilon_j] = 0$ for all k, j. Again in matrix terms, this gives $E[\mathbf{F}] = \mathbf{0}$, $E[\boldsymbol{\epsilon}] = \mathbf{0}$, $\text{Var}[\mathbf{F}] = \mathbf{I}$, and $\text{Var}[\boldsymbol{\epsilon}] = \text{diag}\{\psi_1, \psi_2, \ldots, \psi_p\}$, where each $\mathbf{0}$ is an appropriately dimensioned vector of zeros and $\mathbf{I}$ is the identity matrix. For convenience, denote $\mathbf{D}_{\boldsymbol{\psi}} = \text{diag}\{\psi_1, \psi_2, \ldots, \psi_p\}$. Under these moment assumptions, it can be shown (Exercise 10.5) that $E[Z_j] = 0$ (as designed) and

$$\text{Var}[Z_j] = h_j^2 + \psi_j, \tag{10.4}$$

where $h_j^2 = \sum_{k=1}^{q} b_{jk}^2$.

Equation (10.4) shows that $\text{Var}[Z_j]$ is modeled as a sum of two components. The first, h_j^2, represents the contribution of the common loading coefficients to $\text{Var}[Z_j]$ and is called the *communality* of Z_j. The second, ψ_j, is called the *specific variance* of Z_j, so-named because it gives the variable-specific contribution of each ϵ_j to $\text{Var}[Z_j]$. (One also sees the somewhat awkward term 'uniquenesses' for the ψ_js.) By design, $\text{Var}[Z_j] = 1$. Thus in concert with $\psi_j \geq 0$, (10.4) requires $b_{jk}^2 \leq 1$, that is,

$$-1 \leq b_{jk} \leq 1.$$

(Indeed, (10.4) also gives $h_j^2 = 1 - \psi_j$ for all j.) In addition, the covariance assumptions on $\mathbf{F}$ and $\boldsymbol{\epsilon}$ lead to $\text{Cov}[Z_j, Z_m] = \sum_{k=1}^{q} b_{jk} b_{mk}$ for $j \neq m$ (see Exercise 10.5). Collected together in matrix form, this is

$$\text{Var}[\mathbf{Z}] = \mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{D}_{\boldsymbol{\psi}}. \tag{10.5}$$

As the Z_js are standardized to have unit variance, the correlation between any two is $\text{Corr}[Z_j, Z_m] = \text{Cov}[Z_j, Z_m]/(1)$, using (2.10). Thus the factor-analytic construction from (10.3) produces a prescriptive recipe for modeling the correlation structure of the Z_js, via (10.5). This has substantive consequences. For instance, we see $\text{Corr}[Z_j, Z_m] = \text{Cov}[Z_j, Z_m] = \sum_{k=1}^{q} b_{jk} b_{mk}$ and because $b_{jk}^2 \leq 1$, this correlation will be large (in absolute value) only if the b_{jk}s are themselves large on many of the same factors. Similarly, loadings close to zero indicate limited or possibly meaningless relationships between F_k and Z_j. As a result, loading patterns with selected b_{jk}s near ± 1 and many others near zero offer

very convenient identification of the underlying factors. This, in turn, provides for potential knowledge discovery into the phenomenon under study.

Despite the careful model construction here, there remains an issue of identifiability. Suppose we take a $q \times q$ orthogonal matrix $\mathbf{Q}$ and modify (10.3) into $\mathbf{Z} = \mathbf{B}\mathbf{Q}^{\mathrm{T}}\mathbf{F} + \boldsymbol{\epsilon}$. Then,

$$\mathrm{Var}[\mathbf{Z}] = \mathbf{B}\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\mathbf{B}^{\mathrm{T}} + \mathbf{D}_{\boldsymbol{\psi}} = \mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{D}_{\boldsymbol{\psi}},$$

because $\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}$. Thus despite rotation of the loadings by $\mathbf{Q}^{\mathrm{T}}$, Var[$\mathbf{Z}$] is unchanged. We see that (infinitely) many different loading patterns can be constructed to represent the same correlation structure for $\mathbf{Z}$. This is both a blessing and a curse: it affords the analyst great flexibility to modify a given collection of loadings by rotating them into $\mathbf{B}\mathbf{Q}^{\mathrm{T}}$. One can search repeatedly for as clear an explication as possible on how the factors affect $\mathbf{Z}$. The definition of 'clear' is open to interpretation, however, and this subjectivity can easily be overdone, even abused. Indeed, the 'E' in EFA partly represents this exploratory aspect – also see the following text – and can make the technique seem as much an art as a science. This leads some analysts to view it with skepticism.

10.3.2 Principal factor estimation

To utilize the factor-analytic model (10.3) in practice, estimates must be determined for the components of Var[$\mathbf{Z}$] in (10.5). Assume the number of factors is fixed at some $q < p$. The aim is to employ information in the sample covariance or correlation matrix to estimate $\mathbf{B}\mathbf{B}^{\mathrm{T}}$ and $\mathbf{D}_{\boldsymbol{\psi}}$. Write these latter estimators as $\hat{\mathbf{B}}\hat{\mathbf{B}}^{\mathrm{T}}$ and $\hat{\mathbf{D}}_{\boldsymbol{\psi}}$, respectively. If $\mathbf{Z}_i = [Z_{i1}\, Z_{i2} \cdots\, Z_{ip}]^{\mathrm{T}}$ is the ith vector of observations from a p-variate random sample ($i = 1, \ldots, n$), then the sample correlation matrix from Section A.7 is $\mathbf{R} = \frac{1}{n-1}\sum_{i=1}^{n} \mathbf{Z}_i\mathbf{Z}_i^{\mathrm{T}}$.

A common method for estimating the correlation structure mimics the eigenanalysis used in PCA and, hence, is called *principal factor*, or also *principal axis factor*, estimation. (The overlapping nomenclature may contribute to some of the confusion seen between EFA and PCA.) The general strategy is to modify the sample correlation matrix $\mathbf{R}$ by replacing its diagonal elements (all 1) with estimates of the communalities. Many possibilities exist for these replacements, including some clever manipulations of multiple regressions or multiple correlations among the Z_js (Everitt 2005, Section 4.2). One simple approach appeals to the core relationship in (10.5): viewing $\mathbf{R}$ as an empirical estimator for $\mathrm{Var}[\mathbf{Z}] = \mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{D}_{\boldsymbol{\psi}}$, estimate the communalities as $\hat{h}_j^2 = 1 - \hat{\psi}_j$, where the $\hat{\psi}_j$s are taken from the diagonal of the matrix $\mathbf{R} - \hat{\mathbf{B}}\hat{\mathbf{B}}^{\mathrm{T}}$ for some estimate $\hat{\mathbf{B}}\hat{\mathbf{B}}^{\mathrm{T}}$.

Now, to find $\hat{\mathbf{B}}\hat{\mathbf{B}}^{\mathrm{T}}$, begin with the spectral decomposition of $\mathbf{R}$ but retain only the largest q eigenvalues, along with their corresponding unit eigenvectors. Denote these as λ_k and $\mathbf{u}_k$ ($k = 1, \ldots, q$), respectively. Clearly, if $q = p$, this reproduces the full PCA decomposition, in effect forcing $\psi_j = 0$. For $q < p$, however, collect this reduced set of eigencomponents into the column matrix $[\mathbf{u}_1\sqrt{\lambda_1}\ \ \mathbf{u}_2\sqrt{\lambda_2}\cdots\mathbf{u}_q\sqrt{\lambda_q}]$ and view the reduced spectral decomposition

$$\left[\mathbf{u}_1\sqrt{\lambda_1}\ \ \mathbf{u}_2\sqrt{\lambda_2}\cdots\mathbf{u}_q\sqrt{\lambda_q}\right]\left[\mathbf{u}_1\sqrt{\lambda_1}\ \ \mathbf{u}_2\sqrt{\lambda_2}\cdots\mathbf{u}_q\sqrt{\lambda_q}\right]^{\mathrm{T}} = \sum_{k=1}^{q}\lambda_j\mathbf{u}_k\mathbf{u}_k^{\mathrm{T}}$$

as the initial estimate $\hat{\mathbf{B}}\hat{\mathbf{B}}^{\mathrm{T}}$ (cf. Section A.6.4). From this, recover $\hat{\psi}_j$ from the diagonal of $\mathbf{R} - \hat{\mathbf{B}}\hat{\mathbf{B}}^{\mathrm{T}}$.

We usually iterate this process: replace the recovered estimates $1-\hat{\psi}_j$ into the corresponding diagonal elements of $\mathbf{R}$ and recompute the spectral decomposition for this updated $\mathbf{R}$. Continue until some prespecified convergence criterion is achieved, for example, iterate until tr($\mathbf{R}$) (the sum of $\mathbf{R}$'s diagonal elements; see Section A.1) stabilizes over successive updates. For computing purposes, this form of principal factor estimation is available in **R** via the `fa()` function of the external *psych* package (use the factoring option `fm='pa'`).

Once iteration is complete, the estimated percentage variation explained by the kth factor is $\sum_{j=1}^{p} \hat{b}_{jk}^2/p$ (assuming the data have been standardized), where the $\hat{b}_{jk}$s are taken from the final estimate $\hat{\mathbf{B}}$. The estimated communalities are $\hat{h}_j^2 = \sum_{k=1}^{q} \hat{b}_{jk}^2$.

As an iterative procedure, convergence with principal factors is usually rapid. (Indeed, some authors apply it or its variants with only one iteration.) Like any iterative approach, however, it can suffer certain instabilities. One important example occurs when the iterative estimate of a communality, $\hat{h}_j^2 = 1 - \hat{\psi}_j$, exceeds 1. This forces the corresponding specific variance $\hat{\psi}_j$ below its lower limit of zero, which is senseless. Known as 'Heywood cases' (Heywood 1931), the iterative algorithm must identify and adapt to such untoward occurrences, possibly by resetting the initial conditions or applying some other adjustment. For more on Heywood cases, see Dillon et al. (1987) and Kolenikov and Bollen (2012).

10.3.3 Maximum likelihood estimation

Principal factor estimation can be viewed as 'distribution free,' because it makes no assumptions on the particular distributions of $\mathbf{F}$ or $\boldsymbol{\epsilon}$. This lends it a certain level of robustness for use in practice.

If the analyst is willing to make, say, normal (Gaussian) distribution assumptions on F_k and ϵ_j, the full power of parametric likelihood analysis (Section 5.1) can be brought to bear on estimating $\mathbf{B}$ and $\mathbf{D}_{\boldsymbol{\psi}}$. To wit, suppose $[F_k\ \epsilon_j]^{\mathrm{T}}$ are independent bivariate normal vectors with zero means and diagonal covariance matrices diag$\{1, \psi_j\}$, $j = 1, \ldots, p$; $k = 1, \ldots, q$. Assuming the data have been standardized into z-scores Z_{ij} with sample correlation matrix $\mathbf{R} = \frac{1}{n-1}\sum_{i=1}^{n} \mathbf{Z}_i\mathbf{Z}_i^{\mathrm{T}}$, the log-likelihood may be written as

$$\ell(\mathbf{B}\mathbf{B}^{\mathrm{T}}, \mathbf{D}_{\boldsymbol{\psi}}) = C - \frac{1}{2}\{n \log |\mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{D}_{\boldsymbol{\psi}}| + (n-1)tr([\mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{D}_{\boldsymbol{\psi}}]^{-1}\mathbf{R})\},$$

where $|\mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{D}_{\boldsymbol{\psi}}|$ is the determinant of $\mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{D}_{\boldsymbol{\psi}}$ and C is a constant not related to the unknown parameters (Anderson and Olkin 1985). Maximizing $\ell(\cdot)$ produces a maximum likelihood estimate (MLE) for $\mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{D}_{\boldsymbol{\psi}}$. Unfortunately, rotation indeterminacy remains a problem here, and so some additional constraint must be imposed to complete the optimization. For instance, Lawley and Maxwell (1962) required $\mathbf{B}\mathbf{D}_{\boldsymbol{\psi}}^{-1}\mathbf{B}^{\mathrm{T}}$ to be diagonal and then conducted the constrained maximization via computer iteration. Details are available in, for example, Bartholomew et al. (2011, Section 3.4), Clarke et al. (2009, Section 9.2.1), and Lawley and Maxwell's (1962) own early work.

Given MLEs for $\hat{b}_{jk}$ and $\hat{\psi}_j$ from $\hat{\mathbf{B}}\hat{\mathbf{B}}^{\mathrm{T}}$ and $\hat{\mathbf{D}}_{\boldsymbol{\psi}}$, the MLEs for the communalities are $\hat{h}_j^2 = \sum_{k=1}^{q} \hat{b}_{jk}^2$. The estimated percentage variation explained by the kth factor is $\sum_{j=1}^{p} \hat{b}_{jk}^2/p$ (again, assuming the data have been standardized).

One can also appeal to the normality assumptions and construct a likelihood ratio (LR) test for model adequacy. The test compares the likelihood under the given q-factor model to that under a full p-factor model: if the two are essentially indistinguishable – quantified by

an insignificant LR P-value – the q-factor model may be viewed as adequately reducing the dimension of the problem. Conversely, small P-values from the test indicate that the explainable variability provided by the q factors may not be sufficient to supplant all p factors, and so an additional factor (or factors) might be necessary (Everitt 2005, Section 4.3).

The LR test is useful when determining the number of factors is part of the exploratory process – see the next section – because it can guide selection of q. This necessarily involves calculation of repeated P-values on the same set of data, however, and so some correction for multiple testing (Section 5.5) or other adjustment may be appropriate to avoid overfitting or 'overfactoring' the final model (Hayashi et al. 2007). The LR test, along with the full maximum likelihood (ML) estimation procedure, may be conducted in **R** via the `factanal()` function.

Iteration and convergence with the ML approach can be fairly rapid, although the MLEs are more sensitive to Heywood cases (de Winter and Dodou 2012). Analysts will often employ ML factors in concert with principal factors, in order to study if and how the estimators differ between the two methods. Roughly similar results will generate greater confidence in any achieved dimension reduction.

10.3.4 Selecting the number of factors

In practice, the number of factors, q, may not be known a priori, and if so a large portion of the 'exploratory' nature of the EFA involves determining the number, and final nature, of the reported factors. In the former case, a number of possible strategies exists for choosing q. For instance, if employing principal factor estimation as in Section 10.3.2, one could mimic the Kaiser criterion from PCA and set q equal to the number of eigenvalues greater than 1. As there, a factor whose eigenvalue λ_k is below 1 can be viewed as providing less explainable variability than (at least) one of the original variables. Begin with a full p-factor analysis, initially forcing $\psi_j = 0$ for all j. Then, simply find that $q < p$ such that $\lambda_k \geq 1$ for $k = 1, \ldots, q$. Given this q, recompute the EFA with just q factors in the model.

Similarly, one might select q such that a predetermined percentage of variation is provided by the q-factor model. That is, for a fixed percentage $100\pi\%$ $(0 < \pi < 1)$, and assuming the data have been standardized, select q such that

$$\frac{1}{p}\sum_{j=1}^{p}\sum_{k=1}^{q}\hat{b}_{jk}^2 \geq \pi \; . \tag{10.6}$$

(Notice that $\sum_{k=1}^{q}\hat{b}_{jk}^2 = \hat{h}_j^2$ are the estimated communalities under the q-factor model.) Common choices for the percentage threshold are often seen in the 50–80% range, depending on the domain-specific application.

To select q when employing ML factor estimation, it is common to apply the LR test for adequacy mentioned in Section 10.3.3. Begin with, say, $q_o < p$ factors (typical suggestions start at $q_o = 4$, 5 or 6) and apply the LR test. If its corresponding large-sample P-value is below a predetermined significance level, α, increase to $q_o + 1$ factors and retest. Continue until P exceeds α. Conversely, if with q_o factors P is already larger than the desired significance level, drop to $q_o - 1$ factors and retest. Continue reducing by one factor until P is just larger than α. (As mentioned earlier, this ignores the multiplicity inherent in such a repeated testing strategy. Strict false positive error control is not always an issue here; when it is, some form of multiplicity adjustment should also be applied.)

10.3.5 Factor rotation

Recall from Section 10.3.1 that the EFA model suffers (or benefits, depending on one's point-of-view) from a form of rotation indeterminacy: the base model in (10.3) can be modified via an orthogonal rotation matrix $\mathbf{Q}$ into $\mathbf{Z} = \mathbf{B}\mathbf{Q}^{\mathrm{T}}\mathbf{F} + \boldsymbol{\epsilon}$ without changing the core variance relationship $\mathrm{Var}[\mathbf{Z}] = \mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{D}_{\boldsymbol{\psi}}$ in (10.5). Analysts can take advantage of this indeterminacy to rotate a given q-factor model's fit and explore how the new factor loadings change under the chosen rotation. (Again, an element of the 'exploratory' nature in EFA.) If a rotation can be found which loads each variable heavily on as few factors as possible, while drawing most of the other loading coefficients to zero, each variable is connected with only one or a few factors in a mutually exclusive manner. The result is called 'simple structure' (Thurstone, 1947) where, hopefully, domain-intelligent patterns emerge in the factor loadings. Interpretability of the factors' effects is then eased, even enhanced (see Everitt, 2005, Section 4.5). Beyond dimension reduction, an associated goal in EFA is to affect the pattern of rotated loadings to achieve this simplified interpretability.

As might be expected, many rotation strategies exist. For simplicity, continue to let $\hat{b}_{jk}$ represent the final (estimated) factor loadings over the $j = 1, \ldots, p$ variables and $k = 1, \ldots, q$ factors. Perhaps the most popular rotation strategy is known as the *varimax* approach, proposed by Kaiser (1958). As its name suggests, the goal is to maximize the (sample) variance of each factor's squared loadings, summed over all variables: in effect, for fixed q, maximize the objective quantity

$$\mathcal{D}_{\hat{\mathbf{B}}} = \sum_{j=1}^{p} \left\{ \sum_{k=1}^{q} \hat{b}_{jk}^4 - \frac{1}{q} \left(\sum_{k=1}^{q} \hat{b}_{jk}^2 \right)^2 \right\}.$$

When $\mathcal{D}_{\hat{\mathbf{B}}}$ is large, the loading coefficients will migrate towards 1 or 0. Varimax rotation, therefore, aims to produce focused factors with a small number of high loadings and many loadings at or close to zero. It leans away from production of broad, general factors. For computing purposes, **R** offers varimax rotation through a variety of conduits. Simplest perhaps is the `rotation='varimax'` option in `factanal()`, or the `rotate='varimax'` option in `fa()` from the external *psych* package.

Another alternative is *quartimax* rotation (Neuhaus and Wrigley 1954), where variables load heavily on just a single factor and are loaded trivially or not at all on all the others. The `fa()` function from *psych* can also provide quartimax rotation, via its `rotate='quartimax'` option.

If the analyst is willing to abandon the assumption that the factors are uncorrelated, that is, $\mathrm{Cov}[F_k, F_m] = 0$, then the options for rotation widen. The requirement that the factors are uncorrelated was applied in (10.3) more for convenience than any other reason, and for many data-analytic scenarios, its imposition is not strictly required. Rotations that allow factor covariances to vary from zero are called *oblique*, and they produce a wide variety of loading patterns. On the one hand, this adds increased flexibility to the analyst's EFA toolkit; on the other hand, it requires more care in loading interpretation and more complicated bookkeeping to manage the now-nonzero factor correlations. Also, with greater flexibility comes greater opportunity for hyperinterpretation of factor significance. As discussed above, this has led some to debate the value of factor rotation and EFA in general.

One of the most popular oblique rotation strategies is known as *promax* rotation (Hendrickson and White 1964). The method initializes with a varimax rotation and then uses

increasing exponents on the loadings to push the higher coefficients towards 1 while shrinking lower values towards 0. In **R**, promax rotation is performed by `factanal()` via its `rotation='promax'` option, or via the `rotate='promax'` option in `fa()` from the external *psych* package. For more on oblique rotations in EFA, see Everitt (2005, Section 4.3).

10.3.6 Implementing an EFA

As many authors note, calculation of an EFA (or a PCA, for that matter) is of little value if the correlations among the variables are all at or near zero: the operation then essentially returns the original observed variables. A test statistic for whether the sample correlation matrix **R** differs from an identity matrix **I** was given by Bartlett (1950):

$$B_p^2 = -\left\{(n-1) - \frac{1}{6}(2p+5)\right\} \log |\mathbf{R}| \,, \tag{10.7}$$

where $|\mathbf{R}|$ is the determinant of **R**. In large samples, $B^2 \dot{\sim} \chi^2(\frac{1}{2}p\{p-1\})$ so, for example, an approximate *P*-value for testing if **R** differs from **I** is

$$P\left[\chi^2\left(\frac{1}{2}p\{p-1\}\right) \geq B_p^2\right].$$

This is known as *Bartlett's test for sphericity*; Dziuban and Shirkey (1974) discussed its operating characteristics. In **R**, the test for sphericity is available via the `cortest.bartlett()` function in the external *psych* package, or of course the test statistic can be calculated directly via use of the `det()` function to find $|\mathbf{R}|$.

Another index that provides a check for nonzero correlation is known as the *Kaiser-Meyer-Olkin* (*KMO*) *measure*. Proposed by Kaiser (1970) and modified by Kaiser and Rice (1974), the index gauges the amount of departure from zero among the various correlations in the sample. Begin with the sample correlation matrix $\mathbf{R} = \frac{1}{n-1}\sum_{i=1}^{n} \mathbf{Z}_i\mathbf{Z}_i^{\mathrm{T}}$ and its elements r_{jk}. Let $\mathrm{diag}\{\mathbf{R}^{-1}\}$ be a diagonal matrix built from the diagonal elements of $\mathbf{R}^{-1}$. Then, find $\mathbf{Q} = \mathrm{diag}\{\mathbf{R}^{-1}\}^{-1/2}\mathbf{R}^{-1}\mathrm{diag}\{\mathbf{R}^{-1}\}^{-1/2}$, with elements q_{jk}. The KMO index is

$$\frac{\sum\sum_{j\neq k} r_{jk}^2}{\left(\sum\sum_{j\neq k} r_{jk}^2 + \sum\sum_{j\neq k} q_{jk}^2\right)}$$

In practice, index values above about 80% show strong correlations where an EFA can be effective, values between 50% and 80% show reasonable correlations, while values below 50% indicate little value in pursuing the calculations. Dziuban et al. (1979) gave details on the measure and on its operating characteristics.

If the intricate model formulation embodied by (10.3) and (10.5) is felt to be valid with a set of multivariable data, dimension reduction via EFA is a fairly straightforward process, given modern-day computing capabilities. One standardizes the data; plots the variables for visual overview and inspection; checks for valid correlation via, for example, Bartlett's test for sphericity; decides whether (or not) a normal distribution model can be imposed for ML factor (or principal factor) estimation; fixes a value for the number of factors q or uses the strategies described above for estimating it; and applies a rotation if desired to enhance factor

interpretation, structure detection, and possible knowledge discovery. The method can allow for much more in the way of advanced multivariate data analysis (Bartholomew et al. 2011), but these basic steps constitute the standard features of an EFA.

Example 10.3.1 EFA for psychometric test outcomes. A classic data set in psychometric testing presented by Birren and Morrison (1961) concerns responses of $n = 933$ Caucasian adults, aged 25–64, on the Wechsler Adult Intelligence Scale (WAIS). As conducted in that article, the WAIS test reported outcome scores on 11 different cognitive 'subtests' (listed in Table 10.5), scores on which were taken as the variables for an EFA. Two additional standardized variables, $Z_{12} = \{\text{Age}\}$ and $Z_{13} = \{\text{Years of education}\}$, were also included. One question of interest was whether the $p = 13$ variables could be reduced to a smaller number of factors, along with any insights as to how those factors might affect the original variables.

Table 10.6 presents the correlation matrix, $\mathbf{R}$, for these data as given in Birren and Morrison (1961). Most striking is the consistently negative correlation of $Z_{12} = \{\text{Age}\}$ with all the other variables. Many of these values are so close to zero as to likely be insignificant, but for a few at least, increasing age appears to associate with decreasing outcome response.

Suppose the factor analytic model in (10.3) applies, and for convenience, assume a normal (Gaussian) distribution for the standardized data. A check of the correlation structure via Bartlett's test for sphericity gives $B^2_{13} = 7007.2$ on $(13)(12)/2 = 78$ d.f. (degrees of freedom). The P-value here is well below 10^{-4}. This helps to validate application of an EFA to these data.

To estimate $\mathbf{B}\mathbf{B}^{\mathrm{T}}$ and $\mathbf{D}_{\boldsymbol{\psi}}$, apply ML via, for example, the `factanal()` function in **R**. Suppose the correlation matrix from Table 10.6 has been entered as the numeric matrix `Rmx`. Start with $q_o = 5$ factors (and no rotation) via the command

```
> factanal( covmat=Rmx, n.obs=933, factors=5, rotation='none' )
```

The consequent output (not shown) produces an LR test for adequacy with P-value of 7×10^{-5}, indicating a poor fit. Moving then to $q_o = 6$ factors requires the command

```
> factanal( covmat=Rmx, n.obs=933, factors=6, rotation='none' )
```

with consequent LR P-value $P = 0.143$. Thus a six-factor model seems minimally acceptable. Table 10.7 displays the estimated loadings and summary information from the fit.

The loading pattern in Table 10.7 is unclear. Perhaps use of a varimax rotation via

```
> factanal( covmat=Rmx, n.obs=933, factors=6,
                                          rotation='varimax' )
```

can help to improve factor interpretation. Table 10.8 gives the new factor loadings and other summary information from the fit (notice that the communalities $\hat{h}^2_j$ are unchanged,

Table 10.5 Standardized variables, Z_j, from WAIS study in Example 10.3.1.

Z_1 = Information	Z_5 = Digit span	Z_{10} = Picture arrangement
Z_2 = Comprehension	Z_6 = Vocabulary	Z_{11} = Object assembly
Z_3 = Arithmetic	Z_7 = Digit symbol	Z_{12} = Age
Z_4 = Similarities	Z_8 = Picture completion	Z_{13} = Education
	Z_9 = Block design	

Source: Birren and Morrison (1961).

Table 10.6 Correlation matrix, **R**, for WAIS study data in Example 10.3.1; upper triangular portion only (lower triangle is transpose of upper triangle).

Variable	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8	Z_9	Z_{10}	Z_{11}	Z_{12}	Z_{13}
Z_1	1	0.67	0.62	0.66	0.47	0.81	0.47	0.60	0.49	0.51	0.41	−0.07	0.66
Z_2		1	0.54	0.60	0.39	0.72	0.40	0.54	0.45	0.49	0.38	−0.08	0.52
Z_3			1	0.51	0.51	0.58	0.41	0.46	0.48	0.43	0.37	−0.08	0.49
Z_4				1	0.41	0.68	0.49	0.56	0.50	0.50	0.41	−0.19	0.55
Z_5					1	0.45	0.45	0.42	0.39	0.42	0.31	−0.19	0.43
Z_6						1	0.49	0.57	0.46	0.52	0.40	−0.02	0.62
Z_7							1	0.50	0.50	0.52	0.46	−0.46	0.57
Z_8								1	0.61	0.59	0.51	−0.28	0.48
Z_9									1	0.54	0.59	−0.32	0.44
Z_{10}										1	0.46	−0.37	0.49
Z_{11}											1	−0.28	0.40
Z_{12}												1	−0.29
Z_{13}													1

Source: Birren and Morrison (1961).

Table 10.7 Loadings, $\hat{b}_{jk}$; communalities, $\hat{h}_j^2$; and other summary results from ML exploratory factor analysis for WAIS study data in Example 10.3.1.

Variable, Z_j	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6	$\hat{h}_j^2$
Z_1	0.996	−0.015	−0.057	−0.016	<0.001	<0.001	0.995
Z_2	0.690	0.137	0.188	0.269	0.068	−0.120	0.622
Z_3	0.658	−0.227	0.674	−0.034	−0.011	−0.003	0.940
Z_4	0.680	0.243	0.183	0.154	0.006	−0.047	0.580
Z_5	0.492	0.131	0.318	−0.032	−0.100	−0.012	0.371
Z_6	0.829	0.133	0.117	0.421	−0.031	0.022	0.898
Z_7	0.495	0.500	0.287	−0.078	−0.272	0.127	0.673
Z_8	0.620	0.388	0.211	−0.024	0.210	−0.120	0.638
Z_9	0.516	0.418	0.348	−0.135	0.310	0.100	0.687
Z_{10}	0.533	0.433	0.262	−0.021	0.035	−0.185	0.576
Z_{11}	0.432	0.409	0.265	−0.093	0.232	0.173	0.517
Z_{12}	−0.085	−0.567	−0.204	0.362	0.236	0.119	0.571
Z_{13}	0.675	0.255	0.150	0.002	−0.235	0.052	0.602
$\sum_{j=1}^{p} \hat{b}_{jk}^2$	5.121	1.481	1.092	0.441	0.396	0.138	
% variation	0.394	0.114	0.084	0.034	0.030	0.011	
Cumul. % var.	0.394	0.508	0.592	0.626	0.656	0.667	

Table 10.8 Loadings, $\hat{b}_{jk}$; communalities, $\hat{h}_j^2$; and other summary results from ML exploratory factor analysis after varimax rotation for WAIS study data in Example 10.3.1.

Variable, Z_j	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6	$\hat{h}_j^2$
Z_1	0.794	0.256	0.059	0.274	0.470	−0.002	0.995
Z_2	0.682	0.286	0.055	0.249	−0.034	−0.092	0.622
Z_3	0.373	0.245	0.053	0.857	0.049	−0.004	0.940
Z_4	0.625	0.323	0.193	0.217	0.022	−0.015	0.580
Z_5	0.355	0.211	0.260	0.362	0.042	0.025	0.371
Z_6	0.882	0.246	0.019	0.223	−0.032	0.092	0.898
Z_7	0.387	0.298	0.592	0.189	<0.001	0.219	0.673
Z_8	0.471	0.541	0.259	0.157	0.062	−0.167	0.638
Z_9	0.264	0.708	0.251	0.224	0.049	−0.006	0.687
Z_{10}	0.436	0.408	0.402	0.171	−0.024	−0.168	0.576
Z_{11}	0.235	0.614	0.236	0.138	0.033	0.093	0.517
Z_{12}	0.043	−0.201	−0.725	−0.008	−0.016	0.058	0.571
Z_{13}	0.574	0.190	0.374	0.236	0.147	0.138	0.602
$\sum_{j=1}^{p} \hat{b}_{jk}^2$	3.548	1.921	1.477	1.314	0.257	0.152	
% variation	0.273	0.148	0.114	0.101	0.020	0.012	
Cumul. % var.	0.273	0.421	0.534	0.635	0.655	0.667	

as expected). To find the specific variances for each variable, label the output object from `factanal()` and examine its `uniquenesses` attribute:

```
> wais6vmx.fa <- factanal( covmat=Rmx, n.obs=933, factors=6,
                                              rotation='varimax' )
> wais6vmx.fa$uniquenesses
 [1] 0.0050 0.3785 0.0601 0.4197 0.6288 0.1022 0.3270
 [8] 0.3619 0.3130 0.4237 0.4833 0.4287 0.3979
```

In the rotated loadings from Table 10.8, the first factor loads heavily with $Z_1 = \{\text{Information}\}$, $Z_2 = \{\text{Comprehension}\}$, $Z_4 = \{\text{Similarities}\}$, $Z_6 = \{\text{Vocabulary}\}$ (all verbal ability measures), and $Z_{13} = \{\text{Education}\}$. Thus it may relate to some level of verbal acuity. The second factor loads heavily with $Z_8 = \{\text{Picture completion}\}$, $Z_9 = \{\text{Block design}\}$, and $Z_{11} = \{\text{Object assembly}\}$, suggesting perhaps a visual perception skill factor. The fourth factor loads $Z_3 = \{\text{Arithmetic}\}$ (and little else) heavily and may be a mathematical factor.

The other factors, especially the fifth and sixth, give few large loadings. Accounting also for the small percentage of variation they explain (see the lower rows of Table 10.8), the latter two at least may be superfluous. Indeed, Everitt (2005, Section 4.7) notes that for very large data sets such as seen here, the LR test can be unnecessarily sensitive: it overreacts to slight differences between the observed and predicted correlations and rejects more often than necessary. Reduced models with fewer factors may be quite reasonable and possibly easier to interpret. Exercise 10.6 explores possible avenues along these lines with these data. □

In some settings, the data analyst may wish to find estimates for the common factors, known as 'Factor Scores,' analogous to the PC scores from a PCA. The issue is more deceptive

than it may at first seem, however: the EFA model in (10.2) expresses the manifest variables in terms of the (unknown) factors, not vice versa. Some form of inversion is, therefore, necessary. Indeed, because the factors are assumed random the issue is actually one of prediction, not parameter estimation, and one must proceed carefully. Everitt (2005, Section 4.6) discusses the prediction of Factor Scores in more detail.

Employed strictly as a data reduction strategy EFA has both proponents and detractors, as has been noted throughout this section. It enlists a very particular model to describe the observed variables and requires extensive numerical computation to fit that model. Every time the method is considered for use, the data analyst must ask: does the latent factor model make sense, and is it appropriate for the data at hand? If so, EFA can provide valuable dimension reduction, especially given its additional ability to supply possible interpretation(s) for the reduced factors' impact on the phenomenon under study. It can also provide a springboard to *confirmatory factor analysis* (Jöreskog 1969), where more specific constraints are placed on the manifest variables and the postulated factors. If the latent factor model is inappropriate, however, use of EFA may be simple 'overkill' or worse, could produce misleading interpretations of the data. Analysts should approach its use with informed caution.

10.4 Canonical correlation analysis*

Often viewed as an extension of PCA due to similarities in its dimension-reduction technique, the method of *canonical correlation analysis* (CCA) can be applied when two multivariate *sets* of variables are observed. Rather than focus on observed variation in the data, however, the technique operates on the *correlations* between the two sets of variables.

CCA was developed by Hotelling (1935, 1936). Its basic data structure involves n observations taken on the two sets of variables $x_1, \ldots, x_p$ and $y_1, \ldots, y_q$, producing separate data matrices

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ x_{21} & \cdots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix} \quad \text{and} \quad \mathbf{Y} = \begin{bmatrix} y_{11} & \cdots & y_{1q} \\ y_{21} & \cdots & y_{2q} \\ \vdots & \ddots & \vdots \\ y_{n1} & \cdots & y_{nq} \end{bmatrix}. \tag{10.8}$$

As with the other dimension-reduction techniques studied in this chapter, it is common for the observations to be standardized by centering about their means and dividing by their standard deviations. Denote these here as $z_{ij} = (x_{ij} - \overline{x}_j)/s_{xj}$ and $w_{ij} = (y_{ik} - \overline{y}_k)/s_{yk}$, $i = 1, \ldots, n$; $j = 1, \ldots, p; k = 1, \ldots, q$. Collect the results into data matrices $\mathbf{Z}$ and $\mathbf{W}$, respectively, corresponding to (10.8). From these, find the correlation matrices $\mathbf{R}_{ZZ} = \mathbf{Z}^{\mathrm{T}}\mathbf{Z}/(n-1)$ and $\mathbf{R}_{WW} = \mathbf{W}^{\mathrm{T}}\mathbf{W}/(n-1)$, along with the cross-product matrices $\mathbf{R}_{ZW} = \mathbf{R}_{WZ}^{\mathrm{T}} = \mathbf{Z}^{\mathrm{T}}\mathbf{W}/(n-1)$. For convenience of notation, let $\varsigma = \min\{p, q\}$.

The goal in a CCA is to find linear combinations of $\mathbf{Z}$ and $\mathbf{W}$ whose correlation is a maximum; that is, find $\mathbf{Za}$ and $\mathbf{Wb}$ that maximize $r = \mathrm{Corr}[\mathbf{Za},\mathbf{Wb}]$. The quantities $\mathbf{Za}$ and $\mathbf{Wb}$ are called the (pair of) *canonical variates* of $\mathbf{Z}$ and $\mathbf{W}$. In effect, CCA finds the direction of maximal correlation between a pair of matrices. It involves a multivariate extension of the simpler pairwise correlation coefficient from Section 3.3.3.

Often, a single pair of canonical variates will not be sufficient, so the goal expands to find a collection of canonical variates $\mathbf{Za}_m$ and $\mathbf{Wb}_m$ with correlations r_m such that $r_1 > r_2 > \cdots > r_\tau$ for some $\tau < \varsigma$. This requires additonal constraints, so similar to the approach in a PCA,

we force the within-group covariances to zero and set

$$\text{Cov}[\mathbf{Za}_m, \mathbf{Za}_h] = \text{Cov}[\mathbf{Wb}_m, \mathbf{Wb}_h] = 0,$$

along with

$$\text{Cov}[\mathbf{Za}_m, \mathbf{Wb}_h] = 0,$$

for all $m \neq h$.

This is a constrained maximization problem, the solution to which results in yet another eigenanalysis (Timm 2002, Section 8.7). We set

$$\mathbf{R}_{ZZ}^{-1}\mathbf{R}_{ZW}\mathbf{R}_{WW}^{-1}\mathbf{R}_{WZ}\mathbf{a} = \lambda\mathbf{a}$$

and solve for the eignevalue/eigenvector pair λ and $\mathbf{a}$. One can construct a similar eigenequation to find $\mathbf{b}$, although it is often simpler to apply the equivalency relationship

$$\mathbf{b} = \frac{1}{\sqrt{\lambda}}\mathbf{R}_{WW}^{-1}\mathbf{R}_{WZ}\mathbf{a}\ .$$

From these, find the (ordered) eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_\varsigma$ and set $r_m = \sqrt{\lambda_m}$ for $m = 1, \ldots, \varsigma$. The r_m values are the *canonical correlation coefficients* of $\mathbf{Z}$ and $\mathbf{W}$. (Notice that these are the positive square roots; canonical correlations are constructed to be non-negative.) The largest eigenvalue corresponds to the maximal correlation, with decreasing correlation as m grows. As is common, the squared canonical correlations r_m^2 represent the proportion of variation in $\mathbf{Za}_m$ explained by that in $\mathbf{Wb}_m$.

As applied in a CCA, the dimension reduction strategy translates to operations with a reduced set of canonical variates. That is, choose a collection of canonical variates $\mathbf{Za}_m$ and $\mathbf{Wb}_m$ with their corresponding canonical correlations r_m over $m = 1, \ldots, \tau$ for some $\tau < \varsigma$, such that the collection retains a large proportion of the correlation structure available in the data. For instance, given a fixed percentage $100\pi\%$ ($0 < \pi < 1$) and assuming the data have been standardized, one might select τ such that

$$\frac{\sum_{m=1}^{\tau} r_m^2}{\sum_{h=1}^{\varsigma} r_h^2} \geq \pi. \tag{10.9}$$

Practical values for the threshold $100\pi\%$ will vary, depending on the domain-specific problem under study; common choices are often in the 50–80% range.

The CCA model also allows for tests of the null hypothesis

$$\text{H}_{o\tau}\colon\ \rho_1 \neq 0, \ldots, \rho_\tau \neq 0, \rho_{\tau+1} = \cdots = \rho_\varsigma = 0, \tag{10.10}$$

for any $\tau = 1, \ldots, \varsigma - 1$, where ρ_m is the true, underlying correlation estimated by r_m. If $\text{H}_{o\tau}$ is true, then only the first τ correlations are significantly different from zero and there is no need to retain more than those τ (pairs of) canonical variates.

For testing $\text{H}_{o\tau}$, Bartlett (1938, 1941) gave the statistic

$$G_\tau^2 = -\left\{(n-1) - \frac{1}{2}(p+q+1)\right\} \sum_{m=\tau+1}^{\varsigma} \log(1 - r_m^2). \tag{10.11}$$

In large samples, reject $H_{o\tau}$ at significance level α when $G_\tau^2 \geq \chi_\alpha^2(\nu)$, with $\nu = (p-\tau)(q-\tau)$ d.f. The corresponding large-sample P-value is $P[\chi^2\{(p-\tau)(q-\tau)\} \geq G_\tau^2]$.

If a specific value for τ is not known or anticipated, testing can proceed sequentially: begin at $\tau = \varsigma - 1$ and test $H_{o\tau}$ via (10.11). If the test rejects, stop and conclude the entire collection of ς canonical variates is required. If the test fails to reject, however, decrease τ by 1 and retest using (10.11). If this next test rejects, stop and conclude the reduced collection of $\varsigma - 1$ canonical variates is required. If it fails to reject, continue decreasing τ by one unit and retest, and so on. When the test finally rejects at a given τ, stop and operate with the reduced collection of those $\tau + 1$ canonical variates. If the test fails to reject for all τ, operate with only the largest canonical correlation at $\tau = 1$. (Clearly, this strategy involves multiple testing, similar to the approach for finding the number of factors in an EFA. If control of the consequent false positive error is important for the problem under study, some form of multiplicity adjustment may be required; see Section 5.5.)

In small samples, r_m possesses a bias for estimating ρ_m. (Timm 2002, Section 8.7.c), following Lawley (1959), gave a correction that operates on the eigenvalues/squared correlations. Modify r_m^2 into $\tilde{r}_m^2 = \upsilon_m^2 r_m^2$ with

$$\upsilon_m^2 = 1 - \frac{1}{n-1} \sum_{\substack{h=1 \\ m \neq h}}^{\tau} \frac{r_h^2}{r_m^2 - r_h^2} - \frac{\varsigma - \tau}{(n-1) r_m^2} \,,$$

valid for the largest $\tau < \varsigma$ correlations. If the correction factor calculates to a negative value, leave it undefined. The resulting values of $\tilde{r}_m$ are known as *adjusted canonical correlations*.

In **R**, CCA is performed via the `cancor()` function, although its extent is limited. A variety of external packages extend the basic features of `cancor()`. See http://cran.r-project.org/web/packages/.

Example 10.4.1 Canonical correlation in the Million-Song Dataset. The Million-Song Data Set is a publicly available online database (http://labrosa.ee.columbia.edu/millionsong/) cataloging the audio features of a million popular music tracks from 1922 to 2011 (Bertin-Mahieux et al. 2011). The data provide a useful testbed for a variety of research studies with massive data. Here, consider a formal 'test' subset of the larger database comprising audio features from $n = 51\,630$ tracks, available from the UCI Machine Learning Repository at http://archive.ics.uci.edu/ml. The data are quantifications of 90 audio attributes, $p = 12$ based on timbre average (the $\mathbf{X}$ domain) and $q = 78$ based on timbre covariance (the $\mathbf{Y}$ domain). A selection of the data is displayed in Table 10.9. (The complete set of values is given at http://www.wiley.com/go/piegorsch/data_analytics.)

As an exercise in CCA, the $p + q = 90$ attribute variables in Table 10.9 can be reduced to a smaller set of $\tau < \varsigma = \min\{12, 78\} = 12$ canonical variates. Begin by standardizing the data in the $\mathbf{Z}$ and $\mathbf{W}$ matrices, respectively. (Recall that **R**'s `scale()` function standardizes the columns of a numeric matrix.) From these, find the correlation matrices $\mathbf{R}_{ZZ} = \mathbf{Z}^{\mathrm{T}}\mathbf{Z}/(n-1)$, $\mathbf{R}_{WW} = \mathbf{W}^{\mathrm{T}}\mathbf{W}/(n-1)$, and $\mathbf{R}_{ZW} = \mathbf{R}_{WZ}^{\mathrm{T}} = \mathbf{Z}^{\mathrm{T}}\mathbf{W}/(n-1)$. In **R**, use `cor(Z)`, `cor(W)`, and `cor(Z,W)`, respectively.

The correlation matrix for the complete data can be built from the individual component matrices as a partition:

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{ZZ} & \mathbf{R}_{ZW} \\ \mathbf{R}_{WZ} & \mathbf{R}_{WW} \end{bmatrix}. \tag{10.12}$$

Table 10.9 Selected audio track quantifications from the Million-Song Data Set stratified by timbre average (**X** domain) and timbre covariance (**Y** domain) variables; selection from larger collection of $n = 51630$ tracks.

X: timbre average				**Y**: timbre covariance			
TAvg01	TAvg02	$\cdots$	TAvg12	TCov01	TCov02	$\cdots$	TCov78
45.442	−30.750	$\cdots$	−2.289	17.902	1377.122	$\cdots$	7.173
52.678	−2.889	$\cdots$	3.229	5.668	702.254	$\cdots$	−5.215
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
51.857	59.117	$\cdots$	−6.198	20.166	598.453	$\cdots$	12.174

Source: UCI Machine Learning Repository (http://archive.ics.uci.edu/ml).

The 90×90 matrix of correlations here is too large and complex to read easily. However, it can be visualized via a *heatmap* display as described in Section 4.2.4. Here, the heatmap is a 90×90 graphic whose coloration or grayscaling represents the correlations in **R**. Figure 10.5 displays such a heatmap for (10.12) built from the `heatmap()` function in **R**. Lighter 'hotter' scale indicates values closer to 1, while darker 'cooler' scale indicates values closer to −1; the color-bar along the left of the graphic gives the spectrum from 1 (bottom) to −1 (top).

As might be expected, the multiple correlation patterns in Figure 10.5 are mixed, although a clear 'hotspot' of increased correlation is evident for the first dozen or so timber covariance variables (see labels along the right/bottom). Note that because **R** is a symmetric matrix, the heatmap pattern mirrors above and below the white diagonal.

To conduct the CCA, appeal to `cancor(Z,W)` in **R** produces the (ordered) canonical correlations displayed in Table 10.10. The table also gives the corresponding squared correlations, r_m^2, along with the cumulative percentage variation explained by each (paired) canonical variate. As can be seen, the first $\tau = 3$ (pairs of) canonical variates explain over 50% of the variation from the correlation structure. If we increase to the first $\tau = 6$ canonical variates, we exceed 80%. This suggests that a smaller group of between three and six canonical variates could be sufficient to represent the correlation structure in this set of data.

We can also consider the individual null hypotheses from (10.10) that as a group the first τ correlations are significantly different from zero (and that the remaining $\varsigma - \tau = 12 - \tau$ are not). Table 10.11 displays Bartlett's G_τ^2 statistic (10.11) to test each null hypothesis, for $\tau = 11, \ldots, 1$. We see each G_τ^2 statistic is highly significant when compared to its corresponding χ^2 reference distribution. This is not altogether surprising: when applied to extremely large sample sizes – like those here – χ^2 tests of this sort can be overly sensitive. They overreact to small deviations from the null event and reject more often than necessary. This appears to be the case with these data. □

Just as in PCA, examination of the CCA loading coefficients $\mathbf{a}_m$ and $\mathbf{b}_m$ for each canonical variate can provide some understanding of how the original (standardized) variates associate with each other. One may also extend the analysis to a *sparse CCA*, where many loading coefficients are driven to zero in order to highlight contributions of higher-impact variables (Parkhomenko et al. 2009; Witten and Tibshirani 2009). This shares features with the similar 'sparse' strategy for PCA. (The external *PMA* package can perform sparse CCA in **R**, via its

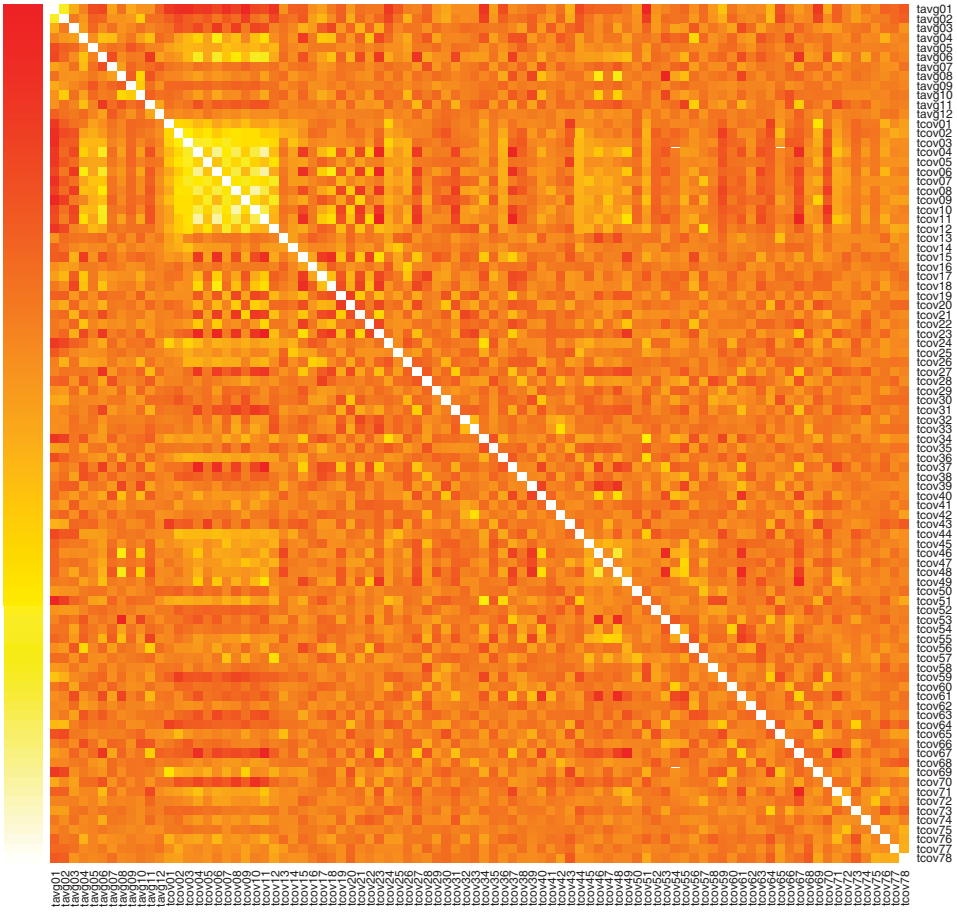

Figure 10.5 Heatmap of 90×90 correlation matrix for standardized data from Table 10.9. Lighter colors indicate correlations near 1, darker colors near -1. Scale at left gives the spectrum from 1 (white, bottom) to -1 (dark, top). Source: Data from http://labrosa.ee.columbia.edu/millionsong/ via http://archive.ics.uci.edu/ml.

`CCA()` function.) Given the multivariate nature of these sorts of analyses, however, interpretation can be tricky and often must mature with the analyst's practice and experience.

A standard recommendation for conducting a CCA is that the sample size minimally satisfy $n \geq p + q + 1$ (Eaton and Perlman 1973). When n is small relative to p and q, however, the method is hampered. One approach that can adjust for excessive numbers of variables is known as *regularized canonical correlation analysis* (RCCA), which in effect introduces penalty terms into the correlation matrices to adjust for excess variables. See González et al. (2008), who also provided the external *CCA* package in **R** for performing RCCA.

The short introduction to CCA given here only scratches the surface of this complex method of multivariate data analytics. Readers interested in further details may refer to

Table 10.10 Ordered canonical correlations from CCA of testbed data in Table 10.9.

m	r_m	r_m^2	Cumlulative percentage of variance (%)*
1	0.9240	0.8538	20.0
2	0.8735	0.7631	37.9
3	0.7936	0.6298	52.7
4	0.7450	0.5490	65.5
5	0.6255	0.3913	74.7
6	0.4957	0.2457	80.5
7	0.4629	0.2143	85.5
8	0.4494	0.2020	90.2
9	0.3617	0.1308	93.3
10	0.3407	0.1161	96.0
11	0.2967	0.0880	98.1
12	0.2854	0.0815	100

* Canonical variates' accumulated percentage contribution to total explainable variability: $\sum_{h=1}^{m} r_h^2 / \sum_{h=1}^{12} r_h^2$.

Table 10.11 Bartlett test statistics, G_τ^2, from CCA of testbed data in Table 10.9.

τ	G_τ^2	d.f.	P-value
11	4 383.63	67	$<10^{-4}$
10	9 137.17	136	$<10^{-4}$
9	15 501.58	207	$<10^{-4}$
8	22 734.47	280	$<10^{-4}$
7	34 373.48	355	$<10^{-4}$
6	46 811.48	432	$<10^{-4}$
5	61 356.48	511	$<10^{-4}$
4	86 960.00	592	$<10^{-4}$
3	128 040.46	675	$<10^{-4}$
2	179 304.87	760	$<10^{-4}$
1	253 586.41	847	$<10^{-4}$

the advance discussions in, for example, Everitt (2005, Section 8.3) and Izenman (2008, Section 7.3).

Exercises

10.1 Return to the PCA for the hazard-loss data in Table 10.2, and explore the final PC scores in greater detail, as follows.

(a) Apply Bartlett's test for sphericity to these data using (10.7). Is there an indication that the correlation structure was rich enough to perform the PCA?

(b) Construct a scatterplot matrix for the first three PC scores studied in Example 10.2.2. What patterns emerge, if any?

(c) As suggested in the example, include the next (fourth) PC. Now, construct a scatterplot matrix display for the four retained PC scores. Also construct all (4 – why is it 4?) possible 3D scatterplots using the four PC scores. (In **R**, the external *scatterplot3d* package may prove useful.)

(d) Do any additional patterns emerge?

10.2 Return to the full wheat kernel data from Table 4.7 with all $p = 7$ variables. Perform a PCA on the seven-variable, standardized data set and determine the extent of any available dimension reduction using the following criteria:

(a) Kaiser's (1958) criterion where $\lambda_j \geq 1$ for $j = 1, \ldots, q$.

(b) Jolliffe's (1972) criterion where $\lambda_j \geq 0.7$ for $j = 1, \ldots, q$.

(c) Selection of the first q eigenvalues to achieve at least 80% explainable variation.

(d) Construct a scree plot and choose q such that the plot 'elbows' at λ_q.

10.3 To explore the geometry of PCs in more detail, return to the $p = 2$ case illustrated in Figure 10.1 using husband's and wife's ages from Table 10.1. With those same data, construct the following alternative graphs.

(a) Center each original variable: find $x_1^* = (x_1 - \overline{x}_1)$ as the centered wife's age and $x_2^* = (x_2 - \overline{x}_2)$ as the centered husband's age. Now plot x_2^* versus x_1^*. Compare this to the plot in Figure 10.1.

(b) Calculate the z-scores via (3.7) for each original variable: find $z_1 = (x_1 - \overline{x}_1)/s_1$ as the standardized wife's age and $z_2 = (x_2 - \overline{x}_2)/s_2$ as the standardized husband's age. Now plot z_2 versus z_1. Compare this to the plot in Figure 10.1.

(c) Perform a PCA on the standardized data and find the 2 PC scores, PC_1 for the wives and PC_2 for the husbands. Plot PC_2 versus PC_1. Compare this to your other plots of these data. Do you see any patterns among them? (*Hint*: Try an overlay of the PC lines on the standardized plot.)

10.4 To explore the mathematics of PCs, set $p = 2$ with standardized variables $\mathbf{z} = [z_1 \; z_2]^T$ (Everitt 2005, Section 3.2). Suppose n data pairs are recorded and the sample correlation matrix is found as

$$\mathbf{R} = \begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix}.$$

(a) Start with the two eigenvalues, λ_1 and λ_2, of $\mathbf{R}$. From (A.4), these solve the determinant equation $|\mathbf{R} - \lambda\mathbf{I}| = 0$. Show that this reduces to solving the quadratic equation $(1 - \lambda)^2 - r^2 = 0$. What are the two solutions?

(b) Next find the corresponding eigenvectors. Start with λ_1 and from (A.3), solve the matrix equation $\mathbf{R}\mathbf{u}_1 = \lambda_1\mathbf{u}_1$ for $\mathbf{u}_1 = [u_{11} \; u_{21}]^T$. Show that this leads to the system of two equations (with two unknowns)

$$u_{11} + ru_{21} = (1 + r)u_{21}$$

and

$$ru_{11} + u_{21} = (1 + r)u_{21}.$$

Solve this system for u_{11} and u_{21}. What interesting result do you achieve? To establish a unique solution, impose the constraint $\mathbf{u}_1^{\mathrm{T}}\mathbf{u}_1 = 1$. Do any arbitrary features still remain? If so, force $u_{11} > 0$.

(c) Mimic Exercise 10.4b to find the eigenvector $\mathbf{u}_2$ corresponding to λ_2.

(d) Use the results above to find the PCs y_1 and y_2 in terms of z_1 and z_2.

10.5 Under the factor analysis model for Z_j from (10.2) or (10.3), and assuming $E[\mathbf{F}] = \mathbf{0}$, $E[\boldsymbol{\epsilon}] = \mathbf{0}$, $\mathrm{Var}[\mathbf{F}] = \mathbf{I}$, and $\mathrm{Var}[\boldsymbol{\epsilon}] = \mathrm{diag}\{\psi_1, \psi_2, \ldots, \psi_p\}$, verify the following for any $j = 1, \ldots, p$:

(a) $E[Z_j] = 0$.

(b) $\mathrm{Var}[Z_j] = h_j^2 + \psi_j$, where $h_j^2 = \sum_{k=1}^{q} b_{jk}^2$ are the communalities. Since $\mathrm{Var}[Z_j]$ is constructed to be 1, verify the relationship between h_j^2 and $\boldsymbol{\psi}_j$.

(c) Assume $\mathrm{Cov}[F_k, \epsilon_j] = 0$ for all k, j. For $m \neq j$, show that $\mathrm{Cov}[Z_j, Z_m] = \sum_{k=1}^{q} b_{jk} b_{mk}$.

(d) Verify that this leads to the matrix expression $\mathrm{Var}[\mathbf{Z}] = \mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{D}_{\boldsymbol{\psi}}$.

10.6 Return to the WAIS test data in Example 10.3.1 and as there employ ML estimation with a varimax rotation. Explore the fit of different q-factor models as follows (Everitt 2005, Section 4.7).

(a) Set $q = 6$ and perform the fit. This should reproduce the results in Table 10.8. From this, compute and plot the residual correlations $\mathbf{R} - (\hat{\mathbf{B}}\hat{\mathbf{B}}^{\mathrm{T}} + \hat{\mathbf{D}}_{\boldsymbol{\psi}})$. (Only the $\binom{13}{2} = 78$ entries in the upper or lower triangular portion are necessary. The `upper.tri()` function in R may prove helpful.) Do the differences appear very small and close to zero?

(b) Now set $q = 3$ and reperform the ML fit with varimax rotation. Again compute and/or plot the residual correlations. Do these seem much different from those from the $q = 6$ fit? If not, are the rotated factors simpler to interpret?

10.7 As part of a study to develop digitized cell imaging techniques for diagnosing cancer status, Street et al. (1995) reported data from $n = 151$ female cancer patients over a series of digitized variables meant to categorize certain target cell tissues. Consider the $p = 11$ cell feature variables listed in the following display, along with their recorded outcomes from these 151 patients. (Only a selection of the measurements appears here. Download the complete data set at http://www.wiley.com/go/piegorsch/data_analytics.)

The first five variables (Size, Radius, Perimeter, Area, and Compactness) may be viewed as 'size' characteristics, the next two (Texture and Smoothness) as 'texture' characteristics, and the last four (Concavity, Concave Points, Symmetry, and Fractal Dimension) as 'shape' characteristics.

Size	Radius	Perimeter	Area	Compact	Texture	Smooth
5.0	18.02	0.1086	0.0633	0.0142	1013.0	3.972
3.0	17.99	0.3001	0.0787	0.0490	1001.0	8.589
⋮	⋮	⋮	⋮	⋮	⋮	
3.5	16.70	0.1012	0.0604	0.0263	885.4	4.243

Concavity	Concave Points	Symmetry	Fractal Dimension
0.0169	37.08	0.120	0.117
0.0300	17.33	0.162	0.265
⋮	⋮	⋮	⋮
0.0196	34.92	0.133	0.132

(a) Plot the data: use a scatterplot matrix to visualize any potential (pairwise) patterns among the original 11 variables.

(b) Standardize the original data across the $p = 11$ variables and find their sample correlation matrix $\mathbf{R}$.

(c) Apply Bartlett's test for sphericity using (10.7) to these data. Is the correlation structure rich enough to perform an EFA?

(d) Apply an EFA to explore if dimension reduction from the original 11 variables is possible. Suppose the factor analytic model in (10.3) applies, and for convenience, assume a normal distribution structure on the standardized data. Estimate the elements of the matrices $\mathbf{B}\mathbf{B}^{\mathrm{T}}$ and $\mathbf{D}_{\boldsymbol{\psi}}$ via ML. Set the number of factors to $q_o = 4$ and with no rotation test for model adequacy via an LR statistic. Continue testing until an adequate reduction in dimension (if any) is found. Make no adjustment for test multiplicity. (*Hint*: To improve iterative stability, use the $\hat{\psi}_j$ 'uniquenesses' from the previous fit as starting values in each new fit.)

(e) Find the loadings from your chosen model. Is any interpretable pattern evident? If not, consider a varimax rotation and reexamine the loading pattern for interpretable features.

(f) Contrast use of the LR test to reduce the dimension of the model with simple calculation of percentage of explainable variability using the squared loading coefficients as in (10.6). Set your threshold to 70%. How does this change your results?

10.8 Fayers and Machin (2007, Section 6.2) reported summary correlations from a study on psychological factors affecting chronic-disease therapy. A sample of $n = 1952$ patients were surveyed for their responses on $p = 14$ different anxiety- and depression-related features of the therapy. Of these variables, the first seven were anxiety related (coded A1, A3, … , A13) and the final seven were depression related (coded D2, D4, … , D14). The observed correlation matrix, $\mathbf{R}$, was found to be the following (upper triangular portion only; lower triangle is symmetric):

	A1	A3	A5	A7	A9	A11	A13	D2	D4	D6	D8	D10	D12	D14
A1	1	0.54	0.56	0.50	0.49	0.34	0.54	0.31	0.34	0.41	0.28	0.30	0.33	0.30
A3		1	0.60	0.41	0.58	0.32	0.60	0.26	0.36	0.37	0.23	0.27	0.32	0.27
A5			1	0.43	0.52	0.32	0.56	0.27	0.36	0.41	0.27	0.27	0.31	0.30
A7				1	0.38	0.34	0.43	0.41	0.43	0.47	0.33	0.34	0.44	0.40
A9					1	0.30	0.57	0.20	0.28	0.32	0.17	0.23	0.26	0.30
A11						1	0.38	0.15	0.18	0.19	0.16	0.17	0.17	0.19
A13							1	0.26	0.33	0.37	0.23	0.33	0.30	0.31
D2								1	0.47	0.50	0.52	0.40	0.59	0.42
D4									1	0.58	0.33	0.38	0.54	0.44
D6										1	0.37	0.41	0.52	0.47
D8											1	0.36	0.44	0.31
D10												1	0.45	0.36
D12													1	0.42
D14														1

(a) Apply Bartlett's test for sphericity using (10.7) to these data. Is the correlation structure rich enough to perform an EFA?

(b) Apply an EFA to explore if dimension reduction from the original 14 variables is possible. Suppose the factor analytic model in (10.3) applies. Estimate the elements of the matrices $\mathbf{B}\mathbf{B}^{\mathrm{T}}$ and $\mathbf{D}_{\boldsymbol{\psi}}$ via principal factors (i.e., principal axis factoring). Set the number of factors to $q_o = 5$, and with no rotation, determine if this five-factor model is adequate via a percentage of explainable variability metric, as in (10.6). Set your threshold to 90%. (You may need to increase the maximum number of iterations in your computer code. Consider using at least 1000 iterations.) If not, increase or decrease the number of factors until an adequate reduction in dimension (if any) is found.

(c) Find the loadings from your chosen model. Is any interpretable pattern evident? If not, consider a varimax rotation and reexamine the loading pattern for interpretable features.

10.9 Hambrick et al. (2010) gave summary correlations from a psychometric study of work multitasking. A sample of $n = 109$ participants were administered tests of multitasking capacity/speed under a suite of either four Baseline ('Bblock') work conditions

	Bblock1	Bblock2	Bblock3	Bblock4	Eblock1	Eblock2	Eblock3	Eblock4
Bblock1	1	0.73	0.68	0.70	0.54	0.53	0.42	0.39
Bblock2		1	0.69	0.63	0.46	0.48	0.45	0.35
Bblock3			1	0.58	0.53	0.47	0.40	0.38
Bblock4				1	0.45	0.42	0.31	0.24
Eblock1					1	0.78	0.68	0.61
Eblock2						1	0.81	0.77
Eblock3							1	0.85
Eblock4								1

or four increased-pace Emergency ('Eblock') conditions. Responses under the $p = 8$ conditions represented the outcome variables. The observed correlation matrix, $\mathbf{R}$, was reported as the matrix displayed above (upper triangular portion only; lower triangle is symmetric).

(a) Apply Bartlett's test for sphericity using (10.7) to these data. Is the correlation structure rich enough to perform an EFA?

(b) Apply an EFA to explore if dimension reduction from the original eight variables is possible. Suppose the factor analytic model in (10.3) applies, and for convenience, assume a normal distribution structure on the standardized data. (Estimate the elements of the matrices $\mathbf{B}\mathbf{B}^{\mathrm{T}}$ and $\mathbf{D}_{\boldsymbol{\psi}}$ via ML.) As there are two broad testing conditions ('B' and 'E'), set the number of factors to $q_o = 2$ and with no rotation test for model adequacy via an LR statistic. Continue testing until an adequate reduction in dimension (if any) is found. Make no adjustment for test multiplicity.

(c) Find the loadings from your chosen model. Is any interpretable pattern evident? If not, consider a varimax rotation and reexamine the loading pattern for interpretable features.

(d) Contrast use of the LR test to reduce the dimension of the model with simple calculation of percentage of explainable variability using the squared loading coefficients as in (10.6). Set your threshold to 80%. How does this change your results?

10.10 CCA is increasingly applied when studying genetic outcomes between different populations, disease states, and so on. Consider the following data, which are expression levels in 19672 genes from $n = 89$ samples of breast cancer tissue described by Witten et al. (2009). CCA can be employed to help identify coregulated sets of genes on different chromosomes. For illustrative use, expression levels are listed here for a subset of $p = 4$ genes from human chromosome 1 (the $\mathbf{X}$ domain) and a subset of $q = 3$ genes from human chromosome 13 (the $\mathbf{Y}$ domain). A selection of the measurements follows. (Download the complete data set at http://www.wiley.com/go/piegorsch/data_analytics.)

$\mathbf{X}$: Chromosome 1 genes				$\mathbf{Y}$: Chromosome 13 genes		
SF3B4	HSPC003	GNPAT	NDUFS2	WASF3	BRCA2	PCCA
9.7058	9.0607	10.626	10.1618	7.6545	5.2597	5.5841
9.7582	8.8349	10.607	10.1444	6.0809	5.0343	6.6120
9.0349	8.7048	10.033	9.5802	5.9279	5.6758	8.1687
⋮	⋮	⋮	⋮	⋮	⋮	⋮
9.5462	9.7489	10.485	10.9294	8.2027	4.9079	7.0484

(a) Perform a CCA. Start by standardizing the data: compute a $\mathbf{Z}$ matrix from $\mathbf{X}$ and a $\mathbf{W}$ matrix from $\mathbf{Y}$.

(b) Calculate the $\varsigma = 3$ canonical correlations, r_m, and their corresponding canonical variates from the correlation matrices of $\mathbf{Z}$ and $\mathbf{W}$.

(c) Find the squared correlations r_m^2, and identify the percentage of explainable variation each canonical term provides, via (10.9). If 70% of total variation is considered a sufficient threshold, what do you conclude about the need for any of the terms?

(d) Calculate the Bartlett statistic from (10.11) at $\tau = 2$ and $\tau = 1$. Operate at $\alpha = 0.05$. (Do not adjust for multiplicity.) What does this suggest about the need for any of the terms?

(e) Construct a plot of the leading canonical variables, that is, plot $\mathbf{Za}_1$ against $\mathbf{Wb}_1$. Do any informative patterns emerge?

11

Techniques for unsupervised learning: clustering and association

11.1 Cluster analysis

An important data analytic technology involves the study of how observations *cluster* into distinct groups or categories. When the categories are known and delineated in advance, statistical learning about the clustering/categorization is a supervised process, as in the classification problem from Chapter 9. When the categories or clusters are unknown, however, learning is unsupervised. *Cluster analysis* is the broad term for this effort: the analysis of multivariate data to identify patterns of clustering they exhibit. The concept evolved in the mid-twentieth century after early works of Driver and Kroeber (1932) and Tryon (1939) established many of the basic themes, with applications in anthropology and psychology. Cluster analysis has since become a popular technique in data mining and informatics.

As in Chapter 10, assume the data are numerical measurements on $p > 1$ different variables of interest taken over a sample or training set of n subjects. Each variable is associated with an individual column vector $\mathbf{x}_j = [x_{1j}\ x_{2j}\ \cdots\ x_{nj}]^T$ $(j = 1, \ldots, p)$. The corresponding $n \times p$ data matrix is $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_p]$. Here again, it will be useful to center the observations about their means, say, $x^*_{ij} = x_{ij} - \bar{x}_j$, producing a column matrix of (mean-corrected) values $\mathbf{X}^*$ from (10.1).

Generally, differences in scale between the p variables can impact the clustering relationship: variables with especially large scales can dominate or skew the pattern relative to those with smaller scales, destabilizing the analysis. To avoid this, we further divide the x^*_{ij}s by their corresponding standard deviations, s_j. This produces standardized z-scores as in (3.7): $z_{ij} = x^*_{ij}/s_j = (x_{ij} - \bar{x}_j)/s_j,\ i = 1, \ldots, n;\ j = 1, \ldots, p$, with consequent, standardized, $n \times p$ data matrix $\mathbf{Z}$.

Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery,
First Edition. Walter W. Piegorsch.

www.wiley.com/go/piegorsch/data_analytics

The fundamental goal of a cluster analysis is to create a set of clusters $\{C_1, C_2, \ldots, C_K\}$ from the p-variate information provided by the standardized observation vectors $\mathbf{z}_i = [z_{i1}\ z_{i2}\ \cdots\ z_{ip}]$ making up the n rows of $\mathbf{Z}$. Ostensibly, observations within a cluster C_k should be more similar to each other and less similar to those in any other cluster $C_{k'}$. (Technically, the notation C_k is used here to contain those *indices* i corresponding to row vectors $\mathbf{z}_i$ located in the kth cluster. Where no confusion exists, however, the terminology will also refer loosely to an observation $\mathbf{z}_i$ as being 'in' C_k.)

How to define similarity within – or, for that matter, dissimilarity between – clusters is, of course, the challenge: many different clustering strategies have been proposed, each with its own selective value and use(s). At their core, however, the various methods all attempt to replicate what the human eye does expertly in two (and sometimes three) dimensions: identify where and how clusters of data group together. Indeed, when $p = 2$ or 3, appeal to 2D or 3D scatterplots and/or scatterplot matrices often provides sufficient visual guidance on which clusters the z_{ij}s inhabit. (Further study of how and what each identified cluster represents then becomes part of the knowledge discovery process.) In higher dimensions, humans seldom visualize patterns as effectively, however, and we turn to statistical techniques for cluster identification. Indeed, these can also prove useful in the simpler two- and three-dimensional settings. For illustrative purposes, consider the following small-scale example.

Example 11.1.1 Automobile characteristics. From a study of US automobile efficiency similar to that in Example 4.2.6, Myatt (2007, Table 6.8) gave a small data set with selected physical characteristics of $n = 8$ eight-cylinder passenger cars from model year 1970. For simplicity of visualization, focus here is restricted to the $p = 2$ variables $x_1 = \{\text{Engine displacement (cu. in.)}\}$ and $x_2 = \{\text{Acceleration (s, from 0 to 60 mph)}\}$. The data appear in Table 11.1; the corresponding z-scores are plotted in Figure 11.1. The figure presents a simple scatterplot (Figure 11.1a) alongside a labeled plot with the model names (Figure 11.1b).

The relationship in Figure 11.1 illustrates the expected drop (improvement) in acceleration times with larger engine displacement for these eight-cylinder US passenger cars. Beyond this, however, two simple clusters quickly visualize: a low-displacement/slow-acceleration cluster to the upper left and a large-displacement/fast-acceleration cluster to the lower right. Translating to the original scale, one sees that large displacements above 400 cu. in. appear to definitively lower acceleration times for these vehicles. Such a 'knowledge discovery' may seem obvious, but it is nonetheless instructive to see the data illustrate this phenomenon. How to replicate this sort of visual identification statistically is the goal of a cluster analysis.

We will return to these data in order to explicate features of the various clustering methods in this section. □

Every cluster analysis depends on how 'distance' is defined in the feature space. The distance metric, also called a *dissimilarity measure*, between any two observations $\mathbf{z}_i$ and $\mathbf{z}_h$ is given by a functional $d(\mathbf{z}_i, \mathbf{z}_h)$ that satisfies standard conditions of a 'distance' for all i and h:

- $d(\mathbf{z}_i, \mathbf{z}_h) \geq 0$,
- $d(\mathbf{z}_i, \mathbf{z}_h) = d(\mathbf{z}_h, \mathbf{z}_i)$ ('symmetry'),
- $d(\mathbf{z}_i, \mathbf{z}_i) = 0$, and
- $d(\mathbf{z}_i, \mathbf{z}_h) \leq d(\mathbf{z}_i, \mathbf{z}_\ell) + d(\mathbf{z}_\ell, \mathbf{z}_h)$ for all $\ell = 1, \ldots, n$ (the *triangle inequality*).

Table 11.1 Data on selected eight-cylinder US automobiles from 1970 model year: Engine Displacement (cu. in.) and Acceleration (s, from 0 to 60 mph).

Make and model	Displacement (cu. in.)	Acceleration (s)
Chevrolet Chevelle	307	12.0
Plymouth Fury	440	8.5
Ford Galaxie 500	429	10.0
Chevrolet Impala	454	9.0
AMC Rebel SST	304	12.0
Plymouth Satellite	318	11.0
Buick Skylark	165	11.5
Ford Torino	140	10.5

Source: Myatt (2007, Table 6.8).

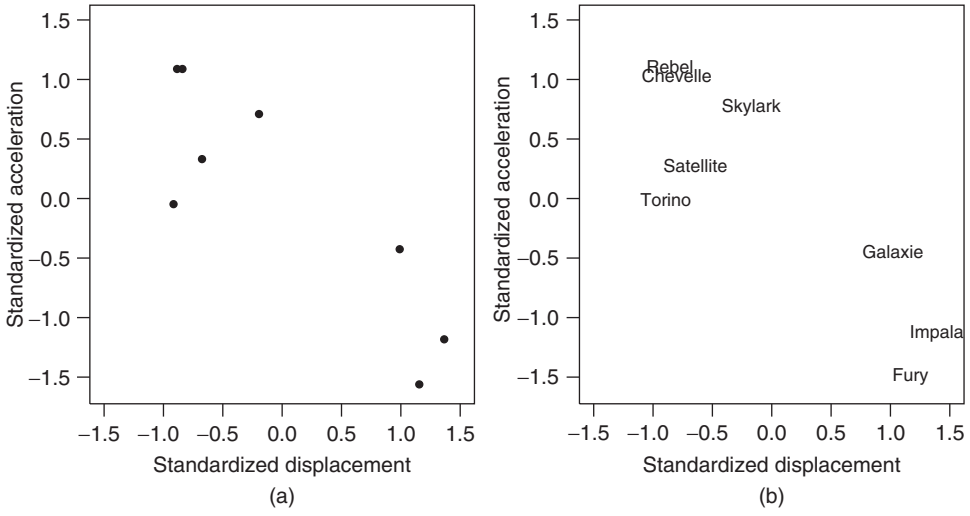

Figure 11.1 Scatterplots of automobile characteristics in Table 11.1 from Example 11.1.1. (a) $z_2 = \{\text{Standardized Acceleration}\}$ plotted against $z_1 = \{\text{Standardized Engine Displacement}\}$. Two clusters are apparent at upper left and lower right. (b) Points labeled by auto model (slight jitter added to distinguish adjacent points). Source: Data from Myatt (2007, Table 6.8).

We collect the observed distances together into a $n \times n$ *distance matrix*, also called a *dissimilarity matrix*, $\mathbf{D}$, with (i, h)th element $d(\mathbf{z}_i, \mathbf{z}_h)$. Notice then that $\mathbf{D} = \mathbf{D}^{\mathrm{T}}$ and that the diagonal of $\mathbf{D}$ contains only zeros. In many cases, numerical or graphical examination of $\mathbf{D}$ (via, e.g., a heatmap from Section 4.2.4) can give useful input on the affinity patterns exhibited in the data.

In passing, it is worth noting that some analysts prefer to operate instead with a *proximity* or *similarity matrix* $\mathbf{S}$ that captures how 'close' observations are to each other. The proximity

elements $s(\mathbf{z}_i, \mathbf{z}_h)$ are constructed to increase as $d(\mathbf{z}_i, \mathbf{z}_h)$ decreases, with the typical convention that $s(\mathbf{z}_i, \mathbf{z}_h) = 1$ when $d(\mathbf{z}_i, \mathbf{z}_h) = 0$. Thus a simple choice for $\mathbf{S}$ is $\mathbf{I} - \mathbf{D}$ after scaling the $d(\mathbf{z}_i, \mathbf{z}_h)$ values to have maximum 'distance' 1. Of course, other forms are possible. For example, one might compute the sample correlation between $\mathbf{z}_i$ and $\mathbf{z}_h$, averaging across the p variables see Hastie et al. (2009, Section 14.3.2).

The distance metric is often taken as Euclidean, that is, the usual 'as-the-crow-flies' distance. Similar to the options for k-nearest neighbor learning from Section 9.3, a number of alternatives can be considered for the distance; Table 11.2 reproduces the options in Table 9.8. Notice that the Minkowski form is actually a general class of metrics, characterized by its positive parameter γ. Special cases include Euclidian ($\gamma = 2$), Manhattan ($\gamma = 1$, sometimes called 'Hamming' distance when the x_{ij}s are quantitative measurements), and Maximum ($\gamma \to \infty$) distances. For Canberra distance, terms with zero numerator and denominator are omitted from the sum; Canberra distance is often intended for use with nonnegative counts. For some additional metrics useful with other forms of data such as binary observations, see Myatt and Johnson (2009, Section 3.6).

Table 11.2 Selected metrics for defining the distance, $d(\mathbf{z}_i, \mathbf{z}_h)$, between two p-dimensional row vectors $\mathbf{z}_i = [z_{i1} \quad z_{i2} \quad \cdots \quad z_{ip}]$ and $\mathbf{z}_h = [z_{h1} \quad z_{h2} \quad \cdots \quad z_{hp}]$.

Name	Distance, d_{ih}
Euclidean	$d(\mathbf{z}_i, \mathbf{z}_h) = \sqrt{\sum_{j=1}^{p} (z_{ij} - z_{hj})^2}$
Manhattan ('city block')	$d(\mathbf{z}_i, \mathbf{z}_h) = \sum_{j=1}^{p} \lvert z_{ij} - z_{hj} \rvert$
Maximum/Tchebychev	$d(\mathbf{z}_i, \mathbf{z}_h) = \max_{j=1,\ldots,p} \{\lvert z_{ij} - z_{hj} \rvert\}$
Minkowski	$d(\mathbf{z}_i, \mathbf{z}_h) = \left\{ \sum_{j=1}^{p} \lvert z_{ij} - z_{hj} \rvert^{\gamma} \right\}^{1/\gamma} \quad (\gamma > 0)$
Canberra	$d(\mathbf{z}_i, \mathbf{z}_h) = \sum_{j=1}^{p} \lvert z_{ij} - z_{hj} \rvert / \lvert z_{ij} + z_{hj} \rvert$

In **R**, the `dist()` function computes distances between the rows of a data matrix. The function can accommodate any of the specific metrics listed in Table 11.2, via its `method=` option; `method='euclidean'` is the default.

Focus in this section is on how the information in $\mathbf{D}$ can be used to cluster the n observations in a quantitative manner. Two distinct clustering strategies are examined: (i) *hierarchical methods* where convenient or evolving subgroups/subcategories appear as the clusters develop and (ii) data segmentation that *partitions* the observations into a set of $K \geq 1$ clearly separated clusters. The latter approach is how most users imagine clustering and cluster analysis – segmenting the data into groups as suggested in Figure 11.1 – while the former process has more of a branching, tree-like nature similar to the decision trees in Section 9.4.1. Each approach is discussed in turn, beginning with hierarchical clustering.

11.1.1 Hierarchical clustering

Hierarchical clustering is based on a simple concept: start with as many clusters as there are observations (so $K = n$) and amalgamate clusters in a stepwise, hierarchical manner.

Add observations to clusters based on how the 'distances' between the clusters change as they grow. This is known as *agglomerative hierarchical clustering*. When operating hierarchically, observations are not to be swapped or moved between clusters: once an observation is added to an agglomerating cluster, it remains there until completion of the process. (One can alternatively reverse the process and start with a single large cluster containing all n observations and then divide clusters apart by removing observations based on the changing distances between clusters. This divisive case is aptly called *divisive hierarchical clustering*. Agglomerative clustering is more common in practice, however, and is highlighted here. Some comments on divisive clustering appear at the end of the section.)

Given the interobservation distances in the dissimilarity matrix $\mathbf{D}$ – and therefore, a fixed choice for the distance metric – a further definition is required for the *linkage* between two clusters C_k and C_m. Linkage here refers to how separation between C_k and C_m is calculated, given the choice of distance metric $d(\mathbf{z}_i, \mathbf{z}_h)$ from Table 11.2. Many authors also refer to the linkage descriptor as (another) 'dissimilarity measure,' now measuring dissimilarities between clusters rather than between observations. (The terminology varies and can be confusing; in practice, users must be careful to recognize which term is being used for which calculation.) To help distinguish the two, denote the linkage dissimilarity measure as $\delta(C_k, C_m)$. Here again, a number of different forms are possible. One of the simplest is known as *single linkage* or *single-link clustering*, where the closest observations between two clusters define the linkage distance (a type of *nearest-neighbor* relationship):

$$\delta_{\text{Sngl}}(C_k, C_m) = \min_{\substack{i \in C_k \\ h \in C_m}} \{d(\mathbf{z}_i, \mathbf{z}_h)\}.$$

Contrastingly, one can take the distance between the farthest two observations, called *complete linkage* or *complete-link clustering*:

$$\delta_{\text{Comp}}(C_k, C_m) = \max_{\substack{i \in C_k \\ h \in C_m}} \{d(\mathbf{z}_i, \mathbf{z}_h)\}.$$

Both single-link and complete-link clustering are invariant to monotone transformations of $d(\mathbf{z}_i, \mathbf{z}_h)$, and when the underlying clusters are well delineated and tightly contained, the two approaches often yield similar outcomes. In other cases, however, they can produce distinctly different results. Since it focuses on only the closest points between two clusters, single linkage can form clusters that are fairly diffuse, elongated, or irregularly shaped (think 'barbells'). Known as the *chaining problem*, this is sometimes viewed as a deficiency of the method. As Clarke et al. (2009, p. 416) noted, however, unusually shaped clusters do occur in nature, so '… it's unclear in general whether such properties are features or bugs.'

At the other extreme, complete linkage tends toward small, tightly contained clusters. Since it is driven by the largest possible distance between two clusters, however, it can be sensitive to outlying observations. Thus it sometimes produces clusters with elements less like each other and more like those in an adjoining cluster, violating a basic premise of cluster analysis.

A compromise between the two strategies is *average linkage* or *average-link clustering*, defined via

$$\delta_{\text{Avg}}(C_k, C_m) = \frac{1}{\nu_k \nu_m} \sum_{i \in C_k} \sum_{h \in C_m} d(\mathbf{z}_i, \mathbf{z}_h), \tag{11.1}$$

where ν_k and ν_m are the numbers of elements in clusters C_k and C_m, respectively. Average linkage can be shown to approximate a measure of integrated cluster dissimilarity as n grows large

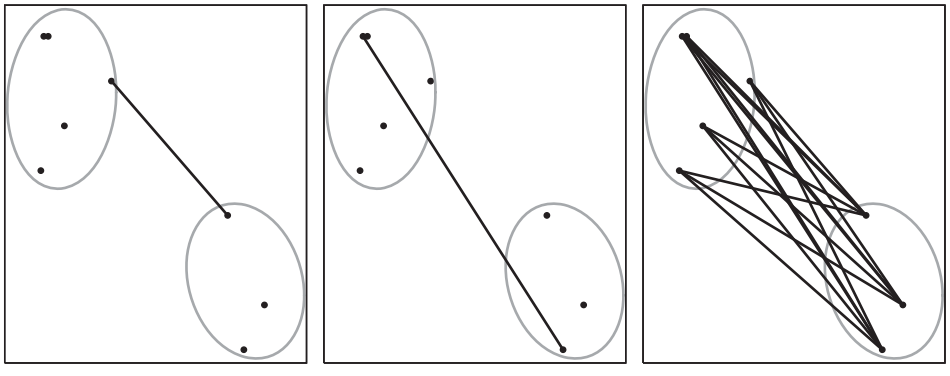

Figure 11.2 Different idealized forms of linkage for hierarchical clustering. (a) Single linkage, (b) complete linkage and (c) average linkage.

(Hastie et al. 2009, Section 14.3.12) and represents a middle ground between the single- and complete-link extremes. Its mediating features make it a popular choice in many applications.

Figure 11.2 illustrates these three linkage scenarios, using two abstracted clusters with the automobile data from Table 11.1.

A number of other options exist for characterizing the linkage distance $\delta(C_k, C_m)$, for use in selected or specialized applications. For instance, define the *centroid* of a cluster as the p-variate arithmetic mean (row) vector across all its elements: $\overline{\mathbf{z}}_k = [\overline{z}_{+1k} \quad \overline{z}_{+2k} \quad \cdots \quad \overline{z}_{+pk}]$, where $\overline{z}_{+jk} = \sum_{i \in C_k} z_{ij} / \nu_k$. Then, *centroid linkage* employs

$$\delta_{\text{Cent}}(C_k, C_m) = d(\overline{\mathbf{z}}_k, \overline{\mathbf{z}}_m)$$

(Clarke et al. 2009, Section 8.2.1). Here, $d(\overline{\mathbf{z}}_k, \overline{\mathbf{z}}_m)$ indicates application of the selected metric from Table 11.2 to find the 'distance' between the centroids $\overline{\mathbf{z}}_k$ and $\overline{\mathbf{z}}_m$. Other forms of linkage distance include the use of medians to define the location of each cluster, or of flexible, parameterized functions to define the intercluster distances (Lance and Williams 1967); see Hubert (2006).

Changing the linkage specification also changes the underlying definition of 'closeness' or 'affinity' between the developing clusters. Thus different choices for the linkage result in different – sometimes substantially different – hierarchical patterns (see Example 11.1.3). From an exploratory perspective, however, this can be as much a blessing as a curse: application of alternative linkage specifications to the same data sometimes uncovers interesting or unexpected features, which in turn helps to advance the knowledge discovery process.

With the architecture defined for the interobservation dissimilarity metric $d(\mathbf{z}_i, \mathbf{z}_h)$ and the intercluster linkage $\delta(C_k, C_m)$, the formal algorithm for agglomerative hierarchical clustering is actually fairly straightforward (Everitt 2005, Section 6.2):

AH.1 Start with $K = n$ clusters, each containing a single, distinct observation: $C_k = \{\mathbf{z}_k\}$, $k = 1, \ldots, n$.

AH.2 Find the two clusters C_{k^*} and C_{m^*} exhibiting the smallest linkage separation $\min_{k,m}\{\delta(C_k, C_m)\}$, and collapse them together. Consequently decrease K by 1.

AH.3 If $K = 1$ stop; otherwise return to step AH.2.

The result of an agglomerative clustering operation is typically displayed as a tree-like classification, called a *dendrogram*, similar to the decision trees in Section 9.4. The agglomerated $K = 1$ 'root' cluster is traditionally located at the top, with the smaller clusters branching down hierarchically and ending in the $K = n$ single-'leaf' clusters. (One can also reverse the display with the root at the bottom of the dendrogram, or with it at the side from left-to-right, etc.) A sliding scale alongside marks the linkage distances, and the dendrogram is constructed so that clusters separated by a given distance δ all align transversely at that δ on the distance scale (see Figure 11.3).

Example 11.1.2 Automobile characteristics (Example 11.1.1, continued). To illustrate the construction of a simple dendrogram, return to the automobile characteristics data from Table 11.1. Recall that this small data set has only $n = 8$ observations, with $p = 2$ outcome variables, $z_2 = \{\text{Standardized acceleration}\}$ plotted against $z_1 = \{\text{Standardized engine displacement}\}$ as in Figure 11.1. In the figure two clear clusters appeared evident.

Expanding on this visualization, consider application of agglomerative, hierarchical clustering. For the underlying distance metric, let $d(\mathbf{z}_i, \mathbf{z}_h)$ be standard Euclidean distance from Table 11.2. To calculate the underlying distances and collect them into a dissimilarity matrix **D**, one can use the **R** command

```
> D <- dist( cbind(z1,z2), method='euclidean', diag=TRUE )
```

where `z1` and `z2` are individual 8×1 column vectors with the standardized displacements and accelerations, respectively, bound together into an 8×2 **Z** matrix via the `cbind()` function. Given the interobservation distances in **D**, **R** can then perform agglomerative hierarchical clustering via its `hclust()` function. To construct the corresponding dendrogram, simply apply **R**'s `plot()` function to the `hclust()` object:

```
> plot( hclust(D, method='average'),
                          ylab=expression(delta), labels=Names )
```

where the `method='average'` option calls for average-link clustering as in (11.1) and `Names` is a character vector containing the model names of the eight automobiles to label the dendrogram leaves. Notice that `hclust()` does not employ the actual (standardized) observations z_{ij} to perform the cluster analysis; only information in the dissimilarity matrix **D** is required.

The resulting dendrogram appears in Figure 11.3. Working up from $\delta = 0$ in the figure, we see {Chevelle, Rebel} aggregate quickly into a two-element cluster, followed by {Impala, Fury}, and then by {Satellite, Torino}. Skylark and Galaxie remain as singletons until about $\delta = 0.76$, when Skylark joins the original {Chevelle, Rebel} cluster. At approximately $\delta = 0.92$, this three-element cluster merges with the {Satellite, Torino} pair, creating the low-displacement/slow-acceleration cluster visualized in Figure 11.1. Skylark remains a singleton until about $\delta = 1$, when it agglomerates with the {Impala and Fury} pair to produce the large-displacement/fast-acceleration cluster in Figure 11.1. These two clusters remain distinct until δ exceeds 2.6, when the entire collection merges into a single agglomerated cluster. Thus for values of an intercluster linkage distance between approximately 1.0 and

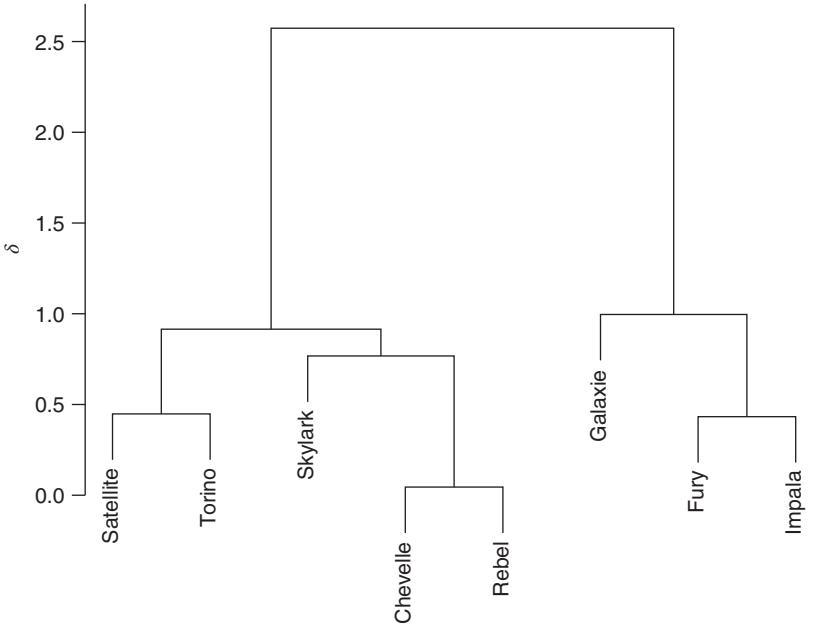

Figure 11.3 Average linkage dendrogram for Automobile Characteristics data in Example 11.1.2. Vertical scale is intercluster linkage distance, $\delta(C_k, C_m)$. Underlying distance metric is Euclidean, from Table 11.2. Source: Data from Myatt (2007, Table 6.8).

2.6 (a substantial portion of the δ scale here), the two-cluster visualization from Figure 11.1 is validated by this average-linkage dendrogram.

Exercise 11.1 explores how the dendrogram and the corresponding clusters change if single-link or complete-link clustering is instead employed with these data. □

Example 11.1.3 Clustering in gene expression data. An important application of cluster analysis occurs with gene expression levels and DNA microarray data (Hastie et al. 2009, Section 14.3.12); cf. Example 5.5.2. Clustering often focuses on the n different genes that represent the observations, where the genes' expression patterns are examined across $p > 1$ conditions such as different disease states and organ/tissue types, ecological species. Alternatively, n individual subjects may be clustered using expression data from p different genes felt to represent presumptive genetic markers. For the latter case, Uhlmann et al. (2012) described a study where $n = 118$ colorectal cancer patients had their relative gene expression ratios sampled for $p = 4$ different genes thought contribute to cancer progression: osteopontin (OPN), cyclooxygenase-2 (COX-2), transforming growth factor β (TGF-β), and matrix metalloproteinase-2 (MMP-2). Among the questions of interest to the investigators was whether clustering among the patients' expression patterns could reveal tumor subgroups or other carcinogenic effects. The data, sanitized to mask personal identifying information and kindly provided by Dr. Martin Sill of the Deutsches Krebsforschungszentrum in Heidelberg, Germany, appear in Table 11.3. (As previously, only a selection of the data is given in the table. The complete set of data is available at http://www.wiley.com/go/piegorsch/data_analytics.)

Table 11.3 Selected values for relative gene expression levels of $p = 4$ putative tumor progression genes, from $n = 118$ colorectal cancer patients.

	Gene			
Patient code no.	OPN	COX-2	Tgf-β	MMP-2
1	8.0528	60.1337	0.5105	6.0677
2	34.6786	221.1466	19.9325	3.4757
3	0.2449	1.8196	0.3016	1.0040
⋮	⋮	⋮	⋮	⋮
117	0.1259	1.3478	0.1125	0.5449
118	1.3660	8.8011	0.1983	1.9355

Source: Uhlmann et al. (2012).

Consider here agglomerative hierarchical clustering of these 118 patients' relative gene expression patterns using the $p = 4$ genes in Table 11.3. Begin by standardizing the original x_{ij}s in the table to z-scores: $z_{ij} = (x_{ij} - \bar{x}_j)/s_j$, $i = 1, \ldots, 118$; $j = 1, \ldots, 4$. Set the distance metric from Table 11.2 to Euclidean, and for now, employ average linkage.

For these settings, a dendrogram in **R** can be produced using commands similar to those from Example 11.1.2:

```
> D <- dist( Z, method='euclidean', diag=TRUE )
> plot( hclust(D, method='average'),
                    ylab=expression(delta), labels=Patient )
```

where `Patient` is a vector containing the patients' reference code numbers.

The average linkage dendrogram appears in Figure 11.4, where substantial initial coupling is evidenced. That is, for small values of δ most early clusters are simple pairs or triplets of patients whose expressions patterns are very close. The agglomeration accelerates, however, so that by only about $\delta = 0.15$ a large cluster of 46 patients has developed – delineated by the lower dashed cut-line in the figure. (See the small rectangle towards the lower center of the dendrogram.) To find the cluster membership labels for each patient, use **R**'s `cutree()` function, for example,

```
> cutree( hclust(D, method='average'), h=0.15 )
```

As δ increases, this core cluster continues to grow. At approximately $\delta = 2.75$ – delineated by the upper dashed cut-line in the figure – it has accumulated all but 10 of the patients under study. (See the larger rectangle in the dendrogram.) Specifics of the tumors and other patient characteristics from either this smaller/earlier cluster or the larger/agglomerated cluster – or, for that matter, from the remaining patients outside the larger cluster(s) – would be natural targets of further knowledge discovery with these data.

To contrast how other forms of linkage perform with these gene expression data, Figure 11.5 displays the results of single-link and complete-link clustering (still using the Euclidian distance metric). Comparing to Figure 11.4, clear structural differences are apparent. Most evident is the disparate pattern given by single linkage, where clusters congeal much faster at small levels of δ. Complete linkage gives a clustering pattern with some similarities to the average linkage result in Figure 11.4; however, obvious differences also

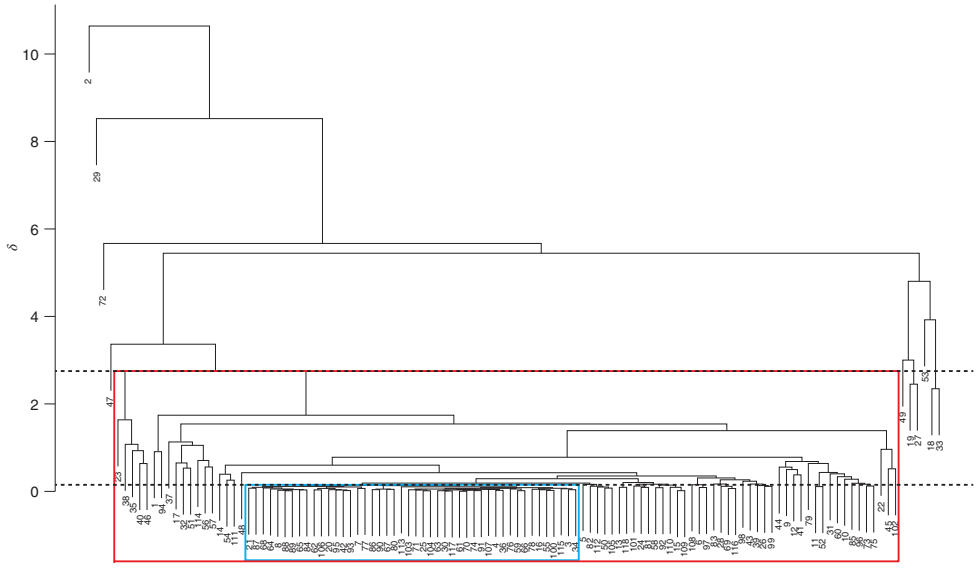

Figure 11.4 Average linkage dendrogram for gene expression data from Example 11.1.3. Underlying distance metric is Euclidean. Dashed lines at $\delta = 0.15$ and $\delta = 2.75$ suggest possible clustering breaks (see text). Corresponding rectangles mark the clusters. Dendrogram leaf digits (tiny terminal numbers) are patient codes. Source: Data from Uhlmann et al. (2012).

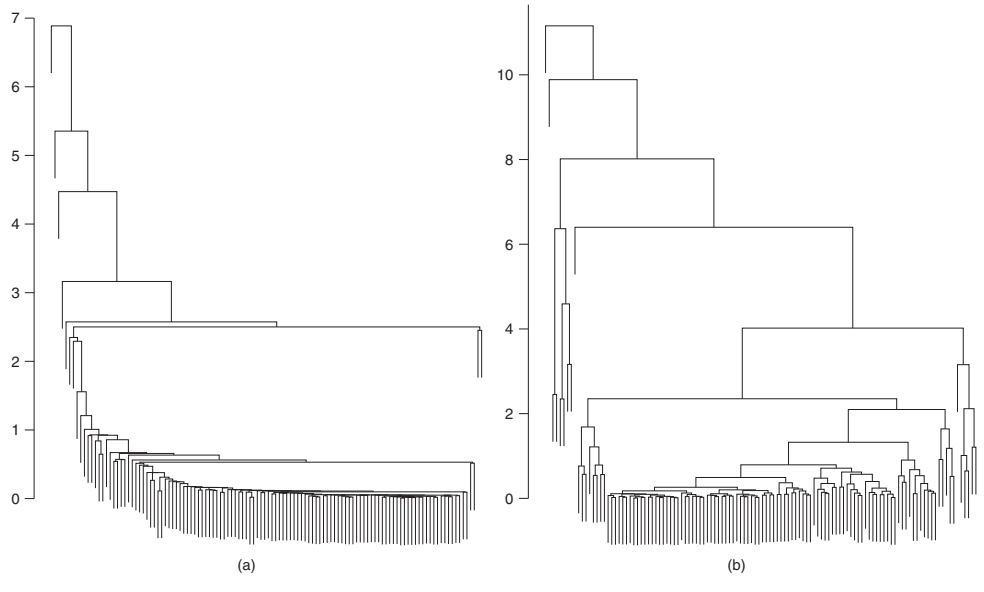

Figure 11.5 (a) Single linkage and (b) complete linkage hierarchical clustering dendrograms for gene expression data from Example 11.1.3. Underlying distance metric is Euclidean. Note difference in linkage distance vertical scales. For clarity of presentation, dendrogram leaf labels are suppressed. Source: Data from Uhlmann et al. (2012).

emerge. In particular, notice the group of patients towards the far left that remains separate from the larger cluster through much higher levels of δ.

On balance, this analysis corroborates the usual caveat that the three linkage methods can exhibit substantial differences in their hierarchical clustering patterns. (Also see Exercise 11.5.) □

As noted above, rather than hierarchically agglomerating from an initial set of K singleton clusters (starting with a single observation each), one can alternatively reverse the process and begin with a single cluster containing all n observations. Such *divisive clustering* hierarchically divides clusters based on widening linkage distances. Algorithms for divisive clustering tend to require more computation than those for agglomerative clustering, and no clear evidence has emerged that the divisive strategy creates more-precise clusters than the agglomerate approach. As a result, agglomerative clustering is often favored in practice. For more on divisive hierarchical clustering, see Clarke et al. (2009, Section 8.2.2).

11.1.2 Partitioned clustering

An alternative strategy for performing cluster analysis is to partition the observations into $K \geq 1$ distinctly encapsulated clusters within the p-variate space under study. (Some users apply the alternative term *segmentation*, when the goal is to partition the observations into a clear set of K focused, if unknown clusters.) As noted above, partitioned clustering is often how a novice analyst envisions the process of finding unsupervised clusters in multivariate data.

Partitioned clustering is somewhat distinct from the hierarchical methodology in the preceding section, and as a result, partitioning/segmentation algorithms are quite different. A critical assumption is that the number of clusters, K, be known in advance. For example, a business may wish to partition its customer base into K segments, corresponding to the K known members of its salesforce (Hastie et al. 2009, Section 14.3.11). Or, preliminary analyses may have identified K clear groupings in the data, such as K general body sizes for a manufacture's clothing lines, and the goal is to specify the cluster memberships for product manufacturing planning (Hand et al. 2001, Section 9.3). As with any cluster analysis, however, the goal remains to form clusters where the data are more similar to each other within a cluster and more different from each other between the K clusters.

Denote a given collection of clusters again as $\{C_1, \ldots, C_K\}$. For a measure of cumulative within-cluster variation, a common suggestion is

$$W(C_1, \ldots, C_K) = \sum_{k=1}^{K} \sum_{\substack{i \in C_k \\ h \in C_k}} d^2(\mathbf{z}_i, \mathbf{z}_h) \,, \tag{11.2}$$

where $d(\mathbf{z}_i, \mathbf{z}_h)$ is a distance metric taken, for example, from Table 11.2. (Note that some authors employ the original distance $d(\mathbf{z}_i, \mathbf{z}_h)$, rather than its square, in (11.2).) The corresponding cumulative between-cluster variation is

$$B(C_1, \ldots, C_K) = \sum_{k=1}^{K} \sum_{\substack{i \in C_k \\ h \notin C_k}} d^2(\mathbf{z}_i, \mathbf{z}_h) \,,$$

where $W(C_1, \ldots, C_K)$ and $B(C_1, \ldots, C_K)$ sum to the total variation

$$T = W(C_1, \ldots, C_K) + B(C_1, \ldots, C_K).$$

The goal then is to minimize $W(C_1, \ldots, C_K)$ or, equivalently, maximize $B(C_1, \ldots, C_K)$ for a given K. (If the domain-specific application calls for optimization of a more specialized target 'score' related to within- and between-cluster variation, a number of possibilities exist, for example, minimize $W(C_1, \ldots, C_K)/B(C_1, \ldots, C_K)$. See Hand et al. (2001, Section 9.4) for an instructive discussion.)

Clarke et al. (2009, Section 8.1.1), following Hastie et al. (2009, Section 14.3.6), noted that if, in (11.2), $d(\mathbf{z}_i, \mathbf{z}_h)$ is taken as Euclidean distance, then the cumulative within-cluster measure can be written as a sum of squared deviations:

$$W(C_1, \ldots, C_K) = 2 \sum_{k=1}^{K} \sum_{i \in C_k} (\mathbf{z}_i - \bar{\mathbf{z}}_k)^2 , \tag{11.3}$$

where, as given earlier, $\bar{\mathbf{z}}_k$ is a row vector representing the kth cluster's centroid

$$\bar{\mathbf{z}}_k = [\bar{z}_{+1k} \quad \bar{z}_{+2k} \quad \cdots \quad \bar{z}_{+pk}] \tag{11.4}$$

with elements $\bar{z}_{+jk} = \sum_{i \in C_k} z_{ij} / \nu_k$ and where ν_k is the number of observations in the kth cluster. Use of Euclidean distance is common here, and focus for the remainder of this section is on minimization of the cumulative within-cluster sum of squares in (11.3).

Ostensibly, minimizing (11.3) for fixed K is a straightforward combinatorial exercise: calculate the cumulative within-cluster sum of squares $W(C_1, \ldots, C_K)$ for all possible assignments of the n observations $\mathbf{z}_i$ into K different clusters, and choose that assignment minimizing $W(C_1, \ldots, C_K)$. This conceptual simplicity conceals some potentially extensive computations, however. The number of possible assignments is

$$\frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^n$$

(Jensen 1969), and for even moderately sized values of n, this quickly becomes intractable. For example, with $n = 25$ and $K = 3$, the number is roughly 10^{11}. In practice, analysts search for the minimum only over a portion of the assignment space. A popular algorithm that achieves this goal is known as K-means clustering (Hastie et al. 2009, Section 14.3.6):

KM.1. Designate a group of $K \geq 2$ different locations in the p-dimensional space under study to serve as the initial cluster 'centroids.' These can be a selection of K individual observations or other well-separated locations suggested by prior knowledge. (Avoid placing the initial points too close, as this can lead to unstable cluster outcomes.)

KM.2. For each observation $\mathbf{z}_i$, $i = 1, \ldots, n$, find the closest centroid $\bar{\mathbf{z}}_k$ and assign $\mathbf{z}_i$ to the cluster centered about that $\bar{\mathbf{z}}_k$.

KM.3. Calculate the new centroids $\bar{\mathbf{z}}_k$ from (11.4) for the resulting K clusters.

KM.4. Iterate through Steps KM.2 and KM.3 until the cluster assignments do not change.

In Step KM.2, 'closeness' between the point $\mathbf{z}_i$ and the centroid $\overline{\mathbf{z}}_k$ is usually quantified by Euclidean distance

$$\left[\sum_{j=1}^{p}(z_{ij}-\overline{z}_{+jk})^2\right]^{1/2}.$$

Notice that in vector notation, this can be written as $[(\mathbf{z}_i^{\mathrm{T}}-\overline{\mathbf{z}}_k^{\mathrm{T}})^{\mathrm{T}}(\mathbf{z}_i^{\mathrm{T}}-\overline{\mathbf{z}}_k^{\mathrm{T}})]^{1/2}$ for the row vector $\mathbf{z}_i$. If strongly heterogeneous correlation exists among the underlying observations, one can adjust by appealing instead to *Mahalanobis distance*

$$\left[(\mathbf{z}_i^{\mathrm{T}}-\overline{\mathbf{z}}_k^{\mathrm{T}})^{\mathrm{T}}\hat{\Sigma}^{-1}(\mathbf{z}_i^{\mathrm{T}}-\overline{\mathbf{z}}_k^{\mathrm{T}})\right]^{\frac{1}{2}}, \tag{11.5}$$

where $\hat{\Sigma}$ is the sample covariance matrix with (j,q)th element

$$s_{jq}=\frac{1}{n-1}\sum_{i=1}^{n}(z_{ij}-\overline{z}_j)(z_{iq}-\overline{z}_q)\ ,$$

$j,q=1,\ \ldots\ ,p$ (Clarke et al. 2009, Section 8.1.1). When the z_{ij}s are z-scores, however, we know $\overline{z}_j=0$ for all j and so some simplification is possible. Indeed, as noted above, the sample covariance matrix for a matrix $\mathbf{Z}$ of standardized z-scores z_{ij} is just the correlation matrix $\mathbf{R}=\mathbf{Z}^{\mathrm{T}}\mathbf{Z}/(n-1)$. Thus (11.5) reduces to $[\mathbf{z}_i\mathbf{R}^{-1}\mathbf{z}_i^{\mathrm{T}}]^{1/2}$.

K-means is a form of 'greedy search' or 'greedy descent' optimization; it attempts to step towards improved values of the cumulative within-cluster sum of squares as it iterates through the developing clusters. From a computational efficiency standpoint, K-means can drastically reduce the execution time required to report a K-cluster solution, and especially for very large p (high dimension) or large n ('big' data), it is often the only practical choice for performing partitioned clustering.

Unfortunately, the K-means algorithm cannot guarantee an optimal solution, and it does occasionally get caught at a local minimum. A common suggestion is to run the algorithm from a number of starting points and choose the clustering solution with the lowest cumulative within-cluster sum of squares. Clarke et al. (2009, Section 8.1.1) recommended using at least $\min\{10,n\}$ different starting positions.

As a computing algorithm, K-means developed in the mid-twentieth century from multiple sources and over a variety of disciplines. As a result, the algorithm exists in different forms and has many modified versions, each focusing on a specific manipulation of the basic greedy descent strategy. Early examples include suggestions by Lloyd (1957, 1982), Forgy (1965), and MacQueen (1967); also see Jain and Dubes (1988, Section 3.3) and Hand et al. (2001, Section 9.7). A version often favored by data analysts is given by Hartigan and Wong (1979), where an observation is prevented from moving between two clusters if the resulting solution increases the cumulative within-cluster sum of squares. The Hartigan and Wong algorithm is in fact the default in **R**'s `kmeans()` function.

Example 11.1.4 Automobile characteristics (Example 11.1.2, continued). To illustrate the K-means algorithm for partitioned cluster analysis, return again to the automobile characteristics data in Table 11.1. Recall that this simple data set has only $n=8$ observations, with $p=2$ outcome variables, $z_2=\{\text{Standardized Acceleration}\}$ plotted against $z_1=\{\text{Standardized Engine Displacement}\}$ as in Figure 11.1. In the figure, two clear clusters

appeared evident, so set $K = 2$. To perform a K-means cluster analysis in **R**, we can apply the commands

```
> Z <- cbind( z1, z2 )
> kmeans( Z, centers=2, nstart=8, algorithm='Hartigan-Wong' )
```

where `Z` is the 8×2 matrix of standardized observations built from the individual-variable (column) vectors. The option `centers=2` instructs **R** to set $K = 2$; because this is a number (as opposed to a vector of initial centroid values) **R** chooses a random set of rows from `Z` as the initial cluster centers. To avoid having this result in a local minimum, the `nstart=8` option calls for eight different initial centroids. The resulting output from `kmeans()` (edited for presentation) is

```
K-means clustering with 2 clusters of sizes 5, 3
Cluster means:
          z1         z2
1 -0.702148  0.6338996
2  1.170247 -1.0564993

Clustering vector:
[1] 1 2 2 2 1 1 1 1

Within cluster sum of squares by cluster:
[1] 1.3297430 0.7390515
 (between_SS / total_SS =  85.2 %)
```

The output provides the final cluster centroids as $\bar{\mathbf{z}}_1 = (-0.702, 0.634)$ and $\bar{\mathbf{z}}_2 = (1.170, -1.056)$. The corresponding individual, within-cluster sums of squares are given as $\sum_{i \in C_1} (\mathbf{z}_i - \bar{\mathbf{z}}_1)^2 = 1.330$ and $\sum_{i \in C_2} (\mathbf{z}_i - \bar{\mathbf{z}}_2)^2 = 0.739$, so that $W(C_1, C_2) = 1.330 + 0.739 = 2.069$. (This latter quantity is available as the `tot.withinss` attribute from the **R** object created by the call to `kmeans`.)

Cluster membership is given by the location of indices in the `Clustering vector`. The sequence `1 2 2 2 1 1 1 1` indicates that cluster C_1 contains automobiles with indices $i = 1, 5, 6, 7, 8$, while cluster C_2 contains those with indices $i = 2, 3, 4$. The data were entered into **R** such that these indices correspond to the top-to-bottom order presented in Table 11.1; thus C_1 contains {Chevelle, Rebel, Satellite, Skylark, Torino} and C_2 contains {Fury, Galaxie, Impala}. This is, of course, the same low-displacement/slow-acceleration versus large-displacement/fast-acceleration delineation identified in Example 11.1.1. It is also the segmentation available in the hierarchical average-linkage dendrogram from Figure 11.3 when values of the linkage lie between $\delta = 1$ and $\delta = 2.6$.

A further partitioning into $K = 3$ clusters is worth a brief investigation here. The effort is simple enough: in **R**, simply change to `centers=3` when invoking `kmeans`. The resulting output (edited for presentation) is

```
K-means clustering with 3 clusters of sizes 3, 3, 2
Cluster means:
          z1         z2
1 -0.6401349  0.9618874
2  1.1702466 -1.0564993
3 -0.7951676  0.1419178
```

```
Clustering vector:
[1] 1 2 2 2 1 3 1 3

Within cluster sum of squares by cluster:
[1] 0.3936573 0.7390515 0.1004234
 (between_SS / total_SS =  91.2 %)
```

The cumulative within-cluster sum of squares has now dropped to $W(C_1, C_2, C_3) = 0.3934 + 0.739 + 0.100 = 1.233$ (which is not surprising: W generally decreases with increasing K), and the ratio of cumulative between-cluster sum of squares to total sum of squares has increased from 85.2% to 91.2%.

From the `Clustering vector` output of `1 2 2 2 1 3 1 3`, we see the trifurcation into $K = 3$ clusters has moved autos with indices $i = 6, 8$ out of C_1 into their own, separate cluster C_3. It has also left the large-displacement/fast-acceleration cluster C_2 unchanged. That is, C_1 now contains only {Chevelle, Rebel, Skylark}, C_2 continues to harbor {Fury, Galaxie, Impala}, and the new C_3 houses {Satellite, Torino}. Referring to Figure 11.1, this has further segmented the low-displacement/slow-acceleration automobiles along an acceleration gradient: C_1 is a very slow-acceleration cluster, while C_3 contains low-displacement vehicles with moderate acceleration.

Intriguingly, this three-cluster partitioning differs from the three-cluster arrangement suggested by the average-linkage dendrogram in Figure 11.3: there, the hierarchical construction retains the five-element low-displacement/slow-acceleration cluster {Chevelle, Rebel, Satellite, Skylark, Torino} but separates Galaxie into a singleton cluster, leaving {Fury, Impala} as a paired high-displacement/very fast-acceleration cluster. This illustrates the common warning that the different strategies for cluster analysis can sometimes produce conflicting results. Still, disparities such as these serve as useful springboards for exploring the underlying features of the data and may lead to new knowledge discoveries.

A graphical display of the separated clusters is known as a *Voronoi tessellation* or a *Voronoi diagram* (Schoenberg 2012). This is a formal construction that uses optimally placed line segments to partition the feature space into disjoint 'cells,' where any point in the *i*th cell is closer to that cell's centroid than to any other centroid. Obviously, this overlaps with many of the goals of a partitioned cluster analysis.

In **R**, the external *deldir* package can produce Voronoi tessellations (also called *Dirichlet tessellations*) via its `deldir()` function. Sample code for the automobile characteristics data is

```
> auto8.km <- kmeans( cbind(z1,z2), centers=3, nstart=8,
                                  algorithm='Hartigan-Wong' )
> require( deldir )
> auto8.dd <- deldir( auto8.km$centers[,1],
                      auto8.km$centers[,2] )
> plot( auto8.dd, wlines='tess', lty='solid' )
```

Figure 11.6 displays the tessellation for the three-cluster partitioning, using information from the three-cluster `kmeans()` output. The original data are overlaid for enhanced visualization.

Tessellations are particularly useful in geographic analytics for partitioning cartographic clusters, because the vertices of the tessellation's polygons represent optimized distances from cluster centroids. They can be used to service nearest-neighbor queries when, for example,

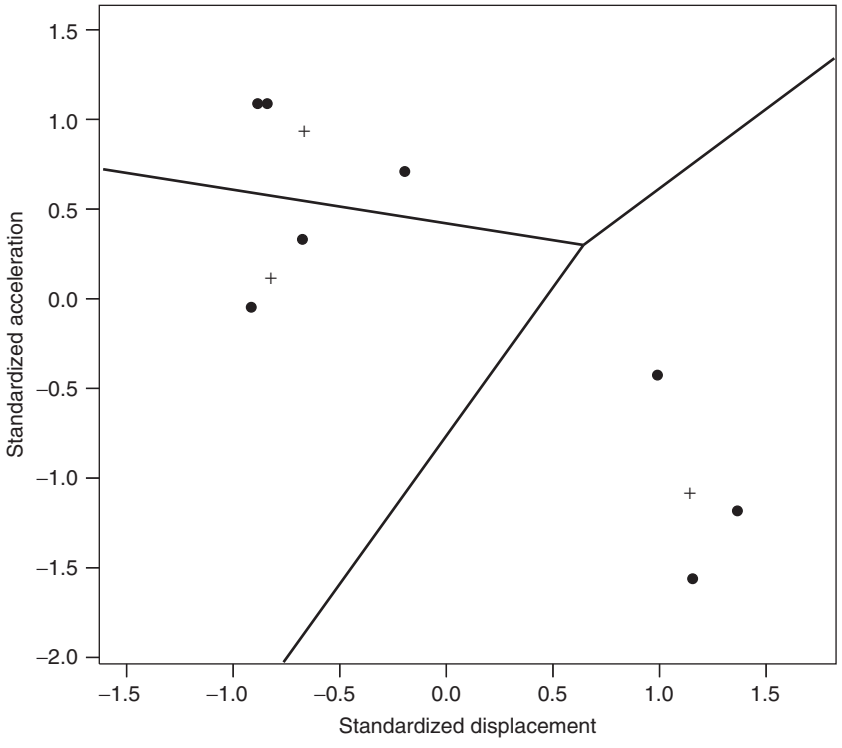

Figure 11.6 Voronoi tessellation for automobile characteristics data in Table 11.1. Three clusters from the K-means analysis in Example 11.1.4 are separated by tessellated line segments to produce cluster boundaries. Crosses (+) indicate cluster centroids. Dark circles (•) are overlaid scatterplot of $z_2 = \{\text{Standardized acceleration}\}$ against $z_1 = \{\text{Standardized engine displacement}\}$ (slight jitter added to distinguish adjacent points). Source: Data from Myatt (2007, Table 6.8).

planning mass-transit circuits, separating water supplies from hazardous waste sites, locating mobile telephone towers. For more details, see Linoff and Berry (2011, Chapter 13). □

When K is unknown, it may still be possible to perform a partitioned cluster analysis, although the effort becomes even more exploratory in nature. One simply conducts the partitioning for each value of $K = 1, \ldots, n-1$ and saves the cumulative within-cluster sum of squares at each K, say, $W_K(C_1, \ldots, C_K)$. Since W_K is generally decreasing in K (indeed, $W_n = 0$), so is any monotonic measure based on W_K, such as $\log\{W_K\}$ or

$$\frac{W_K}{\mathrm{T} - W_K}.$$

A plot of W_K or $\log\{W_K\}$ versus K often shows an abrupt change in the decreasing pattern as $K \to n$. Similar to the scree plot in principal components analysis (Section 10.2.2), if such a precipitous 'elbow' appears, that point may represent a putative choice for the unknown K.

Another popular measure is the 'pseudo-F' statistic (Caliński and Harabasz 1974):

$$F = \frac{(\mathrm{T} - W_K)/(K-1)}{W_K/(n-K)}\,, \tag{11.6}$$

so-named because it imitates the F-ratio in (7.18) from an analysis of variance. When plotted against K, a large local spike in (11.6) indicates that between-cluster variation is high relative to within-cluster variation at that K. This becomes another possibility for the unknown number of clusters and can validate a choice of K when, for example, both an elbow plot and a pseudo-F plot suggest the same value. The pseudo-F is straightforward to calculate directly and is also available in **R** via, for example, the `index.G1()` function of the external *clusterSim* package.

Selection of the number of clusters in an exploratory cluster analysis is an active area of analytic and statistical research, and advances occur on a regular basis. See, for example, Koepke and Clarke (2013), Gopal et al. (2012), and the reference therein.

Example 11.1.5 Home center financial performance. Cluster analysis is a popular methodology for unsupervised learning with business and financial data. For example, Clark (2010) compiled business performance data for calendar year 2009 on $n = 497$ US companies engaged in home center/building supply sales. Recorded were the following variables:

x_1 = Annual sales (in \$M)
x_2 = Sales growth (percentage of increase or decrease from previous calendar year)
x_3 = Number of store units (dedicated to or including home/building supply)
x_4 = Number of employees (staffing home/building supply)

Table 11.4 displays the data. (As previously, only a selection of observations is provided in the table. The complete set of data is available at http://www.wiley.com/go/piegorsch/data_analytics.) Notice that many companies reported decreases in sales from calendar 2008 to 2009, as measured by variable x_2, commensurate with concurrently tight conditions in the US economy. This was part of the focus in this particular study.

Consider here a partitioned cluster analysis via the standard K-means algorithm, to determine if any special (or obvious) patterns exist in these data. Begin by standardizing the $p = 4$

Table 11.4 Selected business performance data for calendar year 2009 on $p = 4$ financial variables, from $n = 497$ US companies engaged in home center/building supply sales.

	Outcome variables			
Company	x_1 = Sales (\$M)	x_2 = % change	x_3 = Units	x_4 = Staff
Home Depot	66 176.0	−7.2	2244	375000
Lowe's	47 220.0	−2.0	1710	238000
Wal-Mart	25 700.0	−1.2	3708	143800
⋮	⋮	⋮	⋮	⋮
Peek's Carpet & Tile	20.5	61.4	9	30
Theut Products	20.0	0.0	7	105

Source: Clark (2010).

variables from Table 11.4 into z_1, z_2, z_3, and z_4. To set a value for the number of clusters, take a series of K-means fits over increasing K up to, say, $K = 6$ and explore some of the measures mentioned earlier. A typical **R** command for each fit could be

```
> Z <- cbind( z1, z2, z3, z4 )
> kmeans( Z, centers=4, nstart=100, algorithm='Hartigan-Wong' )
```

for $K = 4$. As the number of observations is reasonably large here, the command includes `nstart=100` starting values to help to ensure a valid clustering solution. Varying this over other values of K and extracting the overall within-cluster sum of squares W_K (in **R**, this is the `tot.withinss` attribute from the `kmeans()` object) leads to an elbow plot of W_K versus K. Figure 11.7 displays the plot, along with an alternative pseudo-F plot for determining K. Both graphics indicate selection of $K = 4$ for the number of clusters with these data.

The four-cluster solution from `kmeans()` gives the following **R** output (edited for presentation):

```
K-means clustering with 4 clusters of sizes 2, 260, 5, 230
Cluster means:
          [,1]        [,2]       [,3]        [,4]
1  14.49536999  0.07695562  5.5630081 14.34938073
2  -0.07905351  0.85948194 -0.1080104 -0.07761037
3   2.45768662 -0.79002619  8.9206856  2.50025098
4  -0.09010983 -0.95508298 -0.1202032 -0.09139704
Within cluster sum of squares by cluster:
[1] 35.12519 66.59292 62.08312 48.57776
 (between_SS / total_SS =  89.3 %)
```

Notice the very small cluster sizes for cluster C_1 (two elements) and C_3 (five elements). The corresponding 497-element `Clustering vector` (not shown; also available as the `cluster` component from the `kmeans()` object) gives the four-cluster delineation as summarized

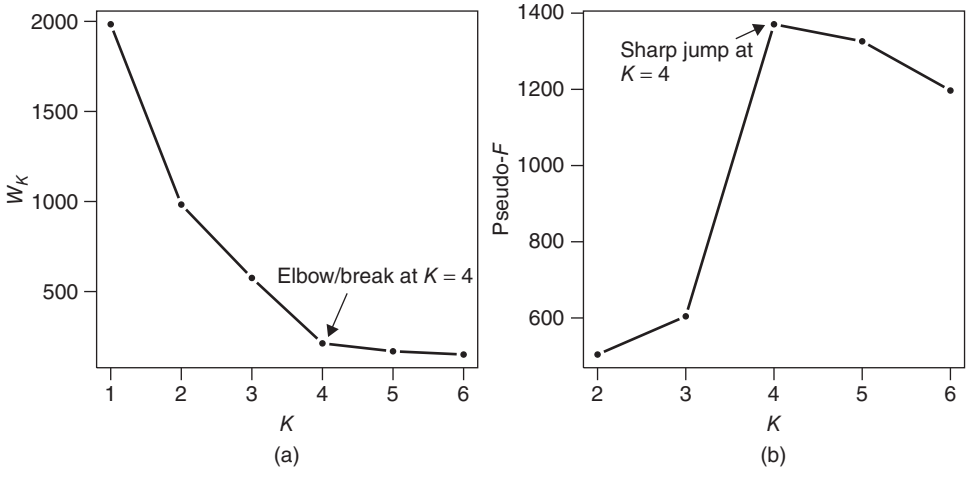

Figure 11.7 (a) Cumulative within-cluster sum of squares W_K and (b) pseudo-F statistic plotted against number of clusters, K, for home center data in Example 11.1.5. Both plots indicate selection of $K = 4$.

Table 11.5 Four-cluster K-means solution for business performance data from Table 11.4.

Cluster	Cluster size	Companies (for larger clusters, only selected elements are given)
C_1	$\nu_1 = 2$	Home Depot, Lowe's
C_2	$\nu_2 = 260$	Menards, Dai-Tile, … , Peek's Carpet & Tile, Theut Products
C_3	$\nu_3 = 5$	Wal-Mart, CCA Global Partners, Sears, Sherwin-Williams, Fastenal
C_4	$\nu_4 = 230$	ProBuild Holdings, Sutherland Lumber, … , 41 Lumber, Allen & Allen

in Table 11.5. We see that the two megastores, Home Depot and Lowe's, are the only components of cluster C_1, which is not surprising given their substantial presence in this market. The other small cluster, C_3, is made up of very large North American companies that either have extensive, specialized commitments to building and hardware supply, such as Fastenal (fasteners and hardware) and Sherwin-Williams (paints/paint supplies), or are large businesses with extensive hardware/home center subunits (Wal-Mart, CCA Global Partners, Sears).

The remaining two clusters in Table 11.5, C_2 and C_4, are almost evenly split in terms of size; each makes up slightly less than half of the total collection. Cluster C_2 appears to house some of the larger companies in terms of physical resources. For example, the following **R** commands and output summarize the separate store staff counts (from Table 11.4) for each cluster, with accompanying output:

```
> HomeK4.km <- kmeans( Z, centers=4, nstart=100 )
> summary( Staff[which(HomeK4.km$cluster == 2)] )
   Min. 1st Qu.  Median    Mean  3rd Qu.    Max.
   14.0   100.0   175.0   660.6    333.8 40000.0

> summary( Staff[which(HomeK4.km$cluster == 4)] )
   Min. 1st Qu.  Median    Mean  3rd Qu.    Max.
   20.0    85.0   130.0   368.4    250.0 11320.0
```

Almost every comparison measure is greater for cluster C_2, and these become substantial for the higher measures such as the upper quartile and maximum. Similar results occur for the separate store unit counts:

```
> summary( Units[which(HomeK4.km$cluster == 2)] )
   Min. 1st Qu.  Median    Mean  3rd Qu.    Max.
   1.00    4.00    7.00   24.55    16.00  930.00

> summary( Units[which(HomeK4.km$cluster == 4)] )
   Min. 1st Qu.  Median    Mean  3rd Qu.    Max.
   1.00    3.00    5.00   20.35    10.00  474.00
```

To visualize the four-cluster solution's effect on the original data, Figure 11.8 presents a scatterplot matrix with clusters labeled in each panel. The plot was constructed using the **R** command

```
> pairs( Home.df,
         panel=function(x,y) {text(x,y,HomeK4.km$cluster)} )
```

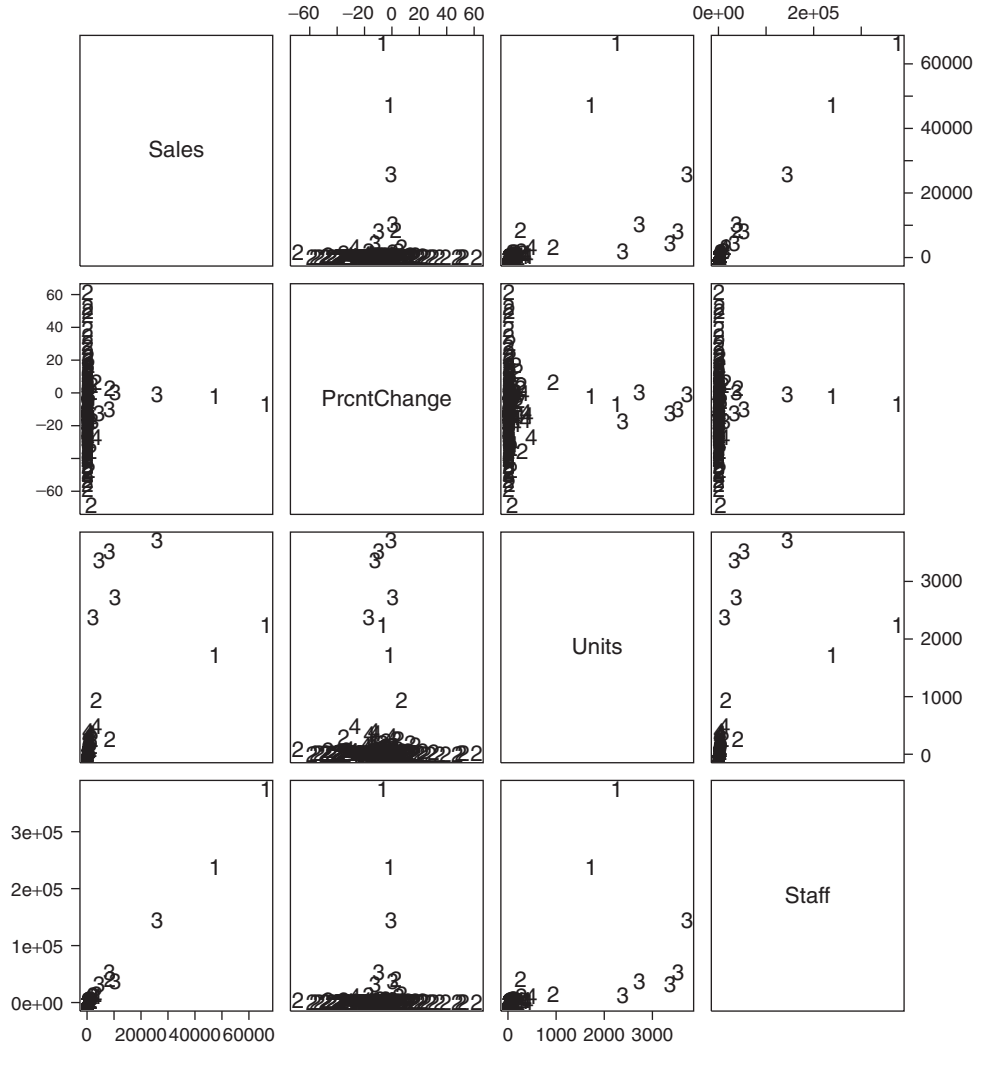

Figure 11.8 Scatterplot matrix for home center data in Example 11.1.5 (original scales) with points labeled via K-means cluster assignments from four-cluster solution. Source: Data from Clark (2010).

where `Home.df` is the original dataframe containing the $p = 4$ input variables and `HomeK4.km` is the four-cluster K-means solution object (see above).

Figure 11.8 shows that the clustering has evident validity: points associated with clusters C_1 and C_3 separate visibly from the rest in essentially every pairwise plot panel. Some points from cluster C_2 are also visible, although the skewed scale for the Sales, Units, and Staff variables tends to compact the C_2 and C_4 observations. Exercise 11.8 analyzes the data using logarithmic scales to improve resolution with these tightly condensed groupings. □

It is important to warn that as with any analytic method that employs arithmetic means, the K-means algorithm is susceptible to outliers. To compensate, alternative algorithms call on more-robust measures to calculate the cluster centers. For example, the K-medoids algorithm replaces the centroid $\overline{\mathbf{z}}_k$ in (11.3) with the *medoid*, $\mathbf{m}_k$, of each kth cluster. The medoid (not to be confused with the median; see next paragraph) of a cluster is defined as the individual observation within the cluster whose average distance from all other observations in the cluster is a minimum. Distance can be Euclidean, Mahalanobis, or any other desired measure of dissimilarity. The medoids are then used to represent the cluster centers, in place of the outlier-vulnerable centroids. K-medoid clustering is available in a number of external **R** packages, including *cluster* and *fpc*.

Another alternative to K-means takes a step further. The *K-medians algorithm* employs a target score based on absolute-value distance, $W_K^* = \sum_{k=1}^{K} \sum_{i \in C_k} | \mathbf{z}_i - \tilde{\mathrm{Q}}_{2k} |$, where $\tilde{\mathrm{Q}}_{2k}$ is a cluster median (Clarke et al. 2009, Section 8.1.2). Expressions for $\tilde{\mathrm{Q}}_{2k}$ can vary, because there is no unique definition of a multidimensional median (see Small 1990). The external **R** package *depth* can calculate various forms of multivariate median, via its `med()` function. A form of K-medians clustering is available in the **R** package *flexclust*, through its `kcca()` function.

If the analyst is willing to adopt a formal probability model for the multivariate observations in $\mathbf{X}$, the cluster analysis can be expanded to incorporate this information. In the process, the results can become much more powerful – but much more dependent on the underlying model – and expand the basic clustering paradigm past its exploratory roots. Such *model-based clustering* often assumes that the multivariate observations are derived from a mixture of K probability density functions (p.d.f.s), where each kth p.d.f. models variation in observations from the kth cluster ($k = 1, \ldots, K$). If the forms of the p.d.f.s are known, such as K p-variate normal densities (Section 2.3.9) with possibly different means, variances, and/or covariances, then maximum likelihood estimation (Section 5.2.4) can be applied to estimate the unknown parameters. If K is itself assumed unknown it can also be estimated, using various model selection procedures. As might be expected, the calculations for model-based clustering appeal to some rather sophisticated statistical techniques, and a full discourse exceeds the scope here. Interested readers can find details in Everitt (2005, Section 6.4) or Clarke et al. (2009, Section 8.3.1).

Clearly, many different techniques are available to perform exploratory cluster analysis. No matter the strategy taken, however, it is important to warn that no catch-all method exists. Each approach has its own strengths and weaknesses. In practice, the underlying clusters can sometimes involve complex forms and shapes: think of a circular cluster completely surrounded by a separate, ringed cluster (often called 'bull's-eye' clustering; cf. Figure 9.10a) or a pair of separate crescents pushed together so that one end of each crescent tucks into the bowl of the other, creating a larger, sigmoidal pattern ('half-moon' clustering). Unusual shapes such as these can be difficult to identify, especially if the number of dimensions p grows very large. (Another consequence of the curse of dimensionality mentioned in Section 7.4.1: in high-dimensional space, the data points become more sparse and our ability to identify connections among them drops precipitously; see Clarke et al. (2009, Section 1.0.1).) There may even be cases where no true clusters exist at all, but the clustering operation still reports potential segmentations in the data. Cluster analysts must approach the calculations with due caution – in some cases, even with a modicum of skepticism! – and be conscious

of the unsupervised aspects the learning effort entails. Hand et al. (2001, p. 296) provided a useful, succinct philosophy (with original emphasis):

> The important lesson … is that we must match the method to the objectives. In particular, we must adopt a cluster analytic tool that is effective at detecting clusters that conform to the definition of what is meant by 'cluster' in the problem at hand. It is perhaps worth adding that we should not be too rigid about [cluster identification]. Data mining, after all, is about discovering the *unexpected*, so we must not be too determined in imposing our preconceptions on the analysis. Perhaps a search for a different kind of cluster structure will throw up things we have not previously thought of.

For more on cluster analysis, see the various references throughout this section, the modern compendium by Everitt et al. (2011), and the classic text by Kaufman and Rousseeuw (1990).

11.2 Association rules/market basket analysis

As we have seen, many different methods exist to describe, summarize, and quantify associations among a series of multidimensional observations. Clustering (as in Section 11.1) studies patterns in terms of distances between objects, classification (Chapter 9) allocates objects to known categories based on observed similarities, correlation (Section 3.3.3) quantifies linear relationships between two variables, scatterplots and other graphical devices (Section 4.2) visualize relationships in two or more dimensions, and so on. Another analytic device for divining how objects connect with one another is known as *association rule learning*; it attempts to identify interesting or informative connections among a set of items or events as they relate in an implicating ('if $\mathcal{A}$ then $\mathcal{B}$') manner.

The fundamental concept for association rules arose in studies of customer buying patterns; hence, the coadunate moniker *market basket analysis* when association rules are applied in consumer studies. An oft-cited scenario is purchasing behavior in grocery stores, where customers buy 'market baskets' of various grocery items – eggs, milk, lettuce, and so on – in assorted patterns. Some purchasing patterns are obvious – say, (almost) always buying salad dressing when buying lettuce – while others can prove surprising and instructive. For instance, a famous example identified a pattern with discount department megastore Wal-Mart's customers: apparently, purchasers of Barbie® dolls often also bought one of three kinds of candy bars. While the underlying motivations for such an association were ambiguous (Palmeri, C., 'Believe in yourself, believe in the merchandise,' *Forbes* **160** (5), 8 September 1997, pp. 118–124), the connotations were clear to Wal-Mart's managers: locate more candy bar displays either closer or on a pathway to the Barbie aisle. (Another, infamous example related late-afternoon purchases of diapers with concomitant acquisitions of beer, although this appears to be more of an urban myth than a success of association rule mining; see Rao, S.S., 'Birth of a legend,' *Forbes* **161** (7), 6 April 1998, pp. 128–130.)

Originated by Agrawal et al. (1993) for the consumer setting, association rules have expanded to many application areas, including biomedicine and genetics, architectural design, geoinformatics, and communication and information systems, to name just a few. In this section, an introduction is given to this rapidly evolving methodology in its most basic form.

11.2.1 Association rules for binary observations

Suppose a database of n 'transactions' (i.e., observations) $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ is recorded over p possible items $\mathcal{I} = \{I_1, I_2, \ldots, I_p\}$. The items could be specific purchases such as milk and lettuce (or diapers and beer!), or other characteristics such as purchaser's sex, new/repeat customer, high-school graduate, and so on. Item categories may also be extended to other settings; for example, in medical studies, one might query disease status, prescribed medications, insured/uninsured, and so on. Collect the data into an $n \times p$ matrix $\mathbf{X}$, where the component observations are binary: $x_{ij} = 1$ if transaction i resulted in purchase/activation of item j, and 0 otherwise.

As a simple touchstone example, consider the hypothetical transaction database for grocery shopping excursions given by the matrix in Figure 11.9. Notice how the $n = 12$ transactions differ across the $p = 9$ items (although transactions T_3 and T_7 are in fact identical).

Patterns of concomitant purchases among the rows of $\mathbf{X}$ are called *itemsets*. For example, an itemset in a grocery store market analysis could be {lettuce, salad dressing} or {eggs, lettuce, salad dressing}. Generically, an itemset is any collection of $k \geq 1$ items $\mathcal{A}_k = \{I_{j_1}, I_{j_2}, \ldots, I_{j_k}\} \subset \mathcal{I}$ thought to exhibit potential associations among the n transactions. The goal is to mine the transaction database and uncover useful relationships – called *association rules* – among items that are (or are not) purchased together.

In the simple setting described here, a formal association rule connects a k-fold itemset $\mathcal{A}_k$ with a singleton itemset $\mathcal{B}_1$ via an implicative association: $\{\mathcal{A}_k \Rightarrow \mathcal{B}_1\}$. That is, purchase/activation of $\mathcal{A}_k$ is presumed to induce purchase/activation of $\mathcal{B}_1$. The left-hand side ('LHS') of the rule is called the *antecedent* and the right-hand side ('RHS') is the *consequent*. (Actually, the directed implication here is artificial, because the database does not provide a causal relationship for the purchase of $\mathcal{A}_k$ *leading to* purchase of $\mathcal{B}_1$. Technically, the data themselves only indicate whether the two itemsets are purchased in the same transaction. Causal motivations must be taken from external knowledge or assumptions on the customers'

Items
I_1 = Eggs
I_2 = Milk
I_3 = Cheese
I_4 = Lettuce
I_5 = Salad dressing
I_6 = Onions
I_7 = Pepper
I_8 = Mustard
I_9 = Ice cream

	Items								
Transactions	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9
T_1	0	0	1	1	1	1	1	1	0
T_2	1	0	0	1	0	0	0	0	0
T_3	0	1	0	0	0	0	0	0	0
T_4	1	0	0	1	1	0	0	0	0
T_5	0	1	1	1	1	1	0	0	1
T_6	0	1	0	1	1	0	1	0	0
T_7	0	1	0	0	0	0	0	0	0
T_8	0	1	1	0	0	0	1	0	0
T_9	1	1	0	1	0	1	0	0	0
T_{10}	1	1	1	1	1	1	0	0	1
T_{11}	0	1	0	1	1	0	0	1	1
T_{12}	0	1	1	0	0	1	1	0	1

Figure 11.9 Hypothetical market basket transaction database for $n = 12$ grocery store customers.

purchasing behaviors. To remain consistent with the association rule literature, however, the '⇒' notation is retained here.) For simplicity, $\mathcal{B}_1$ is restricted here to a single item, although the framework can be extended to include m-fold consequents, $\mathcal{B}_m$.

11.2.2 Measures of rule quality

Viewing the x_{ij}s as realizations of some binary random variable, the itemsets $\mathcal{A}_k$ and $\mathcal{B}_1$ become 'events.' Thus, the corresponding association rule $\{\mathcal{A}_k \Rightarrow \mathcal{B}_1\}$ has some probability of occurrence $P[\mathcal{A}_k \wedge \mathcal{B}_1]$, where the $\wedge$ symbol indicates a 'logical and' operation. An estimate of this probability is available from the database: simply divide the number of times $\{\mathcal{A}_k \Rightarrow \mathcal{B}_1\}$ is observed by the total number of transactions:

$$S(\mathcal{A}_k \Rightarrow \mathcal{B}_1) = \frac{\#\text{ occurrences of } \{\mathcal{A}_k \Rightarrow \mathcal{B}_1\}}{n} .$$

This is known as the *support* of the association rule $\{\mathcal{A}_k \Rightarrow \mathcal{B}_1\}$. (The term is not to be confused with the support space of a probability function (p.m.f. or p.d.f.) from Section 2.1.)

From this, the *accuracy* of the rule $\{\mathcal{A}_k \Rightarrow \mathcal{B}_1\}$ is the ratio of its support to the support of the antecedent $\mathcal{A}_k$:

$$A(\mathcal{A}_k \Rightarrow \mathcal{B}_1) = \frac{S(\mathcal{A}_k \Rightarrow \mathcal{B}_1)}{S(\mathcal{A}_k)} .$$

(This should not be confused with the similarlynamed accuracy measure from classification analytics in Section 9.1.2.) In effect, accuracy can be viewed as an estimate of the conditional probability $P[\mathcal{B}_1|\mathcal{A}_k]$. Strong association rules will have accuracies as large as possible, preferably close to 1.

Accuracy for an association rule is also called *confidence* by many in the machine learning literature. This is, of course, unfortunate usage because it has nothing to do with the confidence intervals of Section 5.3.1.

The support and accuracy of an association rule are measures for its overall quality. Rules that appear often in the transaction database have high $S(\mathcal{A}_k \Rightarrow \mathcal{B}_1)$ and, therefore, will involve large numbers of customers. Acting on a rule whose support is high, say, exceeding a minimum threshold $s_o \in (0, 1)$, has the potential to affect a large portion of the customer base. Further, rules with high accuracy appear better at predicting consequent purchases from their antecedents. Here too, acting on rules where $A(\mathcal{A}_k \Rightarrow \mathcal{B}_1)$ exceeds some minimum accuracy threshold $c_o \in (0, 1)$ brings the potential to affect more customers and increase future transactions (and, in market applications, revenues; see Section 11.2.3.

For example, with the hypothetical transactions in Figure 11.9, consider the simple association rule {lettuce ⇒ salad dressing}. That is, set $\{\mathcal{A}_1\} = \{I_4\}$ and $\{\mathcal{B}_1\} = \{I_5\}$. Inspection of the transaction matrix shows that the support here is $S(I_4 \Rightarrow I_5) = 6/12$ or 50%, a fairly common association in this database. (Support thresholds vary greatly in practice; values of s_o can range from as low as 1% up to 20% or even 50%, depending on the domain-specific targets.) Accuracy of the rule is then

$$A(I_4 \Rightarrow I_5) = \frac{S(I_4 \Rightarrow I_5)}{S(I_4)} = \frac{\left(\frac{6}{12}\right)}{\left(\frac{8}{12}\right)} ,$$

or 75%. In practice, an accuracy threshold of at least $c_o = 50\%$ is typical; thus the rule associating {lettuce ⇒ salad dressing} is fairly accurate. (Indeed, in this hypothetical database, shoppers never purchased salad dressing without also purchasing lettuce.) Or, let $\{\mathcal{A}_2\} = \{I_2, I_3\}$ and $\{\mathcal{B}_1\} = \{I_9\}$, that is, the rule {milk, cheese ⇒ ice cream}. Then $S(I_2, I_3 \Rightarrow I_9) = 3/12 = 25\%$ is less often observed, although its accuracy

$$A(I_2, I_3 \Rightarrow I_9) = \frac{S(I_2, I_3 \Rightarrow I_9)}{S(I_2, I_3)} = \frac{\left(\frac{3}{12}\right)}{\left(\frac{4}{12}\right)}$$

is also 75%. Thus this latter rule is a similarly interesting candidate for further study.

Of additional use is appeal to measures of association/correlation for which objective break-points exist. A popular construct in association rule mining is the *lift* (also called *improvement*). The lift of a rule is the ratio of its accuracy to the support of its consequent:

$$L(\mathcal{A}_k \Rightarrow \mathcal{B}_1) = \frac{A(\mathcal{A}_k \Rightarrow \mathcal{B}_1)}{S(\mathcal{B}_1)} = \frac{S(\mathcal{A}_k \Rightarrow \mathcal{B}_1)}{S(\mathcal{A}_k)S(\mathcal{B}_1)} \,. \tag{11.7}$$

As a measure of rule quality, lift expands on the accuracy $A(\mathcal{A}_k \Rightarrow \mathcal{B}_1)$ by quantifying how 'independent' the antecedent is from the consequent when the latter is viewed in a standalone sense. That is, if the 'event' $\mathcal{A}_k$ were statistically independent from $\mathcal{B}_1$, we would expect the final numerator in (11.7) to equal the final denominator, so that $L(\mathcal{A}_k \Rightarrow \mathcal{B}_1) \approx 1$. When $L(\mathcal{A}_k \Rightarrow \mathcal{B}_1) > 1$, the proposed rule appears to improve over simple independence, while if $L(\mathcal{A}_k \Rightarrow \mathcal{B}_1) < 1$, the rule is less informative than assuming the two itemsets are independent.

For example, in Figure 11.9, while both $\{I_4 \Rightarrow I_5\}$ and $\{I_2, I_3 \Rightarrow I_9\}$ have similar accuracies of 75%, their lifts differ, providing more discrimination. Using (11.7), $L(I_4 \Rightarrow I_5)$ calculates to

$$L(I_4 \Rightarrow I_5) = \frac{\left(\frac{6}{12}\right)}{\left(\frac{8}{12}\right)\left(\frac{6}{12}\right)} = 1.5 \,,$$

a good level of improvement. However,

$$L(I_2, I_3 \Rightarrow I_9) = \frac{\left(\frac{3}{12}\right)}{\left(\frac{4}{12}\right)\left(\frac{4}{12}\right)} = 2.25$$

is even higher: a strong indicator for this latter rule's potential associative value.

11.2.3 The Apriori algorithm

Given these various measures of quality, an exercise in association rule learning would, in theory, simply canvass the target database and identify all those rules whose support or accuracy exceed predetermined thresholds, or whose lifts exceed 1 (or some other improvement threshold), and so on. In practice, however, the size of most transaction databases precludes such a strategy. Databases on the order of $n = 10^4$ are considered small in some circles, and transaction 'warehouses' with n exceeding 10^8 are not uncommon, representing terabytes of

data. Coupling this with item lists containing thousands or even tens of thousands of candidate entries produces transaction matrices that are very large and very sparse. The necessary storage and computational requirements quickly move out of range for many practical users. Some simplification may be available if the various items collect naturally into taxonomies, for example, milk, cheese, and ice cream could all be grouped into 'dairy' and vegetables and fruits into 'produce' (Tufféry 2011, Section 10.2). Of course, this also increases granularity in the item structure, and some associations can become blurred. Item taxonomies are best applied when domain-specific requirements motivate their use.

To compensate for sparseness in the database, itemsets are restricted to meet minimum support thresholds, usually at least $s_o \geq 1\%$, although this can vary depending on the analytic objectives. (At times, the threshold is given in terms of minimum total number of itemsets, i.e., a threshold of ns_o.) The results are often called *frequent itemsets* and represent a focused subset for closer examination. Of course, this can eliminate interesting associations among uncommon items, particularly with low-occurrence consequents. For example, Hastie et al. (2009, Section 14.2.2) suggested, perhaps partly with tongue-in-cheek, how easily one might overlook an intriguing if infrequent association such as {vodka ⇒ caviar} among premium marketers.

Additional minimum thresholds on the accuracy – usually at least $c_o \geq 50\%$ – are next imposed, often along with the natural requirement that lift exceed 1. Many analysts also restrict k to keep the eventual associations from becoming too unwieldy. Upper bounds of $k \leq 4$ or $k \leq 5$ are not unheard-of.

Restriction to frequent itemsets can still produce massive data sets, however, and machine learning analysts have developed clever algorithms for managing transaction databases and searching for association rules. One of the earliest is the *Apriori* algorithm of Agrawal and Srikant (1994). Along with the usual minimum threshold requirements, Apriori takes advantage of a key feature of itemset architecture. It recognizes that addition of an additional item to an existing itemset cannot increase the number of occurrences in the database of the new larger itemset. That is, by adding a new item $I_j \notin \mathcal{A}_k$ to $\mathcal{A}_k$ and enlarging the itemset to $\mathcal{A}_{k+1}$, we always find $S(\mathcal{A}_{k+1}) \leq S(\mathcal{A}_k)$. Apriori employs this downward-closure property to improve execution time when searching frequent itemsets, starting with $k = 1$ and working upwards from there (Hand et al. 2001, Section 13.3.2). Its combination of simplicity and effectiveness makes Apriori a very popular rule mining algorithm. In **R**, Apriori is implemented via the `apriori()` function of the external *arules* package.

Example 11.2.1 **Bank depositor associations.** In the bank marketing study of a European bank's depositors previously discussed in Example 9.2.1, a large database was constructed on the depositors' backgrounds – education, loan status, and so on – and banking activities (Moro et al. 2011). The large collection of data here allows for unsupervised learning into possible associations among the depositors' banking characteristics. After removing depositors with missing/unknown observations, the database used here comprises a total of $n = 43\,193$ depositor records (with any personal, identifying information expunged).

From this, an individual depositor's observation is derived as a set of binary indicators on a series of pertinent outcome variables. For an association rule analysis, each depositor is viewed as a 'transaction,' and their binary indicators for each outcome are the individual 'items.' Table 11.6 lists the items/variables, and Table 11.7 gives the corresponding transaction matrix $\mathbf{X}$. (As previously, only a selection of observations is provided in the data table. The complete set of data is available at http://www.wiley.com/go/piegorsch/data_analytics.)

Table 11.6 Binary outcome variables ('items') for association rule analysis of depositor characteristics, from $n = 43\,193$ customers of a large European bank.

Items	Binary feature(s)	
I_1 = Adult	0 = Below 21 years	1 = At/above 21 years
I_2 = Working	0 = Not working/retired	1 = Working
I_3 = Marital	0 = Single/divorced	1 = Married
I_4 = Post-secondary Education	0 = No college	1 = Some college/degree
I_5 = Credit default	0 = Yes/previous	1 = None
I_6 = Positive average balance	0 = No	1 = Yes
I_7 = Active home loan	0 = No	1 = Yes
I_8 = Active personal loan	0 = No	1 = Yes

Source: Moro et al. (2011).

Table 11.7 Selected elements of binary transaction matrix **X** for association rule analysis of depositor characteristics, from $n = 43\,193$ customers of a large European bank.

Customer code	Items							
	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
T_1	1	1	0	0	0	0	0	1
T_2	1	1	1	1	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
T_{43192}	1	1	1	1	1	1	0	0
T_{43193}	1	1	0	1	1	1	0	0

Source: Moro et al. (2011).
Items are defined in Table 11.6.

To conduct an association rule analysis via the Apriori algorithm, the **X** matrix in Table 11.7 can be entered into **R** and manipulated to accommodate operations by the external *arules* package and its `apriori()` function. The following code gives sample commands, after having imported **X** as the dataframe `bank.df`:

```
> require( arules )
> bank.tr <- as( as.matrix(bank.df), 'transactions' )
> bank.ru <- apriori( bank.tr, parameter =
                  list(supp=0.5, conf=0.8,
                       minlen=2, maxlen=6, target='rules') )
> inspect( bank.ru )
```

These commands perform the following operations:

1. load the *arules* package;
2. apply the `as()` function with the `'transactions'` option to coerce `bank.df` into a 'transactions' object `bank.tr` for use with `apriori()`;
3. apply `apriori()` to the transactions object `bank.tr` and recover the association rules; and
4. inspect the resulting `bank.ru` 'rules' object to study the association rules.

Notice in the call to `apriori()` that various options are supplied as a `list`; these specify minimum levels for the uncovered rules:

- `supp=0.5` sets the minimum support level to 50% (the default is 10%)
- `conf=0.8` sets the minimum accuracy ('confidence') to 80% (the default is also 80%)
- `minlen=2` sets the minimum length of a rule, that is, the number of items (antecedents + consequents), to 2 (the default is `minlen=1`, which can produce rules with empty antecedents)
- `maxlen=6` sets the maximum length of a rule to 6, which in effect forces $k \leq 5$ because `apriori()` only allows for singleton consequents

We find that `bank.ru` gives a set of 55 rules, as detailed by the `inspect()` command. The output (not shown) also gives values of support, accuracy (called 'confidence' by the program), and lift. These quality measures can alternatively be accessed as three columns, respectively, from the *arules* `quality()` function. As expected, all supports are at least 50% and all accuracies are at least 80%. In fact, one of the accuracies is exactly 100%: A(Marital, Positive Avg. Balance $\Rightarrow$ Adult) = 1.000. (This is known as a 'trivial' rule: it is not very surprising that every depositor in this database who was both married and carried a positive average balance was always over 21 years of age.) The lifts for these 55 rules vary between `min(quality(bank.ru)[3])` = 0.994 and `max(quality(bank.ru)[3])` = 1.051. For instance, L(Home Loan $\Rightarrow$ Adult) is 1.001, barely above 1 (another trivial rule).

We can restrict attention only to rules with lifts above 1 by subsetting the rule object:

```
> bankLift1.ru <- bank.ru[quality(bank.ru)[3] > 1]
> inspect( bankLift1.ru )
```

This produces 39 rules. Many remain trivial, although a few interesting results – called 'actionable' rules – appear. These are generally related to I_5 = Credit Default. (Recall that I_5 indicates if a depositor has *no* previous default on record.) The following output is edited from `inspect(bankLift1.ru)`:

```
 index  lhs              rhs          support    conf.    lift
   5    {Marital}     => {CrDefault}  0.59074  0.98343  1.0016
   7    {PosBalanc}   => {CrDefault}  0.83213  0.99307  1.0114
  23    {Adult,
         Marital}     => {CrDefault}  0.59072  0.98343  1.0016
  24    {Working,
         PosBalanc}   => {CrDefault}  0.74714  0.99289  1.0112
  25    {Working,
         CrDefault}   => {PosBalanc}  0.74714  0.84314  1.0062
  27    {Adult,
         PosBalanc}   => {CrDefault}  0.83074  0.99305  1.0114
  28    {Adult,
         CrDefault}   => {PosBalanc}  0.83074  0.84737  1.0113
  38    {Adult, Working,
         PosBalanc}   => {CrDefault}  0.74702  0.99289  1.0112
  39    {Adult, Working,
         CrDefault}   => {PosBalanc}  0.74702  0.84316  1.0062
```

The general message is that working, married adults appear likely to carry a positive average balance and/or avoid defaulting on credit. (Carrying a home loan also enters into the mix.) Perhaps more surprisingly, no other depositor characteristics – education, personal loan, and so on – appear to affect these rule associations as strongly with these data.

We find that this database identifies strong associations between common characteristics of job and personal stability – married, working, and so on – when considering depositors for their credit-worthiness and/or likelihood of carrying positive balances.

Some might argue that such associations border on 'trivial,' because experts in the banking industry surely have recognized this already. It could have been just this sort of analysis, however, that led to those discoveries. □

Evolution of alternative market basket algorithms, along with enhancements to the basic Apriori concept, has produced a variety of frequent itemset search procedures (Motoda and Ohara 2009; Nath et al. 2013). One approach simply *samples* from the frequent itemsets, using methods of statistical random sampling (Section 3.1). If conducted properly, the sample will represent the associative information in the larger transaction database and can reduce the computational burden (Hand et al. 2001, Section 13.3). Indeed, algorithm design for association rule learning is an active research area, with many avenues open for further advancement (Borgelt 2012).

11.2.4 Statistical measures of association quality

As a predictive measure of association rule quality, the lift in (11.7) represents a more objective quantification than the simpler concepts of support and accuracy. Taking this a step further, one can view a rule's {antecedent ⇒ consequent} association structure in a cross-classified manner, similar to the confusion matrices for classification rules in Section 9.1.2. This allows for formal statistical measures to be applied.

Frame the construction as follows: given an antecedent itemset $\mathcal{A}_k$ and a consequent itemset $\mathcal{B}_1$, calculate how many transactions in $\mathbf{X}$ satisfy both itemsets – that is, those transactions where all items in both $\mathcal{A}_k$ and $\mathcal{B}_1$ are purchased. Denote this by Y_{11}. Similarly, calculate how many transactions satisfy neither itemset; denote this by Y_{00}. Lastly, calculate

Y_{10} = the number of transactions where $\mathcal{A}_k$ is satisfied but $\mathcal{B}_1$ is not,

and

Y_{01} = the number of transactions where $\mathcal{B}_1$ is satisfied but $\mathcal{A}_k$ is not.

(The total should equal n, because these four cases fully partition $\mathbf{X}$.) Collect the four counts together into a cross-classified 2×2 contingency table, as in Section 8.3.3. Table 11.8 gives a schematic representation.

We can quantify the association in this 2×2 table via the Pearson X^2 statistic from (8.14). Here, we have

$$X^2 = \sum_{i=0}^{1} \sum_{j=0}^{1} \frac{(Y_{ij} - \hat{Y}_{oij})^2}{\hat{Y}_{oij}}, \tag{11.8}$$

where

$$\hat{Y}_{oij} = \frac{\left(\sum_{j=0}^{1} Y_{ij}\right)\left(\sum_{i=0}^{1} Y_{ij}\right)}{\sum_{i=0}^{1}\sum_{j=0}^{1} Y_{ij}} = \frac{Y_{i+}Y_{+j}}{Y_{++}}$$

Table 11.8 A 2×2 contingency table for cross-classification of association rule outcomes

		Consequent	
		$\mathcal{B}_1$ fail	$\mathcal{B}_1$ met
Antecedent	$\mathcal{A}_k$ fail	Y_{00}	Y_{01}
	$\mathcal{A}_k$ met	Y_{10}	Y_{11}

is the expected (i, j)th cell count when no association exists between the antecedent and consequent conditions (cf. Section 8.3.3). We use the ability of the X^2 statistic to quantify high association in this 2×2 contingency table. As the observed pattern in Table 11.8 deviates from the no-association state, the statistic in (11.8) grows large. (Note, however, that there is no formal hypothesis test being performed here, so that appeal to significance levels and false-positive errors is irrelevant.)

A caveat: the X^2 statistic in (11.8) is unaffected by column transposition. If we were to permute the columns in Table 11.8, X^2 remains the same. (Try it!) As applied to association rules, this translates to a χ^2-equivalence between the rule $\{\mathcal{A}_k \Rightarrow \mathcal{B}_1\}$ and $\{\mathcal{A}_k \Rightarrow \mathcal{B}_1^c\}$, where $\mathcal{B}_1^c$ is the complement of $\mathcal{B}_1$ ('not' $\mathcal{B}_1$). In most settings, however, the rules $\{\mathcal{A}_k \Rightarrow \mathcal{B}_1\}$ and $\{\mathcal{A}_k \Rightarrow \mathcal{B}_1^c\}$ will represent different associations, so this χ^2-equivalence will be spurious. The X^2 measure still has value in detecting strong associations, but analysts should not rely on it exclusively. At a minimum, always also calculate an additional measure, such as the lift. (Although, some transaction rules with equal X^2 values can also have equal lifts.) In fact, any other measure designed to detect association in 2×2 tables could also be calculated to similar effect, such as the (two-sided) hypergeometric P-value from the Fisher exact test in Section 8.3.3. In **R**, the external *arules* package can calculate X^2 for any rule via its `interestMeasure()` function as of course can many other **R** functions, such as `chisq.test()`.

Example 11.2.2 Bank depositor associations (Example 11.2.1, continued). In the bank depositor data from Table 11.7, we can calculate the X^2 statistic for each of the highlighted rules with lifts greater than 1. Applied to the same database and **R** objects from Example 11.2.1, the commands

```
> require( arules )
> indeces <- c(5, 7, 23, 24, 25, 27, 28, 38, 39)
> interestMeasure( bankLift1.ru[indeces],
                   method='chiSquare', transactions=bank.tr )
```

produce the X^2 values summarized in Table 11.9.

As might be expected, all of the X^2 values in Table 11.9 are large. The smaller values for the rules

{Marital ⇒ (no) Credit Default},

{Adult, Marital ⇒ (no) Credit Default},

{Working, (no) Credit Default ⇒ Positive Avg. Balance}, and

{Adult, Working, (no) Credit Default ⇒ Positive Avg. Balance}

Table 11.9 X^2 statistics from (11.8) for selected association rules in Example 11.2.2 with bank depositor data from Table 11.7.

Index	Rule: {Antecedent ⇒ Consequent}	X^2
5	{Marital ⇒ (no) Cred. Default}	8.578
7	{Pos. Balance ⇒ (no) Cred. Default}	1567.356
23	{Adult, Marital ⇒ (no) Cred. Default}	8.570
24	{Working, Pos. Balance ⇒ (no) Cred. Default}	893.382
25	{Working, (no) Cred. Default ⇒ Pos. Balance}	66.896
27	{Adult, Pos. Balance ⇒ (no) Cred. Default}	1548.262
28	{Adult, (no) Cred. Default ⇒ Pos. Balance}	1414.184
38	{Adult, Working, Pos. Balance ⇒ (no) Cred. Default}	892.649
39	{Adult, Working, (no) Cred. Default ⇒ Pos. Balance}	67.396

suggest lesser strength of association. For the others, added potential may lie in exploring their particular associations further. □

Owing in part to their ease of use, association rules are extremely popular in machine learning and knowledge discovery. Consequently, a large literature has developed; see Höppner (2010); Zhang and Wu (2011). For particular emphasis on business and commerce, Linoff and Berry (2011, Chapter 15) provided an extensive introduction.

Exercises

11.1 Return to the automobile characteristics data in Example 11.1.2 and now apply (a) single linkage and (b) complete linkage to construct the dendrogram. (Continue to employ a Euclidean distance metric.) Does either clustering pattern change substantively for this simple data set?

11.2 What happens if you change to Manhattan distance in Exercise 11.1?

11.3 The basic structure of a cluster analysis may be transposed, so that clustering among the *variables* is explored instead of clustering among the observations. The data remain the same, although now the target goal is to find clusters among the p columns of $\mathbf{Z}$.

(a) Write these column vectors as $\mathbf{z}_j$ for $j = 1, \ldots, p$ and consider two different column vectors $\mathbf{z}_j$ and $\mathbf{z}_m$. How would the technical details change for defining a dissimilarity metric $d(\mathbf{z}_j, \mathbf{z}_m)$ in this case? What changes would be necessary to apply a corresponding hierarchical agglomerative cluster analysis with, say, average-link clustering?

(b) How would you apply this reformulated method in practice using, for example, `hclust()` and/or `plot()` in **R**?

11.4 From a bioinformatic study of gene expression profiles in carcinogenesis, Golub et al. (1999) analyzed expression data for 7129 human genes potentially associated with hematopoietic cancers. The profiles were taken from a group of 47 patients with acute lymphoblastic leukemia. A selection of the reported expression levels follows. (Download the complete data set at http://www.wiley.com/go/piegorsch/data_analytics.)

	Patient code no.						
Gene descr./code	1001	1032	1113	1124	$\cdots$	6371	6672
AF1q	399	252	1588	27	$\cdots$	79	267
apoargC	−290	−274	−337	−256	$\cdots$	−116	−182
ard-1	426	371	420	587	$\cdots$	244	569
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
Zyxin	298	307	309	693	$\cdots$	509	417

(a) Conduct an agglomerative hierarchical cluster analysis on these gene expression data, with genes viewed as the n observations and patients viewed as the p variables. (Remember to standardize the data across genes within each jth patient before proceeding with the calculations.) Use Euclidean distance for the underlying distance metric. Separately apply each of (i) single linkage, (ii) complete linkage, and (iii) average linkage for the linkage metric, and comment on any differences you see in the resulting dendrograms.

(b) Transpose the perspective from Exercise 11.4a and now view patients as the n observations and genes as the p variables, as per Exercise 11.3. (Notice that you will have to return to the original expression values and now standardize across patients within each gene.) Continue to employ Euclidean distance, but only construct an average linkage dendrogram.

(c) Use the results from Exercises 11.4a and 11.4b to build a heatmap (cf. Figure 10.5) of the gene expression outcomes based on the hierarchical clusterings. That is, reorder the original data matrix $\mathbf{X}$ by setting an observation's row index equal to the ordered location given by clustering genes hierarchically, and its column index is equal to the ordered location given by clustering patients hierarchically. (The `heatmap()` function in **R** may be useful, particularly with its `Rowv=` and `Colv=` options activated. See `help(heatmap)`.) A popular color scheme with gene expression data is to set negative (underexpressed) observations to green and positive (overexpressed) observations to bright red. If you have such capabilities, try it; however, be aware that this particular scheme may cause problems for viewers with dichromatism.

11.5 Return to the gene expression data in Example 11.1.3 and examine the hierarchical clustering solutions from each of the three linkage strategies in greater detail. In particular, compare the 'early' clustering patterns at small values of δ for the single linkage

and complete linkage solutions in Figure 11.5 with that identified in Figure 11.4. For clusters that develop quickly, is there large overlap in the cluster memberships? (If so, this would indicate possible subgroups of patients with the tumor.)

11.6 In a study of calendar year 2012 business activities for hotel management companies (Anonymous 2013), data were collected for $n = 147$ firms on the number of guest rooms each had under management, along with the firm's total 2012 revenue (in \$M). This gave $p = 2$ outcome variables for study. The data are available online at http://www.wiley.com/go/piegorsch/data_analytics; a sample is given as follows:

Company	Guest rooms	Revenue (\$M)
Interstate Hotels & Resorts	61 205	2600.00
Aimbridge Hospitality	25 908	700.13
GF Management	21 971	520.00
White Lodging Services	20 232	746.64
John Q Hammons	19 000	50.00
⋮	⋮	⋮
Chartwell Hospitality	107	1.75
Allen & O'Hara	96	2.45

Both outcome variables exhibit a substantial right skew (plot histograms or boxplots to verify this), so perform a logarithmic transformation on each and operate with $x_1 = \log(\text{Guest rooms})$ and $x_2 = \log(\text{Revenue})$. Study potential clustering with these log-transformed data as follows:

(a) Plot x_2 versus x_1. Does visual inspection suggest any apparent clustering?

(b) Standardize the data and construct a scatterplot of the resulting z-scores. Do any substantial changes appear from the plot in Exercise 11.6a?

(c) Apply K-means partitioning with Euclidean distance to identify potential clusters in the standardized data. Use an elbow plot of the cumulative within-cluster sum of squares, W_K, to select a value for K. Also plot $\log\{W_K\}$ and the pseudo-F statistic against K to verify if your choice appears warranted.

(d) Return to the scatterplot of the standardized data in Exercise 11.6a and modify the plot to mark the K-means cluster solution from Exercise 11.6c. (Use different colors, or different plot symbols, etc. for each observation's cluster membership.) Overlay a Vonoroi tessellation to clearly segment the clusters.

11.7 In a study of diagnostic factors for human cancer, Wolberg and Mangasarian (1990) reported on $p = 9$ different prognostic variables for a sample of breast cancer patients in the US state of Wisconsin. The data, corresponding to a version of this database discussed in Sugar and James (2003), comprise $n = 683$ observations. A selection of the data is given as follows. (Download the complete data set at http://www.wiley.com/go/piegorsch/data_analytics.)

	Patient code no.					
Outcome variable	1000025	1002945	1015425	···	888820	897471
x_1 = Clump thickness	5	5	3	···	5	4
x_2 = Cell size uniformity	1	4	1	···	10	8
x_3 = Cell shape uniformity	1	4	1	···	10	8
x_4 = Marginal adhesion	1	5	1	···	3	5
x_5 = Epithelial cell size	2	7	2	···	7	4
x_6 = Bare nuclei	1	10	2	···	3	5
x_7 = Bland chromatin	3	3	3	···	8	10
x_8 = Normal nucleoli	1	2	1	···	10	4
x_9 = Mitoses	1	1	1	···	2	1

(a) Standardize the nine outcome variables into z-scores. Apply K-means partitioning with Euclidean distance to identify potential clusters in the data. Use an elbow plot of the cumulative within-cluster sum of squares, W_K, to select a value for K. Also plot the pseudo-F statistic against K to verify if your choice seems warranted. Does anything unusual appear with the pseudo-F plot?

(b) An additional variable recorded for each patient was whether their tumor was benign or malignant. This variable (labeled as x_{10} = 'Class') is also included in the downloadable data set: $x_{10} = 2$ for benign tumors (for 444 of the patients) and $x_{10} = 4$ for malignant tumors (for 239 of the patients). Recode these as 'B' and 'M', respectively. Determine how many of each tumor type are identified in each of the K clusters from the K-means clustering solution you found in Exercise 11.7a. Present the counts in a tabular cross-classification, with rows as clusters and columns as tumor type. Comment on the results.

11.8 Return to the home center data in Example 11.1.5. The skew in the original data suggests the use of a (natural) logarithmic transform to help to improve analytic traction. So, apply a log transform to the x_1 = Sales, x_3 = Units, and x_4 = Staff variables. (Leave the x_2 = percentage of change variable unaltered.)

(a) Reconstruct the labeled scatterplot matrix (Figure 11.8), using the log-Sales, log-Units, and log-Staff variables (and with the original % change variable). Use the cluster labels from the analysis in Example 11.1.5. What patterns appear in the new scatterplots?

(b) Using the log-Sales, log-Units, and log-Staff variables (and with the original % change variable), conduct a new cluster analysis. (Remember to standardize the log-transformed data first.) Apply K-means partitioning with Euclidean distance. Use an elbow plot of the cumulative within-cluster sum of squares, W_K, to select a value for K. Also plot the pseudo-F statistic against K to verify if your choice seems warranted. With your given choice of K, construct another labeled scatterplot matrix to examine the clustering solution. How does this compare to the plot in Exercise 11.8a?

11.9 Association rule learning is often applied in biomedical applications. To illustrate, the following data from $n = 270$ patients on factors affecting heart disease come from the famous 'Statlog' project in machine learning (Michie et al. 1994, Section 9.4.2). For purposes of assessing possible associations in the data, consider the following $p = 10$ dichotomized outcome variables:

Items	Binary feature(s)	
I_1 = Age	0 = Less than 60 years	1 = At/over 60 years
I_2 = Sex	0 = Female	1 = Male
I_3 = Blood pressure (systolic)	0 = Less than 140 mmHg	1 = At/over 140 mmHg
I_4 = Cholesterol	0 = Less than 240 mg/dL	1 = At/over 240 mg/dL
I_5 = Blood sugar	0 = Less than 120 mg/dL	1 = At/over 120 mg/dL
I_6 = Maximum heart rate	0 = Less than 150 bpm	1 = At/over 150 bpm
I_7 = Induced angina	0 = No	1 = Yes
I_8 = Fluoroscopy indicator	0 = None	1 = Some
I_9 = Thallium scan	0 = Normal	1 = Some defect
I_{10} = Disease state	0 = Absent	1 = Present

A selection of the corresponding binary transaction matrix $\mathbf{X}$ is given as follows (the full data are available online at http://www.wiley.com/go/piegorsch/data_analytics):

	Items									
Patient code	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
T_1	1	1	0	0	0	0	0	0	0	1
T_2	1	0	0	0	1	0	0	0	0	0
T_3	1	0	1	0	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
T_{269}	0	1	1	1	0	0	1	1	1	1
T_{270}	1	0	1	1	0	1	0	1	1	1

(a) Apply the Apriori algorithm to this transaction database and identify any possible association rules. Set your minimum support to 2% and your minimum accuracy ('confidence') to 50%, and only consider itemset antecedents of size $k = 5$ or less. What association rules does this uncover? How do their lifts compare?

(b) Also calculate the X^2 statistic from (11.8) for each of these rules. Does this provide additional distinguishability among the rules?

(c) Experiment with the algorithm settings (minimum support, minimum accuracy, etc.) in Exercise 11.9a. What patterns emerge?

11.10 Return to the bank depositor data in Example 11.2.2, and verify the individual calculations for the X^2 statistic with the following rules from Table 11.9. In particular, find the underlying 2×2 table and perform the X^2 calculations directly without appeal to the *arules* package.

(a) {Marital ⇒ (no) Cred. Default}, $X^2 = 8.578$.

(b) {Adult, Marital ⇒ (no) Cred. Default}, $X^2 = 8.570$.

(c) {Working, (no) Cred. Default ⇒ Pos. Balance}, $X^2 = 66.896$.

11.11 Brijs et al. (1999) discussed a market basket analysis for a Belgian retail supermarket involving $n = 88\,162$ customer transactions. These data give each transaction as a row that lists which of the store's products, coded by stock-keeping units (SKUs), were purchased. As is typical, the total number of SKUs for this supermarket is large: $p = 16\,470$. The data have been sanitized: the SKUs are coded to between 0 and 16 469, and all customer information is masked. A selection of the data is given as follows (the complete set of data is available online at http://fimi.ua.ac.be/data/retail.dat):

Transaction number	Item SKUs purchased							
1	0	1	2	3	4	5	6	7
	8	9	10	11	12	13	14	15
	16	17	18	19	20	21	22	23
	24	25	26	27	28	29		
2	30	31	32					
3	33	34	35					
⋮		⋮						
88160	2310	4267						
88161	39	48	2528					
88162	32	39	205	242	1393			

(a) Apply the Apriori algorithm to identify any possible association rules in this transaction database. (If using **R** and the `apriori()` function, the data's input format will require manipulation. Try constructing it as a `list` via the command `strsplit(readLines('retail.csv'),',')`, where `retail.csv` is the comma-separated data file.) Set your minimum support to 10% and your minimum accuracy ('confidence') to 50%, and only consider itemset antecedents of size $k = 5$ or less. What association rules does this uncover? How do their lifts compare?

(b) Also calculate the X^2 statistic from (11.8) for each of these rules. Does this provide additional distinguishability among the rules?

(c) Experiment with the algorithm settings (minimum support, minimum accuracy, etc.) in Exercise 11.11a. Especially for this large and sparse transaction matrix, what patterns emerge?

11.12 Return to the heart disease data in Exercise 11.9b, and verify the individual calculations for the X^2 statistic with the rule whose lift is a maximum among all those generated. In particular, find the underlying 2×2 table and perform the X^2 calculations without appeal to the *arules* package.

11.13 Prove algebraically the contention in Section 11.2.4 that if the rows of the 2×2 contingency table in Table 11.8 are transposed, the X^2 statistic in (11.8) is unaffected.

Appendix A

Matrix manipulation

Matrix and vector operations are a necessary component of many data analytic calculations. This brief appendix reviews the terminology, notation, and selected algebraic aspects most useful in this regard. All quantities are assumed to be real valued. For a complete introduction, including background on many of the results stated in the following, see complete textbook sources such as Gentle (2007) and Schay (2012).

A.1 Vectors and matrices

A vector or a matrix is a collection of numbers assembled together into a structured, rectangular (including square) array. By contrast, a *scalar* is a unidimensional number, denoted by a simple Latin or Greek letter, such as b or β. A *vector* is a single row or column of n scalars, denoted by a lowercase, boldfaced symbol $\mathbf{b}$. In special cases, such as vectors of random variables, uppercase notation $\mathbf{X}$ is used.

The scalar elements of a vector are assigned subscripts to identify their location in the array. A *row vector* arranges these subscripted elements in a row: $[b_1\ b_2\ \cdots\ b_n]$. A *column vector* arranges them in a column:

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}. \tag{A.1}$$

The transposition operator, denoted by superscript T at the end of the vector, transposes a row vector into a column vector, and vice versa. Thus the transposed row vector $[b_1\ b_2\ \cdots\ b_n]^{\mathrm{T}}$ is equivalent to the column vector in (A.1).

A *matrix* is an $n \times p$ rectangular collection of scalars into n rows and p columns, denoted with an uppercase Roman or Greek letter, $\mathbf{B}$ or Θ, and indexed first by the rows and then by

Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery,
First Edition. Walter W. Piegorsch.

www.wiley.com/go/piegorsch/data_analytics

the columns:

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{bmatrix}.$$

For shorthand notation, we use $\mathbf{B} = \{b_{ij}\}$. Now, transposition takes an $n \times p$ matrix into a $p \times n$ matrix: if $\mathbf{B}$ has elements $\{b_{ij}\}$, the transpose $\mathbf{B}^{\mathrm{T}}$ has elements $\{b_{ji}\}$. Repeated transposition reproduces the original matrix: $(\mathbf{B}^{\mathrm{T}})^{\mathrm{T}} = \mathbf{B}$.

It is often convenient to write an $n \times p$ matrix as a collection of its columns viewed as vectors, for example, $\mathbf{B} = [\mathbf{b}_1\ \mathbf{b}_2\ \cdots\ \mathbf{b}_p]$, where $\mathbf{b}_j = [b_{1j}\ b_{2j}\ \cdots\ b_{nj}]^{\mathrm{T}}$ is the jth column. (Alternatively, one can also view $\mathbf{B}$ in terms of its row vectors.) If a column (or row) of a matrix is a linear combination of any other columns (or rows) – for example, $\mathbf{b}_j = w_q\mathbf{b}_q + w_r\mathbf{b}_r$ for $j \neq q, r$ and any scalars w_q and w_r – the column (row) is *linearly dependent*. If not, it is *linearly independent*. The number of linearly independent columns of a matrix $\mathbf{B}$ is called its *rank*, denoted as rank($\mathbf{B}$). One can equivalently define rank in terms of the number of linearly independent rows, because the two quantities are in fact equal (Gentle 2007, Section 3.3). Indeed, rank($\mathbf{B}$) = rank($\mathbf{B}^{\mathrm{T}}$). If rank($\mathbf{B}$) = min$\{n, p\}$, $\mathbf{B}$ is said to have full rank.

If $n = p$ the matrix is *square* of order n. The elements b_{ii} then constitute the *diagonal* of the matrix. The *trace* of an $n \times n$ square matrix is the sum of its diagonal elements:

$$\mathrm{tr}(\mathbf{B}) = \sum_{i=1}^{n} b_{ii}.$$

Elements above the diagonal are the *upper diagonal elements*; those below are the *lower diagonal elements*.

If all the lower (upper) diagonal elements of a square matrix are identically zero, the matrix is called *upper (lower) triangular*. More generally, for any rectangular ($n \times p$) matrix $\mathbf{M}$, if all elements below the leading diagonal $m_{11}, m_{22}, \ldots, m_{qq}$ are zero (for $q = \min\{n, p\}$), the matrix is called *(upper) trapezoidal*.

If the elements of a square matrix satisfy $b_{ij} = b_{ji}$ ($i \neq j$), the matrix is *symmetric* (and then $\mathbf{B} = \mathbf{B}^{\mathrm{T}}$). A symmetric matrix with $b_{ij} = b_{ji} = 0$ for all $i \neq j$ is called a *diagonal matrix*, denoted as diag$\{b_{11}, b_{22}, \ldots, b_{nn}\}$ or simply diag$\{b_{ii}\}$ if the context is clear.

The special $n \times n$ diagonal matrix $\mathbf{I} = \mathrm{diag}\{1, 1, \ldots, 1\}$ is the *identity matrix*. The notation includes the order as a subscript, $\mathbf{I}_n$, if greater specificity is required. Viewed as a matrix of column vectors, $\mathbf{I}_n$ is written as $\mathbf{I}_n = [\mathbf{e}_1\ \mathbf{e}_2\ \cdots\ \mathbf{e}_n]$, where $\mathbf{e}_j$ is the *coordinate unit vector* made up entirely of zeros except for a 1 in its jth element: $\mathbf{e}_1 = [1\ 0\ 0\ \cdots\ 0]^{\mathrm{T}}$, $\mathbf{e}_2 = [0\ 1\ 0\ \cdots\ 0]^{\mathrm{T}}$, and so on. Similar in nature is the column vector made up entirely of ones, denoted herein as $\mathbf{h} = [1\ 1\ 1\ \cdots\ 1]^{\mathrm{T}}$.

A.2 Matrix algebra

Vector and matrix addition is conducted elementwise, as is scalar multiplication. So, for example,

$$\mathbf{b} + w\,\mathbf{d} = \begin{bmatrix} b_1 + w\,d_1 \\ b_2 + w\,d_2 \\ \vdots \\ b_n + w\,d_n \end{bmatrix},$$

where w is any scalar. The operations are commutative, so that $\mathbf{b} + w\mathbf{d} = w\mathbf{d} + \mathbf{b} = \mathbf{b} + \mathbf{d}w$. Notice that the two vectors or matrices being added must have the same dimensions. Subtraction is similar.

Matrix/vector multiplication is more complex. Often called 'Cayley multiplication' after its development by nineteenth century British mathematician Arthur Cayley (see Cayley 1858), the standard definition multiplies elementwise across a column and then down a row, adding the resulting scalar product(s). The operation depends on the row and column dimensions of the product's factors and on the order in which they appear. Thus, multiplying a $1 \times n$ row vector $\mathbf{b}^{\mathrm{T}}$ by an $n \times 1$ column vector $\mathbf{d}$ produces a scalar:

$$\mathbf{b}^{\mathrm{T}}\mathbf{d} = \sum_{i=1}^{n} b_i d_i.$$

(This is also called a *dot product*, with notation $\mathbf{b} \cdot \mathbf{d}$, or also a *Euclidean inner product*.) The number of elements in each of the two vectors must match. Conversely, multiplying an $n \times 1$ column vector by a $1 \times p$ row vector is possible, producing an $n \times p$ matrix:

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} [d_1 \; d_2 \; \cdots \; d_p] = \begin{bmatrix} b_1 d_1 & b_1 d_2 & \cdots & b_1 d_p \\ b_2 d_1 & b_2 d_2 & \cdots & b_2 d_p \\ \vdots & \vdots & \cdots & \vdots \\ b_n d_1 & b_n d_2 & \cdots & b_n d_p \end{bmatrix}.$$

Clearly, vector multiplication is *not* commutative.

Matrix multiplication is an extension of this row $\times$ column operation: the 'product' of an $n \times p$ matrix $\mathbf{B}$ and a $p \times m$ matrix $\mathbf{D}$ has (i,j)th element $\sum_{h=1}^{p} b_{ih} d_{hj}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$. The resulting matrix product, $\mathbf{BD}$, is an $n \times m$ matrix. For the product to exist, the number of columns in $\mathbf{B}$ must equal the number of rows in $\mathbf{D}$. If so, $\mathbf{B}$ and $\mathbf{D}$ are *conformable* (for multiplication). Here again, matrix multiplication is not commutative: $\mathbf{BD}$ need not equal $\mathbf{DB}$. In fact, one or both may not exist, depending on whether or not the individual matrices are conformable. If they are, then $(\mathbf{BD})^{\mathrm{T}} = \mathbf{D}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}$. For two conformable matrices $\mathbf{B}_{n\times p}$ and $\mathbf{D}_{p\times n}$, $\mathrm{tr}(\mathbf{BD}) = \mathrm{tr}(\mathbf{DB})$.

Because order makes a difference in matrix multiplication, special terminology is required. For instance, if $\mathbf{B}$ and $\mathbf{D}$ are conformable to produce the product $\mathbf{BD}$, we say $\mathbf{D}$ has been *premultiplied* by $\mathbf{B}$. Similarly, in $\mathbf{DB}$, $\mathbf{D}$ is *postmultiplied* by $\mathbf{B}$. In ether case, multiplication by the (appropriate) identity matrix always returns the original $n \times p$ matrix: $\mathbf{I}_n\mathbf{B} = \mathbf{B}$ and $\mathbf{B}\mathbf{I}_p = \mathbf{B}$.

In passing, note that other alternative definitions exist for the concept of a matrix product (see Gentle 2007, Section 3.2.9).

When two vectors $\mathbf{b}$ and $\mathbf{d}$ satisfy $\mathbf{b}^{\mathrm{T}}\mathbf{d} = 0$, they are said to be *orthogonal*. (This has a geometric interpretation: orthogonal vectors are perpendicular and cross at right angles in Euclidean space.) A set of nonzero vectors that are mutually orthogonal are also linearly independent.

When two orthogonal vectors $\mathbf{b}$ and $\mathbf{d}$ are scaled to have unit length such that $\mathbf{b}^{\mathrm{T}}\mathbf{b} = 1$ and $\mathbf{d}^{\mathrm{T}}\mathbf{d} = 1$, they are called *orthonormal*. Notice that the coordinate unit vectors, $\mathbf{e}_j$ from Section A.1, are all orthonormal.

Applied to matrices, a square matrix, $\mathbf{Q}$, is *orthogonal* if it satisfies $\mathbf{Q}\mathbf{Q}^{\mathrm{T}} = \mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}$. (The definition of orthogonality can be extended to any $n \times p$ matrix, but this will not be required herein.)

A.3 Matrix inversion

There is no simple definition for matrix division. For a square matrix, there is the concept of a reciprocal called an *inverse matrix*. To define this, some preliminary constructions are required.

Given a square $n \times n$ matrix $\mathbf{B} = \{b_{ij}\}$, the *determinant* of the matrix is a summary scalar useful for various multiplicative operations. The notation is $|\mathbf{B}|$ or sometimes $\det(\mathbf{B})$. Note that the former has nothing to do with the absolute value operator.

The simplest way to define $|\mathbf{B}|$ is in terms of the conditions that it satisfies. Let $\mathbf{B} = [\mathbf{b}_1\ \mathbf{b}_2\ \cdots\ \mathbf{b}_p]$, where $\mathbf{b}_j = [b_{1j}\ b_{2j}\ \cdots\ b_{nj}]^{\mathrm{T}}$ is $\mathbf{B}$'s jth column. Define the determinant $|\mathbf{B}|$ such that

(a) $|\mathbf{I}| = 1$ for $\mathbf{B} = \mathbf{I}$,

(b) $|\mathbf{b}_1\ \cdots\ v\mathbf{b}_j + w\mathbf{d}\ \cdots\ \mathbf{b}_p| = v|\mathbf{B}| + w|\mathbf{b}_1\ \cdots\ \mathbf{d}\ \cdots\ \mathbf{b}_p|$ for scalars v and w and any $n\times 1$ vector $\mathbf{d}$,

(c) $|\cdots\ \mathbf{b}_j\ \cdots\ \mathbf{b}_k\ \cdots| = 0$ if $\mathbf{b}_j = \mathbf{b}_k$, and

(d) $|\mathbf{b}_1\ \cdots\ \mathbf{b}_j\ \mathbf{b}_{j+1}\ \cdots\ \mathbf{b}_p| = -|\mathbf{b}_1\ \cdots\ \mathbf{b}_{j+1}\ \mathbf{b}_j\ \cdots\ \mathbf{b}_p|$

(Schay 2012, Section 6.1). Rules for calculating $|\mathbf{B}|$ from these defining conditions can then be developed. In the simplest case, the determinant of a 2×2 matrix is

$$\begin{vmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{vmatrix} = b_{11}b_{22} - b_{12}b_{21}.$$

For larger n, the determinant is best calculated via computer. For instance, in **R**, use the `det(X)` function where `X` is an $n \times n$ array of class `matrix`.

The following are some special cases for an $n \times n$ matrix $\mathbf{B}$:

- $|w\mathbf{B}| = w^n|\mathbf{B}|$ for any scaler w.
- $|\mathbf{B}^{\mathrm{T}}| = |\mathbf{B}|$.
- If $\mathbf{B}$ is upper or lower triangular, $|\mathbf{B}| = \prod_{i=1}^{n} b_{ii}$.
- Suppose $\mathbf{B}$ is *partitioned* into
$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{22} \end{bmatrix}$$
for square submatrices $\mathbf{B}_{11}$ and $\mathbf{B}_{22}$, where $\mathbf{0}$ is a matrix of zeros. Then $|\mathbf{B}| = |\mathbf{B}_{11}||\mathbf{B}_{22}|$.
- For the $n \times n$ matrix $\mathbf{D}$, $|\mathbf{BD}| = |\mathbf{B}||\mathbf{D}|$.
- If $\mathbf{Q}$ is a square, orthogonal matrix, $|\mathbf{Q}| = \pm 1$.

Note, however, that $|\mathbf{A} + \mathbf{B}| \neq |\mathbf{A}| + |\mathbf{B}|$.

The determinant can be used to identify the inverse of a matrix. A square matrix $\mathbf{B}$ whose determinant $|\mathbf{B}|$ equals zero is called *singular*. If $|\mathbf{B}| \neq 0$, then $\mathbf{B}$ is *nonsingular*. Every square nonsingular matrix $\mathbf{B}$ has an *inverse matrix* $\mathbf{B}^{-1}$ that satisfies

$$\mathbf{B}^{-1}\mathbf{B} = \mathbf{I} = \mathbf{B}\mathbf{B}^{-1}.$$

(Singular matrices do not possess an inverse.) Nonsingular matrices must have full rank.

As with the determinant, expressions for finding an inverse matrix $\mathbf{B}^{-1}$ can become complex (see Schay 2012, Section 2.5). In the simple 2×2 case, the result is

$$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}^{-1} = \frac{1}{b_{11}b_{22} - b_{12}b_{21}} \begin{bmatrix} b_{22} & -b_{12} \\ -b_{21} & b_{11} \end{bmatrix}. \tag{A.2}$$

Notice that the denominator in (A.2) is simply $|\mathbf{B}|$. For larger n, the inverse is often determined via computer. For instance, in **R**, use the `solve(X)` function where `X` is an $n \times n$ array of class `matrix`.

If $\mathbf{B}$ is nonsingular, then

- $(\mathbf{B}^{\mathrm{T}})^{-1} = (\mathbf{B}^{-1})^{\mathrm{T}}$.
- To solve the (conformable) system of linear equations $\mathbf{Bx} = \mathbf{d}$, premultiply by $\mathbf{B}^{-1}$ to find $\mathbf{x} = \mathbf{B}^{-1}\mathbf{d}$.
- If the conformable matrix $\mathbf{D}$ is also nonsingular, $(\mathbf{BD})^{-1} = \mathbf{D}^{-1}\mathbf{B}^{-1}$.
- If $\mathbf{Q}$ is (square and) orthogonal, $\mathbf{Q}^{-1} = \mathbf{Q}^{\mathrm{T}}$.
- And obviously, $\mathbf{I}^{-1} = \mathbf{I}$.

A.4 Quadratic forms

Given an $n \times 1$ vector $\mathbf{d}$ and a symmetric $n \times n$ matrix $\mathbf{B}$, the scalar quantity

$$\mathbf{d}^{\mathrm{T}}\mathbf{B}\,\mathbf{d} = \sum_{i=1}^{n}\sum_{j=1}^{n} d_i b_{ij} d_j$$

is called a *quadratic form*. When $\mathbf{B} = \mathbf{I}$, the quadratic form is just the sum of squares $\mathbf{d}^{\mathrm{T}}\mathbf{d} = \sum_{i=1}^{n} d_i{}^2$, sometimes called the *inertia* of the vector $\mathbf{d}$.

If $\mathbf{d}^{\mathrm{T}}\mathbf{B}\,\mathbf{d} \geq 0$ for any real vector $\mathbf{d}$, $\mathbf{B}$ is termed a *nonnegative definite matrix* (some authors use 'positive semidefinite'). If, further, $\mathbf{d}^{\mathrm{T}}\mathbf{B}\,\mathbf{d} > 0$ for any real vector $\mathbf{d} \neq \mathbf{0}$, $\mathbf{B}$ is *positive definite*. Alternatively, if $\mathbf{d}^{\mathrm{T}}\mathbf{B}\,\mathbf{d} < 0$ for any real vector $\mathbf{d} \neq \mathbf{0}$, $\mathbf{B}$ is *negative definite*.

A special case occurs when $\mathbf{B}$ has the form $\mathbf{X}^{\mathrm{T}}\mathbf{X}$, where $\mathbf{X}$ is an $m \times n$ matrix. Then, the quadratic form $\mathbf{d}^{\mathrm{T}}\mathbf{B}\,\mathbf{d}$ is

$$\mathbf{d}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\,\mathbf{X}\,\mathbf{d} = (\mathbf{Xd})^{\mathrm{T}}\mathbf{Xd}.$$

Notice, however, that $\mathbf{Xd}$ is an $m \times 1$ vector, say $\mathbf{a} = \mathbf{Xd}$. Thus $\mathbf{d}^{\mathrm{T}}\mathbf{B}\,\mathbf{d} = \mathbf{a}^{\mathrm{T}}\mathbf{a} = \sum_{i=1}^{m} a_i^2$, which is clearly nonnegative for any $\mathbf{a}$ (and, hence, for any $\mathbf{d}$). Therefore, we conclude that $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is always nonnegative definite. If, further, $a_i > 0$ for all i, $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ will be positive definite, with inverse matrix $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$. (A symmetric, positive definite matrix is always nonsingular; see the following sections.)

A.5 Eigenvalues and eigenvectors

Suppose $\mathbf{B} = [\mathbf{b}_1\ \mathbf{b}_2\ \cdots\ \mathbf{b}_n]$ is a square $n \times n$ matrix with jth column $\mathbf{b}_j$. Let $\mathbf{u} \neq \mathbf{0}$ be any nonzero $n \times 1$ vector. Notice that the product $\mathbf{Bu}$ is itself an $n \times 1$ vector. In fact, it is a linear

combination of the columns of $\mathbf{B}$, $\mathbf{Bu} = \sum_{i=1}^{n} \mathbf{b}_i u_i$. Suppose further that there exists a scalar $\lambda \neq 0$ relating the vector $\mathbf{Bu}$ to the vector $\mathbf{u}$ via

$$\mathbf{Bu} = \lambda \mathbf{u} \, . \tag{A.3}$$

If (A.3) holds, we say λ is an *eigenvalue* and $\mathbf{u}$ is an *eigenvector* of the square matrix $\mathbf{B}$. Other names for the eigenvalue are 'characteristic value' or 'latent value.' Notice that if $\mathbf{u}$ satisfies (A.3), so will $w\mathbf{u}$ for any scalar $w \neq 0$. Thus eigenvectors are not unique. To establish a standard, eigenvectors are often scaled to $\mathbf{u}^{\mathrm{T}}\mathbf{u} = 1$. These are then called *unit eigenvectors*.

It can be shown that under (A.3) with $\mathbf{u} \neq \mathbf{0}$, the matrix $\mathbf{B} - \lambda\mathbf{I}$ must be singular. That is,

$$|\mathbf{B} - \lambda\mathbf{I}| = 0. \tag{A.4}$$

This is another equation – called the characteristic polynomial of $\mathbf{B}$ – which can be solved for λ. As such, some authors use (A.4) as the definition of an eigenvalue. As the equation defines the roots of an nth order polynomial in λ, from the Fundamental Theorem of Algebra (Borwein and Erdélyi 1995, Section 1.1), there are at most n real, distinct eigenvalues for $\mathbf{B}$. (Note that some roots may be imaginary and/or repeated.)

Other results for an $n \times n$ matrix $\mathbf{B}$ with eigenvalue λ and corresponding eigenvector $\mathbf{u}$ are as follows:

- $w\lambda$ is an eigenvalue of $w\mathbf{B}$ for any nonzero scalar $w \neq 0$.
- λ^2 is an eigenvalue of $\mathbf{B}^2$ with corresponding eigenvector $\mathbf{u}$.
- If $\mathbf{B}$ is nonsingular, $1/\lambda$ is an eigenvalue of $\mathbf{B}^{-1}$, with corresponding eigenvector $\mathbf{u}$.
- If $\mathbf{B}$ is symmetric, all its eigenvalues are real.
- If $\mathbf{B}$ is symmetric with two distinct eigenvalues λ_1 and λ_2, then the eigenvectors, $\mathbf{u}_1$ and $\mathbf{u}_2$, corresponding to these eigenvalues are orthogonal: $\mathbf{u}_1^{\mathrm{T}}\mathbf{u}_2 = \mathbf{u}_2^{\mathrm{T}}\mathbf{u}_1 = 0$.
- If $\mathbf{B}$ has eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, then $\mathrm{tr}(\mathbf{B}) = \sum_{i=1}^{n} \lambda_i$ and $|\mathbf{B}| = \prod_{i=1}^{n} \lambda_i$.

The latter result shows that if $\mathbf{B}$ possesses a zero-valued eigenvalue, it is singular. Further, a symmetric, nonnegative definite matrix has only real, nonnegative eigenvalues, while a symmetric, positive-definite matrix has only real, positive eigenvalues (Gentle 2007, Section 3.8). Notice then that a symmetric, positive-definite matrix has determinant $|\mathbf{B}| = \prod_{i=1}^{n} \lambda_i > 0$ and, therefore, must be nonsingular.

Eigenanalysis is possible by hand, but the computer makes the effort less daunting. For instance, in **R**, use the `eigen(X)` function where `X` is an $n \times n$ array of class `matrix`.

A.6 Matrix factorizations

Given a matrix with certain features – nonnegative definite, symmetric, and so on – a number of factorizations and decompositions have evolved to characterize its structure. A few of these useful for data analytics are presented in this section. For deeper expositions, see, for example, (Gentle 2007, Ch. 5) or Skillicorn (2007).

A.6.1 QR decomposition

A simple, yet very useful decomposition of an $n \times p$ matrix $\mathbf{B}$ is

$$\mathbf{B} = \mathbf{QR},$$

where $\mathbf{Q}$ is a $n \times n$ orthogonal matrix and $\mathbf{R}$ is trapezoidal. If $n = p$ so that $\mathbf{B}$ is square, $\mathbf{R}$ will be upper triangular.

This *QR decomposition* has many uses; one often sees it employed when solving the system of equations $\mathbf{Bx} = \mathbf{d}$. Computing is facilitated in R via the `qr()` function. For more on the QR decomposition, see Gentle (2007, Section 5.7).

A.6.2 Spectral decomposition

Suppose that a square $n \times n$ matrix $\mathbf{B}$ has ordered eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and that these are collected into a diagonal matrix $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$. Take $\mathbf{U}$ as the matrix whose columns are the corresponding eigenvectors: $\mathbf{U} = [\mathbf{u}_1\ \mathbf{u}_2\ \cdots\ \mathbf{u}_n]$. Then, from (A.3), we can write $\mathbf{BU} = \mathbf{U\Lambda}$. From this, if $\mathbf{U}$ is nonsingular, postmultiplying by $\mathbf{U}^{-1}$ achieves the *diagonal factorization* of $\mathbf{B}$:

$$\mathbf{B} = \mathbf{U\Lambda U}^{-1}. \tag{A.5}$$

It can be shown that $\text{tr}(\mathbf{B}) = \text{tr}(\mathbf{\Lambda})$ and $|\mathbf{B}| = |\mathbf{\Lambda}|$. Further, the number of nonzero eigenvalues in $\mathbf{\Lambda}$ is equal to $\text{rank}(\mathbf{B})$.

When $\mathbf{B}$ is symmetric, (A.5) will always apply. If the $\mathbf{u}_j$s are then assumed to be unit eigenvectors, the matrix $\mathbf{U}$ will be orthogonal. That is, $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}$, and we can write $\mathbf{U}^{-1} = \mathbf{U}^{\mathrm{T}}$. Replacing this in (A.5) produces the *spectral decomposition* of a symmetric matrix $\mathbf{B}$:

$$\mathbf{B} = \mathbf{U\Lambda U}^{\mathrm{T}}. \tag{A.6}$$

Many authors automatically default to eigenvectors in unit form and thus, at least for a symmetric matrix, view (A.5) and (A.6) as reformulations of each other. Indeed, because the eigenvalues/vectors play critical roles in both, one often sees the term 'eigendecomposition' used for either factorization. For computing purposes, in R, the various components of the spectral decomposition are easily extracted from the `eigen()` function.

A.6.3 Matrix square root

Cases occur in data analytics where a matrix $\mathbf{B}^{\frac{1}{2}}$ is required to satisfy the relationship

$$(\mathbf{B}^{\frac{1}{2}})^2 = \mathbf{B}, \tag{A.7}$$

for some given matrix $\mathbf{B}$. One is tempted to call $\mathbf{B}^{\frac{1}{2}}$ the 'square root' of $\mathbf{B}$. As matrix multiplication is more complex than scalar multiplication, however, the notion of a matrix square root requires careful development. For the discussion here, focus is on $n \times n$ symmetric matrices where $\mathbf{B}^{\mathrm{T}} = \mathbf{B}$.

Start with a simple case. Suppose $\mathbf{B}$ is a diagonal matrix of the form $\mathbf{B} = \text{diag}\{b_1, b_2, \ldots, b_n\}$ with $b_i \geq 0$ for all i. Then clearly

$$\mathbf{B}^{\frac{1}{2}} = \text{diag}\left\{\sqrt{b_1}, \sqrt{b_2}, \ldots, \sqrt{b_n}\right\} \tag{A.8}$$

always exists and satisfies (A.7). Define (A.8) as the nonnegative definite square root of a diagonal matrix.

Building from this, suppose the symmetric matrix $\mathbf{B}$ is nonnegative definite, with nonnegative, (decreasing) ordered eigenvalues in $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \ldots, \lambda_n\}$ and with corresponding matrix of (unit) eigenvectors $\mathbf{U}$. From (A.6), $\mathbf{B}$ has spectral decomposition $\mathbf{B} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{T}}$. But then, the matrix

$$\mathbf{B}^{\frac{1}{2}} = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^{\mathrm{T}},$$

will satisfy (A.7), using the definition of a diagonal square root in (A.8) for $\mathbf{\Lambda}^{\frac{1}{2}}$. It can be shown that $\mathbf{B}^{\frac{1}{2}}$ is also symmetric and nonnegative definite.

If $\mathbf{B}$ is positive definite, then so is $\mathbf{B}^{\frac{1}{2}}$, with its own inverse matrix $\mathbf{B}^{-\frac{1}{2}} = \mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^{\mathrm{T}}$. Here, $\mathbf{\Lambda}^{-\frac{1}{2}} = \text{diag}\{\lambda_1^{-1/2}, \ldots, \lambda_n^{-1/2}\}$.

For more on square roots of matrices, see Gentle (2007, Section 3.8).

A.6.4 Singular value decomposition

A more general decomposition, of which the spectral decomposition in Section A.6SD can be viewed as a special case, applies to any $n \times p$ matrix $\mathbf{X}$. Known as the *singular value decomposition*, or SVD, it gives a factorization of $\mathbf{X}$ in the form

$$\mathbf{X} = \mathbf{Q}\mathbf{D}\mathbf{V}^{\mathrm{T}}, \tag{A.9}$$

where $\mathbf{Q}$ and $\mathbf{V}$ are $n \times n$ and $p \times p$ orthogonal matrices, respectively, and $\mathbf{D}$ is an $n \times p$ 'diagonal' matrix with entries $d_{11} \geq d_{22} \geq \cdots d_{rr} > 0$ along its leading diagonal up to $r = \text{rank}(\mathbf{X}) \leq \min\{n, p\}$ and with all other entries as zero. The d_{ii}s from $\mathbf{D}$ are known as the *singular values* of $\mathbf{X}$. For computing purposes, in R, the various components of the SVD are available via the `svd()` function.

A useful application of the SVD occurs for the matrix product $\mathbf{B} = \mathbf{X}^{\mathrm{T}}\mathbf{X}$. Notice here that $\mathbf{B}$ is square (of order p) and symmetric. Then from (A.9),

$$\mathbf{B} = (\mathbf{Q}\mathbf{D}\mathbf{V}^{\mathrm{T}})^{\mathrm{T}}\mathbf{Q}\mathbf{D}\mathbf{V}^{\mathrm{T}} = \mathbf{V}\mathbf{D}^{\mathrm{T}}\mathbf{D}\mathbf{V}^{\mathrm{T}},$$

appealing to the orthogonality of $\mathbf{Q}$. Because $\mathbf{B}$ is symmetric, we can compare this with (A.6): clearly, $\mathbf{V} = \mathbf{U}$ is the (column) matrix of unit eigenvectors from $\mathbf{X}^{\mathrm{T}}\mathbf{X}$, while

$$\mathbf{D}^{\mathrm{T}}\mathbf{D} = \text{diag}\{d_{11}^2, d_{22}^2, \ldots, d_{rr}^2, 0, \ldots, 0\} \tag{A.10}$$

is the diagonal matrix of its eigenvalues. (If $\mathbf{X}$ has full column rank such that $\text{rank}(\mathbf{X}^{\mathrm{T}}\mathbf{X}) = p$, then $\mathbf{D}$ will contain no zero entries on its diagonal. Recall that the number of nonzero eigenvalues is the rank of a symmetric matrix.) Clearly then, the eigenvalues are the squares of the singular values in $\mathbf{D}$: $\lambda_i = d_{ii}^2$. These are all nonnegative, reminding us that $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is nonnegative definite (positive definite if $p = r$).

From $\mathbf{D}^{\mathrm{T}}\mathbf{D}$ in (A.10), let

$$\mathbf{W} = \mathbf{U}\mathbf{D}^{\mathrm{T}} = [\mathbf{u}_1 d_{11} \;\; \mathbf{u}_2 d_{22} \;\; \cdots \;\; \mathbf{u}_p d_{pp}]^{\mathrm{T}} = [\mathbf{u}_1 \sqrt{\lambda_1} \;\; \mathbf{u}_2 \sqrt{\lambda_2} \;\; \cdots \mathbf{u}_p \sqrt{\lambda_p}\,]^{\mathrm{T}},$$

where $\mathbf{u}_i$ is the ith (unit) eigenvector of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$. One possible realization of the SVD for $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is then the product $\mathbf{W}\mathbf{W}^{\mathrm{T}}$, where $\mathbf{W}$ is the column matrix of eigenvectors multiplied (up to a sign) by the square roots of their eigenvalues.

A.7 Statistics via matrix operations

Many summary statistics can be represented in terms of matrix operations. To see this, suppose a univariate random sample from Section 3.1, $X_i,\ i = 1, \ldots, n$, is collected into a column vector $\mathbf{X} = [X_1 \; X_2 \; \cdots \; X_n]^{\mathrm{T}}$. Recall that the column vector made up entirely of ones is denoted by $\mathbf{h} = [1 \; 1 \; \cdots \; 1]^{\mathrm{T}}$. Then, the sample mean is a scaled dot product of the two: $\overline{X} = \frac{1}{n}\mathbf{h} \cdot \mathbf{X} = \frac{1}{n}\mathbf{h}^{\mathrm{T}}\mathbf{X}$. With this, the *corrected* (by the mean) vector of observations is

$$\mathbf{X}^* = \mathbf{X} - \frac{\mathbf{h}^{\mathrm{T}}\mathbf{X}}{n}\mathbf{h} = \begin{bmatrix} X_1 - \overline{X} \\ X_2 - \overline{X} \\ \vdots \\ X_n - \overline{X} \end{bmatrix}.$$

The corresponding sample variance is the sum of squares of these elements, divided by $n - 1$:

$$S^2 = \frac{1}{n-1}(\mathbf{X}^*)^{\mathrm{T}}\mathbf{X}^*.$$

A p-variate sample, where each component is represented by the column vector $\mathbf{X}_j = [X_{1j} \; X_{2j} \cdots X_{nj}]^{\mathrm{T}}, j = 1, \ldots, p$, can be similarly collected into an $n \times p$ matrix

$$\mathbf{X} = [\mathbf{X}_1 \; \mathbf{X}_2 \; \cdots \; \mathbf{X}_p] = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix}.$$

The corresponding matrix of mean-corrected observations is

$$\mathbf{X}^* = \begin{bmatrix} X_{11} - \overline{X}_1 & X_{12} - \overline{X}_2 & \cdots & X_{1p} - \overline{X}_p \\ X_{21} - \overline{X}_1 & X_{22} - \overline{X}_2 & \cdots & X_{2p} - \overline{X}_p \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} - \overline{X}_1 & X_{n2} - \overline{X}_2 & \cdots & X_{np} - \overline{X}_p \end{bmatrix}. \tag{A.11}$$

For summarizing variation, the sample analog to the population covariance matrix in (2.12) is the *sample covariance matrix*. This locates the sample variances S_j^2 along the diagonal, and the sample covariances from Section 3.3.3

$$S_{jk} = \frac{1}{n-1}\sum_{i=1}^{n}(X_{ij} - \overline{X}_j)(X_{ik} - \overline{X}_k)$$

($j \neq k$) as the off-diagonal elements:

$$\hat{\boldsymbol{\Sigma}} = \begin{bmatrix} S_1^2 & S_{12} & \cdots & S_{1p} \\ S_{21} & S_2^2 & \cdots & S_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ S_{p1} & S_{p2} & \cdots & S_p^2 \end{bmatrix}. \tag{A.12}$$

Note that because $S_{jk} = S_{kj}$ ($j \neq k$), $\hat{\boldsymbol{\Sigma}}$ is symmetric. Indeed, let $\mathbf{C} = (n-1)^{-\frac{1}{2}}\mathbf{X}^*$. Then (A.12) can be written compactly as $\hat{\boldsymbol{\Sigma}} = \mathbf{C}^{\mathrm{T}}\mathbf{C} = \frac{1}{n-1}(\mathbf{X}^*)^{\mathrm{T}}\mathbf{X}^*$.

From (A.12), one can construct the corresponding *sample correlation matrix*:

$$\mathbf{R} = \begin{bmatrix} 1 & r_{12} & \cdots & r_{1p} \\ r_{21} & 1 & \cdots & r_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} & \cdots & 1 \end{bmatrix},$$

where $r_{jk} = S_{jk}/(S_j S_k)$ is the sample correlation coefficient between X_j and X_k, as in (3.9). Again, because $r_{jk} = r_{kj}$ ($j \neq k$), $\mathbf{R}$ is symmetric. If one collects the z-scores $z_{ij} = (X_{ij} - \overline{X}_j)/S_j$ from (3.7) into the matrix $\mathbf{Z} = \{z_{ij}\}$, then $\mathbf{R} = \mathbf{Z}^{\mathrm{T}}\mathbf{Z}/(n-1)$. Note that some authors use $\mathbf{Z}^{\mathrm{T}}\mathbf{Z}/n$ for the correlation matrix; clearly, this will make little difference when n is very large. In **R**, `cor(X)` calculates a correlation matrix when the input argument `X` is a numeric vector, matrix, or data frame. The `cor()` function uses $n-1$ in its denominator.

Appendix B

Brief introduction to R

R is an integrated suite of operators, functions, and user-contributed packages that conducts statistical and mathematical calculations (R Core Team 2014). The program is an open source computational environment developed by Ihaka and Gentleman (1996) and built upon the **S** language. (For background on **S**, see Becker (1994).) The goal of this appendix is to provide a brief overview of **R** and to establish guidelines for how it is used throughout the rest of this textbook. It is not intended to supplant the many quality sources on **R**, including introductions such as Dalgaard (2008) and Verzani (2005), the concise online guide by Owen (http://cran.r-project.org/doc/contrib/Owen-TheRGuide.pdf), and of course the main Comprehensive **R** Archive Network (CRAN) web site and its online manual at http://cran.r-project.org/doc/manuals/R-intro.html. Readers should refer to sources such as these for a deeper introduction to the language.

The program operates on Windows®, Apple OS, and Linux systems. At the time of this writing, the current version number is 3.1.0, available for both 32-bit and 64-bit platforms. It is that version, using 64-bit format, from which the bulk of the **R** material in this text was prepared.

R's open source architecture makes it highly popular and, of course, free. It is a highly effective-yet-flexible form of modern statistical software, provides a powerful graphical subsystem, and has an extensive user network that contributes subordinate external packages to the program's ongoing evolution (see Section B.4).

On most systems, when base **R** is started, it opens a window in which the various commands can be entered and the (nongraphical) output is reported. This is known as the **R** *workspace*. Separate windows will open to display output from the graphic subsystem. The workspace and all **R** variables and objects created during the session can be saved for future retrieval. As with any computer software, it is advisable to save workspace results regularly and often.

Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery,
First Edition. Walter W. Piegorsch.

www.wiley.com/go/piegorsch/data_analytics

Base **R** is command-line driven, and in this textbook, code presented to generate statistical and mathematical calculations employs the command-line format. For distinguishability, this is shown in standard `typewriter text`, either as inline commands or as standalone displays. In the latter case, code to be entered at the command line will have a leading entry mark using **R**'s default 'greater than' sign, `>`. Thus a command to give the sample mean, say, of a vector of numbers `X` will be displayed as

```
> mean( X )
```

Note that all **R** code is case sensitive. `mean(X)` and `mean(x)` will give different values unless `X` and `x` have exactly the same arithmetic mean.

R output will also be presented in `typewriter text`, again either inline or as a standalone display. In the latter case, to distinguish from commands/code, no leading `>` will appear, although, in many cases, **R** will lead with a bracketed numeral to indicate what element of its output stream immediately follows. For instance, if there is only one element to follow, the output leads with `[1]` (illustrated later). Where necessary, output may be edited for presentation purposes.

The basic function to learn about any **R** feature or function is `help()`, where the particular command of interest is placed between the parenthesis marks. For instance,

```
> help( mean )
```

gives a help page on the `mean()` function to calculate arithmetic means. An equivalent shortcut places the command of interest after a questions mark, as in

```
> ?mean
```

If **R** cannot interpret the query, it gives a `No documentation` error.

To quit or end an **R** session, the base command is

```
> q()
```

although each separate operating system may have equivalent menu bar or mouse shortcuts.

R uses special expressions to indicate certain types of values. A number that cannot be calculated, for example, 0/0, is 'not a number,' or `NaN`. Calculations that include an `NaN` will themselves result in an `NaN`. This differs, however, from a number whose limit is (unambiguously) ∞, say, 1/0. If **R** recognizes the infinite value, it will assign it the special expression `Inf`. Any calculations that involve an `Inf` attempt to parse the limiting infinite value. This can result in another `Inf`, a finite value, or even an `NaN`, depending on the particulars. Missing values are possible and are expressed as `NA`. Note that this differs from `NaN`, above.

As any **R** user will affirm, the best way to learn **R** is to use **R**: experiment with the various commands (and options), sort through the `help()` files (some of which can be admittedly opaque), and apply the functions and methods to data. Besides its use as a computing environment, **R** is also an effective tool for learning and understanding data-analytic thinking.

The following sections give some examples to help get started.

B.1 Data entry and manipulation

Data can be entered into the current workspace in various ways. The simplest is with the `c()` (for 'concatenate') command, for example,

```
> c( 1, -1.2, pi )   #these are three numbers
```

creates a vector with three entries, the last of which is the internally held constant $\pi = 3.14159265...$ (The # hash symbol is **R**'s comment character: all text after a # is ignored until the next carriage return.) This vector can be assigned to an object or variable that retains the three values for future use. **R**'s default assignment operator is the left arrow ←, that is, assign the quantity on the right to the quantity on the left (a very programming-oriented construction). Most keyboards do not carry a left-arrow key, however, so **R** uses the two keystrokes <- (i.e., 'less than' and 'minus') with no space between them. As each of these symbols carries its own (different!) meaning when applied separately in **R**, users must apply caution when entering them. In its newer versions, **R** allows equivalent use of the equal sign = for assignment. (The two expressions can act differently in certain cases, so caution is advised. See the **R** online manual.) Thus, for example, to assign the three numbers above to the variable X, use either of the following:

```
> X <- c(1, -1.2, pi)
> X =  c(1, -1.2, pi)    #same assignment
```

To display the contents of an existing object in the workspace, use the `print()` function. To list the current contents of the workspace, including all assigned variables, type `ls()`.

It is good practice to keep the workspace compact, so that excess/temporary objects do not build up. The `remove()` function, also simply `rm()`, deletes unwanted variables, objects, and functions. (Needless to say, this should be used with caution: there is no 'unremove' function.)

Larger data sets can be entered using other, more powerful functions. The `scan()` function scans in keyboard input continuously until two successive carriage returns are entered. This is useful for entering long streams of data. It can also be applied in conjunction with file system paths that direct to an external file, for example, in Windows, the command

```
> X <- scan( "C:/datafile.txt" )
```

finds the ASCII text file `datafile.txt` located on the user's `C:` drive and assigns its contents into the variable X. (Notice use of the forward slash.) If the file is located deep in the file system, dialog-box navigation can be invoked via `file.choose()`:

```
> X <- scan( file.choose() )
```

Inbuilt functions can perform common assignment tasks. For instance, `seq(from=,to=, by=)` produces sequences of numbers, while `rep()` repeats blocks of material a given number of times. To wit,

```
> seq( from=1, to=13 )        #default is by=1
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13

> 1:13               #shortcut for seq( 1,13 )
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13

> seq( from=13, to=4, by=-1 )
 [1]  13 12 11 10  9  8  7  6  5  4

> seq( from=3, to=7.5, by=0.5 )
 [1] 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5

> rep( 13, times=4 )
 [1] 13 13 13 13
```

For multiple-variable databases or tables of data, a series of `read` functions can be used. For instance, `read.table(file.choose())` will read a variety of different database formats, here using dialog-box navigation; see `help(read.table)`. **R** can, in fact, read and write data in many forms, including interfaces with other popular software programs. For more details, see the CRAN introduction at http://cran.r-project.org/doc/manuals/R-intro.html #Reading-data-from-files or any of the introductory sources mentioned earlier.

Once entered, variables can be manipulated in a variety of ways. Summary functions or other operations applied to vectors produce either scalar- or vector-valued results. For instance, given the three numbers in `X`, above, we find

```
> length( X )
 [1] 3

> sum( X )
 [1] 2.941593

> round( sum(X), digits=4 )
 [1] 2.9416
```

```
> mean( X )
 [1] 0.9805309

> rep( X, times=2 )
 [1]  1.000000 -1.200000
 [3]  3.141593  1.000000
 [5] -1.200000  3.141593
```

R operates seamlessly with vector- and matrix-valued objects; indeed, it is an object-oriented programming environment. Where necessary, a single variable containing a stream of numbers is treated as a column vector (even though it displays as a row). To create an $n \times p$ matrix, use the `matrix()` command; see `help(matrix)`. For example,

```
> M <- matrix( seq( 2,7 ), nrow=2, ncol=3 )
```

produces the 2×3 matrix

```
      [,1] [,2] [,3]
[1,]    2    4    6
[2,]    3    5    7
```

Notice that data entry is by column; this can be modified via options in `matrix()`. **R** can also 'bind' two or more vectors together into a matrix, via the `cbind()` and `rbind()` commands. The former binds columns, the latter binds rows. To coerce an **R** object into a matrix, use `as.matrix()`.

A special kind of array object in **R** is known as a *data frame*. Data frames are used to collect p different $n \times 1$ vectors and/or $n \times p$ matrices together when their ith rows relate to one another. This is particularly useful in statistical applications: suppose a response variable Y is associated with two predictor variables X_1 and X_2. A data frame connects the ith observation in each vector in order to facilitate future statistical calculations. (Thus, any data frame's column vectors must all have the same length. Missing values, as `NA`, are permitted.)

In **R**, the `data.frame()` command constructs data frames. For example,

```
> Y <- c( 3, 9, 5.5, -1, -1.1 )
> X1 <- seq( from=3, to=5, by=0.5 )
> X2 <- c( 540.4, 100.2, 221.0, 490.1, NA )
> data.df <- data.frame( Y, X1, X2 )
```

yields

```
       Y   X1     X2
  1  3.0  3.0  540.4
  2  9.0  3.5  100.2
  3  5.5  4.0  221.0
  4 -1.0  4.5  490.1
  5 -1.1  5.0     NA
```

The first column of the output is the row number. The remainder is the actual data frame with the columns aligned by row. Notice that column names correspond to the entry variables.

Elements within **R** vectors, matrices, data frames, etc. are referenced by square brackets. For instance, in the matrix `M` from above, the (2,3)rd element lies in the second row and the third column and is referenced as `M[2,3]`. Notice that row indices precede column indices. The entire third column is simply `M[,3]`, while the entire second row is `M[2,]`. Indexing for a data frame is similar: for example, calling `data.df[4,3]` above reports the value `490.1`.

A minus sign used in a bracketed index indicates deletion; for example, to remove the second and third rows from `data.df`, use

```
> data.df[ -c(2,3), ]
       Y   X1     X2
  1   3.0  3.0  540.4
  4  -1.0  4.5  490.1
  5  -1.1  5.0     NA
```

R's object-oriented structure allows for creative use of the bracketed indexing feature to created subsetted objects; see http://cran.r-project.org/doc/manuals/R-intro.html#Index-vectors. For instance, to identify those elements in a vector below a specific threshold, embed the threshold as a logical query (cf. Table B.1):

```
> Y[ Y<0 ]
 [1] -1.0  -1.1
```

Any **R** object, including data frames, possesses a series of internal attributes that describe and summarize its contents. To view these, apply the `attributes()` command, for example,

```
> attributes( data.df )
 $names
 [1] "Y"  "X1" "X2"

 $row.names
 [1] 1 2 3 4 5

 $class
 [1] "data.frame"
```

Users can access subcomponents from any **R** object for further calculation. For most of the objects seen herein, the `$` symbol acts as a separator between the object name and the subcomponent name:

```
>  data.df$Y
 [1]  3.0  9.0  5.5 -1.0 -1.1
```

Combine this with bracketed indexing, for example, to access individual elements of any multicomponent object:

```
>  data.df$X2[ c(2,4) ]
 [1] 100.2  490.1
```

B.2 A turbo-charged calculator

Many authors affirm, sometimes tongue-in-cheek, that at its core **R** is 'a turbo-charged calculator.' That is, given a series of mathematical operations, **R** will report their result as reliably as a typical hand calculator – with, often, higher floating-point accuracy. Consider the following commands, functions (which should be obvious), and their resulting outputs:

```
> 13 + 4
[1] 17

> 13 * 4
[1] 52

> 13^2
[1] 169

> sqrt( 13 )
[1] 3.605551
```

```
> abs( 4 - 13 )
[1] 9

> cos( pi )
[1] -1

> log( 13 - 4 )
[1] 2.197225

> factorial(4)
[1] 24
```

Here, `log()` is the natural logarithm unless specified otherwise with an optional argument – see `help(log)` – and the `factorial()` function simply applies (2.19). Notice that the (scalar) multiplication operator is the usual `*`.

Vector and matrix arithmetic is performed with special commands. Matrix addition and subtraction is elementwise, as in Section A.2, so just apply the + and – operators, respectively. Matrix multiplication requires a special operator, however: `%*%`. (In all cases, of course, the operands must be conformable.) Inverses of square, nonsingular matrices are found with the `solve()` function. When needed, transposition is achieved with the `t()` function. Thus, for example, with the 3×1 vector `X` and the 2×3 matrix `M` seen earlier, we have

```
>  M %*% X
          [,1]
  [1,] 16.04956
  [2,] 18.99115

> X %*% M
  Error in X %*% M : non-conformable arguments

> t(X) %*% t(M)
          [,1]     [,2]
  [1,] 16.04956 18.99115
```

The `diag()` function creates diagonal matrices from input values (among other features). So, for example,

```
> D <- diag(X)
> print(D)
        [,1]  [,2]      [,3]
   [1,]    1   0.0  0.000000
   [2,]    0  -1.2  0.000000
   [3,]    0   0.0  3.141593

> solve(D)
        [,1]      [,2]     [,3]
   [1,]    1   0.00000  0.00000
   [2,]    0  -0.83333  0.00000
   [3,]    0   0.00000  0.31831
```

Readers may verify that `solve(D)%*%D` is indeed the 3×3 identity matrix $\mathbf{I}$.

Besides its ability to produce basic mathematical calculations, **R** also provides an interface to a powerful graphics subsystem. (Almost all the plots in this text were constructed using **R**'s graphical capabilities.) At its core is the basic `plot()` command. For example, to plot an ordinate variable `Y` against an abscissa variable `X`, use `plot(X,Y)` or equivalently `plot(Y ~ X)`. In either case, order is important.

Plot output in **R** can be manipulated in a variety of ways, and this allows for some imaginative graphical displays. For more details, see http://cran.r-project.org/doc/manuals/R-intro.html#Graphics and Murrell (2011).

B.3 R functions

B.3.1 Inbuilt R functions

As illustrated earlier, **R** contains a large collection of inbuilt functions for mathematical and statistical calculations. These all use parentheses to sequester their arguments, although some functions such as `ls()` or `q()` may not require any arguments. The parentheses are still necessary, however. In this textbook, an **R** function will, in most cases, be referred to with its parentheses, but if the use is generic no actual arguments or argument placeholders will be included. When specific arguments or inputs are required, these will be listed.

To learn about any function, simply apply `help()`, as suggested above. This usually reports the required and/or optional arguments each functions takes and gives further detail on how these are used. If a particular option is specified formally in the function's help screen, the specification represents the default value for that option. Argument or option position is assumed as given. For example, `help(log)` reports that the `log()` function has the complete form `log(x, base=exp(1))`. That is, the first argument to `log()` is some value `x`, and if a second argument is given, it is the desired base of the logarithmic calculation. If no second argument is given, the function defaults to the natural log with `base=exp(1)`.

Some help screens will show an ellipsis (. . .) in a function's description. This simply indicates that further arguments can be passed to the function from other functions or operations, giving the programmer added flexibility.

Position in a function call can be overridden by specifying the arguments or options by their individual names, in any order. The default position is imposed only if the argument(s)

is given as a standalone value. Thus with the `log()` function, the following calls are all equivalent:

```
> log( pi )
 [1] 1.14473

> log( x=pi )  #from help(log): specify the operand via "x="
 [1] 1.14473

> log( base=exp(1), x=pi )
 [1] 1.14473
```

A semicolon (`;`) in **R** acts as a separator, allowing for multiple commands or operations to be given before a carriage return. Thus, for example,

```
> x <- sqrt( pi ); print( x+1 )
```

and

```
> x <- sqrt( pi )
> print( x+1 )
```

both produce the same result (2.772454, for the record).

Special forms of *logical operations* can also be performed, either within a function that calls for a logical argument or simply as a standalone logical query. The basic logical operators are given in Table B.1.

The outcome of a logical operation is either TRUE or FALSE. This may seem obvious, but in fact, **R** treats these as actual values: `TRUE` or `FALSE`. Shorthand equivalents are `T` or `F`. (Thus, it is usually unwise to name a separate variable as `T` or `F` – popular letters in statistical parlance! – because these could confuse future logical operations.) For instance, to determine if the fourth element of a vector `X2` equals a certain number, say 14.2, write `X2[4]==14.2`. Include a logical OR to expand the query:

```
> print( X2 )
 [1] 540.4  100.2  221.0  490.1    NA

> ( X2[4] == 14.2 ) | ( X2[4] < 2.72 )    #logical OR via "|"
[1] FALSE
```

Combining logical operators with element referencing via square brackets gives **R** programming particular pliancy. For example, the `is.na()` function reports the indices of those elements missing (‘`NA`’) in an object. These can then be applied to that object (or any other object) to excise the missing observations:

Table B.1 Logical operators in **R**.

Operator	Operation	Operator	Operation
`==`	Equal to	`!=`	Not equal to
`<`	Less than	`<=`	Less than or equal to
`>`	Greater than	`>=`	Greater than or equal to
`&`	Logical AND	`|`	Logical OR

```
> mean( X2 )
[1] NA

> is.na( X2 )
 [1] FALSE FALSE FALSE FALSE  TRUE

> X2[ is.na(X2)==F ]
 [1] 540.4 100.2 221.0 490.1

> mean( X2[is.na(X2)==F] )
 [1] 337.925
```

By the way, this particular calculation can be accomplished simply by using the `na.rm=` option (for 'NA remove'):

```
> mean( X2, na.rm=TRUE )
 [1] 337.925
```

See `help(mean)`.

B.3.2 Flow control

Users can apply standard loops and conditional execution for flow control of **R** calculations. The program has the usual `for()`, `while()`, and `repeat()` loop capabilities. It also allows for `if()` statements, where the argument to the `if()` command is a logical query; for example, to set the fifth element of the vector `Y` equal to zero if the corresponding element of `X2` is missing, use

```
> if( is.na(X2[5])==T ) { Y[5]=0 }
```

(The braces contain the operation to be executed if the argument is `TRUE`. These are not necessary if the executable operation can be contained on a single input line.)

The `else` command can be combined with `if()` to enhance flow control. A special `ifelse()` command is also available; see http://cran.r-project.org/doc/manuals/R-intro.html#Loops-and-conditional-execution.

B.3.3 User-defined functions

R allows programmers to construct their own *user-defined functions* via the `function()` command. The standard syntax is

```
> functionName <- function( arg_1, arg_2, ... ) {
                  executable_1
                  executable_2
                  ⋮
                  }   #end of function
```

where `functionName` is the user's choice for the name of the function, each `arg` is an input argument to the function, and the `executable` statements contained by braces are various lines of **R** code to be executed. For the function to actually return a result, the desired output quantity must be written as the last line of the executable statements,

or be reported via the `return()` subfunction. See http://cran.r-project.org/doc/manuals/R-intro.html#Writing-your-own-functions for details.

One can also take advantage of **R**'s capabilities as a programming language, although details on this are generally outside the scope of this text. Advanced users may gain from the treatments by Jones et al. (2009) and Gentleman (2009). Also see Venables and Ripley (2000).

B.4 R packages

Another powerful feature in **R** is its use of libraries and packages to assemble specialized functions. These are stored as separate compendiums for more efficient access. For example, the core *stats* package contains the basic statistical functions; it loads automatically when **R** starts. (Packages are referenced in this text using *italic* naming. Use these to refer to the CRAN web site at http://cran.r-project.org/doc/manuals/R-intro.html#Packages for full information.) Other packages can be more exotic, such as the *lattice* package for trellis graphics. To see the full list of available packages for a local installation, type `library()` after starting **R**.

Except for the base collection, most packages do not load automatically and must be explicitly invoked. Either of the commands `library()` or `require()` can load an installed package. For example, to load the *lattice* package, type `library(lattice)` or `require(lattice)`. To detach the loaded package, type `detach(package:lattice)`.

Beside the base offerings, a wealth of external, add-on packages has been developed by the **R** user community. Most are available from the CRAN web site at http://cran.r-project.org/web/packages/. Each package provides a targeted and sometimes highly specific set of statistical and mathematical functions, many of which are useful in modern data analytics. Where appropriate, specific examples are highlighted throughout the main chapters of this textbook.

External packages must first be installed (just once) on the computer's local storage device before they can be loaded into an active workspace. To install an external package, follow the instructions in `help(install.packages)`. (The program may first require that a specific CRAN repository or mirror site be specified for online access.) The process is facilitated on certain platforms with menu-driven options; for example, modern Windows installations provide a 'Packages' menu for package retrieval and management. Once installed locally, a package is loaded for case-by-case use with `library()` or `require()`, as above.

A few distinctive packages are worth particular mention. First is a series designed for use with exceptionally large data sets, as would be common in large-scale data analytics (the 'big data' setting). When **R** is employed on desktop computers and standalone workstations, it will generally be limited to the machine's available RAM. Depending on the local setup, a massive data set can tax the program's abilities. In response, developers have contributed specialized functions that break the data into smaller portions and then map these to RAM in a sequential or segmented manner. An archetypal example is the external *ff* package; its extension *ffbase* provides an extensive toolkit of functions for use on massive data sets, including arithmetic operators, data manipulation, summary statistics, and data transformations. Also useful is the external *biglm* package, which performs linear (Chapters 6 and 7) and generalized linear (Chapter 8) regression with massive data. Other advances along these lines include the external *bigmemory* package and its extension, *biganalytics*. Development of packages such as these for massive data continues apace and will likely prove a valuable asset in **R**'s ongoing evolution.

For users uncomfortable with the command-line nature of base **R**, the external *Rcmdr* package – also known as 'R Commander' – provides graphical user interface (GUI) capability for an **R** session. The package is popular for its intuitive feel, its selection of statistical functions, and, especially, its value as a tool for introducing basic statistical methods with **R**. See Wilson (2012) for a comparative review. *Rcmdr* can also integrate with another useful tool, RStudio®. Although not itself an **R** package, RStudio is an application programming interface (API) that oversees **R** sessions; manages the workspace; debugs, edits, and executes **R** code; and produces pertinent graphical output. It is particularly useful for integrated code/software development and can be accessed in either open source or commercial editions. More detail is available at http://www.rstudio.com/.

Readers are cautioned that any references herein to external **R** packages represent neither endorsements for nor approbation of those packages. As with any application of an open source product, users operate at their own gain and risk.

References

Abdi H and Williams LJ (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics* **2**(4), 433–459.

Abrahantes JC, Sotto C, Molenberghs G, Vromman G and Bierinckx B (2011). A comparison of various software tools for dealing with missing data via imputation. *Journal of Statistical Computation and Simulation* **81**(11), 1653–1675.

Agrawal R, Imielinski T and Swami AN (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data – SIGMOD '93, Washington, DC, May 26–28, 1993* (eds. Buneman P and Jajodia S), pp. 207–216. ACM Press, New York.

Agrawal R and Srikant R (1994). Fast algorithms for mining association rules in large databases. In *VLDB '94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago, Chile* (eds. Bocca JB, Jarke M and Zaniolo C), pp. 487–499. Morgan Kaufmann, San Francisco, CA.

Agresti A (2013). *Categorical Data Analysis*, 3rd edn. John Wiley & Sons, Inc., Hoboken, NJ.

Agresti A and Coull BA (1998). Approximate is better than "exact" for interval estimation of binomial proportions. *American Statistician* **52**(2), 119–126.

Ahmad IA and Ran IS (2004). Data based bandwidth selection in kernel density estimation with parametric start via kernel contrasts. *Journal of Nonparametric Statistics* **16**(6), 841–877.

Akaike H (1973). Information theory and an extension of the maximum likelihood principle. In *Proceedings of the 2nd International Symposium on Information Theory (Tsahkadsor, 1971)* (eds. Petrov BN and Csáki B), pp. 267–281. Akadémiai Kiadó, Budapest.

Allen DM (1974). The relationship between variable selection and data augmentation and a method for prediction. *Technometrics* **16**(1), 125–127.

Anderson TW and Olkin I (1985). Maximum-likelihood estimation of the parameters of a multivariate normal distribution. *Linear Algebra and its Applications* **70**, 147–171.

Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery,
First Edition. Walter W. Piegorsch.

www.wiley.com/go/piegorsch/data_analytics

Anonymous (1949). News. *Mathematical Tables and Other Aids to Computation* **3**(28), 544–547.

Anonymous (2012). P&GJ's annual pipeline report: 500 leading gas distribution utilities. *Pipeline & Gas Journal* **239**(11), 22–30.

Anonymous (2013). 2013 hotel management survey: top third-party management companies. *Hotel Management* **288**(3), 26–36.

Armitage P (1955). Tests for linear trends in proportions and frequencies. *Biometrics* **11**(3), 375–386.

Astuti E and Yanagawa T (2002). Trend test for count data with extra-Poisson variability. *Biometrics* **58**(2), 398–402.

Austin E, Pan W and Shen X (2013). Penalized regression and risk prediction in genome-wide association studies. *Statistical Analysis and Data Mining* **6**(4), 315–328.

Babu GJ and Singh K (1983). Inference on means using the bootstrap. *Annals of Statistics* **11**(3), 999–1003.

Bailer AJ and Oris JT (1997). Estimating inhibition concentrations for different response scales using generalized linear models. *Environmental Toxicology and Chemistry* **16**(7), 1554–1560.

Barnett V and Lewis T (1995). *Outliers in Statistical Data*, 3rd edn. John Wiley & Sons, Ltd, Chichester.

Bartholomew D, Knott M and Moustaki I (2011). *Latent Variable Models and Factor Analysis: A Unified Approach*, 3rd edn. John Wiley & Sons, Ltd, Chichester.

Bartlett MS (1938). Further aspects of the theory of multiple regression. *Proceedings of the Cambridge Philosophical Society: Mathematical and Physical Sciences* **34**(1), 33–40.

Bartlett MS (1941). The statistical significance of canonical correlations. *Biometrika* **32**(1), 29–37.

Bartlett MS (1950). Tests of significance in factor analysis. *British Journal of Statistical Psychology* **3**(2), 77–85.

Bayes T (1763). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London* **53**, 370–418.

Becker RA (1994). A brief history of S. In *Computational Statistics: Papers Collected on the Occasion of the 25th Conference on Statistical Computing at Schloß Reisensburg* (eds. Dirschedl P and Ostermann R), pp. 81–110. Physica-Verlag, Heidelberg.

Beirlant J, Dudewicz EJ, Györfi L and van der Meulen EC (1997). Nonparametric entropy estimation: an overview. *International Journal of Mathematical and Statistical Sciences* **6**(1), 17–39.

Belsley DA, Kuh E and Welsch RE (1980). *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. John Wiley & Sons, Inc., New York.

Benjamini Y (1988). Opening the box of a boxplot. *American Statistician* **42**(4), 257–262.

Benjamini Y and Hochberg Y (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B (Methodological)* **57**(1), 289–300.

Berger RL (1996). More powerful tests from confidence interval *p* values. *American Statistician* **50**(4), 314–318.

Berk R, Brown LD, Buja A, Zhang K and Zhao L (2013). Valid post-selection inference. *Annals of Statistics* **41**(2), 802–837.

Bertin-Mahieux T, Ellis DP, Whitman B and Lamere P (2011). The million song dataset. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), Miami, FL, October 24–28, 2011* (eds. Klapuri A and Leider C), pp. 591–596. University of Miami, Miami, FL.

Birren JE and Morrison DF (1961). Analysis of the WAIS subtests in relation to age and education. *Journal of Genontology* **16**(4), 363–369.

Blæsild P and Granfeldt J (2002). *Statistics with Applications in Biology and Geology*. Chapman & Hall/CRC Press, Boca Raton, FL.

Blyth CR and Still HA (1983). Binomial confidence intervals. *Journal of the American Statistical Association* **78**(381), 108–116.

Bonferroni CE (1936). Teoria statistica delle classi e calcolo delle probabilityà. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze* **8**, 3–62.

Borgelt C (2012). Frequent item set mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2**(6), 437–456.

Borwein P and Erdélyi T (1995). *Polynomials and Polynomial Inequalities*. Springer-Verlag, New York.

Box GEP and Cox DR (1964). An analysis of transformations (with discussion). *Journal of the Royal Statistical Society, Series B (Methodological)* **26**(2), 211–252.

Box GEP, Jenkins GM and Reinsel GC (2008). *Time Series Analysis: Forecasting and Control*, 4th edn. John Wiley & Sons, Inc., Hoboken, NJ.

Bradley RA and Srivastava SS (1979). Correlation in polynomial regression. *American Statistician* **33**(1), 11–14.

Breiman L (1996). Bagging predictors. *Machine Learning* **24**(2), 123–140.

Breiman L (2001). Random forests. *Machine Learning* **45**(1), 5–32.

Breiman L, Friedman J, Olshen RA and Stone CJ (1984). *Classification and Regression Trees*. Wadsworth, Belmont, CA.

Bretz F, Hothorn T and Westfall P (2011). *Multiple Comparisons Using R*. Chapman & Hall, Boca Raton, FL.

Brijs T, Swinnen G, Vanhoof K and Wets G (1999). The use of association rules for product assortment decisions: A case study. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining – SIGKDD '99, San Diego, CA, August 15–18, 1999* (eds. Fayyad U, Chaudhuri S and Madigan D), pp. 254–260. ACM Press, New York.

Broberg P (2003). Statistical methods for ranking differentially expressed genes. *Genome Biology* **4**(6), Article no. R41.

Brown LD (1986). *Fundamentals of Statistical Exponential Families, with Applications in Statistical Decision Theory*, *Institute of Mathematical Statistics Lecture Notes – Monograph Series*, vol. **9**. Institute of Mathematical Statistics, Hayward, CA.

Brown LD, Cai TT and DasGupta A (2001). Interval estimation for a binomial proportion (with discussion). *Statistical Science* **16**(2), 101–133.

Brown LD, Cai TT and DasGupta A (2002). Confidence intervals for a binomial proportion and asymptotic expansions. *Annals of Statistics* **30**(1), 160–201.

Buja A, Cook D, Hofmann H, Lawrence M, Lee EK, Swayne DF and Wickham H (2009). Statistical inference for exploratory data analysis and model diagnostics. *Philosophical Transactions of the Royal Society of London, Series A: Mathematical, Physical and Engineering Sciences* **367**(1906), 4361–4383.

Buonaccorsi JP (2012). Fieller's theorem. In *Encyclopedia of Environmetrics*, Vol. **2**, 2nd edn. (eds. El-Shaarawi AH and Piegorsch WW), pp. 1015–1017. John Wiley & Sons, Ltd, Chichester.

Caliński T and Harabasz J (1974). A dendrite method for cluster analysis. *Communications in Statistics* **3**(1), 1–27.

Cardoso MG (2013). Logical discriminant models. In *Quantitative Modeling in Marketing and Management* (eds. Moutinho L and Huarng KH), pp. 223–253. World Scientific Publishing, Singapore.

Carroll RJ and Ruppert D (1988). *Transformation and Weighting in Regression*. Chapman & Hall, New York.

Carroll SS (1998). Modelling abiotic indicators when obtaining spatial predictions of species richness. *Environmental and Ecological Statistics* **5**(3), 257–276.

Casella G and Berger RL (2002). *Statistical Inference*, 2nd edn. Duxbury Press, Pacific Grove, CA.

Cattell RB (1966). The Scree test for the number of factors. *Multivariate Behavioral Research* **1**(2), 245–276.

Cavanaugh JE (1999). A large-sample model selection criterion based on Kullback's symmetric divergence. *Statistics & Probability Letters* **42**(4), 333–343.

Cayley A (1858). A memoir on the theory of matrices. *Philosophical Transactions of the Royal Society of London* **148**(1), 17–37.

Chadeau-Hyam M, Hoggart CJ, O'Reilly PF, Whittaker JC, De Iorio M and Balding DJ (2008). Fregene: Simulation of realistic sequence-level data in populations and ascertained samples. *BMC Bioinformatics* **9**(9), Article No. 364.

Charytanowicz M, Niewczas J, Kulczycki P, Kowalski PA, Łukasik S and Żak S (2010). A complete gradient clustering algorithm for features analysis of X-ray images. In *Information Technologies in Biomedicine*, Vol. **2**, AISC **69** (eds. Piętka E and Kawa J), pp. 15–24. Springer-Verlag, Berlin.

Chen Z, Bai Z and Sinha BK (2004). *Ranked Set Sampling: Theory and Applications*, *Lecture Notes in Statistics*, Vol. **176**. Springer, New York.

Chen Y, Du P and Wang Y (2014). Variable selection in linear models. *Wiley Interdisciplinary Reviews: Computational Statistics* **6**(1), 1–9.

Christensen R (2011). *Plane Answers to Complex Questions. The Theory of Linear Models*, 4th edn. Springer, New York.

Claeskens G and Hjort NL (2003). The focused information criterion (with discussion). *Journal of the American Statistical Association* **98**(464), 900–945.

Clark K (2010). Top 500: the race to recovery. *Home Channel News* **36**(6), 23–45.

Clarke B, Fokoué E and Zhang HH (2009). *Principles and Theory for Data Mining and Machine Learning*. Springer, Dordrecht.

Clausius R (1865). Über verschiedene für die Anwendung bequeme Formen der Hauptgleichungen der mechanischen Wärmetheorie. *Annalen der Physik* **201**(7), 353–400.

Clemmensen L, Hastie T, Witten D and Ersbøll B (2011). Sparse discriminant analysis. *Technometrics* **53**(4), 406–413.

Cleveland WS (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association* **74**(368), 829–836.

Cleveland WS (1993). *Visualizing Data*. Hobart Press, Summit, NJ.

Cleveland WS and Devlin SJ (1988). Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American Statistical Association* **83**(403), 596–610.

Cleveland WS, Grosse E and Shyu WM (1992). Local regression models. In *Statistical Models in S* (eds. Chambers JM and Hastie TJ), pp. 309–376. Wadsworth & Brooks/Cole, Pacific Grove, CA.

Clopper CJ and Pearson ES (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika* **26**(4), 404–413.

Cochran WG (1954). Some methods for strengthening the common χ^2 tests. *Biometrics* **10**(4), 417–451.

Coles S (2001). *An Introduction to Statistical Modeling of Extreme Values*. Springer-Verlag, London.

Cook RD (1977). Detection of influential observations in linear regression. *Technometrics* **19**(1), 15–18.

Cook RD and Weisberg S (1982). *Residuals and Influence in Regression*. Chapman & Hall, London.

Cordeiro GM and McCullagh P (1991). Bias reduction of maximum likelihood estimates. *Journal of the Royal Statistical Society, Series B (Methodological)* **53**(3), 629–643.

Cox DR (1958). The regression analysis of binary sequences (with discussion). *Journal of the Royal Statistical Society, Series B (Methodological)* **20**(2), 215–242.

Cox DR (1961). Tests of separate families of hypotheses. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability,* Vol. **I** (ed. Neyman J), pp. 105–123. University California Press, Berkeley, CA.

Cox DR (1962). Further results on tests of separate families of hypotheses. *Journal of the Royal Statistical Society, Series B (Methodological)* **24**(2), 406–424.

Cox DR (1988). Some aspects of conditional and asymptotic inference: A review. *Sankhyā, Series A* **50**(3), 314–337.

Cox DR (2013). A return to an old paper: 'Tests of separate families of hypotheses' (with discussion). *Journal of the Royal Statistical Society, Series B (Statistical Methodology)* **75**(2), 207–215.

Crivelli A, Firinguetti L, Montaño R and Muñóz M (1995). Confidence intervals in ridge regression by bootstrapping the dependent variable: a simulation study. *Communications in Statistics – Simulation and Computation* **24**(3), 631–652.

Cule E and De Iorio M (2013). Ridge regression in prediction problems: automatic choice of the ridge parameter. *Genetic Epidemiology* **37**(7), 704–714.

Cule E, Vineis P and De Iorio M (2011). Significance testing in ridge regression for genetic data. *BMC Bioinformatics* **12**(9), Article No. 372.

Dagliyan O, Uney-Yuksektepe F, Kavakli IH and Turkay M (2011). Optimization based tumor classification from microarray gene expression data. *PLoS ONE* **6**(2), Article No. e14579.

Dalgaard P (2008). *Introductory Statistics with R*, 2nd edn. Springer-Verlag, New York.

Davis CE and Steinberg SM (2006). Quantile estimation. In *Encyclopedia of Statistical Sciences*, Vol. **10** (eds. Kotz S, Read CB, Balakrishnan N and Vidakovic B), pp. 6704–6707. John Wiley & Sons, Inc, Hoboken, NJ.

Davison AC and Hinkley DV (1997). *Bootstrap Methods and Their Application*, *Cambridge Series in Statistical and Probabilistic Mathematics*, Vol. **1**. Cambridge University Press, Cambridge.

Decrouez G and Robinson AP (2012). Confidence intervals for the weighted sum of two independent binomial proportions. *Australian & New Zealand Journal of Statistics* **54**(3), 281–299.

Deutsch RC and Piegorsch WW (2012). Benchmark dose profiles for joint-action quantal data in quantitative risk assessment. *Biometrics* **68**(4), 1313–1322.

DiCiccio TJ and Romano JP (1988). A review of bootstrap confidence intervals (with discussion). *Journal of the Royal Statistical Society, Series B (Methodological)* **50**(3), 338–370 (corr. vol. **51**(3), 470).

Digby P and Kempton R (1987). *Multivariate Analysis of Ecological Communities*. Chapman & Hall, London.

Dillon WR, Kumar A and Mulani N (1987). Offending estimates in covariance structure analysis: comments on the causes of and solutions to Heywood cases. *Psychological Bulletin* **101**(1), 126–135.

Dixon SJ and Brereton RG (2009). Comparison of performance of five common classifiers represented as boundary methods: Euclidean Distance to Centroids, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Learning Vector Quantization and Support Vector Machines, as dependent on data structure. *Chemometrics and Intelligent Laboratory Systems* **95**(1), 1–17.

Driver HE and Kroeber AL (1932). Quantitative expression of cultural relationships. *University of California Publications in American Archaeology and Ethnology* **31**(4), 211–256.

Dudoit S and van der Laan MJ (2008). *Multiple Testing Procedures with Applications to Genomics*. Springer, New York.

Dziuban CD and Shirkey EC (1974). When is a correlation matrix appropriate for factor analysis? Some decision rules. *Psychological Bulletin* **81**(6), 358–361.

Dziuban CD, Shirkey EC and Peeples TO (1979). An investigation of some distributional characteristics of the measure of sampling adequacy. *Educational and Psychological Measurement* **39**(3), 543–549.

Eaton ML and Perlman MD (1973). The non-singularity of generalized sample covariance matrices. *Annals of Statistics* **1**(4), 710–717.

Efron B and Gong G (1983). A leisurely look at the bootstrap, the jackknife, and cross-validation. *American Statistician* **37**(1), 36–48.

Eklund G and Seeger P (1965). Massignifikansanalys. *Statistisk Tidskrift (Statistical Review), Series 3* **4**(5), 355–365.

Elder JF and Pregibon D (1996). A statistical perspective on knowledge discovery in databases. In *Advances in Knowledge Discovery and Data Mining* (eds. Fayyad UM, Piatetsky-Shapiro G, Smyth P and Uthurusamy R), pp. 83–113. American Association for Artificial Intelligence/MIT Press, Cambridge, MA.

Esposito Vinzi V and Russolillo G (2013). Partial least squares algorithms and methods. *Wiley Interdisciplinary Reviews: Computational Statistics* **5**(1), 1–19.

Everitt BS (2005). *An R and S-Plus® Companion to Multivariate Analysis*. Springer-Verlag, London.

Everitt BS, Landau S, Leese M and Stahl D (2011). *Cluster Analysis*, 5th edn. John Wiley & Sons, Ltd, Chichester.

Ezzikouri S, El Feydi AE, Chafik A, Afifi R, El Kihal L, Benazzouz M, Hassar M, Pineau P and Benjelloun S (2008). Genetic polymorphism in the manganese superoxide dismutase gene is associated with an increased risk for hepatocellular carcinoma in HCV-infected Moroccan patients. *Mutation Research* **649**(1-2), 1–6.

Faraway JJ (2006). *Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models*. Chapman & Hall/CRC, Boca Raton, FL.

Farcomeni A (2008). A review of modern multiple hypothesis testing, with particular attention to the false discovery proportion. *Statistical Methods in Medical Research* **17**(4), 347–388.

Fayers PM and Machin D (2007). *Quality of Life: The Assessment, Analysis and Interpretation of Patient-Reported Outcomes*, 2nd edn. John Wiley & Sons, Ltd, Chichester.

Feller W (1968). *An Introduction to Probability Theory and its Applications*, Volume **I**, 3rd edn. John Wiley & Sons, New York.

Fieller EC (1940). The biological standardization of insulin (with discussion). *Supplement to the Journal of the Royal Statistical Society* **7**(1), 1–64.

Firth D (1993). Bias reduction of maximum likelihood estimates. *Biometrika* **80**(1), 27–38 (corr. **82**(3), 667).

Fisher RA (1912). On an absolute criterion for fitting frequency curves. *Messenger of Mathematics* **41**, 155–160.

Fisher RA (1921). On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron* **1**(1), 3–32.

Fisher RA (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London, Series A: Containing Papers of a Mathematical or Physical Character* **222**(594–604), 309–368.

Fisher RA (1926). The arrangement of field experiments. *Journal of the Ministry of Agriculture of Great Britain* **33**(6), 503–515.

Fisher RA (1935). The logic of inductive inference (with discussion). *Journal of the Royal Statistical Society* **98**(1), 39–82.

Fisher RA (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics* **7**(2), 179–188.

Fix E and Hodges JL (1951). Discriminatory analysis, nonparametric discrimination: Consistency properties. Technical Report No. 4, Project 21-49-004, US Air Force School of Aviation Medicine, Randolph Field, TX.

Fong DYT (2001). Data management and quality assurance. *Drug Information Journal* **35**(3), 839–844.

Forbes C, Evans M, Hastings N and Peacock B (2010). *Statistical Distributions*, 4th edn. John Wiley & Sons, Inc., Hoboken, NJ.

Forgy EW (1965). Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *Biometrics* **21**(3), 768–769.

Frank A and Asuncion A (2010). UCI Machine Learning Repository http://archive.ics.uci.edu/ml, School of Information and Computer Sciences, University of California – Irvine.

Frank IE and Friedman JH (1993). A statistical view of some chemometrics regression tools (with discussion). *Technometrics* **35**(2), 109–148.

Freedman DA (1983). A note on screening regression equations. *American Statistician* **37**(2), 152–155.

Freund Y and Schapire RE (1996). Experiments with a new boosting algorithm. In *Machine Learning, Proceedings of the 13th International Conference (ICML '96), Bari, Italy, July 3-6, 1996* (ed. Saitta L), pp. 487–499. Morgan Kaufmann, San Francisco, CA.

Freund Y and Schapire RE (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* **55**(1), 119–139.

Friedman JH, Hastie T, Höfling H and Tibshirani R (2007). Pathwise coordinate optimization. *Annals of Applied Statistics* **1**(2), 302–332.

Friendly M (1991). *SAS System for Statistical Graphics*. SAS Institute, Inc., Cary, NC.

Friendly M (2013). The generalized ridge trace plot: visualizing bias and precision. *Journal of Computational and Graphical Statistics* **22**(1), 50–68.

Fu WJ (1998). Penalized regressions: the bridge versus the Lasso. *Journal of Computational and Graphical Statistics* **7**(3), 397–416.

Furnival GM (1971). All possible regressions with less computation. *Technometrics* **13**(2), 403–408.

Furnival GM and Wilson RW (1974). Regressions by leaps and bounds. *Technometrics* **16**(4), 499–511.

Galambos J and Simonelli I (1996). *Bonferroni-Type Inequalities with Applications*. Springer-Verlag, New York.

Gammon K (2009). Belle Curves. *Wired Magazine* **17**(17.02), Article No. 12.

Garwood F (1936). Fiducial limits for the Poisson distribution. *Biometrika* **28**(3/4), 437–442.

Gatu C and Kontoghiorghes EJ (2006). Branch-and-bound algorithms for computing the best-subset regression models. *Journal of Computational and Graphical Statistics* **15**(1), 139–156.

Gelman A (2004). Exploratory data analysis for complex models (with discussion). *Journal of Computational and Graphical Statistics* **13**(4), 755–787.

Gelman A and Unwin A (2013). Infovis and statistical graphics: different goals, different looks (with discussion). *Journal of Computational and Graphical Statistics* **22**(1), 2–49.

Gentle JE (2003). *Random Number Generation and Monte Carlo Methods*, 2nd edn. Springer-Verlag, New York.

Gentle JE (2007). *Matrix Algebra: Theory, Computations, and Applications in Statistics*. Springer, New York.

Gentleman R (2009). *R Programming for Bioinformatics*. Chapman & Hall/CRC Press, Boca Raton, FL.

Gijbels I and Prosdocimi I (2010). Loess. *Wiley Interdisciplinary Reviews: Computational Statistics* **2**(5), 590–599.

Givens GH and Hoeting JA (2013). *Computational Statistics*, 2nd edn. John Wiley & Sons, Inc., Hoboken, NJ.

Goeman JJ and Solari A (2011). Multiple testing for exploratory research (with discussion). *Statistical Science* **26**(4), 584–612.

Golub GH, Heath M and Wahba G (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics* **21**(2), 215–223.

Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD and Lander ES (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286**(5439), 531–537.

González I, Déjean S, Martin PGP and Baccini A (2008). CCA: an **R** package to extend canonical correlation analysis. *Journal of Statistical Software* **23**(12), 1–14.

Gopal V, Fuentes C and Casella G (2012). *bayesclust*: an **R** package for testing and searching for significant clusters. *Journal of Statistical Software* **47**(14), 1–21.

Halonen M, Stern DA, Wright AL, Taussig LM and Martinez FD (1997). *Altemaria* as a major allergen for asthma in children raised in a desert environment. *American Journal of Respiratory and Critical Care Medicine* **155**(4), 1356–1361.

Hambrick DZ, Oswald FL, Darowski ES, Rench TA and Brou R (2010). Predictors of multitasking performance in a synthetic work paradigm. *Applied Cognitive Psychology* **24**(8), 1149–1167.

Hampel D (2008). Estimation of differential entropy for positive random variables and its application in computational neuroscience. In *Mathematical Modeling of Biological Systems*, Volume **II** (eds. Deutsch A, Bravo de la Parra R, de Boer R, Diekmann O, Jagers P, Kisdi E, Kretzschmar M, Lansky P and Metz H), pp. 213–224. Birkhäuser, Basel.

Hand DJ (2009a). Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine Learning* **77**(1), 103–123.

Hand DJ (2009b). Naïve Bayes. In *The Top Ten Algorithms in Data Mining* (eds. Wu X and Kumar V), pp. 163–177. Chapman & Hall/CRC, Boca Raton, FL.

Hand DJ (2012). Assessing the performance of classification methods. *International Statistical Review* **80**(3), 400–414.

Hand DJ, Blunt G, Kelly MG and Adams MN (2000). Data mining for fun and profit. *Statistical Science* **15**(2), 111–131.

Hand DJ, Daly F, McConway K, Lunn AD, and Ostrowski, E (eds.) (1994). *A Handbook of Small Data Sets*. Chapman & Hall, New York.

Hand DJ, Mannila H and Smyth P (2001). *Principles of Data Mining*. MIT Press, Cambridge, MA.

Hand DJ and Yu K (2001). Idiot's Bayes – Not so stupid after all? *International Statistical Review* **69**(3), 385–398.

Hartigan JA and Wong MA (1979). A K-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **28**(1), 100–108.

Hartley HO (1938). Studentization and large-sample theory. *Supplement to the Journal of the Royal Statistical Society* **5**(1), 80–88.

Hastie T, Tibshirani R and Friedman J (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edn. Springer, New York.

Hauck WW and Donner A (1977). Wald's test as applied to hypotheses in logit analysis. *Journal of the American Statistical Association* **72**(360, part 1), 851–853 (corr. vol. **75**(370), 482).

Hayashi K, Bentler PM and Yuan KH (2007). On the likelihood ratio test for the number of factors in exploratory factor analysis. *Structural Equation Modeling: A Multidisciplinary Journal* **14**(3), 505–526.

Hazards & Vulnerability Research Institute (2013). The Spatial Hazard Events and Losses Database for the United States (SHELDUS), Version 10.0 http://www.sheldus.org, Hazards & Vulnerability Research Institute, University of South Carolina, Columbia, SC.

Henderson HV (2004). Interactive and dynamic graphics in statistical consulting. *Journal of Agricultural, Biological, and Environmental Statistics* **9**(4), 402–431.

Hendrickson AE and White PO (1964). Promax: a quick method for rotation to oblique simple structure. *British Journal of Statistical Psychology* **17**(1), 65–70.

Heywood HB (1931). On finite sequences of real numbers. *Proceedings of the Royal Society of London: Series A, Containing Papers of a Mathematical or Physical Character* **134**(824), 486–501.

Hirji KF (2005). *Exact Analysis of Discrete Data*. Chapman & Hall/CRC Press, Boca Raton, FL.

Hoaglin DC, Iglewicz B and Tukey JW (1986). Performance of some resistant rules for outlier labeling. *Journal of the American Statistical Association* **81**(396), 991–999.

Hochberg Y and Tamhane AC (1987). *Multiple Comparison Procedures*. John Wiley & Sons, Inc., New York.

Hoerl AE and Kennard RW (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **12**(1), 55–67.

Hoerl AE, Kennard RW and Baldwin KF (1975). Ridge regression: some simulations. *Communications in Statistics* **4**(2), 105–123.

Hogg RV and Tanis EA (2010). *Probability and Statistical Inference*, 8th edn. Pearson Prentice Hall, Upper Saddle River, NJ.

Höppner F (2010). Association rules. In *Data Mining and Knowledge Discovery Handbook* (ed. Maimon O and Rokach L), pp. 299–319. Springer, New York.

Horgan JM (2009). *Probability with R: An Introduction with Computer Science Applications*. John Wiley & Sons, Inc., Hoboken, NJ.

Hosmer D and Lemeshow S (2013). *Applied Logistic Regression*, 3rd edn. John Wiley & Sons, Inc., New York.

Hotelling H (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* **24**(6,7), 417–441, 498–520.

Hotelling H (1935). The most predictable criterion. *Journal of Educational Psychology* **26**(2), 139–142.

Hotelling H (1936). Relations between two sets of variates. *Biometrika* **28**(3 and 4), 321–377.

Hsu JC (1996). *Multiple Comparisons: Theory and Methods*. Chapman & Hall, Boca Raton, FL.

Huang B, Cook D and Wickham H (2012). tourGui: a gWidgets GUI for the tour to explore high-dimensional data using low-dimensional projections. *Journal of Statistical Software* **49**(6), 1–12.

Huang LS and Smith RL (1999). Meteorologically-dependent trends in urban ozone. *Environmetrics* **10**(1), 103–118.

Hubert LJ (2006). Hierarchical cluster analysis. In *Encyclopedia of Statistical Sciences*, Vol. **5** (eds. Kotz S, Read CB, Balakrishnan N and Vidakovic B), pp. 3142–3148. John Wiley & Sons, Inc., Hoboken, NJ.

Huberty CJ and Olejnik S (2006). *Applied MANOVA and Discriminant Analysis*, 2nd edn. John Wiley & Sons, Inc., Hoboken, NJ.

Hughes-Hallett D, Gleason AM, McCallum WG, Connally E, Flath DE, Kalayciôglu S, Lahme B, Frazer Lock P, Lomen DO, Lovelock D, Lozano GI, Morris J, Mumford D, Osgood BG, Patterson CL, Quinney D, Rhea K, Spiegler AH, Tecosky-Feldman J and Tucker TW (2013). *Calculus: Single and Multivariable*, 6th edn. John Wiley & Sons, Inc., Hoboken, NJ.

Hurvich CM and Tsai CL (1989). Regression and time series model selection in small samples. *Biometrika* **76**(2), 297–307.

Hwang YT, Chu SK and Ou ST (2011). Evaluations of FDR-controlling procedures in multiple hypothesis testing. *Statistics and Computing* **21**(4), 569–583.

Hyndman RJ, Koehler AB, Ord JK and Snyder RD (2008). *Forecasting with Exponential Smoothing: The State Space Approach*. Springer, Berlin.

Ihaka R and Gentleman R (1996). R: a language for data analysis and graphics. *Journal of Computational and Graphical Statistics* **5**(3), 299–314.

Irwin JO (1935). Tests of significance for differences between percentages based on small numbers. *Metron* **12**, 83–94.

Izenman AJ (2008). *Modern Multivariate Statistical Techniques. Regression, Classification, and Manifold Learning*. Springer, New York.

Jain AK and Dubes RC (1988). *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ.

James G, Witten D, Hastie T and Tibshirani R (2013). *An Introduction to Statistical Learning with Applications in R*. Springer, New York.

Jensen RE (1969). A dynamic programming algorithm for cluster analysis. *Operations Research* **17**(6), 1034–1057.

Johnson NL, Kemp AW and Kotz S (2005). *Univariate Discrete Distributions*, 3rd edn. John Wiley & Sons, Inc., New York.

Johnson NL, Kotz S and Balakrishnan N (1994). *Continuous Univariate Distributions*, Volume **1**, 2nd edn. John Wiley & Sons, Inc., New York.

Johnson NL, Kotz S and Balakrishnan N (1995). *Continuous Univariate Distributions,* Volume **2**, 2nd edn. John Wiley & Sons, Inc., New York.

Johnson NL, Kotz S and Balakrishnan N (1997). *Discrete Multivariate Distributions*. John Wiley & Sons, Inc., New York.

Johnson BA, Tateishia R and Hoana NT (2013). A hybrid pansharpening approach and multiscale object-based image analysis for mapping diseased pine and oak trees. *International Journal of Remote Sensing* **34**(20), 6969–6982.

Jolliffe IT (1972). Discarding variables in a principal component analysis. I. Artificial data. *Journal of the Royal Statistical Society, Series C (Applied Statistics)* **21**(2), 160–173.

Jolliffe IT (2002). *Principal Component Analysis*, 2nd edn. Springer-Verlag, New York.

Jones O, Maillardet R and Robinson A (2009). *Introduction to Scientific Programming and Simulation Using R*. Chapman & Hall/CRC Press, Boca Raton, FL.

Jones MC, Marron JS and Sheather SJ (1996). A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association* **91**(433), 401–407.

Jöreskog KG (1969). A general approach to confirmatory maximum likelihood factor analysis. *Psychometrika* **34**(2), 183–202.

Jørgensen B (2012). Generalized linear models. In *Encyclopedia of Environmetrics*, Volume **3**, 2nd edn. (eds. El-Shaarawi AH and Piegorsch WW), pp. 1152–1159. John Wiley & Sons, Ltd, Chichester.

Kadane JB and Lamberth J (2009). Are blacks egregious speeding violators at extraordinary rates in New Jersey? *Law, Probability & Risk* **8**(2), 139–152.

Kaiser HF (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika* **23**(3), 187–200.

Kaiser HF (1970). A second generation little jiffy. *Psychometrika* **35**(4), 401–415.

Kaiser HF and Rice J (1974). Little Jiffy, Mark IV. *Educational and Psychological Measurement* **34**(1), 111–117.

Kang SH and Ahn CW (2008). Tests for the homogeneity of two binomial proportions in extremely unbalanced 2×2 contingency tables. *Statistics in Medicine* **27**(14), 2524–2535.

Kantardzic M (2003). *Data Mining: Concepts, Models, Methods, and Algorithms*. IEEE Press, Piscataway, NJ.

Karatzoglou A, Meyer D and Hornik K (2006). Support vector machines in **R**. *Journal of Statistical Software* **15**(9), 1–28.

Kaufman L and Rousseeuw PJ (1990). *Finding Groups in Data. An Introduction to Cluster Analysis*. John Wiley & Sons, Inc., Hoboken, NJ.

Khuri AI (2003). *Advanced Calculus with Applications in Statistics*, 2nd edn. John Wiley & Sons, Inc., Hoboken, NJ.

Koepke H and Clarke B (2013). A Bayesian criterion for cluster stability. *Statistical Analysis and Data Mining* **6**(4), 346–374.

Kohavi R (1996). Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (eds. Simoudis E, Han J and Fayyad U), pp. 202–207. American Association for Artificial Intelligence Press, Menlo Park, CA.

Kolenikov S and Bollen KA (2012). Testing negative error variances: Is a Heywood case a symptom of misspecification? *Sociological Methods & Research* **41**(1), 124–167.

Kotz S, Balakrishnan N and Johnson NL (2000). *Continuous Multivariate Distributions*, *Models and Applications*, Volume **1**, 2nd edn. John Wiley & Sons, New York.

Kruppa J, Liu Y, Biau G, Kohler M, Knig IR, Malley JD and Ziegler A (2014). Probability estimation with machine learning methods for dichotomous and multicategory outcome: theory. *Biometrical Journal* **56**(4), 534–563.

Kuss O (2013). The danger of dichotomizing continuous variables: A visualization. *Teaching Statistics* **35**(2), 78–79.

Kutner MH, Nachtsheim CJ, Neter J and Li W (2005). *Applied Linear Statistical Models*, 5th edn. McGraw-Hill-Irwin, Boston, MA.

Kvam PH (2011). Comparing Hall of Fame baseball players using most valuable player ranks. *Journal of Quantitative Analysis in Sports* **7**(3), Article No. 19.

Kvam PH and Vidakovic B (2007). *Nonparametric Statistics with Applications to Science and Engineering*. John Wiley & Sons, Inc., Hoboken, NJ.

Lance GN and Williams WT (1967). A general theory of classificatory sorting strategies. 1. Hierarchical systems. *Computer Journal* **9**(4), 373–380.

Lange K (2013). *Optimization*, 2nd edn. Springer, New York.

Lawley DN (1959). Tests of significance in canonical analysis. *Biometrika* **46**(1 and 2), 59–66.

Lawley DN and Maxwell AE (1962). Factor analysis as a statistical method. *Journal of the Royal Statistical Society, Series D (The Statistician)* **12**(3), 209–229.

Leemis LM (1986). Relationships among common univariate distributions. *American Statistician* **40**(2), 143–146.

Lehmann EL and Casella G (1998). *Theory of Point Estimation*, 2nd edn. Springer-Verlag, New York.

Levy KJ and Narula SC (1974). Shortest confidence intervals for the ratio of two normal variances. *Canadian Journal of Statistics* **2**(1), 83–87.

Lewis M (2003). *Moneyball. The Art of Winning an Unfair Game*. W.W. Norton & Company, New York.

Ley E (1996). On the peculiar distribution of the U.S. stock indexes' digits. *American Statistician* **50**(4), 311–313.

Liao JG, Wu Y and Lin Y (2010). Improving Sheather and Jones' bandwidth selector for difficult densities in kernel density estimation. *Journal of Nonparametric Statistics* **22**(1-2), 105–114.

Likert R (1932). A technique for the measurement of attitudes. *Archives of Psychology* no. 140, 1–55.

Lin CP and Bhattacherjee A (2010). Extending technology usage models to interactive hedonic technologies: a theoretical model and empirical test. *Information Systems Journal* **20**(2), 163–181.

Lindsay B and Liu J (2009). Model assessment tools for a model false world. *Statistical Science* **24**(3), 303–318.

Linoff GS and Berry MJ (2011). *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management* 3rd edn. Wiley Publishing, Inc., Indianapolis, IN.

Liu W (2010). *Simultaneous Inference in Regression*, *Monographs on Statistics and Applied Probability*, Volume **118**. Chapman & Hall/CRC, Boca Raton, FL.

Lloyd SP (1957). Quantization for least mean squares error (abstract). *Annals of Mathematical Statistics* **28**(4), 1059.

Lloyd SP (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory* **IT-28**(2), 129–137.

Loh WY (2010). Tree-structured classifiers. *Wiley Interdisciplinary Reviews: Computational Statistics* **2**(3), 364–369.

Lohr SL (2010). *Sampling: Design and Analysis*, 2nd edn. Duxbury Press, Pacific Grove, CA.

Lukas MA (2012). Regularization methods. In *Encyclopedia of Environmetrics*, Volume **5**, 2nd edn. (eds. El-Shaarawi AH and Piegorsch WW), pp. 2181–2185. John Wiley & Sons, Ltd, Chichester.

Lydersen S, Fagerland MW and Laake P (2009). Recommended tests for association in 2×2 tables. *Statistics in Medicine* **28**(7), 1159–1175.

MacQueen J (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, Volume **I** (eds. Le Cam LM and Neyman J), pp. 281–297. University of California Press, Berkeley, CA.

Mallows CL (1973). Some comments on C_P. *Technometrics* **15**(4), 661–675.

Mammone A, Turchi M and Cristianini N (2009). Support vector machines. *Wiley Interdisciplinary Reviews: Computational Statistics* **1**(3), 283–289.

Mansouri K, Ringsted T, Ballabio D, Todeschini R and Consonni V (2013). Quantitative structure-activity relationship models for ready biodegradability of chemicals. *Journal of Chemical Information and Modeling* **53**(4), 867–878.

Margolin BH, Resnick MA, Rimpo JY, Archer P, Galloway SM, Bloom AD and Zeiger E (1986). Statistical analyses for in vitro cytogenetic assays using Chinese hamster ovary cells. *Environmental Mutagenesis* **8**(2), 183–204.

Martinez AR (2010). Natural language processing. *Wiley Interdisciplinary Reviews: Computational Statistics* **2**(3), 352–357.

McCullagh P and Nelder JA (1989). *Generalized Linear Models*, 2nd edn. Chapman & Hall, London.

McDonald L (2014). Florence Nightingale, statistics and the Crimean War. *Journal of the Royal Statistical Society, Series A (Statistics in Society)* **177**(3), 569–586.

McGill R, Tukey JW and Larsen WA (1978). Variations of box plots. *American Statistician* **32**(1), 12–16.

Michie D, Spiegelhalter D and Taylor, CC (ed.) (1994). *Machine Learning. Neural and Statistical Classification*. Ellis Horwood, Upper Saddle River, NJ.

Moguerza JM and Muñoz A (2006). Support vector machines with applications. *Statistical Science* **21**(3), 322–336.

Moore DS (2010). *The Basic Practice of Statistics*, 5th edn. W.H. Freeman & Co., New York.

Moro S, Laureano R and Cortez P (2011). Using data mining for bank direct marketing: an application of the CRISP-DM methodology. In *Proceedings of the European Simulation and Modelling Conference – ESM '2011* (eds. Novais P, Machado J, Analide C and Abelha A), pp. 117–121. EUROSIS, Guimaräes, Portugal.

Motoda H and Ohara K (2009). Apriori. In *The Top Ten Algorithms in Data Mining* (eds. Wu X and Kumar V), pp. 61–92. Chapman & Hall/CRC, Boca Raton, FL.

Murrell P (2011). *R Graphics*, 2nd edn. Chapman & Hall/CRC Press, Boca Raton, FL.

Myatt GJ (2007). *Making Sense of Data. A Practical Guide to Exploratory Data Analysis and Data Mining*. John Wiley & Sons, Inc., Hoboken, NJ.

Myatt GJ and Johnson WP (2009). *Making Sense of Data II: A Practical Guide to Data Visualization, Advanced Data Mining Methods, and Applications*. John Wiley & Sons, Inc., Hoboken, NJ.

Myers R, Montgomery D, Vining G and Robinson T (2012). *Generalized Linear Models with Applications in Engineering and the Sciences*. John Wiley & Sons, Inc., Hoboken, NJ.

Narula SC (1979). Orthogonal polynomial regression. *International Statistical Review* **47**(1), 31–36.

Nath B, Bhattacharyya DK and Ghosh A (2013). Incremental association rule mining: a survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **3**(3), 157–169.

Nelder JA and Wedderburn RWM (1972). Generalized linear models. *Journal of the Royal Statistical Society, Series A (General)* **135**, 370–384.

Neuhaus JO and Wrigley C (1954). The Quartimax method. An analytic approach to orthogonal simple structure. *British Journal of Statistical Psychology* **7**(2), 81–91.

Nguyen DV, Arpat AB, Wang N and Carroll RJ (2002). DNA microarray experiments: biological and technological aspects. *Biometrics* **58**(4), 701–717.

Nightingale F (1858). *Notes on Matters Affecting the Health, Efficiency, and Hospital Administration of the British Army*. Harrison & Sons, London.

Obenchain RL (1977). Classical *F*-tests and confidence regions for ridge regression. *Technometrics* **19**(4), 429–439.

O'Fallon BD, Wooderchak-Donahue W and Crockett DK (2013). A support vector machine for identification of single-nucleotide polymorphisms from next-generation sequencing data. *Bioinformatics* **29**(11), 1361–1366.

Palmeri C (1997). Believe in yourself, believe in the merchandise. *Forbes* **160**(5), 118–124.

Parkhomenko E, Tritchler D and Beyene J (2009). Sparse canonical correlation analysis with application to genomic data integration. *Statistical Applications in Genetics and Molecular Biology* **8**(1), Article No. 1.

Parzen E (1962). On estimation of a probability density function and mode. *Annals of Mathematical Statistics* **33**(3), 1065–1076.

Pearson K (1896). Contributions to the mathematical theory of evolution. – III. Regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London, Series A: Containing Papers of a Mathematical or Physical Character* **187**, 253–318.

Pearson K (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science, 5th series* **50**(302), 157–175.

Pearson K (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science, 6th series* **2**(11), 559–572.

Pepe MS (2003). *The Statistical Evaluation of Medical Tests for Classification and Prediction*. Oxford University Press, Oxford.

Phillips LD ((2005).). Bayesian belief networks. In *Encyclopedia of Statistics in Behavioral Science*, Volume **1** (eds. Everitt BS and Howell D), pp. 130–134. John Wiley & Sons, Ltd, Chichester.

Piegorsch WW and Bailer AJ (1997). *Statistics for Environmental Biology and Toxicology*. Chapman & Hall/CRC Press, Boca Raton, FL.

Piegorsch WW and Bailer AJ (2005). *Analyzing Environmental Data*. John Wiley & Sons, Ltd, Chichester.

Piegorsch WW, Cutter SL and Hardisty F (2007). Benchmark analysis for quantifying urban vulnerability to terrorist incidents. *Risk Analysis* **27**(6), 1411–1425.

Pierchala CE and Surti J (2009). Control charts as a tool for data quality control. *Journal of Official Statistics* **25**(2), 167–191.

Piette J and Jensen ST (2012). Estimating fielding ability in baseball players over time. *Journal of Quantitative Analysis in Sports* **8**(3), Article No. 1463.

Pimentel MA, Clifton DA, Clifton L and Tarassenko L (2014). A review of novelty detection. *Signal Processing* **99**, 215–249.

Playfair W (1786). *The Commercial and Political Atlas: Representing, by Means of Stained Copper-Plate Charts, the Exports, Imports, and General Trade of England, at a Single View. To Which are Added, Charts of the Revenue and Debts of Ireland, Done in the Same Manner by James Correy*, 1st edn. Debrett, Robinson, and Sewell, London.

Potscher BM (1991). Effects of model selection on inference. *Econometric Theory* **7**(2), 163–185.

Press SJ and Wilson S (1978). Choosing between logistic regression and discriminant analysis. *Journal of the American Statistical Association* **73**(364), 699–705.

Przyborowski J and Wilenski H (1935). Statistical principles of routine work in testing clover seed for dodder. *Biometrika* **27**(3/4), 273–292.

Quinlan JR (1996). Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research* **4**, 77–90.

Rao SS (1998). Birth of a legend. *Forbes* **161**(7), 128–130.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rizzo ML (2008). *Statistical Computing with R*. Chapman & Hall/CRC Press, Boca Raton, FL.

Rosenblatt M (1956). Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics* **27**(3), 832–837.

Rowley J (2007). The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of Information Science* **33**(2), 163–180.

Roy V and Kaiser MS (2013). Posterior propriety for Bayesian binomial regression models with a parametric family of link functions. *Statistical Methodology* **13**, 25–41.

Royston JP (1982). An extension of Shapiro and Wilk's *W* test for normality to large samples. *Journal of the Royal Statistical Society, Series C (Applied Statistics)* **31**(2), 115–124.

Salcedo-Sanz S, Rojo-Álvarez JL, Martínez-Ramón M and Camps-Valls G (2014). Support vector machines in engineering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **4**(3), 234–267.

Salzberg S (1988). Exemplar-based learning: theory and implementation. Technical Report TR-10-88, Center for Research in Computing Technology, Aiken Computation Laboratory, Harvard University, Cambridge, MA.

Satterthwaite FE (1946). An approximate distribution of estimates of variance components. *Biometrics* **2**(6), 110–114.

Schaefer RL (1986). Alternative estimators in logistic regression when the data are collinear. *Journal of Statistical Computation and Simulation* **25**(1-2), 75–91.

Schapire RE (1989). The strength of weak learnability. In *30th Annual Symposium on Foundations of Computer Science (FOCS), Oct. 30 – Nov. 1, 1989, Research Triangle Park, NC*, pp. 28–33. IEEE Computer Society Press, Los Alamitos, CA.

Schay G (2012). *A Concise Introduction to Linear Algebra*. Birkhäuser, Boston, MA.

Scheffé H (1953). A method for judging all contrasts in the analysis of variance. *Biometrika* **40**(1/2), 87–104.

Schoenberg FP (2012). Tessellations. In *Encyclopedia of Environmetrics*, Volume **6**, 2nd edn. (eds. El-Shaarawi AH and Piegorsch WW), pp. 2707–2709. John Wiley & Sons, Ltd, Chichester.

Schölkopf B and Smola AJ (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA.

Schwager SJ (1984). Bonferroni sometimes loses. *American Statistician* **38**(3), 192–197.

Schwarz G (1978). Estimating the dimension of a model. *Annals of Statistics* **6**(2), 461–464.

Scott DW (1979). On optimal and data-based histograms. *Biometrika* **66**(3), 605–610.

Scott DW (1992). *Multivariate Density Estimation. Theory, Practice, and Visualization*. John Wiley & Sons, Inc., New York.

Seeger P (1968). A note on a method for the analysis of significance en masse. *Technometrics* **10**(3), 586–593.

Seewald AK (2007). An evaluation of Naive Bayes variants in content-based learning for spam filtering. *Intelligent Data Analysis* **11**(5), 497–524.

Shannon CE (1948). A mathematical theory of communication. *Bell System Technical Journal* **27**(3,4), 379–423, 623–656.

Shapiro SS and Wilk MB (1965). An analysis of variance test for normality: complete samples. *Biometrika* **52**(3/4), 591–611.

Shi Y, Zhang JH, Jiang M, Zhu LH, Tan HQ and Lu B (2010). Synergistic genotoxicity caused by low concentration of titanium dioxide nanoparticles and p,p′-DDT in human hepatocytes. *Environmental and Molecular Mutagenesis* **51**(3), 192–204.

Silvapulle MJ (1981). On the existence of maximum likelihood estimators for the binomial response models. *Journal of the Royal Statistical Society, Series B (Methodological)* **43**(3), 310–313.

Silver N (2012). *The Signal and the Noise: Why So Many Predictions Fail – But Some Don't*. Penguin Press, New York.

Silverman BW (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London.

Simes RJ (1986). An improved Bonferroni procedure for multiple tests of significance. *Biometrika* **73**(3), 751–754.

Skillicorn D (2007). *Understanding Complex Datasets: Data Mining with Matrix Decompositions*. Chapman & Hall/CRC, Boca Raton, FL.

Small CG (1990). A survey of multidimensional medians. *International Statistical Review* **58**(3), 263–277.

Smith HF (1936). The problem of comparing the results of two experiments with unequal errors. *Journal of the Council of Scientific and Industrial Research* **9**, 211–212.

Smith JW, Everhart JE, Dickson WC, Knowler WC and Johannes RS (1988). Using the ADAP learning algorithm to forecast the onset of *diabetes mellitus*. In *Proceedings of the 12th Annual Symposium on Computer Applications and Medical Care (November 6–9, 1988, Washington D.C.)* (ed. Greenes RA), pp. 261–265. IEEE Computer Society Press, Los Alamitos, CA.

Spearman C (1904a). "General intelligence," objectively determined and measured. *American Journal of Psychology* **15**(2), 201–292.

Spearman C (1904b). The proof and measurement of association between two things. *American Journal of Psychology* **15**(1), 72–101.

Spiegelhalter DJ, Best NG, Carlin BP and van der Linde A (2002). Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society, Series B (Statistical Methodology)* **64**(4), 583–639.

Spouge JL (1994). Computation of the gamma, digamma, and trigamma functions. *SIAM Journal on Numerical Analysis* **31**(3), 931–944.

Stahl F and Jordanov I (2012). An overview of the use of neural networks for data mining tasks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2**(3), 193–208.

Stewart J and Kennelly PJ (2010). Illuminated choropleth maps. *Annals of the Association of American Geographers* **100**(3), 513–534.

Stigler SM (1983). Who discovered Bayes theorem? *American Statistician* **37**(4), 290–296.

Storey JD (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)* **64**(3), 479–498.

Storey JD (2003). The positive false discovery rate: a Bayesian interpretation and the *q*-value. *Annals of Statistics* **31**(6), 2013–2035.

Storey JD (2011). False discovery rate. In *International Encyclopedia of Statistical Science* (ed. Lovric M), pp. 504–508. Springer-Verlag, Berlin.

Street WN, Mangasarian OL and Wolberg WH (1995). An inductive learning approach to prognostic prediction. In *Machine Learning. Proceedings of the 12th International Conference on Machine Learning, Tahoe City, CA, July 9–12, 1995* (eds. Prieditis A and Russell SJ), pp. 522–530. Morgan Kaufmann, San Francisco, CA.

Stuart A and Ord JK (1994). *Kendall's Advanced Theory of Statistics, Volume 1. Distribution Theory*, 6th edn. Arnold, London.

Student (1908). The probable error of a mean. *Biometrika* **6**(1), 1–25.

Sturges HA (1926). The choice of a class interval. *Journal of the American Statistical Association* **21**(153), 65–66.

Sugar CA and James GM (2003). Finding the number of clusters in a dataset: an information-theoretic approach. *Journal of the American Statistical Association* **98**(463), 750–763.

Sundberg R (2012). Shrinkage regression. In *Encyclopedia of Environmetrics*, Volume **5**, 2nd edn. (eds. El-Shaarawi AH and Piegorsch WW), pp. 2450–2453. John Wiley & Sons, Ltd, Chichester.

Suzuki N, Rubin D, Lidman C, Aldering G, Amanullah R, Barbary K, Barrientos L, Botyanszki J, Brodwin M, Connolly N, Dawson K, Dey A, Doi M, Donahue M, Deustua S, Eisenhardt P, Ellingson E, Faccioli L, Fadeyev V, Fakhouri H, Fruchter A, Gilbank D, Gladders M, Goldhaber G, Gonzalez A, Goobar A, Gude A, Hattori T, Hoekstra H, Hsiao E, Huang X, Ihara Y, Jee M, Johnston D, Kashikawa N, Koester B, Konishi K, Kowalski M, Linder E, Lubin L, Melbourne J, Meyers J, Morokuma T, Munshi F, Mullis C, Oda T, Panagia N, Perlmutter S, Postman M, Pritchard T, Rhodes J, Ripoche P, Rosati P, Schlegel D, Spadafora A, Stanford S, Stanishev V, Stern D, Strovink M, Takanashi N, Tokita K, Wagner M, Wang L, Yasuda N, Yee H and The Supernova Cosmology Project (2012). The Hubble space telescope cluster supernova survey. V. Improving the dark-energy constraints above $z > 1$ and building an early-type-hosted supernova sample. *Astrophysical Journal* **746**(1), Article No. 85.

Takeuchi K (1976). Distribution of informational statistics and a criterion of model fitting (in Japanese). *Suri-Kagaku (Mathematical Sciences)* **153**, 12–18.

Tarone RE (1986). Correcting tests for trend in proportions for skewness. *Communications in Statistics – Theory and Methods* **15**(2), 317–328.

Tarone RE and Gart JJ (1980). On the robustness of combined tests for trends in proportions. *Journal of the American Statistical Association* **75**(369), 110–116.

Tate RF and Klett GW (1959). Optimal confidence intervals for the variance of a normal distribution. *Journal of the American Statistical Association* **54**(287), 674–682.

Taussig LM, Wright AL, Holberg CJ, Halonen M, Morgan WJ and Martinez FD (2003). Tucson Children's Respiratory Study: 1980 to present. *Journal of Allergy and Clinical Immunology* **111**(4), 661–675.

Thelwall M, Haustein S, Larivière V and Sugimoto CR (2013). Do altmetrics work? Twitter and ten other social web services. *PLoS ONE* **8**(5), Article No. e64841.

Thompson SK (2012). *Sampling*, 3rd edn. John Wiley & Sons, Inc., Hoboken, NJ.

Thurstone LL (1931). Multiple factor analysis. *Psychological Review* **38**(5), 406–427.

Thurstone LL (1947). *Multiple-Factor Analysis: A Development and Expansion of "The Vectors of the Mind"*. University of Chicago Press, Chicago.

Tibshirani R (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B (Methodological)* **58**(1), 267–288.

Tikhonov AN (1963). Solution of incorrectly formulated problems and regularization method. *Soviet Mathematics–Doklady* **4**, 1035–1038.

Timm NH (2002). *Applied Multivariate Analysis*. Springer-Verlag, New York.

Toraason M, Lynch DW, DeBord DG, Singh N, Krieg E, Butler MA, Toennis CA and Nemhauser JB (2006). DNA damage in leukocytes of workers occupationally exposed to 1-bromopropane. *Mutation Research* **603**(1), 1–14.

Tryon RC (1939). *Cluster Analysis*. Edwards Bros., Ann Arbor, MI.

Tsanas A and Xifara A (2012). Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings* **49**, 560–567.

Tufféry S (2011). *Data Mining and Statistics for Decision Making*. John Wiley & Sons, Ltd, Chichester.

Tufte E (2001). *The Visual Display of Quantitative Information*, 2nd edn. Graphics Press, Cheshire, CT.

Tukey JW (1972). Some graphic and semigraphic displays. In *Statistical Papers in Honor of George W. Snedecor* (ed. Bancroft TA), pp. 293–316. Iowa State University Press, Ames, IA.

Tukey JW (1977). *Exploratory Data Analysis*. Addison-Wesley, Reading, MA.

Uhlmann ME, Georgieva M, Sill M, Linnemann U and Berger MR (2012). Prognostic value of tumor progression-related gene expression in colorectal cancer patients. *Journal of Cancer Research and Clinical Oncology* **138**(10), 1631–1640.

Upadhyay R (2014). A Case Study from Banking (Part 3) – Logistic Regression. http://www.bigdatanews.com/profiles/blogs/a-case-study-from-banking-part-3-logistic-regression-1. Accessed 20 March 2014.

Væth M (1985). On the use of Wald's test in exponential families. *International Statistical Review* **53**(2), 199–214.

Vapnik VN (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

Vapnik VN (1998). *Statistical Learning Theory*. John Wiley & Sons, Inc., New York.

Vapnik VN (2000). *The Nature of Statistical Learning Theory*, 2nd edn. Springer-Verlag, New York.

Vasicek O (1976). A test for normality based on sample entropy. *Journal of the Royal Statistical Society, Series B (Methodological)* **38**(1), 54–59.

Venables WN and Ripley BD (2000). *S Programming*. Springer-Verlag, New York.

Venables WN and Ripley BD (2002). *Modern Applied Statistics with S-Plus*, 4th edn. Springer-Verlag, New York.

Verzani J (2005). *Using R for Introductory Statistics*. Chapman & Hall/CRC, Boca Raton, FL.

Vidaurre D, Bielza C and Larrañaga P (2013). A survey of L_1 regression. *International Statistical Review* **81**(3), 361–387.

de Ville B (2013). Decision trees. *Wiley Interdisciplinary Reviews: Computational Statistics* **5**(6), 448–455.

Vise A (2012). CCJ Top 250: getting stronger. For the largest trucking companies, revenue growth outpaces capacity growth. *Commercial Carrier Journal* **169**(8), 60–92.

Wald A (1943). Tests of statistical hypotheses concerning several parameters when the number of observations is large. *Transactions of the American Mathematical Society* **54**(3), 426–482.

Wehrens R and Buydens LMC (2007). Self- and super-organizing maps in R: The kohonen package. *Journal of Statistical Software* **21**(5), 1–19.

Wei LJ and Cowan CD (2006). Selection bias. In *Encyclopedia of Statistical Sciences*, Volume **11** (eds. Kotz S, Read CB, Balakrishnan N and Vidakovic B), pp. 7524–7526. John Wiley & Sons, Inc., Hoboken, NJ.

Welch BL (1938). The significance of the difference between two means when the population variances are unequal. *Biometrika* **29**(3/4), 350–362.

Whittle P (1958). On the smoothing of probability density functions. *Journal of the Royal Statistical Society, Series B (Methodological)* **20**(2), 334–343.

Wickham H (2012). Statistical graphics. In *Encyclopedia of Environmetrics*, Volume **6**, 2nd edn. (eds. El-Shaarawi AH and Piegorsch WW), pp. 2649–2658. John Wiley & Sons, Ltd, Chichester.

Wickham H (2013). Graphical criticism: Some historical notes. *Journal of Computational and Graphical Statistics* **22**(1), 38–44.

Wilkinson L (1999). Dot plots. *American Statistician* **53**(3), 276–281.

Wilkinson L (2005). *The Grammar of Graphics*, 2nd edn. Springer, New York.

Wilks SS (1938). The large-sample distribution of the likelihood ratio for testing composite hypotheses. *Annals of Mathematical Statistics* **9**(1), 60–62.

Wilson EB (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association* **22**(158), 209–212.

Wilson J (2012). Statistical computing with R: selecting the right tool for the job – R Commander or something else? *Wiley Interdisciplinary Reviews: Computational Statistics* **4**(6), 518–526.

Wilson PD and Tonascia J (1971). Tables for shortest confidence intervals on the standard deviation and variance ratio from normal distributions. *Journal of the American Statistical Association* **66**(336), 909–912.

de Winter JCF and Dodou D (2012). Factor recovery by principal axis factoring and maximum likelihood factor analysis as a function of factor pattern and sample size. *Journal of Applied Statistics* **39**(4), 695–710.

Witten DM and Tibshirani RJ (2009). Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical Applications in Genetics and Molecular Biology* **8**(1), Article No. 28.

Witten DM, Tibshirani RJ and Hastie T (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics* **10**(3), 515–534.

Wolberg WH and Mangasarian OL (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences of the United States of America* **87**(23), 9193–9196.

Wolf JR (1856). Mittheilungen über die Sonnenflecken? *Vierteljahrsschrift der Naturforschen Gesellschaft in Zurich* **1**, 151–161.

Wood SN (2006). *Generalized Additive Models: An Introduction with R*. Chapman & Hall/CRC Press, Boca Raton, FL.

Working H and Hotelling H (1929). Applications of the theory of error to the interpretation of trends. *Journal of the American Statistical Association, Supplement: Proceedings of the American Statistical Association* **24**, 73–85.

Wu TT and Lange K (2008). Coordinate descent algorithms for lasso penalized regression. *Annals of Applied Statistics* **2**(1), 224–244.

Yates F (1948). The analysis of contingency tables based on quantitative characters. *Biometrika* **35**(1/2), 176–181.

Yau N (2011). *Visualize This: The FlowingData Guide to Design, Visualization, and Statistics*. Wiley Publishing, Inc., Indianapolis, IN.

Yeh IC (1998). Modeling of strength of high performance concrete using artificial neural networks. *Cement and Concrete Research* **28**(12), 1797–1808.

Young FW, Valero-Mora PM and Friendly M (2006). *Visual Statistics: Seeing Data with Dynamic Interactive Graphics*. John Wiley & Sons, Inc., Hoboken, NJ.

Zabell SL (2008). On Student's 1908 article "The probable error of a mean" (with discussion). *Journal of the American Statistical Association* **103**(481), 1–20.

Zhang HH, Ahn J, Lin X and Park C (2006). Gene selection using support vector machines with non-convex penalty. *Bioinformatics* **22**(1), 88–95.

Zhang S and Wu X (2011). Fundamentals of association rules in data mining and knowledge discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **1**(2), 97–116.

Zou H and Hastie T (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)* **67**(2), 301–320 (corr. **67**(5), 768).

Index

+ subscript, 26
68–95–99.7% rule, 57

A-B testing, 159, 287
accuracy
    in association rule learning, 397
    in classification analytics, 296
AdaBoost algorithm, 321
addition rule, *see under* probability, rules
adjusted R^2, *see under* coefficient of determination
agglomerative hierarchical clustering, *see under* cluster analysis
Agresti-Coull confidence interval, 133
Akaike Information Criterion (AIC), *see under* information criteria
alias, *see under* predictor variable
alternative hypothesis, 138
    one-sided, 139
    two-sided, 139
altmetrics, 157
analysis of deviance table, 264
analysis of variance, 242
    multivariate, xv
    one-factor, 242–243
    two-factor, 244
ANOVA, *see* analysis of variance
Apriori algorithm, 399
artificial neural networks, xv
association rules
    accuracy, 397
        minimum threshold, 399
    antecedent, 396
    χ^2 statistic, 402–403
    consequent, 396
    itemsets, 396
        frequent, 399
        sampling from, 402
    lift, 398
    support, 397
    transaction database, 396
asymptotic distribution, 23
    of estimator, 118
    of sample mean, 35, 118
attribute section, *see* variable selection

backward elimination, *see under* variable selection
bagging, 321, 322
balanced design, 242
bar chart, 83, 91–94
    with error bars, 92–94
Bayes classification rule, 303
Bayes' Rule, 12
    for p.d.f.s, 16
Bayes, T., 12
Bayesian belief network, xv
Bayesian Information Criterion (BIC), *see under* information criteria
Bernoulli trial, 24
best linear unbiased estimator, 166, 199
best subset selection, *see under* variable selection

Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery, First Edition. Walter W. Piegorsch.

www.wiley.com/go/piegorsch/data_analytics

bias
  of estimator, 118
  sampling, 50
bias-variance trade-off
  in classification, 308, 314, 326
  in ridge regression, 232, 233
Big Data, 3, 65, 386, 430
binning, 66–67
  for histogram, 83
  normal reference rule, 66
  Sturges' rule, 66
binomial coefficient, 23
binomial distribution, 23–26
  closure under addition, 26, 119
  entropy of, 47
  in exponential family, 43
  p.m.f., 23
biodiversity, 288
bioinformatics, 5, 333, 405
bivariate normal distribution, *see* normal (Gaussian) distribution, bivariate
BLUE, *see* best linear unbiased estimator
body mass index, 342
Boltzman-Shannon entropy, *see* entropy
Bonferroni correction, *see under* multiplicity adjustment
Bonferroni's inequality, 149
boosting, 321
bootstrap interval, *see* confidence interval, bootstrap
bootstrap resampling, 137
box-and-whisker plot, 80
boxplot, 57, 79–81
  multiple, 95–96
  notched, 79
  whiskers in, 79
branch-and-bound algorithm, 216
bridge regression, 241
bubble plot, 99–101, 112
bull's-eye cluster, 326, 394

c.d.f., *see* cumulative distribution function
canonical correlation analysis, 361–363
  adjusted canonical correlations, 363
  Bartlett statistic, 362
    bias correction, 363
  canonical variates, 361
    eigendecomposition, 362
    regularized, 365
canonical correlation coefficient, 362
CART, *see* classification and regression trees
case deletion, 178, 183, 214, 233, 239
Cayley multiplication, *see* matrix multiplication
Cayley, A., 413
CCA, *see* canonical correlation analysis
Central Limit Theorem, 35–36
central tendency, 53
centroid, 378
characteristic polynomial, *see under* matrix
characteristic value, *see* eigenvalue
χ^2 distribution, 32
  closure under addition, 32
  dervied from normal, 37
  upper-α critical point, 37, 127
χ^2 statistic, 263, 275
  for 2×2 table, 277, 402
choropleth map, 104
classification analytics, 292–328
  accuracy, 296
  Bayesian, 302–303
    naive, 308
  confusion matrix, 296, 336
  decision trees, 312–315
  k-nearest neighbor, 308–309
    tuning parameter for, 308
  logistic regression in, 292
  misclassification error, 296, 325
  nonparametric, 308, 312
  nonseparable data, 325
    slack variable, 325
    soft margin hyperplane, 325
  sensitivity, 296
  separable data, 322
    classification rule, 323
    margin between, 322
  specificity, 296
  support vector machines, 328
  support vector methods, 322
  tree-based methods, 312, 320
classification and regression trees, 321
Clausius, R., 18

CLT, *see* Central Limit Theorem
cluster analysis, 373–394
  agglomerative hierarchical, 376–379
    average-link, 377, 378
    centroid link, 378
    chaining problem, 377
    complete-link, 377, 378
    dendrogram, 379
    single-link, 377, 378
  between-cluster variation, 384, 390
  dissimilarity matrix, 375
  distance metric, 376, 384
  divisive hierarchical, 377, 384
  exploratory, 389–390
  model-based, 394
  partitioned, 384–385
    elbow plot, 389
    K-means, 385–386
    K-medians, 394
    K-medoids, 394
    pseudo-F plot, 390
  pseudo-F statistic, 390
  similarity matrix, 375
  within-cluster sum of squares, 385, 386, 391
Cochran-Armitage trend test
  for counts, 288
  for proportions, 271–272
  generalized, 289
  Tarone skewness adjustment, 272
coefficient of determination (R^2), 170
  adjusted, 214
  multiple, 200
coefficient of variation, 29, 281
collinearity, *see* multicollinearity
confidence band, 173–175
confidence bound
  one-sided, 124
confidence coefficient, *see* confidence level
confidence ellipse
  for regression coefficients, 171–172, 201
  Wald form, 149
confidence interval, 123
  bootstrap, 138
    percentile method, 138
  for binomial probability, 133–134, 148
    Agresti-Coull, 133
    Wilson type, 134
    with continuity correction, 133
  for classification probability, 293
  for correlation coefficient, 189
  for difference in normal means
    with equal variances, 130
    with paired samples, 130
    with unequal variances, 128–129
  for mean response
    joint, 202
    multiple linear, 202
    simple linear, 170
  for normal mean, 124–127
  for normal mean and variance (joint), 150–151
  for normal variance, 127
    optimal, 127, 155
    ratio of, 131
  for Poisson mean, 135
  for ratio, 136–137, 158
  for regression coefficient
    joint, 201
    multiple linear, 201
    simple linear, 168, 170
  frequentist interpretation, 124
  from Delta method, 136
  from pivotal quantity, 127
  likelihood ratio, 135
    profile, 265
  tautology, 139
  Wald, 131–132
    in generalized linear model, 264
confidence level, 123
  simultaneous, 149
confidence rectangle
  for regression coefficients, 171
confidence surface, 202
confidence vs. probability, 124
confusion matrix, *see under* classification analytics
consecutive-dose average spacing, 267
contingency table, 274
  2×2, 277

contingency table (*continued*)
for association rules, 402
χ^2 statistic, 275
implementaion rules, 276
sparse, 276
standardized residuals
from, 277
continuous data, 52
contour plot, 229
control group, 50
convenience sampling, *see under* sampling
convergence
in distribution, 23
in probability, 23
of sample mean, *see under* asymptotic distribution
of transformed variable, *see* Delta method
convex optimization, 324
coordinate descent algorithm, 239
correlation coefficient
as measure of association, 21, 59, 187
population, 20–21, 188
sample, 59, 188
computing formulae, 59
correlation matrix, 104, 420
for standardized variates, 420
covariance
population, 20
of independent variables, 45
sample, 59
covariance matrix, 21–22
for bivariate normal, 36
of estimator, 117
of matrix transformation, 199
sample, 419
coxcomb plot, *see* polar area plot
C_p statistic, 214
critical point
χ^2, 37, 127
F, 40
standard normal, 33, 124
t, 39, 125
critical region, *see* rejection region
cross-validation, 233, 329
generalized, 233
K-fold, 239, 240
L-fold, 315
leave-one-out, 239, 311
cumulative distribution function, 13
joint, 16
curse of dimensionality, 309, 329, 394
CV, *see* cross-validation
cv, *see* coefficient of variatione

data analysis, 52
exploratory, 7
inferential, 115
data analytics, xiii, 4
data mining, 5
data quality, 5
disortion of, 6–7
data transformation, *see* transformation
data, types of
continuous, 52
discrete, 52
interval, 52
nominal, 51
ordinal, 51
qualitative, 51
quantitative, 51
ratio, 52
decile, 19
decision boundary, 296, 303, 304, 322, 332
decision trees, 312–315
branches and leaves, 312
cost-complexity pruning, 314
entropy impurity, 313, 337
Gini impurity, 312, 313, 337
ID3/C4.5/C5.0 algorithms, 314
misclassification impurity, 313, 314, 337
one-standard error rule, 315
pruning, 314–315
weakest-link, 315
splitting criteria, 312
tuning parameter for, 314
degrees of freedom, 32, 37, 38
effective, 233
error, 167, 200
F ratio, 40
Delta method, 36
multivariate, 136
dendrogram, 312, 379

as heatmap add-on, 104
density estimation, *see* kernel density estimator
density function, *see* probability density function
design matrix, 199
deviance function, 261
as impurity measure, 314
as residual sum of squares, 261
χ^2 approximation, 262
diagnostics, *see* regression diagnostics
diagonal matrix, *see under* matrix
DIKW pyramid, 4
dimension reduction, 342
via canonical correlation analysis, 361
via exploratory factor analysis, 351
via principal components, 345–346
Dirichlet tessellation, *see* Voronoi tessellation
discrepancy measure
for hypothesis testing, 202–203
via deviance function, 261–262
discrete data, 52
discriminant analysis, 292
Bayesian, 302–303
Gaussian, 303–304
linear, 297–299
logistic, 292
quadratic, 307
scores in, 292
sparse, 302
discriminant function
linear, 293, 299
discrimination, *see* discriminant analysis
disjoint events, *see under* events
dispersion
as entropy, 18, 62
parameter, 41, 259
dissimilarity measure, 374
distance metric, 374–375
Canberra, 309, 376
Euclidean, 309, 376
Hamming, 309, 376
Mahalanobis, 386
Manhattan, 309, 376
maximum, 309, 376
Tchebychev, 309, 376
distribution, 13
Bernoulli, 24
binomial, 23–26
χ^2, 32
exponential, 29–30
exponential family, 41–43
F, 40–41
gamma, 30–32
geometric, 27
hypergeometric, 278
multinomial, 278, 314
negative binomial, 28
normal (Gaussian), 32–35
bivariate, 36
multivariate, 37, 303, 336
Pareto, 48
Poisson, 26–27
t, 38–40
uniform
continuous, 29
discrete, 28
univariate, 23
dot plot
Cleveland-type, 78
Wilkinson-type, 77–79
dot product, 413

ecoinformatics, 5
EDA, *see* exploratory data analysis
EFA, *see* exploratory factor analysis
eigenanalysis, 415–416
eigendecomposition, *see under* matrix
eigenvalue, 416
and singular matrix, 416
eigenvector, 416
unit, 416
elastic net regularization, *see under* regularization
ensemble learning, 321
entropy, 18
in decision trees, 313
sample, 62–63
error bars, *see under* bar chart
error inflation, 149
error sum of squares, *see* residual sum of squares
estimability constraint

estimability constraint (*continued*)
corner-point
one-factor, 243
two-factor, 244, 275
zero-sum
one-factor, 243
two-factor, 244, 275
estimating equation, 119, 260
estimator
asymptotically unbiased, 118
best linear unbiased, 166, 199
biased, 118
interval, 123
least squares, 119
maximum likelihood, 120
moment (method of), 119
nonparametric, 63
point, 117
unbiased, 118
Euclidean distance, *see under* distance metric
events, 10
complementary, 11
disjoint, 11
independent, 12
mutually exclusive, 11
EWMA, *see* smoothing, exponential weighted average
exact test
for 2×2 table, 278–279
for binomial probability, 147
for $R \times C$ table, 279
expectation operator, *see* expected value
expected value, 17
as linear operator, 18
as weighted average, 17
bivariate, 20
conditional, 20
marginal, 20
of linear combination, 22
population mean, 17
experimentwise error, *see* false positive error, familywise
exploratory data analysis, 151, 292
exploratory factor analysis, 351–358
Bartlett test for sphericity, 357
communality, 352
eigendecomposition, 353
factor scores, 360
factor selection
Kaiser criterion, 355
likelihood ratio test, 354–355
percent explainable variance, 355
Kaiser-Meyer-Olkin measure, 357
latent factor in, 351
loading coefficient, 351, 352
manifest variable, 351
maximum likelihood estimation for, 354
principal factor estimation, 353–354
rotation
oblique, 356
promax, 356
quartimax, 356
varimax, 356
specific variances, 352
uniquenesses, 352
exponential distribution, 29–30
c.d.f., 29
coefficient of variation, 29
entropy of, 47
in exponential family, 48
memoryless property, 30
p.d.f., 29
survival function, 30
exponential family, 41–43, 258
dispersion parameter, 41, 259
mean relationship, 42
natural parameter, 41, 259
binomial, 43, 266
normal (Gaussian), 42
Poisson, 273
probability function, 41, 259
extended, 42
scale parameter, 41, 259
variance function, 42, 259
extrapolation, 175
extreme values, 79

F distribution, 40–41
dervied from normal, 40
relation to χ^2, 41
relation to t, 41
upper-α critical point, 40

F ratio, 40
F-test
  for ANOVA main effect, 243
  for regression coefficient, 203
    polynomial, 211
  in analysis of deviance, 265
Facebook, 157, 286
factor analysis
  confirmatory, 361
  exploratory, 351
factorial operator, 23
false discovery rate, 152–153
  marginal, 154
  positive, 154
false negative error, 139, 296
false positive error, 139, 296
  familywise, 149
    vs. false discovery rate, 152
  pointwise, 149
FDR, *see* false discovery rate
feature assessment, 152
feature extraction, 346, 352
feature selection, 211, 302
feature variable, 164, 291
  centered, 341
  scaled, 309
  standardized, 344
fences (inner), 61
  in boxplot, 79
fences (outer), 61
FET, *see* Fisher Exact Test
Fieller's theorem, 137
Fisher Exact Test, 279
  with unbalanced design, 279
Fisher information matrix, 116–117, 123, 149
  expected, 117
Fisher information number, 116
  for binomial sample, 132, 154
Fisher *Z* transform
  for correlation coefficient, 189
Fisher, R.A., 116
Fisher-Irwin test, *see* Fisher exact test
fitted values
  in multiple linear regression, 199–200
  in simple linear regression, 164
five-number summary, 57, 79, 95
forecasting, 70
forward stepwise regression, *see under* variable selection
fraud detection, 6
frequency polygon, 95
FWER, *see* false positive error, familywise

gamma distribution, 30–32
  closure under addition, 31
  in exponential family, 281, 289
  p.d.f., 30, 281
    alternative form, 281, 289
gamma function, 28
  recursive relationship, 28
Gauss–Markov theorem, 166, 199
Gaussian distribution, *see* normal distribution
Gaussian radial basis function, *see under* kernel function
GCV, *see* cross-validation, generalized
gene expression data, 153, 291, 333, 371, 380, 405
generalized additive model, xv
generalized linear model, 258
  analysis of deviance, 264–265
    with unknown dispersion, 265
  deviance function, 261
    scaled, 261
  dispersion parameter
    χ^2-based estimator, 263, 281
    deviance-based estimator, 262
  for simple linear regression, 283
  gamma, 281
  inverse link, 259
  link function, 259
    complementary log-log, 266
    diagnostic rule-of-thumb, 266
    identity, 259
    logarithmic, 259, 274, 281
    logit, 266
    probit, 266
  log-linear, 274, 280, 281
  logistic, 266
  residual
    deviance, 263
    standardized, 263
  variance function, 259, 263

genome-wide association studies, 152
geoinformatics, 5, 395
geometric distribution, 27
  c.d.f., 27
  memoryless property, 46
  p.m.f., 27
geometric series (finite), 27, 70
GIGO principle, 5
GLiM, *see* generalized linear model
Gosset, W.S., 38
greedy search algorithm, 312, 386

H_o, *see* null hypothesis
H_a, *see* alternative hypothesis
hat matrix, 200
  diagonal elements, 181, 200
  in generalized linear model, 263
  in ridge regression, 232
  trace of, 200
heatmap, 103–104
  for correlation matrix, 104, 364
  temporal, 113
Hessian matrix, 116
Heywood cases, 354
high-dimensional data, 270, 302, 341, 386
hinges, 57
histogram, 83–85
  back-to-back, 87
  bin width, 83
  class intervals, 83
  normal reference rule for, 83
  Sturges' rule for, 83
Hotelling transformation, *see under* principal component analysis
hyperbolic tangent, 190
hypergeometric distribution, 278
hyperplane, 304
hypothesis test, 138–140
  for ANOVA interaction, 244
  for ANOVA main effect, 243, 244
    sequential, 245
  for binomial probability, 147–148
  for canonical correlations, 362
  for correlation coefficient, 188
    rank-based, 190–191
  for difference in normal means
    with equal variances, 144
    with paired data, 144–145
    with unequal variances, 142–143
  for normal mean, 140–141
  for normal variance, 142
  for ratio of normal variances, 145
  for regression coefficient
    joint, 202–203
    multiple linear, 202
    partial, 202
    polynomial, 211
    simple linear, 168
  likelihood ratio, 146–147
    in generalized linear model, 264–265
    instablity with contingency tables, 275
    via deviance difference, 262
  one-sided, 139
  P-value for, 139
  power of, 139
  rejection region, 139
  sensitivity, 139
  significance level, 139
  tautology, 139
  two-sided, 139
  Wald, 145–146
    in generalized linear model, 264
    instability with binomial data, 273
  with contingency tables, 275–276

i.i.d., 35, 49
identity matrix, *see under* matrix
Idiot's Bayes classification, *see* naïve Bayes classification
imputation, *see under* missing data
indicator function, 23
inference, statistical, 7, 115
infographics, 76
informatics, 5
  medical, 5
information age, 3
information criteria, 215–216
  Akaike (AIC), 215
    corrected, 215
  Bayesian (BIC), 215

information matrix, *see* Fisher information matrix
infovis, 76, 110
inner product, Euclidean, 413
input variable, *see* predictor variable or feature variable
interquartile range
  population, 19
  sample, 57
inverse hyperbolic tangent, 189
inverse matrix, *see under* matrix
IQR, *see* interquartile range
itemsets, *see under* association rules
IWLS, *see* least squares, weighted, iterative

K-means clustering, *see under* cluster analysis, partitioned
k-nearest neighbor classifier, *see under* classification analytics
Kaiser criterion, *see under* principal component analysis
Karhunen-Loéve transformation, *see under* principal component analysis
KDD, *see under* knowledge discovery
kernel density estimator, 85–86
  bandwidth, 85
    as standard deviation, 86
    optimal, 86
  kernel function, 85
kernel function
  bisquare, 225
  Cauchy, 329
  dimension-transforming, 328
  Epanechnikov, 85
  Gaussian, 85, 86, 111, 328
  Laplace, 329
  linear, 328
  second-order, 328
  sigmoid, 329
  thin-plate spline, 329
  triangular, 85
  tricube, 224
  tuning parameter for, 329
kernel transformation, *see under* support vector methods
knowledge discovery, xiii, 3
  in databases (KDD), 3

Lagrange multiplier, 231, 299, 344
Laplace radial basis function, *see under* kernel function
Lasso, 238–239
  as variable selector, 238
  coefficient profile plot, 239–240
  cross-validation plot, 240
  tuning parameter for, 238
    cross-validation, 238
    selection of, 238
latent factor, *see under* exploratory factor analysis
latent value, *see* eigenvalue
leaps and bounds algorithm, 216
learning with a teacher, *see* supervised learning
learning without a teacher, *see* unsupervised learning
least squares
  method of, 119–120
  penalized, 231
  weighted, 120
    for linear regression, 184–185, 200–201
    for loess smoothing, 224
    iterative, 260
least squares estimator
  in multiple linear regression, 199
    covariance matrix for, 200
    sampling distribution, 199
    standard error, 200
    unbiased, 199
  in simple linear regression, 164, 191
    sampling distribution, 166, 192
    standard error, 167
    unbiased, 166, 192
  weighted
    in multiple linear regression, 201
    in simple linear regression, 185
leverage, *see under* regression diagnostics
lift, *see under* association rules
likelihood function, 116

likelihood ratio, 135
asymptotic distribution, 135
likelihood ratio test, *see under* hypothesis test
Likert scale, 51
line graph, 94
linear combination
of random variables, 22
linear discriminant analysis, *see under* discriminant analysis
linear discriminant function, *see under* discriminant function
linear predictor
in generalized linear model, 259
in multiple linear regression, 198
one-factor, 242
two-factor, 244, 274
link function, *see under* generalized linear model
locally weighted scatterplot smoothing, *see* loess
loess, 224–225
bisquare kernel, 225
robust iteration, 225, 227–228
smoothing parameter for, 224
tricube kernel, 224
variance diagnostic, 253
with multiple predictors, 227–228
log-likelihood function, 116
log-linear model, *see* regression, log-linear
log-odds ratio, 266
logarithmic transformation, *see under* transformation
logistic regression, *see* regression, logistic
logit, 43, 266
lowess, *see* loess
LR, *see* likelihood ratio

Mahalanobis distance, 386
Mallows' C_p statistic, 214
matrix, 411
characteristic polynomial of, 416
conformable, 413
determinant, 414
diagonal, 412
diagonal factorization of, 417
eigendecomposition for, 417
full rank, 412
identity (**I**), 412
ill conditioned, 230
inverse, 414–415
nonnegative definite, 415
nonsingular, 414
orthogonal, 413
positive definite, 415
rank of, 412
singular, 414
singular values of, 418
square, 412
square root of, 417–418
symmetric, 412
trace of, 412
transposition of, 412
trapezoidal, 412
triangular, 412
matrix multiplication, 413
maximum likelihood, 120–123
invariance property, 123, 135
maximum likelihood estimator, 120–122
asymptotic distribution, 123
covariance matrix, 123
for binomial probability, 132
for normal mean, 121
for normal variance, 121
for Poisson mean, 154
in generalized linear models, 260
in linear regression, 164, 192, 199, 201
maximum, sample, 54, 79
mean
population, 17
sample, 35, 53
mean squared error
for regression model, 167, 200
in analysis of variance, 243
of estimator, 118
median
population, 19, 54
for discrete distribution, 19
Q_2 notation for, 19
sample, 54–55, 79
robust to outliers, 72
medical informatics, *see under* informatics
method of moments, 118–119

mFDR, *see* false discovery rate, marginal
microarray data, 153, 291, 322, 380
Million Song Dataset, 363
minimal simultaneous coverage, 149
minimum, sample, 54, 79
misclassification error
  in decision trees, 313
  in supervised classification, 296
missing data, xv, 6
  imputation, 6
MLE, *see* maximum likelihood estimator
MLR, *see* regression, multiple linear
MOM, *see* method of moments
Monte Carlo method, 137
moving average, *see under* smoothing
MSE, *see* mean squared error
multicollinearity, 208–210, 342
  in polynomial regression, 210
  remedies for, 210
    Lasso, 238
    ridge regression, 232
  variance inflation factor for, 209
multinomial distribution, 278, 314
multiple comparisons, *see* multiple inferences
multiple inferences, 148–151
multiple linear regression, *see* regression, multiple linear
multiplication rule, 11
  for independent events, 12
  for probability functions, 21
multiplicity adjustment, 149–153
  Bonferroni, 149–151
  in linear regression, 171, 203, 206, 247
    for mean response, 173
  P-values, 151
  Simes', 153
multiplicity correction, *see* multiplicity adjustment
multivariate normal, *see* normal (Gaussian) distribution, multivariate

natural parameter, *see under* exponential family
naïve Bayes classification, 308
negative binomial distribution, 28
  closure under addition, 28
  in exponential family, 48
  p.m.f., 28
    alternative form, 28
nested model, 203
neural networks, xv
Nightingale, F., 94
no-effect hypothesis, *see* null hypothesis
no-free-lunch theorem, 150, 174, 203
  and multiplicity adjustment, 151
noise-to-signal ratio, 65
nonsingular matrix, *see under* matrix
norm, 238
  L^2, 323, 329
normal (Gaussian) distribution, 32–35
  bivariate, 36
    conditional p.d.f.s, 197
    p.d.f., 36
  c.d.f. $\Phi(z)$, 32
  closure under addition, 34
  entropy of, 47
  in exponential family, 42
  interquartile range, 34
  multivariate, 37
    log-likelihood ratio, 303, 336
    p.d.f., 37, 303
  p.d.f., 32
  quantiles, 33
  standard, N(0, 1), 32
  upper-α critical point, 33, 124
normal equations
  for multiple linear regression, 199
  for simple linear regression, 164
  weighted, 184
normal probability plot, *see* quantile plot, normal
normal quantile plot, *see* quantile plot, normal
normal reference rule, *see under* binning
notches, *see under* boxplot
novelty detection, xv
null hypothesis, 138

one-standard error rule, *see under* decision trees
opportunity sampling, *see under* sampling
order statistics, 54

orthogonal matrix, *see under* matrix
orthogonal polynomials, 210
outlier, 60–62
    in boxplot, 79, 81
overdispersion, 288
overfitting, 216, 315, 326, 329, 355
oversmoothing, 66

P-value, 139
p.d.f., *see* probability density function
p.m.f., *see* probability mass function
parameter, 23, 49
parameter vector, 115
Pareto distribution, 48
    in exponential family, 48
partial test, *see under* hypothesis test, for regression coefficient
partition, *see under* sample space
Parzen-Rosenblatt-Whittle estimator, *see* kernel density estimator
PCA, *see* principal component analysis
Pearson product-moment correlation, *see* correlation coefficient
Pearson, K., 59
percentile, 19
percentile method, *see under* confidence interval, bootstrap
pFDR, *see* false discovery rate, positive
pie chart, 90–91
pivotal quantity, *see under* confidence interval
Poisson distribution, 26–27
    closure under addition, 27
    in exponential family, 48
    p.m.f., 26
Poisson postulates, 26
polar area plot, 94
polynomial regression, *see* regression, polynomial
positive definite matrix, *see under* matrix
posterior classification probability, 302
power, *see under* hypothesis test
prediction sum of squares, 214
predictor variable, 164, 198, 291
    aliased, 208
    centered, 210, 231, 255
    qualitative, 242
    standardized, 231, 235, 238
PRESS, *see* prediction sum of squares
principal component analysis, 342–346
    eigendecomposition, 345, 367
    Hotelling transformation, 345
    Kaiser criterion, 345
        Jolliffe adjustment, 346
    Karhunen-Loéve transformation, 345
    scree plot, 346
    sparse, 349
principal component regression, 210, 349
principal components, 342
    loadings, 345
    percent explainable variance, 345, 346
    variances equal to eigenvalues, 345
    visualized, 344, 367
prior classification probability, 302
probability
    axioms, 11
    frequency interpretation, 10
    rules, 11–12
        addition rule, 11
        Bayes' Rule, 12
        complement rule, 11
        conditionality rule, 11
        Law of Total Probability, 12
        multiplication rule, 11
    theory of, 10
probability density function, 14
probability distribution, *see* distribution
probability function, 13
    bivariate, 16
    conditional, 16
        joint, 21
    joint, 16, 21
        for random sample, 116
    marginal, 16
        joint, 21
    skewed, 15–16
    symmetric, 15
    unimodal, 15
probability mass function, 13
probit regression, *see* regression, probit
profile likelihood interval, *see under* confidence interval, likelihood ratio

promax rotation, *see under* exploratory factor analysis, rotation
pruning, *see under* decision trees
psychometric testing, 104, 351, 358, 370

Q-Q plot, *see* quantile-quantile plot
QR decomposition, 417
quadratic discriminant analysis, *see under* discriminant analysis
quadratic form, 415
quality assurance/quality control, 5
quantal response, 271
quantile
  function, 57, 58
  population, 19, 34, 87
  sample, 57–58, 87
  standard normal, 33
quantile plot, 88–89
  normal, 88
quantile-quantile (Q-Q) plot, 96–97
quantitative structure–activity relationship (QSAR), 337
quartile
  population, 19
    first (lower), 19
    third (upper), 19
  sample
    first (lower), 57, 79
    third (upper), 57, 79
quartimax rotation, *see under* exploratory factor analysis, rotation
quintile, 19

R, 24, 421–422
  assignment keystrokes (<-), 423
  bracketed indexing, 425–426, 428
  comment character (#), 423
  data entry, 422–425
  data frame, 424
  flow control, 429
  functions
    inbuilt, 427–428
    user-defined, 429–430
  graphics subsystem, 427
  `Inf`, 422
  logical operators, 428
  `NA`, 422
`NaN`, 422
packages, 430–431
  arules, 399, 400, 403
  biglm, 430
  biganalytics, 430
  bigmemory, 430
  binom, 134
  boot, 138
  car, 209
  CCA, 365
  class, 309–311
  clusterSim, 390
  deldir, 388
  e1071, 329
  ElemStatLearn, 254
  ellipse, 173
  ffbase, 430
  ff, 430
  forecast, 70
  genridge, 233, 256
  ggplot2, 104
  ggsubplot, 94
  glmnet, 233, 239, 240, 242, 271
  kernlab, 330
  klaR, 304, 305
  knnGarden, 309
  lattice, 430
  leaps, 212, 217
  map, 336
  maptree, 315
  MASS, 84, 233, 235, 265, 281, 288, 300, 305, 307
  multcomp, 151
  multtest, 151
  PMA, 349, 364
  psych, 87, 190, 354, 356, 357
  psychometric, 190
  qcc, 70
  randomForest, 322
  rgl, 99, 338, 348
  ridge, 233, 234
  rpart, 315–317, 322, 337
  rpart.plot, 315, 317
  scatterplot3d, 99, 317, 367
  scrime, 309
  stats, 430
  svmpath, 329

R packages (*continued*)
tree, 315
UsingR, 78
vcd, 185
vioplot, 112
R Commander, 431
RStudio, 431
workspace, 421
random forests, 322
random sample, 49
simple, 50
vector notation for, 116
random sampling, *see* sampling, random
random variable, 12
asymptotic behavior, 22
continuous, 13
discrete, 13
independent, 16
linear combinations of, 22
moment of, 18, 45
multivariate, 21
sequence of, 22
sum of, 22
transformation of, 16
vector notation for, 21
weighted average of, 22
randomization, 50
rank correlation, 190
rank of a matrix, *see under* matrix
receiver operating characteristic, *see* ROC curve
regression
bridge, 241
gamma, 281
log-linear, 274, 280
logistic, 266
for supervised classification, 292, 333, 335
regularized, 270
sparse, 270
multiple linear, 198
matrix formulation, 199
nonparametric, 321
Poisson, 274
polynomial, 210–211
probit, 266
ridge, 232
simple linear, 163–164
matrix formulation, 251
support vector, 332
with constant coefficient of variation, 281
regression coefficients, 164, 198
vector of, 199
regression diagnostics, 175
Cook's Distance, 182–183
hat value, 181, 200
influential observation, 182–183
leverage, 181
Mallows' C_p, 214
outlier detection, 177–179
prediction sum of squares, 214
raw deleted residual, 178
residual loess plot, 253
residual plot, 176
residual Q-Q plot, 177
Studentized deleted residual, 178–179
exceedance level, 179
Studentized residual, 178
regression tree, 321
regularization, 230
elastic net, 241–242
L_1 penalty, 238
Tikhonov, 231
via Lasso, 238–239
via ridge regression, 232–233
regularization parameter
decision tree, 315
via cross-validation, 315
penalized least squares, 231
support vector, 326
regularization path, *see under* support vector methods
rejection region, 139
research hypothesis, *see* alternative hypothesis
residual, 262
deviance, 263
in multiple linear regression, 200
in simple linear regression, 167
Pearson, 263
raw deleted, 178
standardized

in generalized linear model, 263
Studentized, 178
Studentized deleted, 178–179
exceedance level, 179
residual plot, 176
heterogeneous variance in, 176
nonlinear pattern in, 176
residual sum of squares, 167, 200, 243
response variable, 164
response vector, 199
ridge regression, 232–233
bias-variance trade-off, 232
hat matrix, 232
ridge trace, 233
standard errors in, 236
tuning parameter for, 232
selection of, 233
ridge trace plot, 233
ROC curve, 297
rose plot, *see* polar area plot
R^2, *see* coefficient of determination
R^2_p plot, *see under* variable selection
rug plot, 81, 84

S language, 421
S-plus, 24
sabermetrics, 185
sample mean, 35, 53
asymptotic distribution, 35
asymptotic distribution of, 118
from normal distribution, 35, 118
sensitive to outliers, 54
standard error of, 118
sample size, 49
sample space, 10
partition of, 11
sample variance, 56
computing formulae, 56
sampling
convenience, 7, 9
opportunity, 7
probability-based, 6
random, 49–51
selection bias in, 7
systematic bias in, 50
sampling distribution, 117
sampling frame, 6
SAS, 24
Satterthwaite correction, *see* Welch-Satterthwaite correction
saturated model, 261, 275
scalar, 411
scatterplot, 98–99
three-dimensional, 99
scatterplot matrix, 101–102
schematic plot, 80
Schwarz's Bayesian Criterion, *see* information criteria, Bayesian
Scott's normal reference rule, *see under* binning
scree plot, *see under* principal component analysis
selection bias, *see under* sampling
self-organizing maps, xv
sensitivity
of classification rule, 296
of hypothesis test, 139
separate families of hypotheses, 203
Shannon entropy, *see* entropy
Shapiro-Wilk test, 155, 156
shrinkage regression, 231, 238
signal processing, 322
signal-vs.-noise problem, 65, 298
significance level, 139
signum function, 325
Simes' adjustment, *see under* multiplicity adjustment
simple linear model, *see* regression, simple linear
simple linear regression, *see* regression, simple linear
simple random sampling, *see* random sample, simple
simultaneous confidence region, 149
single nucleotide polymorphism data, 234–236, 239, 322
singular matrix, *see under* matrix
singular value decomposition, 345, 418–419
skew, *see* probability function, skewed
in probability function, 15
in random sample, 54
SLR, *see* regression, simple linear

smoothing, 65
averaging window, 68
double exponential, 70
exponential, 69–70
exponential weighted average, 70
loess, 224
with multiple predictors, 227–228
moving average, 68
retrospective, 68
symmetric, 68
via kernel density estimator, 85
SNP, *see* single nucleotide polymorphism
socioinformatics, 5
sparse discriminant analysis, *see under* discriminant analysis
sparse logistic regression, *see under* regression, logistic
sparse PCA, *see under* principal component analysis
Spearman's rank correlation, *see* rank correlation
specificity, 296
spectral decomposition, 345, 353, 417
SPSS, 24
SRS, *see* random sample, simple
SSE, *see* residual sum of squares
standard deviation
population, 18
sample, 56
standard error, 117
of binomial proportion, 133
of sample mean, 118
standard normal, *see* normal (Gaussian) distribution, standard
standardized variate, 32
statistical inference, 7, 115
statistical learning
supervised, 163
unsupervised, 341
Statlog project , 408
stem-and-leaf plot, *see* stemplot
stemplot, 81–83
back-to-back, 83
stochastic simulation, *see* Monte Carlo Method
strip chart, 77, 79
strip plot, *see* strip chart
Student, *see* Gosset, W.S.
Student's *t*-distribution, 38–40
Student's *t*-test, 140
Studentized deleted residual, *see under* regression diagnostics
Studentized variate, 125
Sturges' rule, *see under* binning
subset selection, *see* variable selection, (best) subset
sum of squares
for ANOVA main effect, 243
partial, 244
residual, 167, 200, 243
sequential, 244
supervised learning, 163
classification analytics, 291
regression analysis, 163
support space, *see* sample space
support vector, 322, 326
nonmargin, 326
support vector machines, 328
support vector methods, 322
classification rule, 325
kernel transformation, 326–329
kernel trick, 329
maximum margin classifier, 322
optimal separating hyperplane, 322, 324, 328
regularization path, 326
robust to outliers, 324
supporting hyperplane, 322
tuning parameter in, 325
support vector regression, 332
survival probability, 29
SVD, *see* singular value decomposition
SVM, *see* support vector machines
symmetric matrix, *see under* matrix
synergy, 267
synthetic data generation, *see* Monte Carlo Method

t distribution, 38–40
dervied from normal, 38
upper-α critical point, 39, 125

t-test, 140
for difference in normal means
F-test equivalent, 243
with equal variances, 144
with paired data, 144–145
with unequal variances, 142–143
for normal mean, 140–141, 147
for regression coefficient
multiple linear, 202
polynomial, 211
simple linear, 168
tessellation, *see* Voronoi tessellation
test data, 233, 291
text mining, xv
Tikhonov regularization, *see under* regularization
time series, 67, 105
ARIMA model, 70
financial, 68
plot, 106–109
Total Probability, Law of, *see under* probability, rules
total sum of squares, 170
trace of a matrix, *see under* matrix
trace plot, 106
training data, 163, 233, 291
transformation, 16, 64–65
arc-sine/square-root, 64
Box-Cox, 64–65, 74, 249, 257
logarithmic, 64
to reduce skew, 64
logit, 64
power, 64–65, 74, 249, 257
square root, 64
transposition operator, 411
trellis graphics, 430
trend test, *see* Cochran-Armitage trend test
triangular matrix, *see under* matrix
trimmed mean, 55
tuning parameter
for kernel function, 329
for Lasso, 238
in penalized least squares, 231
in ridge regression, 232
with decision trees, 314
with support vector methods, 325
Twitter, 157, 250, 286
Type I error, *see* false positive error
Type II error, *see* false negative error

UCI Machine Learning Repository, xiv
unbalanced design, 242, 245
unbiased estimator, *see* estimator, unbiased
uncertainty
of estimator, 117, 167
uniform distribution
continuous, 29
entropy of, 47
discrete, 28
unsupervised learning, 341
association rules, 395
cluster analysis, 373
dimension reduction, 342
exploratory factor analysis, 351
market basket analysis, 395
principal component analysis, 342
upper-α critical point
χ^2, 37, 127
F, 40
standard normal, 124
t, 39, 125

variable, *see* random variable
variable selection, 211–216
(best) subset, 216–217
backward elimination, 219
forward selection, 219
forward stepwise, 218–219
inference after, 211
R_p^2 plot in, 212
stepwise, 217
via AIC/BIC, 215, 222
via Lasso, 238
variance
of estimator, 117
of linear combination, 22
of sum of independent variables, 22
population, 18
sample, 56
variance inflation factor, 209

variance-bias tradeoff, *see* bias-variance tradeoff
variance-covariance matrix, *see* covariance matrix
varimax rotation, *see under* exploratory factor analysis, rotation
vector, 411
  column, 411
  coordinate unit (**e**), 412
  intertia of, 415
  linearly independent, 412
  orthogonal, 413
  orthonormal, 413
  row, 411
  transposition operator for, 411
vector addition, 413
vector multiplication, 413
VIF, *see* variance inflation factor
violin plot, 112
visualization
  multivariate, 90
  univariate, 76
Voronoi tessellation, 388–389

Wald interval, *see under* confidence interval
Wald test, *see under* hypothesis test
Wald, A., 131
weakest-link pruning, *see under* decision trees
weighted average
  expected value of, 22
  of observations, 53
  of random variables, 22
weighted least squares, *see* least squares, weighted
weighting inversely to variance, 120, 185
Welch-Satterthwaite correction, 128, 143
whiskers, *see under* boxplot
Wilk-Shapiro test, *see* Shapiro-Wilk test
Wilson interval, *see under* confidence interval, for binomial probability
window width, 85
Winsorized mean, 55–56
wordle plot, 100

z-score, 57