

THOMAS W. MILLER

FACULTY DIRECTOR OF NORTHWESTERN UNIVERSITY'S
PREDICTIVE ANALYTICS PROGRAM

MODELING
TECHNIQUES
IN
PREDICTIVE
ANALYTICS

A GUIDE TO DATA SCIENCE

PYTHON EDITION

Modeling Techniques in Predictive Analytics with Python and R

A Guide to Data Science

THOMAS W. MILLER

Associate Publisher: Amy Neidlinger
Executive Editor: Jeanne Glasser
Operations Specialist: Jodi Kemper
Cover Designer: Alan Clements
Managing Editor: Kristy Hart
Project Editor: Andy Beaster
Senior Compositor: Gloria Schurick
Manufacturing Buyer: Dan Uhrig

©2015 by Thomas W. Miller
Published by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458

Pearson offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact U.S. Corporate and Government Sales, 1-800-382-3419, corpsales@pearsoned.com. For sales outside the U.S., please contact International Sales at international@pearsoned.com.

Company and product names mentioned herein are the trademarks or registered trademarks of their respective owners.

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

First Printing October 2014

ISBN-10: 0-13-3892069

ISBN-13: 978-0-13-389206-2

Pearson Education LTD.
Pearson Education Australia PTY, Limited.
Pearson Education Singapore, Pte. Ltd.
Pearson Education Asia, Ltd.
Pearson Education Canada, Ltd.
Pearson Educacin de Mexico, S.A. de C.V.
Pearson Education—Japan
Pearson Education Malaysia, Pte. Ltd.
Library of Congress Control Number: 2014948913

Contents

Preface	v
Figures	xi
Tables	xv
Exhibits	xvii
1 Analytics and Data Science	1
2 Advertising and Promotion	16
3 Preference and Choice	33
4 Market Basket Analysis	43
5 Economic Data Analysis	61
6 Operations Management	81
7 Text Analytics	103
8 Sentiment Analysis	135
9 Sports Analytics	187

10	Spatial Data Analysis	211
11	Brand and Price	239
12	The Big Little Data Game	273
A	Data Science Methods	277
A.1	Databases and Data Preparation	279
A.2	Classical and Bayesian Statistics	281
A.3	Regression and Classification	284
A.4	Machine Learning	289
A.5	Web and Social Network Analysis	291
A.6	Recommender Systems	293
A.7	Product Positioning	295
A.8	Market Segmentation	297
A.9	Site Selection	299
A.10	Financial Data Science	300
B	Measurement	301
C	Case Studies	315
C.1	Return of the Bobbleheads	315
C.2	DriveTime Sedans	316
C.3	Two Month's Salary	321
C.4	Wisconsin Dells	325
C.5	Computer Choice Study	330
D	Code and Utilities	335
	Bibliography	379
	Index	413

Preface

“All right . . . all right . . . but apart from better sanitation, the medicine, education, wine, public order, irrigation, roads, a fresh water system, and public health . . . what have the Romans ever done for us?”

—JOHN CLEESE AS REG IN *Life of Brian* (1979)

I was in a doctoral-level statistics course at the University of Minnesota in the late 1970s when I learned a lesson about the programming habits of academics. At the start of the course, the instructor said, “I don’t care what language you use for assignments, as long as you do your own work.”

I had facility with Fortran but was teaching myself Pascal at the time. I was developing a structured programming style—no more GO TO statements. So, taking the instructor at his word, I programmed the first assignment in Pascal. The other fourteen students in the class were programming in Fortran, the lingua franca of statistics at the time.

When I handed in the assignment, the instructor looked at it and asked, “What’s this?”

“Pascal,” I said. “You told us we could program in any language we like as long as we do our own work.”

He responded, “Pascal. I don’t read Pascal. I only read Fortran.”

Today's world of data science brings together information technology professionals fluent in Python with statisticians fluent in R. These communities have much to learn from each other. For the practicing data scientist, there are considerable advantages to being multilingual.

Sometimes referred to as a "glue language," Python provides a rich open-source environment for scientific programming and research. For computer-intensive applications, it gives us the ability to call on compiled routines from C, C++, and Fortran. Or we can use Cython to convert Python code into optimized C. For modeling techniques or graphics not currently implemented in Python, we can execute R programs from Python. We can draw on R packages for nonlinear estimation, Bayesian hierarchical modeling, time series analysis, multivariate methods, statistical graphics, and the handling of missing data, just as R users can benefit from Python's capabilities as a general-purpose programming language.

Data and algorithms rule the day. Welcome to the new world of business, a fast-paced, data-intensive world, an open-source environment in which competitive advantage, however fleeting, is obtained through analytic prowess and the sharing of ideas.

Many books about predictive analytics or data science talk about strategy and management. Some focus on methods and models. Others look at information technology and code. This is a rare book that does all three, appealing to business managers, modelers, and programmers alike.

We recognize the importance of analytics in gaining competitive advantage. We help researchers and analysts by providing a ready resource and reference guide for modeling techniques. We show programmers how to build upon a foundation of code that works to solve real business problems. We translate the results of models into words and pictures that management can understand. We explain the meaning of data and models.

Growth in the volume of data collected and stored, in the variety of data available for analysis, and in the rate at which data arrive and require analysis, makes analytics more important with each passing day. Achieving competitive advantage means implementing new systems for information management and analytics. It means changing the way business is done.

Literature in the field of data science is massive, drawing from many academic disciplines and application areas. The relevant open-source code is growing quickly. Indeed, it would be a challenge to provide a comprehensive guide to predictive analytics or data science.

We look at real problems and real data. We offer a collection of vignettes with each chapter focused on a particular application area and business problem. We provide solutions that make sense. By showing modeling techniques and programming tools in action, we convert abstract concepts into concrete examples. Fully worked examples facilitate understanding.

Our objective is to provide an overview of predictive analytics and data science that is accessible to many readers. There is scant mathematics in the book. Statisticians and modelers may look to the references for details and derivations of methods. We describe methods in plain English and use data visualization to show solutions to business problems.

Given the subject of the book, some might wonder if I belong to either the classical or Bayesian camp. At the School of Statistics at the University of Minnesota, I developed a respect for both sides of the classical/Bayesian divide. I have high regard for the perspective of empirical Bayesians and those working in statistical learning, which combines machine learning and traditional statistics. I am a pragmatist when it comes to modeling and inference. I do what works and express my uncertainty in statements that others can understand.

This book is possible because of the thousands of experts across the world, people who contribute time and ideas to open source. The growth of open source and the ease of growing it further ensures that developed solutions will be around for many years to come. Genie out of the lamp, wizard from behind the curtain—rocket science is not what it used to be. Secrets are being revealed. This book is part of the process.

Most of the data in the book were obtained from public domain data sources. Major League Baseball data for promotions and attendance were contributed by Erica Costello. Computer choice study data were made possible through work supported by Sharon Chamberlain. The call center data of “Anonymous Bank” were provided by Avi Mandelbaum and Ilan Guedj. Movie information was obtained courtesy of The Internet Movie Database, used with permission. IMDb movie reviews data were organized by Andrew L.

Mass and his colleagues at Stanford University. Some examples were inspired by working with clients at ToutBay of Tampa, Florida, NCR Comten, Hewlett-Packard Company, Site Analytics Co. of New York, Sunseed Research of Madison, Wisconsin, and Union Cab Cooperative of Madison.

We work within open-source communities, sharing code with one another. The truth about what we do is in the programs we write. It is there for everyone to see and for some to debug. To promote student learning, each program includes step-by-step comments and suggestions for taking the analysis further. All data sets and computer programs are downloadable from the book's website at <http://www.ftpress.com/miller/>.

The initial plan for this book was to translate the R version of the book into Python. While working on what was going to be a Python-only edition, however, I gained a more profound respect for both languages. I saw how some problems are more easily solved with Python and others with R. Furthermore, being able to access the wealth of R packages for modeling techniques and graphics while working in Python has distinct advantages for the practicing data scientist. Accordingly, this edition of the book includes Python and R code examples. It represents a unique dual-language guide to data science.

Many have influenced my intellectual development over the years. There were those good thinkers and good people, teachers and mentors for whom I will be forever grateful. Sadly, no longer with us are Gerald Hahn Hinkle in philosophy and Allan Lake Rice in languages at Ursinus College, and Herbert Feigl in philosophy at the University of Minnesota. I am also most thankful to David J. Weiss in psychometrics at the University of Minnesota and Kelly Eakin in economics, formerly at the University of Oregon. Good teachers—yes, great teachers—are valued for a lifetime.

Thanks to Michael L. Rothschild, Neal M. Ford, Peter R. Dickson, and Janet Christopher who provided invaluable support during our years together at the University of Wisconsin–Madison and the A.C. Nielsen Center for Marketing Research.

I live in California, four miles north of Dodger Stadium, teach for Northwestern University in Evanston, Illinois, and direct product development at ToutBay, a data science firm in Tampa, Florida. Such are the benefits of a good Internet connection.

I am fortunate to be involved with graduate distance education at Northwestern University's School of Professional Studies. Thanks to Glen Fogerty, who offered me the opportunity to teach and take a leadership role in the predictive analytics program at Northwestern University. Thanks to colleagues and staff who administer this exceptional graduate program. And thanks to the many students and fellow faculty from whom I have learned.

ToutBay is an emerging firm in the data science space. With co-founder Greg Blence, I have great hopes for growth in the coming years. Thanks to Greg for joining me in this effort and for keeping me grounded in the practical needs of business. Academics and data science models can take us only so far. Eventually, to make a difference, we must implement our ideas and models, sharing them with one another.

Amy Hendrickson of T_EXnology Inc. applied her craft, making words, tables, and figures look beautiful in print—another victory for open source. Thanks to Donald Knuth and the T_EX/L^AT_EX community for their contributions to this wonderful system for typesetting and publication.

Thanks to readers and reviewers of the initial R edition of the book, including Suzanne Callender, Philip M. Goldfeder, Melvin Ott, and Thomas P. Ryan. For the revised R edition, Lorena Martin provided much needed feedback and suggestions for improving the book. Candice Bradley served dual roles as a reviewer and copyeditor, and Roy L. Sanford provided technical advice about statistical models and programs. Thanks also to my editor, Jeanne Glasser Levine, and publisher, Pearson/FT Press, for making this book possible. Any writing issues, errors, or items of unfinished business, of course, are my responsibility alone.

My good friend Brittney and her daughter Janiya keep me company when time permits. And my son Daniel is there for me in good times and bad, a friend for life. My greatest debt is to them because they believe in me.

Thomas W. Miller
Glendale, California
August 2014

This page intentionally left blank

Figures

1.1	Data and models for research	3
1.2	Training-and-Test Regimen for Model Evaluation	6
1.3	Training-and-Test Using Multi-fold Cross-validation	7
1.4	Training-and-Test with Bootstrap Resampling	8
1.5	Importance of Data Visualization: The Anscombe Quartet	10
2.1	Dodgers Attendance by Day of Week	19
2.2	Dodgers Attendance by Month	19
2.3	Dodgers Weather, Fireworks, and Attendance	20
2.4	Dodgers Attendance by Visiting Team	21
2.5	Regression Model Performance: Bobbleheads and Attendance	23
3.1	Spine Chart of Preferences for Mobile Communication Services	36
4.1	Market Basket Prevalence of Initial Grocery Items	47
4.2	Market Basket Prevalence of Grocery Items by Category	49
4.3	Market Basket Association Rules: Scatter Plot	50
4.4	Market Basket Association Rules: Matrix Bubble Chart	51
4.5	Association Rules for a Local Farmer: A Network Diagram	53
5.1	Multiple Time Series of Economic Data	63
5.2	Horizon Plot of Indexed Economic Time Series	65
5.3	Forecast of National Civilian Employment Rate (percentage)	67
5.4	Forecast of Manufacturers' New Orders: Durable Goods (billions of dollars)	67
5.5	Forecast of University of Michigan Index of Consumer Sentiment (1Q 1966 = 100)	68
5.6	Forecast of New Homes Sold (millions)	68
6.1	Call Center Operations for Monday	83
6.2	Call Center Operations for Tuesday	83
6.3	Call Center Operations for Wednesday	84
6.4	Call Center Operations for Thursday	84

6.5	Call Center Operations for Friday	85
6.6	Call Center Operations for Sunday	85
6.7	Call Center Arrival and Service Rates on Wednesdays	86
6.8	Call Center Needs and Optimal Workforce Schedule	89
7.1	Movie Taglines from The Internet Movie Database (IMDb)	104
7.2	Movies by Year of Release	106
7.3	A Bag of 200 Words from Forty Years of Movie Taglines	108
7.4	Picture of Text in Time: Forty Years of Movie Taglines	109
7.5	Text Measures and Documents on a Single Graph	110
7.6	Horizon Plot of Text Measures across Forty Years of Movie Taglines	112
7.7	From Text Processing to Text Analytics	113
7.8	Linguistic Foundations of Text Analytics	114
7.9	Creating a Terms-by-Documents Matrix	116
8.1	A Few Movie Reviews According to Tom	136
8.2	A Few More Movie Reviews According to Tom	137
8.3	Fifty Words of Sentiment	139
8.4	List-Based Text Measures for Four Movie Reviews	141
8.5	Scatter Plot of Text Measures of Positive and Negative Sentiment	142
8.6	Word Importance in Classifying Movie Reviews as Thumbs-Up or Thumbs-Down	146
8.7	A Simple Tree Classifier for Thumbs-Up or Thumbs-Down	147
9.1	Predictive Modeling Framework for Picking a Winning Team	188
9.2	Game-day Simulation (offense only)	194
9.3	Mets' Away and Yankees' Home Data (offense and defense)	195
9.4	Balanced Game-day Simulation (offense and defense)	196
9.5	Actual and Theoretical Runs-scored Distributions	198
9.6	Poisson Model for Mets vs. Yankees at Yankee Stadium	200
9.7	Negative Binomial Model for Mets vs. Yankees at Yankee Stadium	201
9.8	Probability of Home Team Winning (Negative Binomial Model)	203
10.1	California Housing Data: Correlation Heat Map for the Training Data	215
10.2	California Housing Data: Scatter Plot Matrix of Selected Variables	216
10.3	Tree-Structured Regression for Predicting California Housing Values	218
10.4	Random Forests Regression for Predicting California Housing Values	219
11.1	Computer Choice Study: A Mosaic of Top Brands and Most Valued Attributes	242
11.2	Framework for Describing Consumer Preference and Choice	244

11.3	Ternary Plot of Consumer Preference and Choice	244
11.4	Comparing Consumers with Differing Brand Preferences	245
11.5	Potential for Brand Switching: Parallel Coordinates for Individual Consumers	247
11.6	Potential for Brand Switching: Parallel Coordinates for Consumer Groups	248
11.7	Market Simulation: A Mosaic of Preference Shares	251
12.1	Work of Data Science	274
A.1	Evaluating Predictive Accuracy of a Binary Classifier	286
B.1	Hypothetical Multitrait-Multimethod Matrix	303
B.2	Conjoint Degree-of-Interest Rating	306
B.3	Conjoint Sliding Scale for Profile Pairs	306
B.4	Paired Comparisons	307
B.5	Multiple-Rank-Orders	307
B.6	Best-worst Item Provides Partial Paired Comparisons	308
B.7	Paired Comparison Choice Task	310
B.8	Choice Set with Three Product Profiles	310
B.9	Menu-based Choice Task	312
B.10	Elimination Pick List	313
C.1	Computer Choice Study: One Choice Set	332
D.1	A Python Programmer's Word Cloud	338
D.2	An R Programmer's Word Cloud	338

This page intentionally left blank

Tables

1.1	Data for the Anscombe Quartet	9
2.1	Bobbleheads and Dodger Dogs	18
2.2	Regression of Attendance on Month, Day of Week, and Bobblehead Promotion	24
3.1	Preference Data for Mobile Communication Services	34
4.1	Market Basket for One Shopping Trip	44
4.2	Association Rules for a Local Farmer	52
6.1	Call Center Shifts and Needs for Wednesdays	87
6.2	Call Center Problem and Solution	88
8.1	List-Based Sentiment Measures from Tom’s Reviews	140
8.2	Accuracy of Text Classification for Movie Reviews (Thumbs-Up or Thumbs-Down)	144
8.3	Random Forest Text Measurement Model Applied to Tom’s Movie Reviews	145
9.1	New York Mets’ Early Season Games in 2007	191
9.2	New York Yankees’ Early Season Games in 2007	192
10.1	California Housing Data: Original and Computed Variables	213
10.2	Linear Regression Fit to Selected California Block Groups	217
10.3	Comparison of Regressions on Spatially Referenced Data	220
11.1	Contingency Table of Top-ranked Brands and Most Valued Attributes	243
11.2	Market Simulation: Choice Set Input	250
11.3	Market Simulation: Preference Shares in a Hypothetical Four-brand Market	252
C.1	Hypothetical profits from model-guided vehicle selection	318
C.2	DriveTime Data for Sedans	319
C.3	DriveTime Sedan Color Map with Frequency Counts	320
C.4	Diamonds Data: Variable Names and Coding Rules	324

C.5	Dells Survey Data: Visitor Characteristics	328
C.6	Dells Survey Data: Visitor Activities	329
C.7	Computer Choice Study: Product Attributes	331
C.8	Computer Choice Study: Data for One Individual	333

Exhibits

1.1	Programming the Anscombe Quartet (Python)	13
1.2	Programming the Anscombe Quartet (R)	15
2.1	Shaking Our Bobbleheads Yes and No (Python)	27
2.2	Shaking Our Bobbleheads Yes and No (R)	30
3.1	Measuring and Modeling Individual Preferences (Python)	38
3.2	Measuring and Modeling Individual Preferences (R)	40
4.1	Market Basket Analysis of Grocery Store Data (Python)	56
4.2	Market Basket Analysis of Grocery Store Data (R)	58
5.1	Working with Economic Data (Python)	70
5.2	Working with Economic Data (R)	76
6.1	Call Center Scheduling (Python)	91
6.2	Call Center Scheduling (R)	96
7.1	Text Analysis of Movie Taglines (Python)	120
7.2	Text Analysis of Movie Taglines (R)	127
8.1	Sentiment Analysis and Classification of Movie Ratings (Python)	151
8.2	Sentiment Analysis and Classification of Movie Ratings (R)	167
9.1	Team Winning Probabilities by Simulation (Python)	209
9.2	Team Winning Probabilities by Simulation (R)	210
10.1	Regression Models for Spatial Data (Python)	222
10.2	Regression Models for Spatial Data (R)	229
11.1	Training and Testing a Hierarchical Bayes Model (R)	255
11.2	Preference, Choice, and Market Simulation (R)	260
D.1	Evaluating Predictive Accuracy of a Binary Classifier (Python)	339
D.2	Text Measures for Sentiment Analysis (Python)	340
D.3	Summative Scoring of Sentiment (Python)	342
D.4	Conjoint Analysis Spine Chart (R)	343
D.5	Market Simulation Utilities (R)	351
D.6	Split-plotting Utilities (R)	352

D.7	Wait-time Ribbon Plot (R)	355
D.8	Movie Tagline Data Preparation Script for Text Analysis (R)	367
D.9	Word Scoring Code for Sentiment Analysis (R)	372
D.10	Utilities for Spatial Data Analysis (R)	376
D.11	Making Word Clouds (R)	377

1

Analytics and Data Science

Mr. Maguire: "I just want to say one word to you, just one word."

Ben: "Yes, sir."

Mr. Maguire: "Are you listening?"

Ben: "Yes, I am."

Mr. Maguire: "Plastics."

—WALTER BROOKE AS MR. MAGUIRE AND DUSTIN HOFFMAN
AS BEN (BENJAMIN BRADDOCK) IN *The Graduate* (1967)

While earning a degree in philosophy may not be the best career move (unless a student plans to teach philosophy, and few of these positions are available), I greatly value my years as a student of philosophy and the liberal arts. For my bachelor's degree, I wrote an honors paper on Bertrand Russell. In graduate school at the University of Minnesota, I took courses from one of the truly great philosophers, Herbert Feigl. I read about science and the search for truth, otherwise known as epistemology. My favorite philosophy was logical empiricism.

Although my days of "thinking about thinking" (which is how Feigl defined philosophy) are far behind me, in those early years of academic training I was able to develop a keen sense for what is real and what is just talk.

A *model* is a representation of things, a rendering or description of reality. A typical model in data science is an attempt to relate one set of variables to another. Limited, imprecise, but useful, a model helps us to make sense of the world. A model is more than just talk because it is based on data.

Predictive analytics brings together management, information technology, and modeling. It is designed for today's data-intensive world. Predictive analytics is data science, a multidisciplinary skill set essential for success in business, nonprofit organizations, and government. Whether forecasting sales or market share, finding a good retail site or investment opportunity, identifying consumer segments and target markets, or assessing the potential of new products or risks associated with existing products, modeling methods in predictive analytics provide the key.

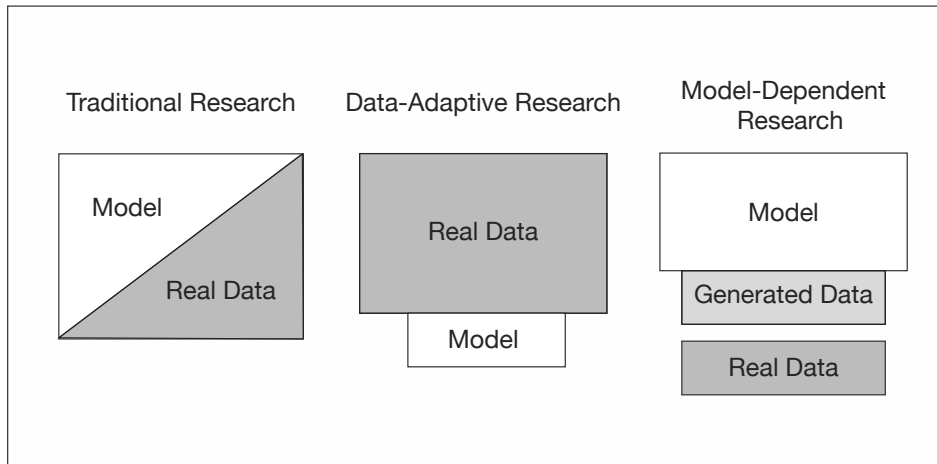
Data scientists, those working in the field of predictive analytics, speak the language of business—accounting, finance, marketing, and management. They know about information technology, including data structures, algorithms, and object-oriented programming. They understand statistical modeling, machine learning, and mathematical programming. Data scientists are methodological eclectics, drawing from many scientific disciplines and translating the results of empirical research into words and pictures that management can understand.

Predictive analytics, as with much of statistics, involves searching for meaningful relationships among variables and representing those relationships in models. There are response variables—things we are trying to predict. There are explanatory variables or predictors—things that we observe, manipulate, or control and might relate to the response.

Regression methods help us to predict a response with meaningful magnitude, such as quantity sold, stock price, or return on investment. Classification methods help us to predict a categorical response. Which brand will be purchased? Will the consumer buy the product or not? Will the account holder pay off or default on the loan? Is this bank transaction true or fraudulent?

Prediction problems are defined by their width or number of potential predictors and by their depth or number of observations in the data set. It is the number of potential predictors in business, marketing, and investment analysis that causes the most difficulty. There can be thousands of potential

Figure 1.1. Data and models for research



predictors with weak relationships to the response. With the aid of computers, hundreds or thousands of models can be fit to subsets of the data and tested on other subsets of the data, providing an evaluation of each predictor. Predictive modeling involves finding good subsets of predictors. Models that fit the data well are better than models that fit the data poorly. Simple models are better than complex models.

Consider three general approaches to research and modeling as employed in predictive analytics: traditional, data-adaptive, and model-dependent. See figure 1.1. The traditional approach to research, statistical inference, and modeling begins with the specification of a theory or model. Classical or Bayesian methods of statistical inference are employed. Traditional methods, such as linear regression and logistic regression, estimate parameters for linear predictors. Model building involves fitting models to data and checking them with diagnostics. We validate traditional models before using them to make predictions.

When we employ a data-adaptive approach, we begin with data and search through those data to find useful predictors. We give little thought to theories or hypotheses prior to running the analysis. This is the world of machine learning, sometimes called statistical learning or data mining. Data-adaptive methods adapt to the available data, representing nonlinear relationships and interactions among variables. The data determine the model.

Data-adaptive methods are data-driven. As with traditional models, we validate data-adaptive models before using them to make predictions.

Model-dependent research is the third approach. It begins with the specification of a model and uses that model to generate data, predictions, or recommendations. Simulations and mathematical programming methods, primary tools of operations research, are examples of model-dependent research. When employing a model-dependent or simulation approach, models are improved by comparing generated data with real data. We ask whether simulated consumers, firms, and markets behave like real consumers, firms, and markets. The comparison with real data serves as a form of validation.

It is often a combination of models and methods that works best. Consider an application from the field of financial research. The manager of a mutual fund is looking for additional stocks for a fund's portfolio. A financial engineer employs a data-adaptive model (perhaps a neural network) to search across thousands of performance indicators and stocks, identifying a subset of stocks for further analysis. Then, working with that subset of stocks, the financial engineer employs a theory-based approach (CAPM, the capital asset pricing model) to identify a smaller set of stocks to recommend to the fund manager. As a final step, using model-dependent research (mathematical programming), the engineer identifies the minimum-risk capital investment for each of the stocks in the portfolio.

Data may be organized by observational unit, time, and space. The observational or cross-sectional unit could be an individual consumer or business or any other basis for collecting and grouping data. Data are organized in time by seconds, minutes, hours, days, and so on. Space or location is often defined by longitude and latitude.

Consider numbers of customers entering grocery stores (units of analysis) in Glendale, California on Monday (one point in time), ignoring the spatial location of the stores—these are cross-sectional data. Suppose we work with one of those stores, looking at numbers of customers entering the store each day of the week for six months—these are time series data. Then we look at numbers of customers at all of the grocery stores in Glendale across six months—these are longitudinal or panel data. To complete our study, we locate these stores by longitude and latitude, so we have spatial

or spatio-temporal data. For any of these data structures we could consider measures in addition to the number of customers entering stores. We look at store sales, consumer or nearby resident demographics, traffic on Glendale streets, and so doing move to multiple time series and multivariate methods. The organization of the data we collect affects the structure of the models we employ.

As we consider business problems in this book, we touch on many types of models, including cross-sectional, time series, and spatial data models. Whatever the structure of the data and associated models, prediction is the unifying theme. We use the data we have to predict data we do not yet have, recognizing that prediction is a precarious enterprise. It is the process of extrapolating and forecasting. And model validation is essential to the process.

To make predictions, we may employ classical or Bayesian methods. Or we may dispense with traditional statistics entirely and rely upon machine learning algorithms. We do what works.¹ Our approach to predictive analytics is based upon a simple premise:

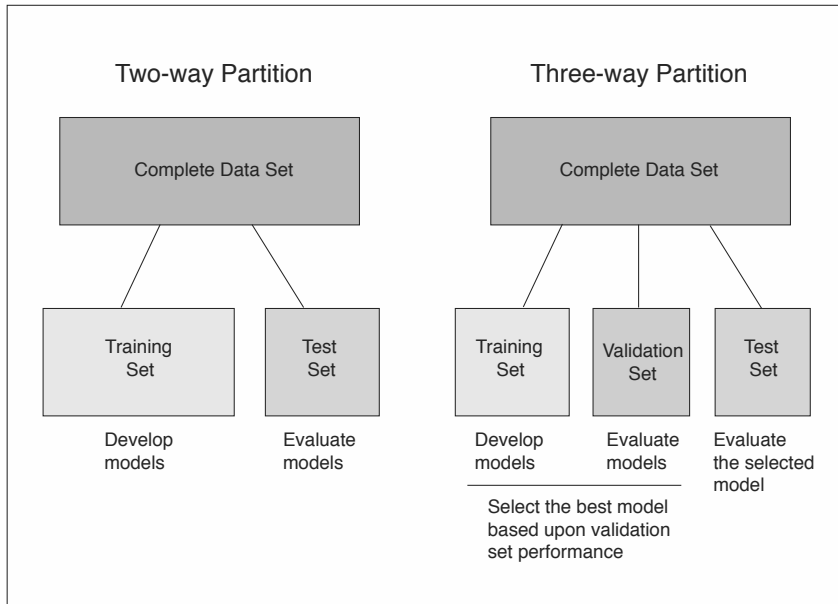
The value of a model lies in the quality of its predictions.

We learn from statistics that we should quantify our uncertainty. On the one hand, we have confidence intervals, point estimates with associated standard errors, significance tests, and p -values—that is the classical way. On the other hand, we have posterior probability distributions, probability intervals, prediction intervals, Bayes factors, and subjective (perhaps diffuse) priors—the path of Bayesian statistics. Indices such as the Akaike information criterion (AIC) or the Bayes information criterion (BIC) help us to judge one model against another, providing a balance between goodness-of-fit and parsimony.

Central to our approach is a *training-and-test regimen*. We partition sample data into training and test sets. We build our model on the training set and

¹ Within the statistical literature, Seymour Geisser (1929–2004) introduced an approach best described as *Bayesian predictive inference* (Geisser 1993). Bayesian statistics is named after Reverend Thomas Bayes (1706–1761), the creator of Bayes Theorem. In our emphasis upon the success of predictions, we are in agreement with Geisser. Our approach, however, is purely empirical and in no way dependent upon classical or Bayesian thinking.

Figure 1.2. Training-and-Test Regimen for Model Evaluation



evaluate it on the test set. Simple two- and three-way data partitioning are shown in figure 1.2.

A random splitting of a sample into training and test sets could be fortuitous, especially when working with small data sets, so we sometimes conduct statistical experiments by executing a number of random splits and averaging performance indices from the resulting test sets. There are extensions to and variations on the training-and-test theme.

One variation on the training-and-test theme is multi-fold cross-validation, illustrated in figure 1.3. We partition the sample data into M folds of approximately equal size and conduct a series of tests. For the five-fold cross-validation shown in the figure, we would first train on sets B through E and test on set A . Then we would train on sets A and C through E , and test on B . We continue until each of the five folds has been utilized as a test set. We assess performance by averaging across the test sets. In leave-one-out cross-validation, the logical extreme of multi-fold cross-validation, there are as many test sets as there are observations in the sample.

Figure 1.3. Training-and-Test Using Multi-fold Cross-validation

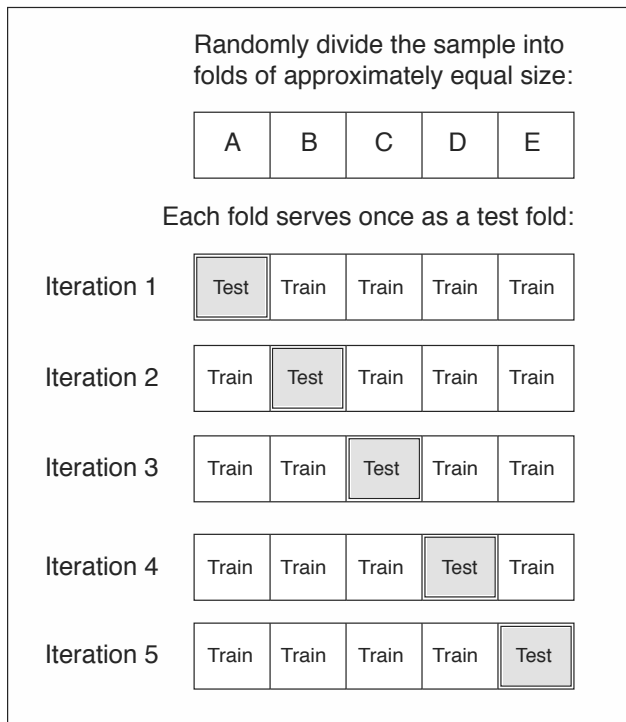
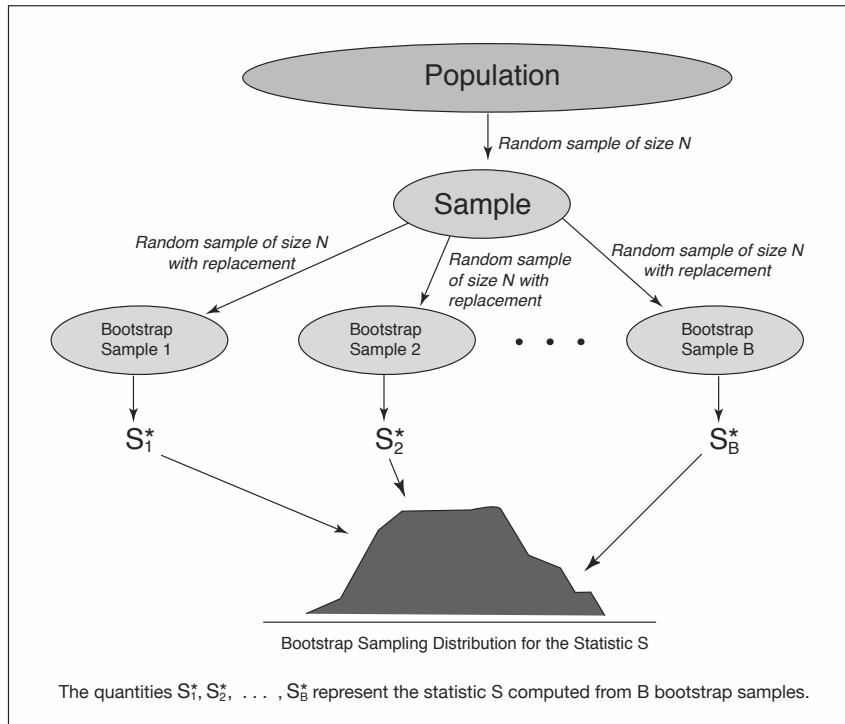


Figure 1.4. Training-and-Test with Bootstrap Resampling



Another variation on the training-and-test regimen is the class of bootstrap methods. If a sample approximates the population from which it was drawn, then a sample from the sample (what is known as a resample) also approximates the population. A bootstrap procedure, as illustrated in figure 1.4, involves repeated resampling with replacement. That is, we take many random samples with replacement from the sample, and for each of these resamples, we compute a statistic of interest. The bootstrap distribution of the statistic approximates the sampling distribution of that statistic. What is the value of the bootstrap? It frees us from having to make assumptions about the population distribution. We can estimate standard errors and make probability statements working from the sample data alone. The bootstrap may also be employed to improve estimates of prediction error within a leave-one-out cross-validation process. Cross-validation and bootstrap methods are reviewed in Davison and Hinkley (1997), Efron and Tibshirani (1993), and Hastie, Tibshirani, and Friedman (2009).

Table 1.1. Data for the Anscombe Quartet

Set I		Set II		Set III		Set IV	
x1	y1	x2	y2	x3	y3	x4	y4
10	8.04	10	9.14	10	7.46	8	6.58
8	6.95	8	8.14	8	6.77	8	5.76
13	7.58	13	8.74	13	12.74	8	7.71
9	8.81	9	8.77	9	7.11	8	8.84
11	8.33	11	9.26	11	7.81	8	8.47
14	9.96	14	8.10	14	8.84	8	7.04
6	7.24	6	6.13	6	6.08	8	5.25
4	4.26	4	3.10	4	5.39	19	12.50
12	10.84	12	9.13	12	8.15	8	5.56
7	4.82	7	7.26	7	6.42	8	7.91
5	5.68	5	4.74	5	5.73	8	6.89

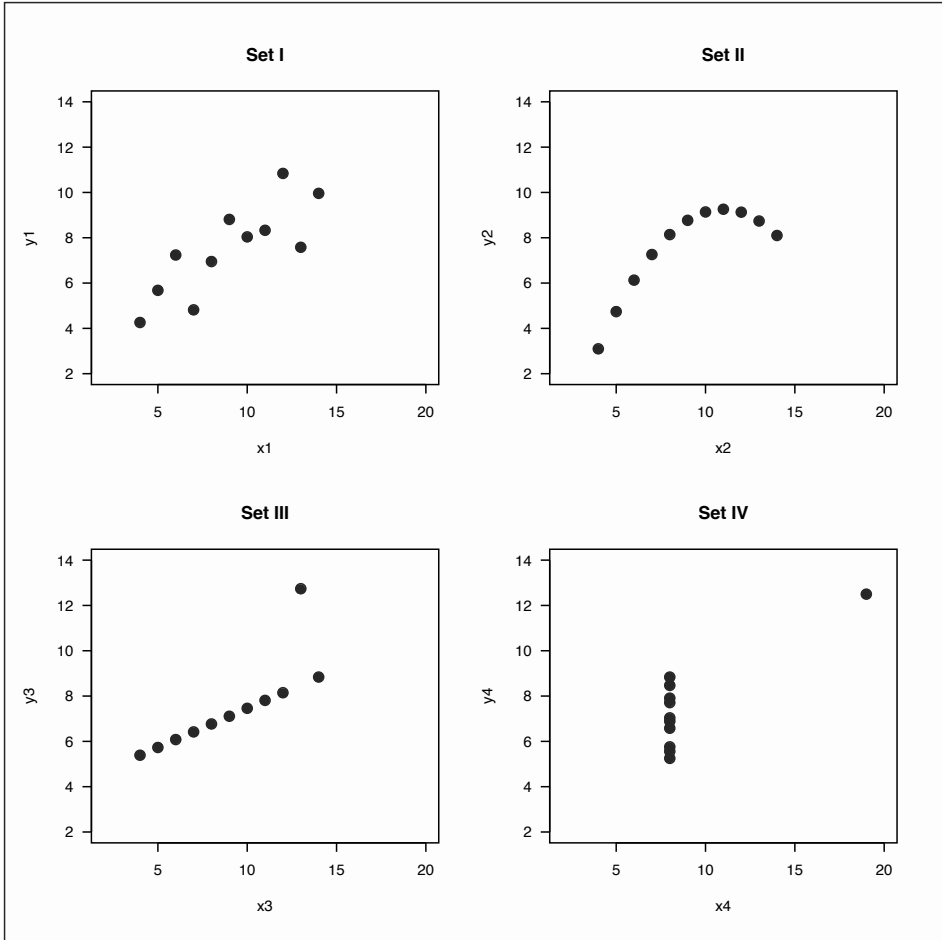
Data visualization is critical to the work of data science. Examples in this book demonstrate the importance of data visualization in discovery, diagnostics, and design. We employ tools of exploratory data analysis (discovery) and statistical modeling (diagnostics). In communicating results to management, we use presentation graphics (design).

There is no more telling demonstration of the importance of statistical graphics and data visualization than a demonstration that is affectionately known as the Anscombe Quartet. Consider the data sets in table 1.1, developed by Anscombe (1973). Looking at these tabulated data, the casual reader will note that the fourth data set is clearly different from the others. What about the first three data sets? Are there obvious differences in patterns of relationship between x and y ?

When we regress y on x for the data sets, we see that the models provide similar statistical summaries. The mean of the response y is 7.5, the mean of the explanatory variable x is 9. The regression analyses for the four data sets are virtually identical. The fitted regression equation for each of the four sets is $\hat{y} = 3 + 0.5x$. The proportion of response variance accounted for is 0.67 for each of the four models.

Following Anscombe (1973), we would argue that statistical summaries fail to tell the story of data. We must look beyond data tables, regression coefficients, and the results of statistical tests. It is the plots in figure 1.5 that tell the story. The four Anscombe data sets are very different from one another.

Figure 1.5. Importance of Data Visualization: The Anscombe Quartet



The Anscombe Quartet shows that we must look at data to understand them. Python and R programs for the Anscombe Quartet are provided at the end of this chapter in exhibits 1.1 and 1.2, respectively.

Visualization tools help us learn from data. We explore data, discover patterns in data, identify groups of observations that go together and unusual observations or outliers. We note relationships among variables, sometimes detecting underlying dimensions in the data.

Graphics for exploratory data analysis are reviewed in classic references by Tukey (1977) and Tukey and Mosteller (1977). Regression graphics are covered by Cook (1998), Cook and Weisberg (1999), and Fox and Weisberg (2011). Statistical graphics and data visualization are illustrated in the works of Tufte (1990, 1997, 2004, 2006), Few (2009), and Yau (2011, 2013). Wilkinson (2005) presents a review of human perception and graphics, as well as a conceptual structure for understanding statistical graphics. Cairo (2013) provides a general review of information graphics. Heer, Bostock, and Ogievetsky (2010) demonstrate contemporary visualization techniques for web distribution. When working with very large data sets, special methods may be needed, such as partial transparency and hexbin plots (Unwin, Theus, and Hofmann 2006; Carr, Lewin-Koh, and Maechler 2014; Lewin-Koh 2014).

Python and R represent rich programming environments for data visualization, including interfaces to visualization applications on the World Wide Web. Chun (2007), Beazley (2009), and Beazley and Jones (2013) review the Python programming environment. Matloff (2011) and Lander (2014) provide useful introductions to R. An R graphics overview is provided by Murrell (2011). R lattice graphics, discussed by Sarkar (2008, 2014), build upon the conceptual structure of an earlier system called S-Plus Trellis™ (Cleveland 1993; Becker and Cleveland 1996). Wilkinson's (2005) "grammar of graphics" approach has been implemented in the Python ggplot package (Lamp 2014) and in the R ggplot2 package (Wickham and Chang 2014), with R programming examples provided by Chang (2013). Cairo (2013) and Zeileis, Hornik, and Murrell (2009, 2014) provide advice about colors for statistical graphics. Ihaka et al. (2014) show how to specify colors in R by hue, chroma, and luminance.

These are the things that data scientists do:

- **Finding out about.** This is the first thing we do—information search, finding what others have done before, learning from the literature. We draw on the work of academics and practitioners in many fields of study, contributors to predictive analytics and data science.
- **Preparing text and data.** Text is unstructured or partially structured. Data are often messy or missing. We extract features from text. We define measures. We prepare text and data for analysis and modeling.
- **Looking at data.** We do exploratory data analysis, data visualization for the purpose of discovery. We look for groups in data. We find outliers. We identify common dimensions, patterns, and trends.
- **Predicting how much.** We are often asked to predict how many units or dollars of product will be sold, the price of financial securities or real estate. Regression techniques are useful for making these predictions.
- **Predicting yes or no.** Many business problems are classification problems. We use classification methods to predict whether or not a person will buy a product, default on a loan, or access a web page.
- **Testing it out.** We examine models with diagnostic graphics. We see how well a model developed on one data set works on other data sets. We employ a training-and-test regimen with data partitioning, cross-validation, or bootstrap methods.
- **Playing what-if.** We manipulate key variables to see what happens to our predictions. We play what-if games in simulated marketplaces. We employ sensitivity or stress testing of mathematical programming models. We see how values of input variables affect outcomes, pay-offs, and predictions. We assess uncertainty about forecasts.
- **Explaining it all.** Data and models help us understand the world. We turn what we have learned into an explanation that others can understand. We present project results in a clear and concise manner. These presentations benefit from well-constructed data visualizations.

Let us begin.

Exhibit 1.1. Programming the Anscombe Quartet (Python)

```
# The Anscombe Quartet (Python)
# demonstration data from
# Anscombe, F. J. 1973, February. Graphs in statistical analysis.
# The American Statistician 27: 1721.

# prepare for Python version 3x features and functions
from __future__ import division, print_function

# import packages for Anscombe Quartet demonstration
import pandas as pd # data frame operations
import numpy as np # arrays and math functions
import statsmodels.api as sm # statistical models (including regression)
import matplotlib.pyplot as plt # 2D plotting

# define the anscombe data frame using dictionary of equal-length lists
anscombe = pd.DataFrame({'x1' : [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
    'x2' : [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
    'x3' : [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
    'x4' : [8, 8, 8, 8, 8, 8, 8, 19, 8, 8, 8],
    'y1' : [8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68],
    'y2' : [9.14, 8.14, 8.74, 8.77, 9.26, 8.1, 6.13, 3.1, 9.13, 7.26, 4.74],
    'y3' : [7.46, 6.77, 12.74, 7.11, 7.81, 8.84, 6.08, 5.39, 8.15, 6.42, 5.73],
    'y4' : [6.58, 5.76, 7.71, 8.84, 8.47, 7.04, 5.25, 12.5, 5.56, 7.91, 6.89]})

# fit linear regression models by ordinary least squares
set_I_design_matrix = sm.add_constant(anscombe['x1'])
set_I_model = sm.OLS(anscombe['y1'], set_I_design_matrix)
print(set_I_model.fit().summary())

set_II_design_matrix = sm.add_constant(anscombe['x2'])
set_II_model = sm.OLS(anscombe['y2'], set_II_design_matrix)
print(set_II_model.fit().summary())

set_III_design_matrix = sm.add_constant(anscombe['x3'])
set_III_model = sm.OLS(anscombe['y3'], set_III_design_matrix)
print(set_III_model.fit().summary())

set_IV_design_matrix = sm.add_constant(anscombe['x4'])
set_IV_model = sm.OLS(anscombe['y4'], set_IV_design_matrix)
print(set_IV_model.fit().summary())

# create scatter plots
fig = plt.figure()
set_I = fig.add_subplot(2, 2, 1)
set_I.scatter(anscombe['x1'], anscombe['y1'])
set_I.set_title('Set I')
set_I.set_xlabel('x1')
set_I.set_ylabel('y1')
set_I.set_xlim(2, 20)
set_I.set_ylim(2, 14)
```



```
set_II = fig.add_subplot(2, 2, 2)
set_II.scatter(anscombe['x2'],anscombe['y2'])
set_II.set_title('Set II')
set_II.set_xlabel('x2')
set_II.set_ylabel('y2')
set_II.set_xlim(2, 20)
set_II.set_ylim(2, 14)

set_III = fig.add_subplot(2, 2, 3)
set_III.scatter(anscombe['x3'],anscombe['y3'])
set_III.set_title('Set III')
set_III.set_xlabel('x3')
set_III.set_ylabel('y3')
set_III.set_xlim(2, 20)
set_III.set_ylim(2, 14)

set_IV = fig.add_subplot(2, 2, 4)
set_IV.scatter(anscombe['x4'],anscombe['y4'])
set_IV.set_title('Set IV')
set_IV.set_xlabel('x4')
set_IV.set_ylabel('y4')
set_IV.set_xlim(2, 20)
set_IV.set_ylim(2, 14)

plt.subplots_adjust(left=0.1, right=0.925, top=0.925, bottom=0.1,
                    wspace = 0.3, hspace = 0.4)
plt.savefig('fig_anscombe_Python.pdf', bbox_inches = 'tight', dpi=None,
            facecolor='w', edgecolor='b', orientation='portrait', papertype=None,
            format=None, transparent=True, pad_inches=0.25, frameon=None)

# Suggestions for the student:
# See if you can develop a quartet of your own,
# or perhaps just a duet, two very different data sets
# with the same fitted model.
```

Exhibit 1.2. Programming the Anscombe Quartet (R)

```
# The Anscombe Quartet (R)

# demonstration data from
# Anscombe, F. J. 1973, February. Graphs in statistical analysis.
# The American Statistician 27: 1721.

# define the anscombe data frame
anscombe <- data.frame(
  x1 = c(10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5),
  x2 = c(10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5),
  x3 = c(10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5),
  x4 = c(8, 8, 8, 8, 8, 8, 8, 19, 8, 8, 8),
  y1 = c(8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68),
  y2 = c(9.14, 8.14, 8.74, 8.77, 9.26, 8.1, 6.13, 3.1, 9.13, 7.26, 4.74),
  y3 = c(7.46, 6.77, 12.74, 7.11, 7.81, 8.84, 6.08, 5.39, 8.15, 6.42, 5.73),
  y4 = c(6.58, 5.76, 7.71, 8.84, 8.47, 7.04, 5.25, 12.5, 5.56, 7.91, 6.89))

# show results from four regression analyses
with(anscombe, print(summary(lm(y1 ~ x1, data = anscombe))))
with(anscombe, print(summary(lm(y2 ~ x2, data = anscombe))))
with(anscombe, print(summary(lm(y3 ~ x3, data = anscombe))))
with(anscombe, print(summary(lm(y4 ~ x4, data = anscombe))))

# place four plots on one page using standard R graphics
# ensuring that all have the same scales
# for horizontal and vertical axes
pdf(file = "fig_anscombe_R.pdf", width = 8.5, height = 8.5)
par(mfrow=c(2,2), mar=c(5.1, 4.1, 4.1, 2.1))
with(anscombe, plot(x1, y1, xlim=c(2,20), ylim=c(2,14), pch = 19,
  col = "darkblue", cex = 1.5, las = 1, xlab = "x1", ylab = "y1"))
title("Set I")
with(anscombe, plot(x2, y2, xlim=c(2,20), ylim=c(2,14), pch = 19,
  col = "darkblue", cex = 1.5, las = 1, xlab = "x2", ylab = "y2"))
title("Set II")
with(anscombe, plot(x3, y3, xlim=c(2,20), ylim=c(2,14), pch = 19,
  col = "darkblue", cex = 1.5, las = 1, xlab = "x3", ylab = "y3"))
title("Set III")
with(anscombe, plot(x4, y4, xlim=c(2,20), ylim=c(2,14), pch = 19,
  col = "darkblue", cex = 1.5, las = 1, xlab = "x4", ylab = "y4"))
title("Set IV")
dev.off()

# par(mfrow=c(1,1),mar=c(5.1, 4.1, 4.1, 2.1)) # return to plotting defaults
```

Index

A

accuracy, *see* classification, predictive accuracy
advertising, 16–33
Akaike information criterion (AIC), 5
Alteryx, 289, 337
ARIMA model, *see* time series analysis
arules, *see* R package, arules
arulesViz, *see* R package, arulesViz
association rule, 46–48, 294

B

bag-of-words approach, *see* text analytics
bar chart, *see* data visualization
base rate, *see* classification, predictive accuracy
Bayes information criterion (BIC), 5
Bayes' theorem, *see* Bayesian statistics, Bayes' theorem
Bayesian statistics, 5, 221, 241, 254, 275, 282, 283, 298
 Bayes' theorem, 283
benchmark study, *see* simulation
best-worst scaling, 308
biclustering, 294
big data, 273, 279
biologically-inspired methods, 290
biplot, *see* data visualization
black box model, 289
block clustering, *see* biclustering
bootstrap method, 8
box plot, *see* data visualization
brand equity research, 239–272
bubble chart, *see* data visualization

C

call center scheduling, *see*
 scheduling, workforce scheduling
car, *see* R package, car
caret, *see* R package, caret

censoring, 214, 315
choice study, 33
 menu-based, 312
classical statistics, 5, 281, 283
 null hypothesis, 281
 power, 282
 statistical significance, 281, 282
classification, 2, 12, 135, 144, 285, 287, 289
 predictive accuracy, 286, 287, 339, 342
classification tree, *see* tree-structured model
cluster, *see* R package, cluster
cluster analysis, 119, 289, 290, 292
coefficient of determination, 285
collaborative filtering, 294
column-oriented database, *see* database
 system, non-relational
complexity, of model, 288
computational linguistics, *see* text analytics,
 natural language processing
confidence interval, *see* classical statistics,
 confidence interval
confusion matrix, *see* classification, predictive
 accuracy
conjoint analysis, 37, 306
content analysis, *see* text analytics, content
 analysis
corpus, *see* text analytics
correlation heat map, *see* data visualization,
 heat map
credit scoring, 300
cross-sectional study, *see* data organization
cross-validation, 6, 288
cutoff rule, *see* classification, predictive
 accuracy
cvTools, *see* R package, cvTools

D

data mining, *see* data-adaptive research
data munging, *see* data preparation

data organization, 5, 66
 data partitioning, 6
 data preparation, 280
 missing data, 280
 data science, 1–12, 277, 278
 data visualization, 8
 bar chart, 47, 49
 biplot, 110
 box plot, 17, 19
 bubble chart, 51
 density plot, 243, 245
 diagnostics, 23, 287
 dot chart, 146, 219
 heat map, 202, 203, 214, 215
 histogram, 106, 195, 198, 200, 201
 horizon plot, 62, 64, 65, 111, 112
 lattice plot, 11, 17, 20, 21, 23
 line graph, 86, 89
 mosaic plot, 241, 242
 multiple time series plot, 63
 network diagram, 53
 parallel coordinates, 246, 248
 ribbon plot, 82–85, 355
 scatter plot, 50
 scatter plot matrix, 214, 216
 spine chart, 35–38, 40, 343
 strip plot, 17, 21
 ternary plot, 241, 243, 244
 time series plot, 67, 68
 tree diagram, 147, 218
 word cloud, 119, 337, 338
 data-adaptive research, 3, 4
 database system, 279, 280
 non-relational, 279, 280
 relational, 279, 280
 density plot, *see* data visualization
 dependent variable, *see* response
 discrete event simulation, *see* simulation,
 discrete event simulation
 document annotation, *see* text analytics,
 document annotation
 document database, *see* database system,
 non-relational
 dot chart, *see* data visualization
 duration analysis, *see* survival analysis

E

e1071, *see* R package, e1071
 economic analysis, 61–80
 indexing, 62
 elimination pick list, 313
 empirical Bayes, 275, *see* Bayesian statistics
 Erlang C, *see* queueing model
 explanatory model, 278

explanatory variable, 2, 3, 285
 exploratory data analysis, 17

F

false negative, *see* classification, predictive
 accuracy
 false positive, *see* classification, predictive
 accuracy
 financial data analysis, 4, 300
 forecast, *see* R package, forecast
 forecasting, 66–69, 218
 four Ps, *see* marketing mix model
 four-fold table, *see* classification, predictive
 accuracy

G

game-day simulation, *see* simulation,
 game-day
 General Inquirer, 148
 generalized linear model, 285, 288
 generative grammar, *see* text analytics
 genetic algorithms, 290
 geographically weighted regression, 218
 ggplot2, *see* R package, ggplot2
 graph database, *see* database system,
 non-relational
 graphics, *see* data visualization
 grid, *see* R package, grid
 group filtering, *see* collaborative filtering

H

heuristics, 290
 hierarchical Bayes, *see* Bayesian statistics
 hierarchical model, 221, 275
 histogram, *see* data visualization
 horizon plot, *see* data visualization

I

IBM, 289, 337
 independent variable, *see* explanatory variable
 integer programming, *see* mathematical
 programming
 interaction effect, 287
 interval estimate, *see* statistic, interval estimate
 item analysis, psychometrics, 143

K

Kappa, *see* classification, predictive accuracy
 key-value store, *see* database system,
 non-relational
 KNIME, 289

L

latent Dirichlet allocation, *see* text analytics, latent Dirichlet allocation

latent semantic analysis, *see* text analytics, latent semantic analysis

lattice, *see* R package, lattice

lattice plot, *see* data visualization

latticeExtra, *see* R package, latticeExtra

leading indicator, 62, 69

least-squares regression, *see* regression

lexical table, *see* text analytics, terms-by-documents matrix

line graph, *see* data visualization

linear least-squares regression, *see* regression

linear model, 285, 288

linear predictor, 285

linguistics, *see* text analytics, natural language processing

lmtest, *see* R package, lmtest

log-linear models, 292

logical empiricism, 1

logistic regression, 3, 143, 285

longitudinal study, *see* data organization

lpSolve, *see* R package, lpSolve

lubridate, *see* R package, lubridate

M

machine learning, 289, 290, *see* data-adaptive research

map-reduce, *see* database system, non-relational

mapproj, *see* R package, mapproj

maps, *see* R package, maps

market basket analysis, 43–60, 294

market response model, 26

market segmentation, *see* segmentation

market simulation, *see* simulation

marketing mix model, 25

Markov chain Monte Carlo, *see* Bayesian statistics, Markov chain Monte Carlo

mathematical programming, 4, 81, 89, 300

integer programming, 88

sensitivity testing, 89

matplotlib, *see* Python package, matplotlib

matrix bubble chart, *see* data visualization, bubble chart

mean-squared error (MSE), *see* root mean-squared error (RMSE)

measurement, 301–314

construct validity, 301

content validity, 149

convergent validity, 302

discriminant validity, 302

face validity, 149

multitrait-multimethod matrix, 301, 303

reliability, 301

meta-analysis, 275

metadata, *see* text analytics

Microsoft, 337

missing data, *see* data preparation, missing data

model validation, *see* training-and-test regimen

model-dependent research, 3, 4

morphology, *see* text analytics

mosaic plot, *see* data visualization

multicollinearity, 212, 214

multidimensional scaling, 107, 109, 119, 292, 295, 296

multilevel models, *see* hierarchical models

multiple imputation, *see* data preparation, missing data

multiple time series plot, *see* data visualization, time series plot

multivariate methods, 119, 295

N

natural language processing, *see* text analytics

natural language toolkit, *see* Python package, nltk

nearest-neighbor model, 220, 221, 294

network diagram, *see* data visualization

neural network, 4

nltk, *see* Python package, nltk

non-relational database, *see* database system, non-relational

NoSQL, *see* database system, non-relational

numpy (NumPy), *see* Python package, numpy

O

operations management, 81–102

optimization, 290

constrained, 88

organization of data, *see* data, organization

os, *see* Python package, os

over-fitting, 214, 220, 287

P

p-value, *see* statistic, p-value

paired comparisons, 307, 310

pandas, *see* Python package, pandas

parallel coordinates plot, *see* data visualization

parametric models, 287

parsing, *see* text analytics, text parsing

patsy, *see* Python package, patsy

perceptual map, *see* data visualization

philosophy, 1

point estimate, *see* statistic, point estimate

- Poisson regression, 284
 - power, *see* classical statistics, power
 - predictive analytics, 1–12
 - definition, 2
 - predictive model, 278
 - predictor, *see* explanatory variable
 - preference scaling, 296
 - preference study, 33
 - pricing research, 239–272
 - principal component analysis, 290, 295
 - privacy, 292
 - probability
 - binomial distribution, 197
 - negative binomial distribution, 197, 199, 202
 - Poisson distribution, 197, 199, 202
 - probability cutoff, *see* classification, predictive accuracy
 - probability heat map, *see* data visualization, heat map
 - probability interval, *see* Bayesian statistics, probability interval
 - process simulation, *see* simulation, process simulation
 - product positioning, 295, 296
 - promotion, 16–33
 - proxy, *see* R package, proxy
 - Python package
 - datetime, 70
 - matplotlib, 13, 27, 70, 120, 151
 - nlTK, 120, 151
 - numpy, 13, 27, 38, 120, 151, 209, 222
 - os, 151
 - pandas, 13, 27, 38, 70, 120, 151, 222
 - patsy, 38, 151
 - re, 120, 151
 - rpy2, 56
 - scipy, 27, 120, 209, 222
 - sklearn, 120, 151, 222
 - statsmodels, 13, 27, 38, 70, 151, 222
- Q**
- quantmod, *see* R package, quantmod
 - queueing, *see* R package, queueing
 - queueing model, 81, 82, 87
- R**
- R package
 - arules, 56, 58
 - arulesViz, 56, 58
 - car, 30
 - caret, 167, 255, 260
 - ChoiceModelR, 255, 260
 - cluster, 127
 - cvTools, 229
 - e1071, 167
 - forecast, 76
 - ggplot2, 91, 96, 127, 167, 260
 - grid, 91, 96, 127, 167
 - lattice, 30, 210, 229, 260
 - latticeExtra, 76, 127, 167
 - lmtest, 76
 - lpSolve, 91, 96
 - lubridate, 76, 91, 96
 - mapproj, 229
 - maps, 229
 - proxy, 127
 - quantmod, 76
 - queueing, 91, 96
 - randomForest, 167, 229
 - RColorBrewer, 56, 58
 - rpart, 167, 229
 - rpart.plot, 167, 229
 - spgwr, 229
 - stringr, 127, 167
 - support.CEs, 40
 - tm, 127, 167
 - vcd, 260
 - wordcloud, 127, 377
 - R-squared, 285
 - random forest, 144–146, 214, 219
 - randomForest, *see* R package, randomForest
 - RColorBrewer, *see* R package, RColorBrewer
 - re, *see* Python package, re
 - recommender systems, 293, 294
 - regression, 2, 3, 12, 22, 24, 25, 143, 214, 217, 284, 288
 - nonlinear regression, 288
 - robust methods, 288
 - time series regression, 66
 - regression tree, *see* tree-structured model
 - regular expressions, *see* Python package, re
 - regularized regression, 288
 - relational database, *see* database system, relational
 - reliability, *see* measurement
 - response, 2, 284
 - ribbon plot, *see* data visualization
 - risk analytics, 300
 - robust methods, *see* regression
 - ROC curve, *see* classification, predictive accuracy
 - root mean-squared error (RMSE), 285
 - rpart, *see* R package, rpart
 - rpart.plot, *see* R package, rpart.plot
 - rpy2, *see* Python package, rpy2
 - RStudio, 337

S

sales forecasting, *see* forecasting

sampling
 sampling variability, 282

SAS, 289, 337

scatter plot, *see* data visualization

scatter plot matrix, *see* data visualization

scheduling, 290
 workforce scheduling, 81–102

scipy (SciPy), *see* Python package, scipy

segmentation, 297, 298

semantics, *see* text analytics

semi-supervised learning, 290

sentiment analysis, 135–187

shrinkage estimators, 288

significance, *see* classical statistics, statistical significance

simulation, 189, 190, 193, 288, 300
 benchmark study, 144, 218, 288, 289
 discrete event simulation, 81, 89, 90
 game-day, 188, 190, 193, 194
 market simulation, 246, 250, 252
 process simulation, 81, 82
 what-if analysis, 12

site selection, 218, *see* spatial data analysis

sklearn (SciKit-Learn), *see* Python package, sklearn

smoothing methods, 288
 splines, 288

social filtering, *see* collaborative filtering

social network analysis, 291, 292

spatial data analysis, 211–238
 site selection, 299
 spatio-temporal model, 212, 221

spatio-temporal model, *see* spatial data analysis, spatio-temporal model

spgwr, *see* R package, spgwr

spine chart, *see* data visualization

sports analytics, 187–211

SQL, *see* database system, relational

state space model, *see* time series analysis

statistic
 interval estimate, 281
 p-value, 281
 point estimate, 281
 test statistic, 281

statistical experiment, *see* simulation

statistical graphics, *see* data visualization

statistical learning, *see* data-adaptive research

statistical significance, *see* classical statistics, statistical significance

statistical simulation, *see* simulation

statsmodels, *see* Python package, statsmodels

stringr, *see* R package, stringr

strip plot, *see* data visualization

supervised learning, 117, 284, 290

support vector machines, 144

support.CEs, *see* R package, support.CEs

survey research, 314

survival analysis, 300

syntax, *see* text analytics

T

tag, *see* text analytics, metadata

target marketing, 297, 298

terms-by-documents matrix, *see* text analytics

ternary plot, *see* data visualization

test statistic, *see* statistic, test statistic

text analytics, 103–134
 bag-of-words approach, 106, 111
 content analysis, 148
 corpus, 107
 document annotation, 314
 generative grammar, 113, 114
 latent Dirichlet allocation, 290
 latent semantic analysis, 290
 metadata, 105
 morphology, 114
 natural language processing, 106, 111, 113, 150
 semantics, 114
 stemming, 115
 syntax, 114
 terms-by-documents matrix, 107, 115, 116
 text feature, 314
 text parsing, 105, 113
 text summarization, 117
 thematic analysis, 148, 290

text feature, *see* text analytics, text feature

text measure, 105, 106, 111, 148, 149, 314, 340

text mining, *see* text analytics

thematic analysis, *see* text analytics, thematic analysis

time series analysis, 61
 ARIMA model, 66
 multiple time series, 63
 state space model, 66

time series plot, *see* data visualization

tm, *see* R package, tm

traditional research, 3

training-and-test regimen, 5, 6, 8, 12, 22, 23, 144, 214, 218, 220, 240

transformation, *see* variable transformation

tree diagram, *see* data visualization

tree-structured model
 classification, 145, 147
 regression, 214, 218

trellis plot, *see* data visualization, lattice plot

tripplot, *see* data visualization, ternary plot

U

unit of analysis, 5

unsupervised learning, 117, 290

V

validation, *see* training-and-test regimen

validity, *see* measurement

variable transformation, 212, 287

vcd, *see* R package, vcd

W

wait-time ribbon, *see* data visualization, ribbon plot

web analytics, 291

Weka, 55

what-if analysis, *see* simulation

wordcloud, *see* R package, wordcloud and data visualization, word cloud