Modeling Techniques in Predictive Analytics with Python and R

A Guide to Data Science

THOMAS W. MILLER
# 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
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Spatial Data Analysis</td>
<td>211</td>
</tr>
<tr>
<td>11</td>
<td>Brand and Price</td>
<td>239</td>
</tr>
<tr>
<td>12</td>
<td>The Big Little Data Game</td>
<td>273</td>
</tr>
<tr>
<td>A</td>
<td>Data Science Methods</td>
<td>277</td>
</tr>
<tr>
<td>A.1</td>
<td>Databases and Data Preparation</td>
<td>279</td>
</tr>
<tr>
<td>A.2</td>
<td>Classical and Bayesian Statistics</td>
<td>281</td>
</tr>
<tr>
<td>A.3</td>
<td>Regression and Classification</td>
<td>284</td>
</tr>
<tr>
<td>A.4</td>
<td>Machine Learning</td>
<td>289</td>
</tr>
<tr>
<td>A.5</td>
<td>Web and Social Network Analysis</td>
<td>291</td>
</tr>
<tr>
<td>A.6</td>
<td>Recommender Systems</td>
<td>293</td>
</tr>
<tr>
<td>A.7</td>
<td>Product Positioning</td>
<td>295</td>
</tr>
<tr>
<td>A.8</td>
<td>Market Segmentation</td>
<td>297</td>
</tr>
<tr>
<td>A.9</td>
<td>Site Selection</td>
<td>299</td>
</tr>
<tr>
<td>A.10</td>
<td>Financial Data Science</td>
<td>300</td>
</tr>
<tr>
<td>B</td>
<td>Measurement</td>
<td>301</td>
</tr>
<tr>
<td>C</td>
<td>Case Studies</td>
<td>315</td>
</tr>
<tr>
<td>C.1</td>
<td>Return of the Bobbleheads</td>
<td>315</td>
</tr>
<tr>
<td>C.2</td>
<td>DriveTime Sedans</td>
<td>316</td>
</tr>
<tr>
<td>C.3</td>
<td>Two Month’s Salary</td>
<td>321</td>
</tr>
<tr>
<td>C.4</td>
<td>Wisconsin Dells</td>
<td>325</td>
</tr>
<tr>
<td>C.5</td>
<td>Computer Choice Study</td>
<td>330</td>
</tr>
<tr>
<td>D</td>
<td>Code and Utilities</td>
<td>335</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>379</td>
</tr>
<tr>
<td></td>
<td>Index</td>
<td>413</td>
</tr>
</tbody>
</table>
“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 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 TeXnology Inc. applied her craft, making words, tables, and figures look beautiful in print—another victory for open source. Thanks to Donald Knuth and the TeX/LaTeX 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
6.5 Call Center Operations for Friday
6.6 Call Center Operations for Sunday
6.7 Call Center Arrival and Service Rates on Wednesdays
6.8 Call Center Needs and Optimal Workforce Schedule
7.1 Movie Taglines from The Internet Movie Database (IMDb)
7.2 Movies by Year of Release
7.3 A Bag of 200 Words from Forty Years of Movie Taglines
7.4 Picture of Text in Time: Forty Years of Movie Taglines
7.5 Text Measures and Documents on a Single Graph
7.6 Horizon Plot of Text Measures across Forty Years of Movie Taglines
7.7 From Text Processing to Text Analytics
7.8 Linguistic Foundations of Text Analytics
7.9 Creating a Terms-by-Documents Matrix
8.1 A Few Movie Reviews According to Tom
8.2 A Few More Movie Reviews According to Tom
8.3 Fifty Words of Sentiment
8.4 List-Based Text Measures for Four Movie Reviews
8.5 Scatter Plot of Text Measures of Positive and Negative Sentiment
8.6 Word Importance in Classifying Movie Reviews as Thumbs-Up or Thumbs-Down
8.7 A Simple Tree Classifier for Thumbs-Up or Thumbs-Down
9.1 Predictive Modeling Framework for Picking a Winning Team
9.2 Game-day Simulation (offense only)
9.3 Mets’ Away and Yankees’ Home Data (offense and defense)
9.4 Balanced Game-day Simulation (offense and defense)
9.5 Actual and Theoretical Runs-scored Distributions
9.6 Poisson Model for Mets vs. Yankees at Yankee Stadium
9.7 Negative Binomial Model for Mets vs. Yankees at Yankee Stadium
9.8 Probability of Home Team Winning (Negative Binomial Model)
10.1 California Housing Data: Correlation Heat Map for the Training Data
10.2 California Housing Data: Scatter Plot Matrix of Selected Variables
10.3 Tree-Structured Regression for Predicting California Housing Values
10.4 Random Forests Regression for Predicting California Housing Values
11.1 Computer Choice Study: A Mosaic of Top Brands and Most Valued Attributes
11.2 Framework for Describing Consumer Preference and Choice
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.3</td>
<td>Ternary Plot of Consumer Preference and Choice</td>
<td>244</td>
</tr>
<tr>
<td>11.4</td>
<td>Comparing Consumers with Differing Brand Preferences</td>
<td>245</td>
</tr>
<tr>
<td>11.5</td>
<td>Potential for Brand Switching: Parallel Coordinates for Individual Consumers</td>
<td>247</td>
</tr>
<tr>
<td>11.6</td>
<td>Potential for Brand Switching: Parallel Coordinates for Consumer Groups</td>
<td>248</td>
</tr>
<tr>
<td>11.7</td>
<td>Market Simulation: A Mosaic of Preference Shares</td>
<td>251</td>
</tr>
<tr>
<td>12.1</td>
<td>Work of Data Science</td>
<td>274</td>
</tr>
<tr>
<td>A.1</td>
<td>Evaluating Predictive Accuracy of a Binary Classifier</td>
<td>286</td>
</tr>
<tr>
<td>B.1</td>
<td>Hypothetical Multitrait-Multimethod Matrix</td>
<td>303</td>
</tr>
<tr>
<td>B.2</td>
<td>Conjoint Degree-of-Interest Rating</td>
<td>306</td>
</tr>
<tr>
<td>B.3</td>
<td>Conjoint Sliding Scale for Profile Pairs</td>
<td>306</td>
</tr>
<tr>
<td>B.4</td>
<td>Paired Comparisons</td>
<td>307</td>
</tr>
<tr>
<td>B.5</td>
<td>Multiple-Rank-Orders</td>
<td>307</td>
</tr>
<tr>
<td>B.6</td>
<td>Best-worst Item Provides Partial Paired Comparisons</td>
<td>308</td>
</tr>
<tr>
<td>B.7</td>
<td>Paired Comparison Choice Task</td>
<td>310</td>
</tr>
<tr>
<td>B.8</td>
<td>Choice Set with Three Product Profiles</td>
<td>310</td>
</tr>
<tr>
<td>B.9</td>
<td>Menu-based Choice Task</td>
<td>312</td>
</tr>
<tr>
<td>B.10</td>
<td>Elimination Pick List</td>
<td>313</td>
</tr>
<tr>
<td>C.1</td>
<td>Computer Choice Study: One Choice Set</td>
<td>332</td>
</tr>
<tr>
<td>D.1</td>
<td>A Python Programmer’s Word Cloud</td>
<td>338</td>
</tr>
<tr>
<td>D.2</td>
<td>An R Programmer’s Word Cloud</td>
<td>338</td>
</tr>
<tr>
<td>Table</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1.1</td>
<td>Data for the Anscombe Quartet</td>
<td>9</td>
</tr>
<tr>
<td>2.1</td>
<td>Bobbleheads and Dodger Dogs</td>
<td>18</td>
</tr>
<tr>
<td>2.2</td>
<td>Regression of Attendance on Month, Day of Week, and Bobblehead Promotion</td>
<td>24</td>
</tr>
<tr>
<td>3.1</td>
<td>Preference Data for Mobile Communication Services</td>
<td>34</td>
</tr>
<tr>
<td>4.1</td>
<td>Market Basket for One Shopping Trip</td>
<td>44</td>
</tr>
<tr>
<td>4.2</td>
<td>Association Rules for a Local Farmer</td>
<td>52</td>
</tr>
<tr>
<td>6.1</td>
<td>Call Center Shifts and Needs for Wednesdays</td>
<td>87</td>
</tr>
<tr>
<td>6.2</td>
<td>Call Center Problem and Solution</td>
<td>88</td>
</tr>
<tr>
<td>8.1</td>
<td>List-Based Sentiment Measures from Tom’s Reviews</td>
<td>140</td>
</tr>
<tr>
<td>8.2</td>
<td>Accuracy of Text Classification for Movie Reviews (Thumbs-Up or Thumbs-Down)</td>
<td>144</td>
</tr>
<tr>
<td>8.3</td>
<td>Random Forest Text Measurement Model Applied to Tom’s Movie Reviews</td>
<td>145</td>
</tr>
<tr>
<td>9.1</td>
<td>New York Mets’ Early Season Games in 2007</td>
<td>191</td>
</tr>
<tr>
<td>9.2</td>
<td>New York Yankees’ Early Season Games in 2007</td>
<td>192</td>
</tr>
<tr>
<td>10.1</td>
<td>California Housing Data: Original and Computed Variables</td>
<td>213</td>
</tr>
<tr>
<td>10.2</td>
<td>Linear Regression Fit to Selected California Block Groups</td>
<td>217</td>
</tr>
<tr>
<td>10.3</td>
<td>Comparison of Regressions on Spatially Referenced Data</td>
<td>220</td>
</tr>
<tr>
<td>11.1</td>
<td>Contingency Table of Top-ranked Brands and Most Valued Attributes</td>
<td>243</td>
</tr>
<tr>
<td>11.2</td>
<td>Market Simulation: Choice Set Input</td>
<td>250</td>
</tr>
<tr>
<td>11.3</td>
<td>Market Simulation: Preference Shares in a Hypothetical Four-brand Market</td>
<td>252</td>
</tr>
<tr>
<td>C.1</td>
<td>Hypothetical profits from model-guided vehicle selection</td>
<td>318</td>
</tr>
<tr>
<td>C.2</td>
<td>DriveTime Data for Sedans</td>
<td>319</td>
</tr>
<tr>
<td>C.3</td>
<td>DriveTime Sedan Color Map with Frequency Counts</td>
<td>320</td>
</tr>
<tr>
<td>C.4</td>
<td>Diamonds Data: Variable Names and Coding Rules</td>
<td>324</td>
</tr>
</tbody>
</table>
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
<table>
<thead>
<tr>
<th></th>
<th>Modeling Techniques in Predictive Analytics with Python and R</th>
</tr>
</thead>
<tbody>
<tr>
<td>D.7</td>
<td>Wait-time Ribbon Plot (R)</td>
</tr>
<tr>
<td>D.8</td>
<td>Movie Tagline Data Preparation Script for Text Analysis (R)</td>
</tr>
<tr>
<td>D.9</td>
<td>Word Scoring Code for Sentiment Analysis (R)</td>
</tr>
<tr>
<td>D.10</td>
<td>Utilities for Spatial Data Analysis (R)</td>
</tr>
<tr>
<td>D.11</td>
<td>Making Word Clouds (R)</td>
</tr>
</tbody>
</table>
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
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.\footnote{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.} 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 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
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.
Randomly divide the sample into folds of approximately equal size:

\[
\begin{array}{cccccc}
A & B & C & D & E \\
\end{array}
\]

Each fold serves once as a test fold:

- **Iteration 1**
  - Test
  - Train
  - Train
  - Train
  - Train

- **Iteration 2**
  - Train
  - Test
  - Train
  - Train
  - Train

- **Iteration 3**
  - Train
  - Train
  - Test
  - Train
  - Train

- **Iteration 4**
  - Train
  - Train
  - Train
  - Test
  - Train

- **Iteration 5**
  - Train
  - Train
  - Train
  - Train
  - Test
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).
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.


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, payoffs, 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.
Chapter 1. Analytics and Data Science

Exhibit 1.1. Programming the Anscombe Quartet (Python)

```python
# The Anscombe Quartet (Python)
# demonstration data from
# 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, 8, 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.
# The Anscombe Quartet (R)

# demonstration data from
# 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, 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

<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy, see classification, predictive accuracy</td>
<td>285, 286, 339, 342</td>
</tr>
<tr>
<td>advertising</td>
<td>16–33</td>
</tr>
<tr>
<td>Akaike information criterion (AIC)</td>
<td>5</td>
</tr>
<tr>
<td>Alteryx</td>
<td>289, 337</td>
</tr>
<tr>
<td>ARIMA model, see time series analysis</td>
<td></td>
</tr>
<tr>
<td>arules, see R package, arules</td>
<td></td>
</tr>
<tr>
<td>arulesViz, see R package, arulesViz</td>
<td></td>
</tr>
<tr>
<td>association rule</td>
<td>46–48, 294</td>
</tr>
</tbody>
</table>

### B

<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>bag-of-words approach, see text analytics</td>
<td></td>
</tr>
<tr>
<td>bar chart, see data visualization</td>
<td></td>
</tr>
<tr>
<td>base rate, see classification, predictive accuracy</td>
<td>285</td>
</tr>
<tr>
<td>Bayes information criterion (BIC)</td>
<td>5</td>
</tr>
<tr>
<td>Bayes' theorem, see Bayesian statistics, Bayes' theorem</td>
<td>283</td>
</tr>
<tr>
<td>Bayesian statistics</td>
<td>5, 221, 241, 254, 275, 282, 283, 298</td>
</tr>
<tr>
<td>Bayes' theorem</td>
<td>283</td>
</tr>
<tr>
<td>benchmark study, see simulation</td>
<td></td>
</tr>
<tr>
<td>best-worst scaling</td>
<td>308</td>
</tr>
<tr>
<td>biclustering</td>
<td>294</td>
</tr>
<tr>
<td>big data</td>
<td>273, 279</td>
</tr>
<tr>
<td>biologically-inspired methods</td>
<td>290</td>
</tr>
<tr>
<td>biplot, see data visualization</td>
<td></td>
</tr>
<tr>
<td>black box model</td>
<td>289</td>
</tr>
<tr>
<td>block clustering, see biclustering</td>
<td></td>
</tr>
<tr>
<td>bootstrap method</td>
<td>8</td>
</tr>
<tr>
<td>box plot, see data visualization</td>
<td></td>
</tr>
<tr>
<td>brand equity research</td>
<td>239–272</td>
</tr>
<tr>
<td>bubble chart, see data visualization</td>
<td></td>
</tr>
</tbody>
</table>

### C

<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>call center scheduling, see</td>
<td></td>
</tr>
<tr>
<td>scheduling, workforce scheduling</td>
<td></td>
</tr>
<tr>
<td>car, see R package, car</td>
<td></td>
</tr>
<tr>
<td>caret, see R package, caret</td>
<td></td>
</tr>
<tr>
<td>censoring</td>
<td>214, 315</td>
</tr>
<tr>
<td>choice study</td>
<td>33</td>
</tr>
<tr>
<td>menu-based</td>
<td>312</td>
</tr>
<tr>
<td>classical statistics</td>
<td>5, 281, 283</td>
</tr>
<tr>
<td>null hypothesis</td>
<td>281</td>
</tr>
<tr>
<td>power</td>
<td>282</td>
</tr>
<tr>
<td>statistical significance</td>
<td>281, 282</td>
</tr>
<tr>
<td>classification, see classification, predictive accuracy</td>
<td>285</td>
</tr>
<tr>
<td>predictive accuracy</td>
<td>286, 287, 339, 342</td>
</tr>
<tr>
<td>classification tree, see tree-structured model</td>
<td></td>
</tr>
<tr>
<td>cluster, see R package, cluster</td>
<td></td>
</tr>
<tr>
<td>cluster analysis</td>
<td>119, 289, 290, 292</td>
</tr>
<tr>
<td>coefficient of determination</td>
<td>285</td>
</tr>
<tr>
<td>collaborative filtering</td>
<td>294</td>
</tr>
<tr>
<td>column-oriented database, see database</td>
<td></td>
</tr>
<tr>
<td>system, non-relational</td>
<td></td>
</tr>
<tr>
<td>complexity, of model</td>
<td>288</td>
</tr>
<tr>
<td>computational linguistics, see text analytics, natural language processing</td>
<td></td>
</tr>
<tr>
<td>correlation heat map, see data visualization, heat map</td>
<td></td>
</tr>
<tr>
<td>credit scoring</td>
<td>300</td>
</tr>
<tr>
<td>cross-sectional study, see data organization</td>
<td></td>
</tr>
<tr>
<td>cross-validation</td>
<td>6, 288</td>
</tr>
<tr>
<td>cutoff rule, see classification, predictive accuracy</td>
<td>285</td>
</tr>
<tr>
<td>cvTools, see R package, cvTools</td>
<td></td>
</tr>
</tbody>
</table>

### D

<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>data mining, see data-adaptive research</td>
<td></td>
</tr>
<tr>
<td>data munging, see data preparation</td>
<td></td>
</tr>
</tbody>
</table>

413
<table>
<thead>
<tr>
<th>Index</th>
<th>415</th>
</tr>
</thead>
</table>

**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, 89
- 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, rpart.plot
RStudio, 337
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

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
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
syntactic, 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, see text analytics, thematic analysis
time series plot, see data visualization
tm, see R package, tm
time series plot, see data visualization
training-and-test regimen, 5, 6, 8, 12, 22, 23, 144, 214, 218, 220, 240
transformation, see variable transformation
tree diagram, see variable transformation
tree-structured model
classification, 145, 147
regression, 214, 218
trellis plot, see data visualization, lattice plot
U
unit of analysis, 5
unsupervised learning, 117, 290

V
validation, see training-and-test regimen
validity, see measurement
variable transformation, 212, 287

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