MODERN BUSINESS ANALYTICS

Practical Data Science for Decision Making

Matt TADDY   Leslie HENDRIX   Matthew HARDING

Mc Graw Hill

MODERN BUSINESS ANALYTICS

Mc Graw Hill

MODERN BUSINESS ANALYTICS

Practical Data Science for Decision Making

Matt Taddy
Amazon, Inc.

Leslie Hendrix
University of South Carolina

Matthew C. Harding
University of California, Irvine

Mc
Graw
Hill

MODERN BUSINESS ANALYTICS

Published by McGraw Hill LLC, 1325 Avenue of the Americas, New York, NY 10019.

Some ancillaries, including electronic and print components, may not be available to customers outside the United States.

This book is printed on acid-free paper.

1 2 3 4 5 6 7 8 9 LKV 27 26 25 24 23 22

ISBN 978-1-266-10833-4
MHID 1-266-10833-5

Cover Image: *MirageC/Getty Images*

The Internet addresses listed in the text were accurate at the time of publication. The inclusion of a website does not indicate an endorsement by the authors or McGraw Hill Education, and McGraw Hill Education does not guarantee the accuracy of the information presented at these sites.

mheducation.com/highered

ABOUT THE AUTHORS

Courtesy of Matt Taddy

Matt Taddy is the author of *Business Data Science* (McGraw Hill, 2019). From 2008–2018 he was a professor of econometrics and statistics at the University of Chicago Booth School of Business, where he developed their Data Science curriculum. Prior to and while at Chicago Booth, he has also worked in a variety of industry positions including as a principal researcher at Microsoft and a research fellow at eBay. He left Chicago in 2018 to join Amazon as a vice president.

Courtesy of Leslie Hendrix

Leslie Hendrix is a clinical associate professor in the Darla Moore School of Business at the University of South Carolina. She received her PhD in statistics in 2011 and a BS in mathematics in 2005 from the University of South Carolina. She has received two university-wide teaching awards for her work in teaching business analytics and statistics courses and is active in the research and teaching communities for analytics. She was instrumental in founding the Moore School's newly formed Data Lab and currently serves as the assistant director.

Courtesy of Matthew C. Harding

Matthew C. Harding is a professor of economics and statistics at the University of California, Irvine. He holds a PhD from MIT and an M.Phil. from Oxford University. Dr. Harding conducts research on econometrics, consumer finance, health policy, and energy economics and has published widely in leading academic journals. He is the founder of Ecometricx, LLC, a big data and machine learning consulting company, and cofounder of FASTlab.global Institute, a nonprofit focusing on education and evidence-based policies in the areas of fair access and sustainable technologies.

BRIEF CONTENTS

CONTENTS

PREFACE

What Is This Book About?

The practice of data analytics is changing and modernizing. Innovations in computation and machine learning are creating new opportunities for the data analyst: exposing previously unexplored data to scientific analysis, scaling tasks through automation, and allowing deeper and more accurate modeling. Spreadsheet models and pivot tables are being replaced by code scripts in languages like R and Python. There has been massive growth in digitized information, accompanied by development of systems for storage and analysis of this data. There has also been an intellectual convergence across fields—machine learning and computer science, statistics, and social sciences and economics—that has raised the breadth and quality of applied analysis everywhere. This is the *data science approach to analytics*, and it allows leaders to go deeper than ever to understand their operations, products, and customers.

This book is a primer for those who want to gain the skills to use data science to help make decisions in business and beyond. The modern business analyst uses tools from machine learning, economics, and statistics to not only track what has happened but predict the future for their businesses. Analysts may need to identify the variables important for business policy, run an experiment to measure these variables, and mine social media for information about public response to policy changes. A company might seek to connect small changes in a recommendation system to changes in customer experience and use this information to estimate a demand curve. And any analysis will need to scale to companywide data, be repeatable in the future, and quantify uncertainty about the model estimates and conclusions.

This book focuses on business and economic applications, and we expect that our core audience will be looking to apply these tools as data scientists and analysts inside companies. But we also cover examples from health care and other domains, and the practical material that you learn in this book applies far beyond any narrow set of business problems.

This is not a book about *one of* machine learning, economics, or statistics. Rather, this book pulls from all of these fields to build a toolset for modern business analytics. The material in this book is designed to be useful for *decision making*. Detecting patterns in past data can be useful—we will cover a number of pattern recognition topics—but the necessary analysis for deeper business problems is about *why* things happen rather than what has happened. For this reason, this book will spend the time to move beyond correlation to causal analysis. This material is closer to economics than to the mainstream of data science, which should help you have a bigger practical impact through your work.

We can't cover everything here. This is not an encyclopedia of data analysis. Indeed, for continuing study, there are a number of excellent books covering different areas of contemporary machine learning and data science. For example, Hastie et al. (2009) is a comprehensive modern statistics reference and James et al. (2021) is a less advanced text from a similar viewpoint. You can view this current text as a stepping stone to a career of continued exploration and learning in statistics and machine learning. We want you to leave with a set of best practices that make you confident in what to trust, how to use it, and how to learn more.

GUIDED TOUR

This book is based on the *Business Data Science* text by Taddy (2019), which was itself developed as part of the MBA data science curriculum at the University of Chicago Booth School of Business. This new adaptation creates a more accessible and course-ready textbook, and includes a major expansion of the examples and content (plus an appendix tutorial on computing with R). Visit Connect for digital assignments, code, datasets, and additional resources.

Practical Data Science for Decision Making

Our target readership is anyone who wants to get the skills to use modern large-scale data to make decisions, whether they are working in business, government, science, or anywhere else.

In the past 10 years, we've observed the growth of a class of generalists who can understand business problems and also dive into the (big) data and run their own analyses. There is a massive demand for people with these capabilities, and this book is our attempt to help grow more of these sorts of people. You may be reading this book from a quantitative undergraduate course, as part of your MBA degree at a business school, or in a data science or other graduate program. Or, you may just be reading the book on your own accord. As data analysis has become more crucial and exciting, we are seeing a boom in people switching into data analysis careers from a wide variety of backgrounds. Those self-learners and career-switchers are as much our audience here as students in a classroom.

All of this said, this is not an *easy* book. We have tried to avoid explanations that require calculus or advanced linear algebra, but you will find the book a tough slog if you do not have a solid foundation in first-year mathematics and probability. Since the book includes a breadth of material that spans a range of complexity, we begin each chapter with a summary that outlines each section and indicates their difficulty according to a *ski-hill* scale:

● The easiest material, requiring familiarity with some transformations like logarithms and exponents, and an understanding of the basics of probability.

■ Moderately difficult material, involving more advanced ideas from probability and statistics or ideas that are going to be difficult to intuit without some linear algebra.

◆ The really tough stuff, involving more complex modeling ideas (and notation) and tools from linear algebra and optimization.

The black diamond material is not necessary for understanding future green or blue sections, and so instructors may wish to set their courses to cover the easy and moderately difficult sections while selecting topics from the hardest sections.

The book is designed to be self-contained, such that you can start with little prerequisite background in data science and learn as you go. However, the pace of content on introductory probability and statistics and regression is such that you may struggle if this is your first-ever course on these ideas. If you find this to be the case, we recommend spending some time working through a dedicated introductory statistics book to build some of this understanding before diving into the more advanced data science material.

It is also important to recognize that data science can be learned only by doing. This means writing the code to run analysis routines on really messy data. We will use the R scripting language for all of our examples. All example code and data is available online, and one of the most important skills you will get out of this book will be an advanced education in this powerful and widely used statistical software. For those who are completely new to R, we have also included an extensive R primer. The skills you learn here will also prepare you well for learning how to program in other languages, such as Python, which you will likely encounter in your business analysis career.

This is a book about how to *do* modern business analytics. We will lay out a set of core principles and best practices that come from statistics, machine learning, and economics. You will be working through many real data analysis examples as you learn by doing. It is a book designed to prepare scientists, engineers, and business professionals to use data science to improve their decisions.

An Introductory Example

Before diving into the core material, we will work through a simple finance example to illustrate the difference between data processing or description and a deeper *business analysis.* Consider the graph in Figure 0.1. This shows seven years of monthly returns for stocks in the S&P 500 index (a return is the difference between the current and previous price divided by the prior value). Each line ranging from bright yellow to dark red denotes an individual stock's return series. Their weighted average—the value of the S&P 500—is marked with a bold line. Returns on three-month U.S. treasury bills are in gray.

This is a fancy plot. It looks cool, with lots of different lines. It is the sort of plot that you might see on a computer screen in a TV ad for some online brokerage platform. *If only I had that information, I'd be rich!*

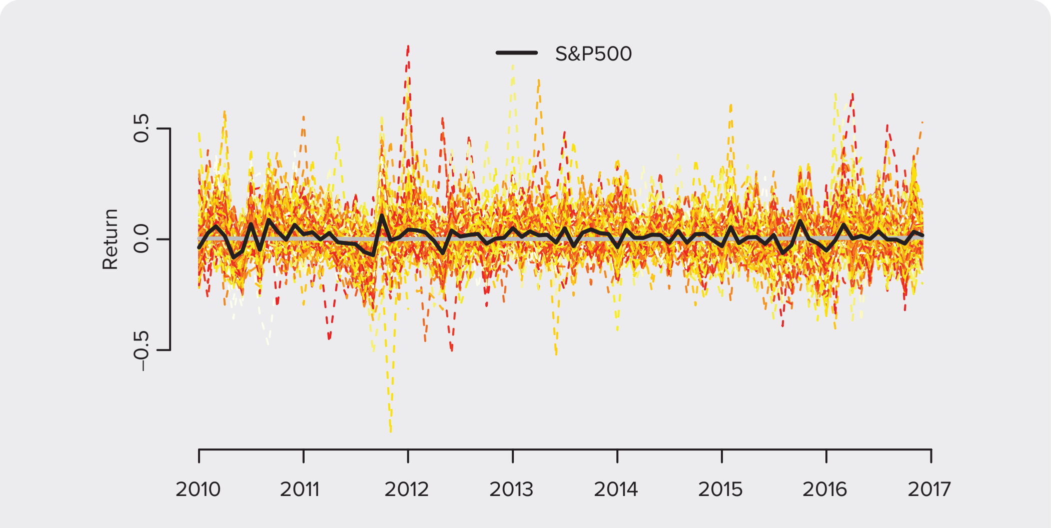


FIGURE 0.1 A fancy plot: monthly stock returns for members of the S&P 500 and their average (the bold line). *What can you learn?*

But what can you actually learn from Figure 0.1? You can see that returns do tend to bounce around near zero (although the long-term average is reliably much greater than zero). You can also pick out periods of higher volatility (variance) where the S&P 500 changes more from month to month and the individual stock returns around it are more dispersed. That's about it. You don't learn *why* these periods are more volatile or when they will occur in the future. More important, you can't pull out useful information about any individual stock. There is a ton of *data* on the graph but little useful information.

Instead of plotting raw data, let's consider a simple *market model* that relates individual stock returns to the market average. The capital asset pricing model (CAPM) regresses the returns of an individual asset onto a measure of overall market returns, as shown here:

$$r_{jt} = \alpha_j + \beta_j m_t + \varepsilon_{jt} \tag{0.1}$$

The *output* r_{jt} is equity j return at time t. The *input* m_t is a measure of the average return—the "market"—at time t. We take m_t as the return on the S&P 500 index that weights 500 large companies according to their market capitalization (the total value of their stock). Finally, ε_{jt} is an *error* that has mean zero and is uncorrelated with the market.

Equation (0.1) is the first regression model in this book. You'll see many more. This is a simple linear regression that should be familiar to most readers. The Greek letters define a line relating each individual equity return to the market, as shown in Figure 0.2. A small β_j, near zero, indicates an asset with low market sensitivity. In the extreme, fixed-income assets like treasury bills have $\beta_j = 0$. On the other hand, a $\beta_j > 1$ indicates a stock that is more volatile than the market, typically meaning growth and higher-risk stocks. The α_j is free money: assets with $\alpha_j > 0$ are adding value regardless of wider market movements, and those with $\alpha_j < 0$ destroy value.

Figure 0.3 represents each stock "ticker" in the two-dimensional space implied by the market model's fit on the seven years of data in Figure 0.1. The tickers are sized proportional to each firm's market capitalization. The two CAPM parameters—[α, β]—tell you a huge amount about the behavior and performance of individual assets. This picture immediately allows you to assess market sensitivity and arbitrage opportunities. For example, the big tech stocks of Facebook (FB), Amazon (AMZN), Apple (AAPL), Microsoft (MSFT), and Google (GOOGL) all have market sensitivity β values close to one. However, Facebook, Amazon, and Apple generated more money independent of the market over this time period compared to Microsoft and Google (which have nearly identical α values and are overlapped on the plot). Note

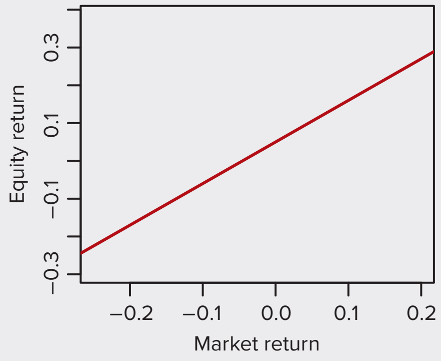


FIGURE 0.2 A *scatterplot* of a single stock's returns against market returns, with the fitted *regression line* for the model of Equation (0.1) shown in red.

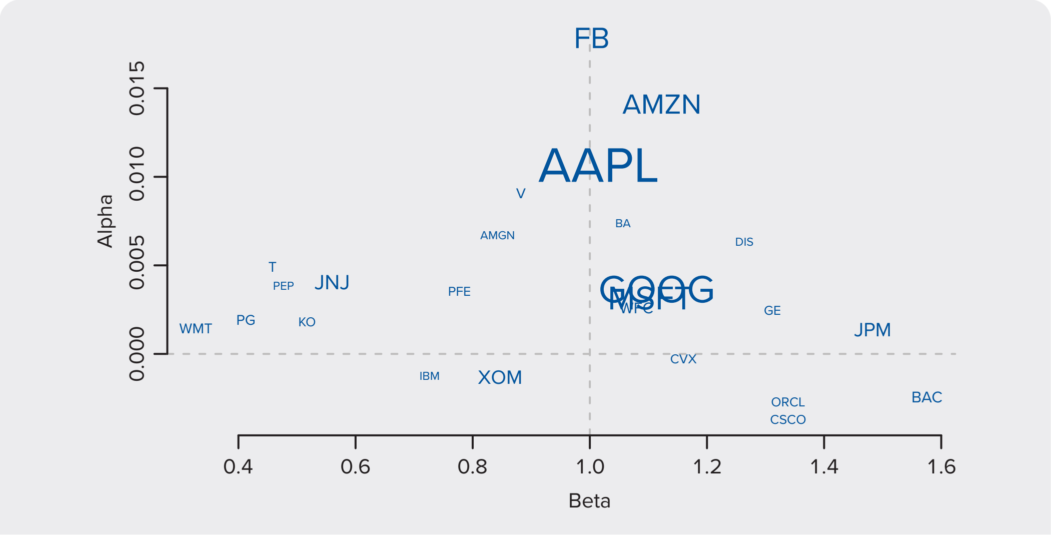


FIGURE 0.3 Stocks positioned according to their fitted market model, where α is money you make regardless of what the market does and β summarizes sensitivity to market movements. The tickers are sized proportional to market capitalization. Production change alpha to α and beta to β in the plot axis labels.

that Facebook's CAPM parameters are estimated from a shorter time period, since it did not have its IPO until May of 2012. Some of the older technology firms, such as Oracle (ORCL), Cisco (CSCO), and IBM, appear to have destroyed value over this period (negative alpha). Such information can be used to build portfolios that maximize mean returns and minimize variance in the face of uncertain future market conditions. It can also be used in strategies like pairs-trading where you find two stocks with similar betas and buy the higher alpha while "shorting" the other.

CAPM is an old tool in financial analysis, but it serves as a great illustration of what to strive toward in practical data science. An interpretable model translates raw data into information that is directly relevant to decision making. The challenge in data science is that the data you'll be working with will be larger and less structured (e.g., it will include text and image data). Moreover, CAPM is derived from assumptions of efficient market theory, and in many applications you won't have such a convenient simplifying framework on hand. But the basic principles remain the same: you want to turn raw data into useful information that has direct relevance to business policy.

Machine Learning

Machine learning (ML) is the field of using algorithms to automatically detect and predict patterns in complex data. The rise of machine learning is a major driver behind data science and a big part of what differentiates today's analyses from those of the past. ML is closely related to modern statistics, and indeed many of the best ideas in ML have come from statisticians. But whereas statisticians have often focused on *model inference*—on understanding the parameters of their models (e.g., testing on individual coefficients in a regression)—the ML community has historically been more focused on the single goal of maximizing *predictive performance* (i.e., predicting future values of some response of interest, like sales or prices).

A focus on prediction tasks has allowed ML to quickly push forward and work with larger and more complex data. If all you care about is predictive performance, then you don't need to worry about whether your model is "true" but rather just test how well it performs when predicting future values. This single-minded focus allows rapid experimentation on alternative models and estimation algorithms. The result is that ML has seen massive success, to the point that you can now expect to have available for almost any type of data an algorithm that will work out of the box to recognize patterns and give high-quality predictions.

However, this focus on prediction means that ML on its own is less useful for many *decision-making* tasks. ML algorithms learn to predict *a future that is mostly like the past.* Suppose that you build an ML algorithm that looks at how customer web browser history predicts how much they spend in your e-commerce store. A purely prediction-focused algorithm will discern what web traffic tends to spend more or less money. It will not tell you what will happen to the spending if you *change* a group of those websites (or your prices) or perhaps make it easier for people to browse the Web (e.g., by subsidizing broadband). That is where this book comes in: we will use tools from economics and statistics in combination with ML techniques to create a platform for using data to make decisions.

Some of the material in this book will be focused on pure ML tasks like prediction and pattern recognition. This is especially true in the earlier chapters on regression, classification, and regularization. However, in later chapters you will use these prediction tools as parts of more structured analyses, such as understanding subject-specific treatment effects, fitting consumer demand functions, or as part of an artificial intelligence system. This typically involves a mix of domain knowledge and analysis tools, which is what makes the data scientist such a powerful figure. The ML tools are useless for policy making without an understanding of the business problems, but a policy maker who can deploy ML as part of their analysis toolkit will be able to make better decisions faster.

Computing with `R`

You don't need to be a software engineer to work as a data scientist, but you need to be able to write and understand computer code. To learn from this book, you will need to be able to read and write in a high-level *scripting* language, in other words, flexible code that can be used to describe recipes for data analysis. In particular, you will need to have a familiarity with `R` (`r-project.org`).

The ability to interact with computers in this way—by typing commands rather than clicking buttons or choosing from a menu—is a basic data analysis skill. Having a script of commands allows you to rerun your analyses for new data without any additional work. It also allows you to make small changes to existing scripts to adapt them for new scenarios. Indeed, making small changes is how we recommend you work with the material in this book. The code for every in-text example is available on-line, and you can alter and extend these scripts to suit your data analysis needs. In the examples for this book, all of the analysis will be conducted in `R`. This is an open-source high-level language for data analysis. `R` is used widely throughout industry, government, and academia. Companies like RStudio sell enterprise products built around `R`. This is not a toy language used simply for teaching purposes—`R` is the real industrial-strength deal.

For the fundamentals of statistical analysis, `R` is tough to beat: all of the tools you need for linear modeling and uncertainty quantification are mainstays. `R` is also relatively forgiving for the novice programmer. A major strength of R is its ecosystem of contributed packages. These are add-ons that increase the capability of core R. For example, almost all of the ML tools that you will use in this book are available via packages. The quality of the packages is more varied than it is for R's core functionality, but if a package has high usage you should be confident that it works as intended.

The Appendix of this book contains a tutorial that is dedicated to getting you started in R. It focuses on the topics and algorithms that are used in the examples in this book. You don't need to be an expert in R to learn from this book; you just need to be able to understand the fundamentals and be willing to mess around with the coded examples. If you have no formal background in coding, worry not: many in the field started out in this position. The learning curve can be steep initially, but once you get the hang of it, the rest will come fast. The tutorial in the Appendix should help you get started. We also provide extensive examples throughout the book, and all code, data, and homework assignments are available through Connect. Every chapter ends with a *Quick Reference* section containing the basic R recipes from that chapter. When you are ready to learn more, there are many great places where you can supplement your understanding of the basics of R. If you simply search for *R* or *R statistics* books on-line, you will find a huge variety of learning resources.

ACKNOWLEDGMENTS

We are grateful for the reviewers who provided feedback on this first edition:

Sue Abdinnour, Wichita State University
Anil Aggarwal, University of Baltimore
Goutam Chakraborty, Oklahoma State University
Rick Cleary, Babson College
John Daniels, Central Michigan University
John Draper, Ohio State University
Janet Frasier, West Virginia University
Phillip C. Fry, Boise State University
Marina Girju, California Baptist University
Richard Hauser, East Carolina University
Kuo-Ting “Ken” Hung, Suffolk University
Aimee Jacobs, California State University, Fresno
Jaehwan Jeong, Radford University
Patrick Johanns, Univeristy of Iowa
Barry King, Butler University
Lauren Kleitz, Xavier University
Su “Sue” Kong, Kutztown University
Min Li, California State University, Sacramento
Jiajuan Liang, University of New Haven
Vic Matta, Ohio University
Ebrahim Mortaz, Pace University
Bob Myers, Georgia Tech University
Robert Nauss, University of Missouri-St. Louis
Arash Negahban, California State University, Chico
Yasin Ozcelik, Fairfield University
Brad Price, West Virginia University
Xingye Qiao, Binghamton University
Roman Rabinovich, Boston University
Bharatendra Rai, UMass-Dartmouth
Balaraman Rajan, California State University, East Bay
Rouzbeh Razavi, Kent State University
John Repede, Queens University of Charlotte
Thomas R. Robbins, East Carolina University
Wendy Swenson Roth, Georgia State University
Seokwoo Song, Weber State University
John R. Stevens, Utah State University
Jie Tao, Fairfield University
Vicar S. Valencia, Indiana University South Bend
Nikhil Varaiya, San Diego State University
Gang Wang, UMass-Dartmouth
K. Matthew Wong, St. John’s University
Chase Wu, New Jersey Institute of Technology
Yajiong “Lucky” Xue, East Carolina University

McGraw Hill connect®

Instructors: Student Success Starts with You

Tools to enhance your unique voice

Want to build your own course? No problem. Prefer to use an OLC-aligned, prebuilt course? Easy. Want to make changes throughout the semester? Sure. And you'll save time with Connect's auto-grading too.

65%
Less Time Grading

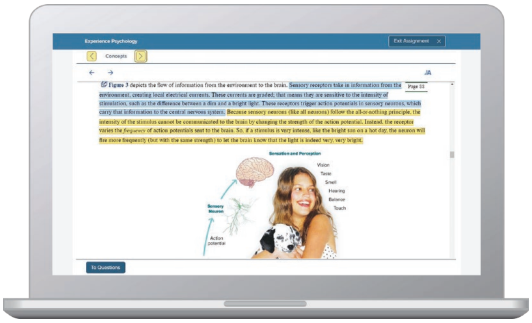


Figure 3 depicts the flow of information from the environment to the brain. Sensory receptors take in information from the environment, creating local electrical currents. These currents are graded; that means they are sensitive to the intensity of stimulation, such as the difference between a dim and a bright light. These receptors trigger action potentials in sensory neurons, which carry that information to the central nervous system. Because sensory neurons (like all neurons) follow the all-or-nothing principle, the intensity of the stimulus cannot be communicated to the brain by changing the strength of the action potential. Instead, the receptor varies the *frequency* of action potentials sent to the brain. So, if a stimulus is very intense, like the bright sun on a hot day, the neuron will fire more frequently (but with the same strength) to let the brain know that the light is indeed very, very bright.

Laptop: McGraw Hill; Woman/dog: George Doyle/Getty Images

Study made personal

Incorporate adaptive study resources like SmartBook® 2.0 into your course and help your students be better prepared in less time. Learn more about the powerful personalized learning experience available in SmartBook 2.0 at **www.mheducation.com/highered/connect/smartbook**

Affordable solutions, added value

Make technology work for you with LMS integration for single sign-on access, mobile access to the digital textbook, and reports to quickly show you how each of your students is doing. And with our Inclusive Access program you can provide all these tools at a discount to your students. Ask your McGraw Hill representative for more information.

Padlock: Jobalou/Getty Images

Solutions for your challenges

A product isn't a solution. Real solutions are affordable, reliable, and come with training and ongoing support when you need it and how you want it. Visit **www.supportateverystep.com** for videos and resources both you and your students can use throughout the semester.

Checkmark: Jobalou/Getty Images

SUPPORT AT every step

Students: Get Learning that Fits You

Effective tools for efficient studying

Connect is designed to help you be more productive with simple, flexible, intuitive tools that maximize your study time and meet your individual learning needs. Get learning that works for you with Connect.

Study anytime, anywhere

Download the free ReadAnywhere app and access your online eBook, SmartBook 2.0, or Adaptive Learning Assignments when it's convenient, even if you're offline. And since the app automatically syncs with your Connect account, all of your work is available every time you open it. Find out more at **www.mheducation.com/readanywhere**

> ***"I really liked this app—it made it easy to study when you don't have your textbook in front of you."***
>
> \- Jordan Cunningham,
> Eastern Washington University

Calendar: owattaphotos/Getty Images

Everything you need in one place

Your Connect course has everything you need—whether reading on your digital eBook or completing assignments for class, Connect makes it easy to get your work done.

Learning for everyone

McGraw Hill works directly with Accessibility Services Departments and faculty to meet the learning needs of all students. Please contact your Accessibility Services Office and ask them to email accessibility@mheducation.com, or visit **www.mheducation.com/about/accessibility** for more information.

Top: Jenner Images/Getty Images, Left: Hero Images/Getty Images, Right: Hero Images/Getty Images

Proctorio
Remote Proctoring & Browser-Locking Capabilities

connect + proctorio

Remote proctoring and browser-locking capabilities, hosted by Proctorio within Connect, provide control of the assessment environment by enabling security options and verifying the identity of the student.

Seamlessly integrated within Connect, these services allow instructors to control students' assessment experience by restricting browser activity, recording students' activity, and verifying students are doing their own work.

Instant and detailed reporting gives instructors an at-a-glance view of potential academic integrity concerns, thereby avoiding personal bias and supporting evidence-based claims.

ReadAnywhere

Read or study when it's convenient for you with McGraw Hill's free ReadAnywhere app. Available for iOS or Android smartphones or tablets, ReadAnywhere gives users access to McGraw Hill tools including the eBook and SmartBook 2.0 or Adaptive Learning Assignments in Connect. Take notes, highlight, and complete assignments offline–all of your work will sync when you open the app with WiFi access. Log in with your McGraw Hill Connect username and password to start learning–anytime, anywhere!

OLC-Aligned Courses

Implementing High-Quality Online Instruction and Assessment through Preconfigured Courseware

In consultation with the Online Learning Consortium (OLC) and our certified Faculty Consultants, McGraw Hill has created pre-configured courseware using OLC's quality scorecard to align with best practices in online course delivery. This turnkey courseware contains a combination of formative assessments, summative assessments, homework, and application activities, and can easily be customized to meet an individual's needs and course outcomes. For more information, visit https://www.mheducation.com/highered/olc.

Tegrity: Lectures 24/7

Tegrity in Connect is a tool that makes class time available 24/7 by automatically capturing every lecture. With a simple one-click start-and-stop process, you capture all computer screens and corresponding audio in a format that is easy to search, frame by frame. Students can replay any part of any class with easy-to-use, browser-based viewing on a PC, Mac, iPod, or other mobile device.

Educators know that the more students can see, hear, and experience class resources, the better they learn. In fact, studies prove it. Tegrity's unique search feature helps students efficiently find what they need, when they need it, across an entire semester of class recordings. Help turn your students' study time into learning moments immediately supported by your lecture. With Tegrity, you also increase intent listening and class participation by easing students' concerns about note-taking. Using Tegrity in Connect will make it more likely you will see students' faces, not the tops of their heads.

Test Builder in Connect

Available within Connect, Test Builder is a cloud-based tool that enables instructors to format tests that can be printed, administered within a Learning Management System, or exported as a Word document of the test bank. Test Builder offers a modern, streamlined interface for easy content configuration that matches course needs, without requiring a download.

Test Builder allows you to:

- access all test bank content from a particular title.
- easily pinpoint the most relevant content through robust filtering options.
- manipulate the order of questions or scramble questions and/or answers.
- pin questions to a specific location within a test.
- determine your preferred treatment of algorithmic questions.
- choose the layout and spacing.
- add instructions and configure default settings.

Test Builder provides a secure interface for better protection of content and allows for just-in-time updates to flow directly into assessments.

Writing Assignment

Available within Connect and Connect Master, the Writing Assignment tool delivers a learning experience to help students improve their written communication skills and conceptual understanding. As an instructor you can assign, monitor, grade, and provide feedback on writing more efficiently and effectively.

Application-Based Activities in Connect

Application-Based Activities in Connect are highly interactive, assignable exercises that provide students a safe space to apply the concepts they have learned to real-world, course-specific problems. Each Application-Based Activity involves the application of multiple concepts, allowing students to synthesize information and use critical thinking skills to solve realistic scenarios.

Mc Graw Hill create® Your Book, Your Way

McGraw Hill's Content Collections Powered by Create® is a self-service website that enables instructors to create custom course materials—print and eBooks—by drawing upon

McGraw Hill's comprehensive, cross-disciplinary content. Choose what you want from our high-quality textbooks, articles, and cases. Combine it with your own content quickly and easily, and tap into other rights-secured, third-party content such as readings, cases, and articles. Content can be arranged in a way that makes the most sense for your course and you can include the course name and information as well. Choose the best format for your course: color print, black-and-white print, or eBook. The eBook can be included in your Connect course and is available on the free ReadAnywhere app for smartphone or tablet access as well. When you are finished customizing, you will receive a free digital copy to review in just minutes! Visit McGraw Hill Create®—www.mcgrawhillcreate.com—today and begin building!

1 REGRESSION

This chapter develops the framework and language of regression: building models that predict response outputs from feature inputs.

- ● **Section 1.1 Linear Regression:** Specify, estimate, and predict from a linear regression model for a quantitative response y as a function of inputs $\mathbf{x}$. Use log transforms to model multiplicative relationships and *elasticities*, and use interactions to allow the effect of inputs to depend on each other.
- ● **Section 1.2 Residuals:** Calculate the residual errors for your regression fit, and understand the key fit statistics *deviance*, R^2, and *degrees of freedom.*
- ■ **Section 1.3 Logistic Regression:** Build logistic regression models for a binary response variable, and understand how logistic regression is related to linear regression as a *generalized linear model.* Translate the concepts of deviance, likelihood, and R^2 to logistic regression, and be able to interpret logistic regression coefficients as effects on the log odds that $y = 1$.
- ■ **Section 1.4 Likelihood and Deviance:** Relate likelihood maximization and deviance minimization, use the generalized linear models to determine residual deviance, and use the `predict` function to integrate new data with the same variable names as the data used to fit your regression.
- ■ **Section 1.5 Time Series:** Adapt your regression models to allow for dependencies in data that has been observed over time, and understand time series concepts including seasonal trends, autoregression, and panel data.
- ◆ **Section 1.6 Spatial Data:** Add spatial fixed effects to your regression models and use Gaussian process models to estimate spatial dependence in your observations.

The vast majority of problems in applied data science require regression modeling. You have a *response* variable (y) that you want to model or predict as a function of a vector of *input features*, or covariates ($\mathbf{x}$). This chapter introduces the basic framework and language of regression. We will build on this material throughout the rest of the book.

Regression is all about understanding the *conditional* probability distribution for "y given $\mathbf{x}$," which we write as p($y|\mathbf{x}$). Figure 1.1 illustrates the conditional distribution in contrast to a *marginal* distribution, which is so named because it corresponds to the unconditional distribution for a single margin (i.e., column) of a data matrix.

A variable that has a probability distribution (e.g., number of bathrooms in Figure 1.1) is called a *random variable*. The *mean* for a random variable is the average of random draws from its probability distribution. While the marginal mean is a simple number, the conditional mean is a function. For example, from Figure 1.1b, you can see that the average home selling price takes different values indexed by the number of bathrooms. The data is distributed randomly around these means, and the way that you model these distributions drives your estimation and prediction strategies.

Conditional Expectation

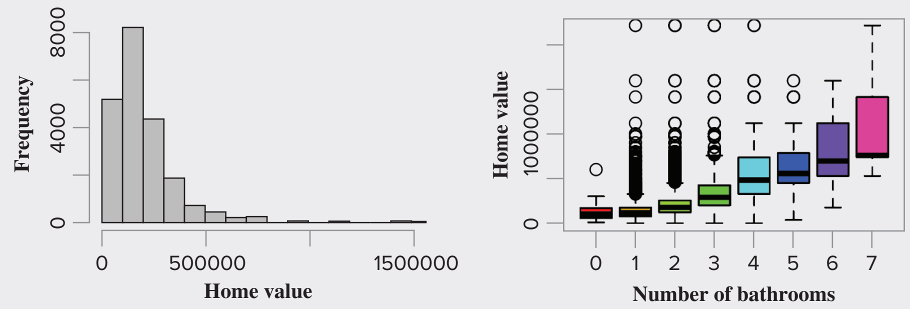
A basic but powerful regression strategy is to build models in terms of *averages* and *lines*. That is, we will model the conditional mean for our output variable as a linear function of inputs. Other regression strategies can sometimes be useful, such as *quantile regression* that models percentiles of the conditional distribution. However for the bulk of applications you will find that *mean* regression is a good approach.

There is some important notation that you need to familiarize yourself with for the rest of the book. We model the conditional mean for y given $\mathbf{x}$ as

$$\mathbb{E}[y\,|\,\mathbf{x}] = f(\mathbf{x}'\boldsymbol{\beta}) \qquad \textbf{(1.1)}$$

where

- $\mathbb{E}[\cdot]$ denotes the taking of the expectation or average of whatever random variable is inside the brackets. It is an extremely important operation, and we will use this notation to define many of our statistical models.



(a) Marginal Distribution

(b) Conditional Distribution

FIGURE 1.1 Illustration of marginal versus conditional distributions for home prices. On the left, we have the marginal distribution for all of the home prices. On the right, home price distributions are conditional on the number of bathrooms.

- The vertical bar | means "given" or "conditional upon," so that $\mathbb{E}[y|\mathbf{x}]$ is read as "the average for y given inputs $\mathbf{x}$."
- $f(\cdot)$ is a "link" function that transforms from the linear model to your response.
- $\mathbf{x} = [1, x_1, x_2, \ldots x_p]$ is the vector of covariates and $\boldsymbol{\beta} = [\beta_0, \beta_1, \beta_2, \ldots \beta_p]$ are the corresponding coefficients.

The *vector* notation, $\mathbf{x}'\boldsymbol{\beta}$, is shorthand for the sum of elementwise products:

$$\mathbf{x}'\boldsymbol{\beta} = [1 x_1\ x_2 \cdots x_p]\begin{bmatrix}\beta_0\\ \beta_1\\ \beta_2\\ \vdots\\ \beta_p\end{bmatrix} = \beta_0 + x_1\beta_1 + x_2\beta_2 + \ldots + x_p\beta_p \qquad \textbf{(1.2)}$$

This shorthand notation will be used throughout the book. Here we have used the convention that $x_0 = 1$, such that β_0 is the intercept.

The link function, $f(\cdot)$, defines the relationship between your linear function $\mathbf{x}'\boldsymbol{\beta}$ and the response. The link function gives you a huge amount of modeling flexibility. This is why models of the kind written in Equation (1.1) are called *generalized linear models* (GLMs). They allow you to make use of linear modeling strategies after some simple transformations of your output variable of interest. In this chapter we will outline the two most common GLMs: linear regression and logistic regression. These two models will serve you well for the large majority of analysis problems, and through them you will become familiar with the general principles of GLM analysis.

1.1 Linear Regression

Linear regression is the workhorse of analytics. It is fast to fit (in terms of both analyst and computational time), it gives reasonable answers in a variety of settings (so long as you know how to ask the right questions), and it is easy to interpret and understand. The model is as follows:

$$\mathbb{E}[y|\mathbf{x}] = \beta_0 + x_1\beta_1 + x_2\beta_2 + \ldots + x_p\beta_p \qquad \textbf{(1.3)}$$

This corresponds to using the link function $f(z) = z$ in Equation (1.1).

With just one input x, you can write the model as $\mathbb{E}[y|x] = \beta_0 + x\beta_1$ and plot it as in Figure 1.2. β_0 is the intercept. This is where the line crosses the y axis when x is 0. β_1 is the

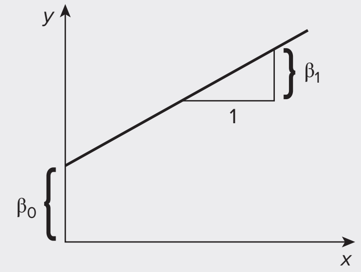


FIGURE 1.2 Simple linear regression with a positive slope β_1. The plotted line corresponds to $\mathbb{E}[y|x]$.

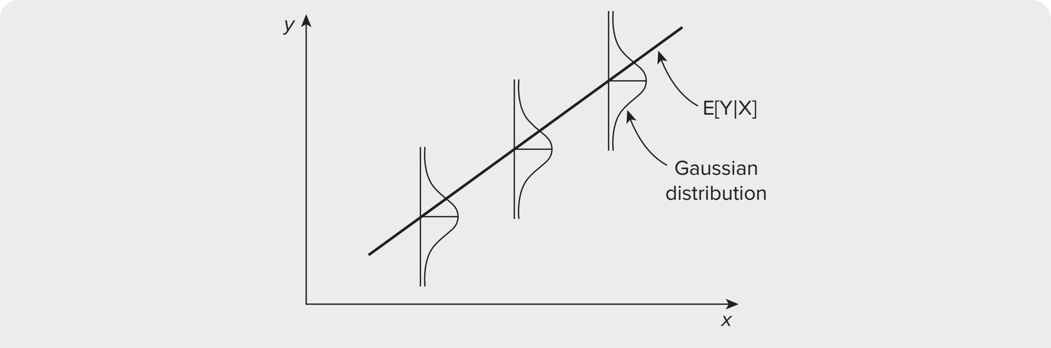


FIGURE 1.3 Using simple linear regression to picture the Gaussian conditional distribution for $y|x$. Here $\mathbb{E}[y|x]$ are the values on the line and the variation parallel to the y axis (i.e., within each narrow vertical strip) is assumed to be described by a Gaussian distribution.

slope and describes how $\mathbb{E}[y|x]$ changes as x changes. If x increases by 1 unit, $\mathbb{E}[y|x]$ changes by β_1. For a two predictor model, we are fitting a plane. Higher dimensions are more difficult to imagine, but the basic intuition is the same.

When fitting a regression model—i.e., when estimating the $\boldsymbol{\beta}$ coefficients—you make some assumptions about the conditional distribution beyond its mean at $\mathbb{E}[y|\mathbf{x}]$. Linear regression is commonly fit for Gaussian (normal) conditional distributions. We write this conditional distribution as

$$y \mid \mathbf{x} \sim \mathrm{N}(\mathbf{x}'\boldsymbol{\beta}, \sigma^2) \tag{1.4}$$

This says that the distribution for y as a function of $\mathbf{x}$ is normally distributed around $\mathbb{E}[y|\mathbf{x}] = \mathbf{x}'\boldsymbol{\beta}$ with variance σ^2. The same model is often written with an additive error term:

$$y = \mathbf{x}'\boldsymbol{\beta} + \varepsilon,\ \varepsilon \sim \mathrm{N}(0, \sigma^2) \tag{1.5}$$

where ε are the “independent” or “idiosyncratic” errors. These errors contain the variations in y that are not correlated with $\mathbf{x}$. Equations (1.4) and (1.5) describe the same model. Figure 1.3 illustrates this model for single-input simple linear regression. The line is the average $\mathbb{E}[y|x]$ and vertical variation around the line is what is assumed to have a normal distribution.

You will often need to transform your data to make the linear model of Equation (1.5) realistic. One common transform is that you need to take a *logarithm* of the response, say, “r,” such that your model becomes

$$\log(r) = \mathbf{x}'\boldsymbol{\beta} + \varepsilon,\ \varepsilon \sim \mathrm{N}(0, \sigma^2) \tag{1.6}$$

Of course this is the same as the model in Equation (1.5), but we have just made the replacement $y = \log(r)$. You will likely also consider transformations for the input variables, such that elements of $\mathbf{x}$ include logarithmic and other functional transformations. This is often referred to as *feature engineering*.

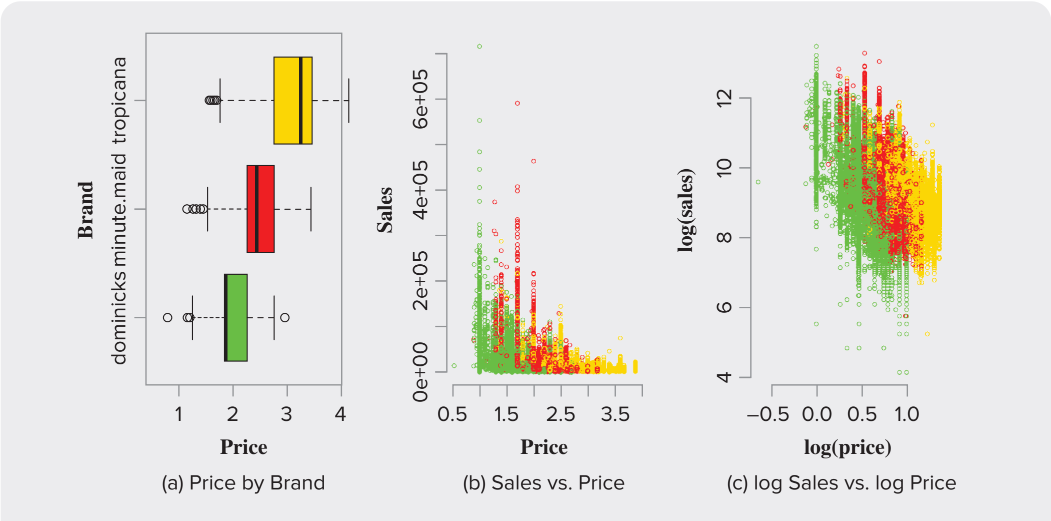
Example 1.1 Orange Juice Sales: Exploring Variables and the Need for a log-log Model As a concrete example, consider sales data for orange juice (OJ) from Dominick’s grocery stores. Dominick’s was a Chicago-area chain. This data was collected in the 1990s and is publicly available from the Kilts Center at the University of Chicago’s Booth School of brands—Tropicana, Minute Maid, Dominick's—at 83 stores in the Chicago area, as well as an indicator, `ad`, showing whether each brand was advertised (in store or flyer) that week.

```
> oj <- read.csv(“oj.csv”,strings=T)
> head(oj)
  sales price     brand  ad
1  8256  3.87 tropicana   0
2  6144  3.87 tropicana   0
3  3840  3.87 tropicana   0
4  8000  3.87 tropicana   0
5  8896  3.87 tropicana   0
6  7168  3.87 tropicana   0
> levels(oj$brand)
[1] “dominicks”  “minute.maid” “tropicana”
```

Notice the argument `strings=T` in `read.csv` as shorthand for "`stringsAsFactors = TRUE`." This converts our `brand` column into a factor variable. This was the default behavior of `read.csv` prior to version 4.0.0 of `R`, but you now need to specify it explicitly. Otherwise you will get an error when you try to make the plots or fit the regression models below.

The code-printout above is our first example showing `R` code and output. We will include a ton of code and output snippets like this throughout the book: they are an integral part of the material. If this output looks unfamiliar to you, you should break here and take the time to work through the `R`-primer in the Appendix.

Figure 1.4 shows the prices and sales broken out by brand. You can see in Figure 1.4a that each brand occupies a different price range: Dominick's is the budget option, Tropicana is the luxury option, and Minute Maid lives between. In Figure 1.4c, sales are clearly decreasing with



(a) Price by Brand

(b) Sales vs. Price

(c) log Sales vs. log Price

FIGURE 1.4

price. This makes sense: demand is *downward* sloping, and if you charge more, you sell less. More specifically, it appears that *log* sales has a roughly linear relationship with *log* price. This is an important point. Whenever you are working with linear (i.e., additive) models, it is crucial that you try to work in the space where you expect to find linearity. For variables that change *multiplicatively* with other factors, this is usually the log scale (see the nearby box for a quick review on logarithms). For comparison, the raw (without log) values in Figure 1.4b show a nonlinear relationship between prices and sales.

log-log Models and Elasticities

Another common scenario models against each other two variables that *both* move multiplicatively. For example, Figure 1.5 shows the national gross domestic product (GDP) versus imports for several countries. Fitting a line to the left panel would be silly; its slope will be entirely determined by small changes in the U.S. values. In contrast, the right panel shows that GDP and imports follow a neat linear relationship in log space.

Returning to our OJ example, Figure 1.4c indicates that this *log-log* model might be appropriate for the orange juice sales versus price analysis. One possible regression model is

$$\log(\texttt{sales}) = \beta_0 + \beta_1 \log(\texttt{price}) + \varepsilon \tag{1.7}$$

Here, `log(sales)` increase by β_1 for every unit increase in `log(price)`. Conveniently, log-log models have a much more intuitive interpretation: sales increase by $\beta_1\%$ for every 1% increase in price. To see this, you need a bit of calculus. Write $y = \exp[\beta_0 + \beta_1 \log(x) + \varepsilon]$ and differentiate with respect to x:

$$\frac{dy}{dx} = \frac{\beta_1}{x} e^{\beta_0 + \beta_1 \log(x) + \varepsilon} = \frac{\beta_1}{x} y \quad \Rightarrow \quad \beta_1 = \frac{dy/y}{dx/x} \tag{1.8}$$

This shows that β_1 is the proportional change in y over the proportional change in x. In economics there is a special name for such an expression: *elasticity*. The concept of elasticity will play an important role in many of our analyses.

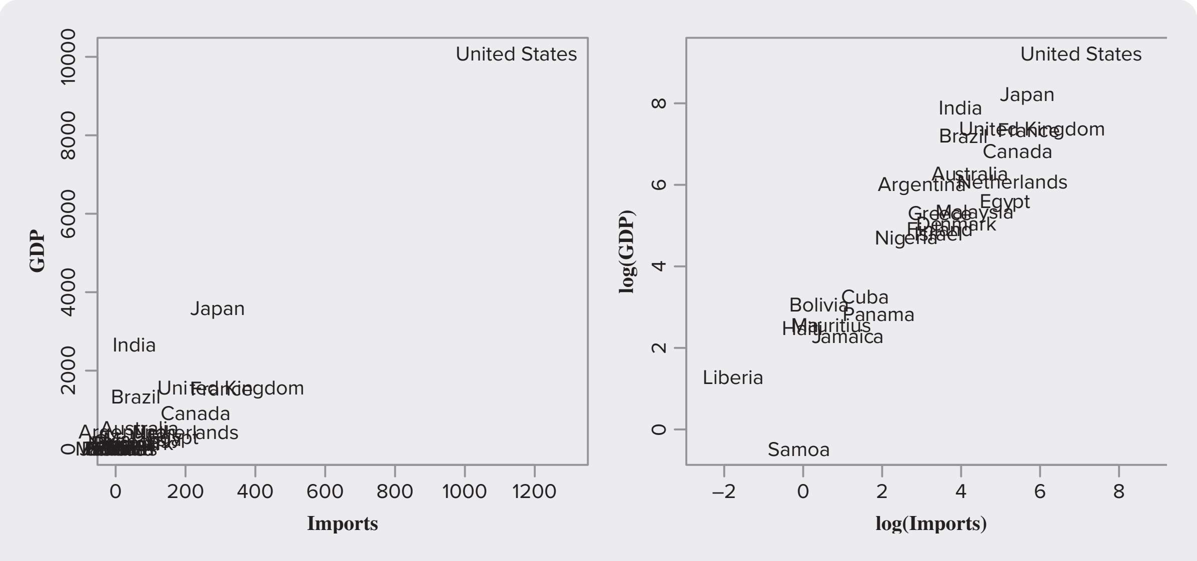


FIGURE 1.5 National GDP against imports, in original and log scale.

Logarithms and Exponents

Recall the logarithm definition:

$$\log(a) = b \Leftrightarrow a = e^b \tag{1.9}$$

Here, $e \approx 2.72$ is "Euler's number" and we refer to e^b as "e to the power of b" or, simply, 'b exponentiated.' We will sometimes write $\exp[b]$ instead of e^b; they mean the same thing. There are other types of logarithms (sometimes base 10 is used in introductory classes), but we will always use the *natural log* defined in Equation (1.9). The base e plays a central role in science and modeling of systems because e^x is its own derivative: $de^x/dx = e^x$ for those readers who know their calculus.

In a simple linear regression for $\log(y)$ on x, β_1 is added to the expected value for $\log(y)$ for each unit increase in x:

$$\log(y) = \beta_0 + \beta_1 x + \epsilon. \tag{1.10}$$

The fact that we are taking the log of y makes this model *multiplicative*. Recall some basic facts about logs and exponents: $\log(ab) = \log(a) + \log(b)$, $\log(a^b) = b \log(a)$, and $e^{a+b} = e^a e^b$. Thus, exponentiating both sides of Equation (1.10) yields

$$y = e^{\beta_0+\beta_1 x+\epsilon} = e^{\beta_0} e^{\beta_1 x} e^{\epsilon} \tag{1.11}$$

Considering $x^* = x + 1$, you get that

$$y^* = e^{\beta_0+\epsilon} e^{\beta_1 x^*} = e^{\beta_0+\epsilon} e^{\beta_1(x+1)} = e^{\beta_0+\beta_1 x+\epsilon} e^{\beta_1} = y e^{\beta_1} \tag{1.12}$$

Therefore, each unit increase in x leads $\mathbb{E}[y|x]$ to be *multiplied* by the factor e^{β_1}.

Example 1.2 Orange Juice Sales: Linear Regression Now that we have established what a *log-log* model will do for us, let's add a bit of complexity to the model from (1.7) to make it more realistic. If you take a look at Figure 1.4c, it appears that the three brands have log-log sales-price relationships that are concentrated around three separate lines. If you suspect that each brand has the same β_1 elasticity but a different intercept (i.e., if all brands have sales that move with price the same way but at the same price some brands will sell more than others), then you would use a slightly more complex model that incorporates both `brand` and `price`:

$$\log(\texttt{sales}) = \alpha_{\texttt{brand}} + \beta \log(\texttt{price}) + \varepsilon \tag{1.13}$$

Here, $\alpha_{\texttt{brand}}$ is shorthand for a separate intercept for each OJ brand, which we could write out more fully as

$$\alpha_{\texttt{brand}} = \alpha_0 \mathbb{1}_{[\texttt{dominicks}]} + \alpha_1 \mathbb{1}_{[\texttt{minute.maid}]} + \alpha_2 \mathbb{1}_{[\texttt{tropicana}]}. \tag{1.14}$$

The indicator functions, $\mathbb{1}_{[v]}$, are one if v is the true factor level and zero otherwise. Hence, Equation (1.13) says that, even though their sales all have the same elasticity to price, the brands can have different sales at the same price due to brand-specific intercepts.

Fitting Regressions with `glm`

To fit this regression in `R` you will use the `glm` function, which is used to estimate the class of generalized linear models that we introduced in Equation (1.1). There is also a `lm` function that fits only linear regression models, so you could use that here also (it takes the same arguments), but we will get in the habit of using `glm` since it works for many different GLMs. The function is straightforward to use: you give it a data frame with the `data` argument and provide a `formula` that defines your regression.

```
> fit <- glm( y ~ var1 + ... + varP, data=mydata)
```

The fitted object `fit` is a list of useful things (type `names(fit)` to see them), and there are functions to access the results. For example,

- `summary(fit)` prints the model, information about residual errors, the estimated coefficients and uncertainty about these estimates (we will cover the uncertainty in detail in the next chapter), and statistics related to model fit.
- `coef(fit)` supplies just the coefficient estimates.
- `predict(fit, newdata=mynewdata)` predicts y where `mynewdata` is a data frame with the same variables as `mydata`.

The formula syntax in the `glm` call is important. The ~ symbol is read as “regressed onto” or “as a function of.” The variable you want to predict, the y response variable, comes before the ~, and the input features, $\mathbf{x}$, come after. This model formula notation will be used throughout the remainder of the book, and we note some common specifications in Table 1.1.

The `R` formula for (1.13) is `log(sales) ~ brand + log(price)`. You can fit this with `glm` using the `oj` data, and then use the `coef` function to view the fitted coefficients. (More on this in Section 1.4.)

```
> fit<-glm( log(sales) ~ brand + log(price), data=oj)
> coef(fit) # fitted coefficients
     (Intercept) brandminute.maid   brandtropicana   log(price)
      10.8288216        0.8701747        1.5299428   -3.1386914
```

There are a few things to notice here. First, you can see that $\hat{\beta} = -3.1$ for the estimated coefficient on log price. Throughout this book we use the convention that $\hat{\theta}$ denotes the estimated value for some parameter θ. So $\hat{\beta}$ is the estimated “sales-price elasticity,” and it says that expected sales drop by about 3% for every 1% price increase. Second, notice that there are distinct model coefficients for Minute Maid and Tropicana but not for Dominick’s. This is due

`y ~ x1`	model by x_1
`y ~ .`	include all other columns
`y ~ .-x3`	include all except x_3
`y ~ .-1`	include all, but no intercept
`y ~ 1`	intercept only
`y ~ x1*x2`	include interaction for x_1 and x_2 and lower order terms
`y ~ x1:x2`	include interaction only
`y ~ .^2`	all possible 2 way interactions and lower order terms

TABLE 1.1 Some common syntax for use in formulas.

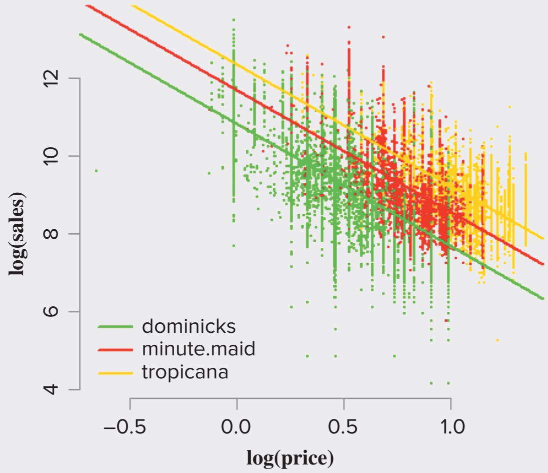


FIGURE 1.6 OJ data and the fitted regression lines (i.e., conditional expectations) for our model from (1.13) that regresses `log(sales)` on `log(price)` and `brand`.

to the way that `R` creates a numeric representation of the factor variables. It treats one of the factor levels as a 'reference level' that is subsumed into the intercept. For details, see the box on model matrices (i.e. design matrices).

The fitted values from the regression in Equation (1.13) are shown in Figure 1.6 alongside the original data. You see three lines shifted according to brand identity. *At the same price,* Tropicana sells more than Minute Maid, which in turn sells more than Dominick's. This makes sense: Tropicana is a luxury product that is preferable at the same price.

Model (Design) Matrices in R

When you regress onto a factor variable, `glm` converts the factor levels into a specific numeric representation. Take a look at rows 100, 200, and 300 from the `oj` data and notice that the `brand` column contains brand names, not numbers.

```
> oj[c(100,200,300),]
    sales price       brand  ad
100  4416  3.19   tropicana   0
200  5440  2.79 minute.maid   0
300 51264  1.39   dominicks   1
```