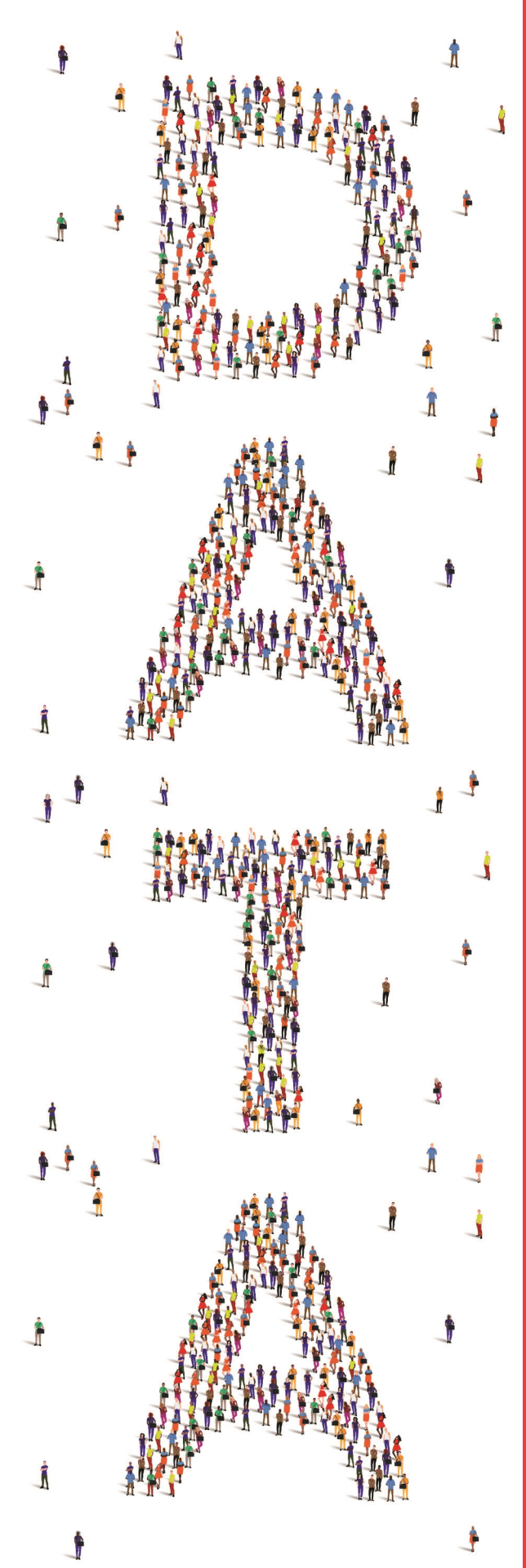


DATA ANALYSIS FOR SOCIAL SCIENCE

A *FRIENDLY* AND *PRACTICAL INTRODUCTION*

ELENA LLAUDET
KOSUKE IMAI

DATA ANALYSIS FOR SOCIAL SCIENCE

DATA ANALYSIS FOR SOCIAL SCIENCE

A FRIENDLY AND PRACTICAL INTRODUCTION

ELENA LLAUDET AND KOSUKE IMAI

PRINCETON UNIVERSITY PRESS
Princeton and Oxford

Published by Princeton University Press
41 William Street, Princeton, New Jersey 08540
99 Banbury Road, Oxford OX2 6JX

press.princeton.edu

ISBN 9780691199429
ISBN (pbk.) 9780691199436
ISBN (e-book) 9780691229348

British Library Cataloging-in-Publication Data is available

Editorial: Bridget Flannery-McCoy and Alena Chekanov
Production Editorial: Mark Bellis
Cover Design: Wanda España
Production: Erin Suydam
Publicity: Kate Hensley and Charlotte Coyne
Copyeditor: Melanie Mallon

Cover Credit: Human Alphabets by Sudarsan Thobias / Shutterstock

This book has been composed in Iwona

Printed on acid-free paper. ∞

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

To my students,
Elena Llaudet

To Christina, Keiji, and Misaki,
Kosuke Imai

CONTENTS

PREFACE

With this book, we hope to make data analysis for the social sciences accessible to everyone. Drawing conclusions from data and being able to evaluate the strengths and weaknesses of social scientific studies are critical skills that should be available to all. Not only can these skills lead to a job as a data scientist, but they also help us better understand and address important issues and problems facing society.

This book project was born when Elena suggested to Kosuke several ways to make more accessible the materials covered in *Quantitative Social Science: An Introduction* (Princeton University Press, 2017; aka QSS). Like QSS, this book teaches the fundamentals of data analysis for social science while analyzing real-world data from published research. This book, however, focuses on a smaller set of essential concepts with an emphasis on reaching students with no prior knowledge of statistics and coding and with minimal background in math. Our goals are to lower the barriers to becoming a data scientist and to share more broadly the excitement of quantitative social science research.

Many people have contributed their knowledge and talents to the production of this book. First and foremost, we would like to thank Kathryn Sargent for the countless hours she spent improving our writing and helping us bring our vision to reality. She has been an integral part of the project from the very beginning, and this book has greatly benefited from her attention to detail, editorial expertise, and good cheer. We are also grateful to all those who have given us feedback, especially our students, early adopters, and reviewers. In particular, we want to thank Alicia Cooperman, Michael Denly, Max Goplerud, Florian Hollenbach, Justin Leinaweaver, Emilee Martichenko, Davi Cordeiro Moreira, Leonid Peisakhin, Sheila Scheuerman, Tyler Simko, Robert Smith, Omar Wasow, and Hye Young You. Our thanks also go to Eric Crahan at Princeton University, who encouraged us to take on this project, and to Bridget Flannery-McCoy and Alena Chekanov, who made sure that the review and production process was as smooth as possible. In addition, Elena would like to offer special thanks to Harvard professor Stephen Ansolabehere for being a constant source of advice, support, and friendship.

Finally, we would like to thank our families and friends for their love and patience throughout this project. Elena thanks her mom, Didi, and brother, Jorge, for always being there for her, despite being on the other side of the Atlantic. She also thanks her friends, especially Bulbul, Baptiste, and Émile, for keeping her fed, sane, and high-spirited during all these years. Kosuke thanks Christina for a lifelong partnership that has made everything, both personal and professional, possible. He also thanks Keiji and Misaki for making sure that their family had many fun moments together, even during the pandemic.

Elena Llaudet and Kosuke Imai
Cambridge, Massachusetts
January 2022

DATA ANALYSIS FOR SOCIAL SCIENCE

1. INTRODUCTION

This book provides a friendly introduction to data analysis for the social sciences. It covers the fundamental methods of quantitative social science research, using plain language and assuming absolutely no prior knowledge of the subject matter.

R symbols, operators, and functions introduced in this chapter: +, −, *, /, <-, ", (), sqrt(), #, setwd(), read.csv(), View(), head(), dim(), $, and mean().

Proceeding step by step, we show how to analyze real-world data using the statistical program R for the purpose of answering a wide range of substantive questions. Along the way, we teach the statistical concepts and programming skills needed to conduct and evaluate social scientific studies. We explain not only how to perform the analyses but also how to interpret the results and identify the analyses' strengths and potential limitations.

Through this book, you will learn how to *measure*, *predict*, and *explain* quantities of interest based on data. These are the three fundamental goals of quantitative social science research. (See outline 1.1.)

WHY DO WE ANALYZE DATA IN THE SOCIAL SCIENCES?

In the social sciences we analyze data to:

- *measure* a quantity of interest, such as the proportion of eligible voters in favor of a particular policy
- *predict* a quantity of interest, such as the likely winner of an upcoming election
- *explain* a quantity of interest, such as the causal effect of attending a private school on student test scores.

OUTLINE 1.1. The three goals of quantitative social science research.

Figuring out whether you aim to measure, predict, and/or explain a quantity of interest should always precede the analysis and often also precede the data collection. As you will learn, the goals of your research will determine (i) what data you need to collect and how, (ii) the statistical methods you use, and (iii) what you pay attention to in the analysis. As you read this book and learn about each goal in detail, the distinctions will become clearer. Here we provide a brief preview.

To measure a quantity of interest such as a population characteristic, we often use survey data, that is, information collected on a sample of individuals from the target population. To analyze the data, we may compute various descriptive statistics, such as mean and median, and create visualizations like histograms and scatter plots. The validity of our conclusions depends on whether the sample is representative of the target population. To measure the proportion of eligible voters in favor of a particular policy, for example, our conclusions will be valid if the sample of voters surveyed is representative of *all* eligible voters.

To predict a quantity of interest, we typically use a statistical model such as a linear regression model to summarize the relationship between the predictors and the outcome variable of interest. The stronger the association between the predictors and the outcome variable, the better the predictive model will usually be. To predict the likely winner of an upcoming election, for example, if economic conditions are strongly associated with the electoral outcomes of candidates from the incumbent party, we may be able to use the current unemployment rate as our predictor.

To explain a quantity of interest such as the causal effect of a treatment on an outcome, we need to find or create a situation in which the group of individuals who received the treatment is comparable, in the aggregate, to the group of individuals who did not. In other words, we need to eliminate or control for all confounding variables, which are variables that affect both (i) the likelihood of receiving the treatment and (ii) the outcome variable. For example, when estimating the causal effect of attending a private school on student test scores, family wealth is a potential confounding variable. Students from wealthier families are more likely to attend a private school and also more likely to receive after-school tutoring, which might have a positive impact on their test scores. To produce valid estimates of causal effects, we may conduct a randomized experiment, which eliminates all confounding variables by assigning the treatment at random. In the current example, we would achieve this by using a lottery to determine which students attend private schools and which do not. Alternatively, if we cannot conduct a randomized experiment and need to rely on observational data instead, we would need to use statistical methods to control for all confounding variables such as family wealth. Otherwise, we would not know what portion of the difference in average test scores between private and public school students was the result of the type of school attended and what portion was the result of family background.

1.1 BOOK OVERVIEW

The book consists of seven chapters.

Chapter 1 is the introductory chapter, which lays the groundwork for the forthcoming data analyses.

Chapters 2 through 5 each introduce one or two published social scientific studies. In these chapters, we show how to analyze real-world datasets to answer different kinds of substantive questions. Specifically, we teach how to use several quantitative methods to measure, predict, and explain quantities of interest. (See outline 1.2, which indicates how each chapter relates to the three goals of quantitative social science research.)

BOOK OUTLINE

Chapter	Goal
1. Introduction	
2. Estimating Causal Effects with Randomized Experiments	Explain
3. Inferring Population Characteristics via Survey Research	Measure
4. Predicting Outcomes Using Linear Regression	Predict
5. Estimating Causal Effects with Observational Data	Explain
6. Probability	
7. Quantifying Uncertainty	All Three

OUTLINE 1.2. Book outline showing how each chapter relates to the three goals of quantitative social science research.

As you can see, chapters 2 and 5 are both about explanation, also known as causal inference. They teach how to estimate causal effects using different types of data. Since the methods differ, they are presented in separate chapters.

The book progresses from simple to more complex methods. Chapter 2 shows how to estimate causal effects using data from a randomized experiment. Chapter 3 is about measurement and teaches how to infer the characteristics of an entire population from a sample of survey respondents. Chapter 4 is about prediction and demonstrates how to use simple linear regression. Chapter 5 shows how to estimate causal effects with observational data and teaches multiple linear regression, the most complicated method we see in the book.

In chapter 6, we cover basic probability, and in chapter 7 we complete some of the analyses from chapters 2 through 5 by quantifying the uncertainty of our empirical findings. A more detailed description of each chapter is below.

1.2 CHAPTER SUMMARIES

In the current introductory chapter, we discuss why data analysis is a required skill among social scientists. We also explain how to get our computers ready, and we familiarize ourselves with RStudio and R, the two programs we will use. Then, we learn to load and make sense of data and practice computing and interpreting means.

In chapter 2, we define and learn how to estimate causal effects using data from a randomized experiment. As the working example, we analyze data from one of the largest experiments in U.S. education policy research, Project STAR, to determine whether attending a small class improves student performance.

In chapter 3, we use survey research to measure population characteristics. In addition, we learn how to visualize and summarize the distribution of single variables as well as the relationship between two variables. To illustrate these concepts, we analyze data related to the 2016 British referendum on withdrawing from the European Union, a decision popularly known as Brexit.

In chapter 4, we learn how to predict outcomes using simple linear regression models. For practice, we analyze data from 170 countries in order to predict growth in gross domestic product (GDP) using night-time light emissions as measured from space.

In chapter 5, we return to estimating causal effects, but this time using observational data. We define confounding variables, examine how their presence complicates the estimation of causal effects, and learn how to use multiple linear regression models to help mitigate the potential bias these variables introduce. To illustrate how this works step by step, we estimate the effects of Russian TV reception on the 2014 Ukrainian parliamentary elections. In this context, we introduce the concepts of internal and external validity. We then discuss the pros and cons of randomized experiments and of observational studies.

In chapter 6, we shift our focus away from data analysis to cover basic probability. We learn about random variables and their distributions as well as the distinction between population parameters and sample statistics. We then discuss the two large sample theorems that enable us to measure statistical uncertainty.

In chapter 7, we use everything we have learned in the preceding chapters and show how to quantify the uncertainty in our empirical findings in order to draw conclusions at the population level. In particular, we show how to quantify the uncertainty in (i) population inferences, (ii) predictions, and (iii) causal effect estimates. As illustrations, we complete some of the analyses we started in chapters 2 through 5.

1.3 HOW TO USE THIS BOOK

This is no ordinary textbook on data analysis. It is intentionally designed to accommodate readers with a variety of math and programming backgrounds.

The book uses a two-column layout: a main column and a side column or margin.

The main column contains the essential material and code, which are intended for all readers, except for the sections labeled FORMULA IN DETAIL. These contain more advanced material and are clearly identified so that you can easily skip them if you so choose.

In the margin are various types of notes and figures, each with a different purpose:

- At the beginning of each chapter, we list the R functions, symbols, and operators that will be introduced. You can look through the list to get a sense of what will be covered. (See, for example, the list for this chapter shown on the first page, and note that we always display code in cyan.)
- TIPs include supplemental material, such as additional explanations, answers to common questions, notes on best practices, and recommendations.
- RECALLs remind you of relevant information mentioned earlier in the book. These reminders are particularly helpful when the book is read only a few pages at a time, such as over the course of a semester.
- To help you review the core concepts, which are shown in **bold red** in the main text, we repeat their definitions in the margin. These notes are displayed in red.
- To help you with R functions, symbols, and operators, the first time these are introduced, we include in the margin an explanation of how they work and provide an example. These explanations are displayed in a cyan-colored frame.

At the end of each chapter, in place of the usual list of supplementary exercises, we include CHEATSHEETS to help you review the core concepts as well as the R functions, symbols, and operators covered.

Supplementary chapter-specific exercises, categorized by degree of difficulty, are available at http://press.princeton.edu/dss.

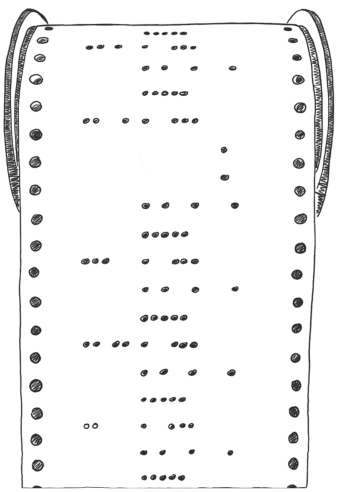
Finally, at the end of the book, we include three separate indexes for concepts, mathematical notation, and R-related topics.

1.4 WHY LEARN TO ANALYZE DATA?

As a social scientist, sooner or later you will need to rely on data to (i) measure the characteristics of a certain population of interest, (ii) make predictions, and/or (iii) make or evaluate decisions involving cause-and-effect relationships. What proportion of a population is in favor of a particular policy? Who is the candidate most likely to win an upcoming election? Shall we implement a particular policy to boost economic growth? You will want to be able to answer these types of questions either by analyzing data yourself or by understanding and assessing someone else's data analysis.

Even if you are not planning to become a social scientist, it is useful for you to know how to analyze data and/or how to distinguish a good quantitative study from a poorly conducted one. These are highly marketable skills. Recent advancements in computing power and the proliferation of data have increased the demand for data analysts who can inform decision makers in the public and private sectors alike.

The analytical skills you will learn by making your way through this book can also be used to improve everyday decisions, from choosing a candidate to vote for to determining the best way to increase your productivity. Perhaps most importantly, by learning the strengths and limitations of different quantitative methods, you will become less vulnerable to arguments based on faulty inferences from data. In the era of big data, we all stand to benefit from becoming savvy consumers of quantitative research, even if we do not all become skilled researchers ourselves.



1.4.1 LEARNING TO CODE

For the purpose of analyzing data, we write and run code. Code contains instructions that a computer can implement. These instructions consist of sequences of clearly defined steps written in a particular programming language. In this book, we code in R, which is a programming language used by many data analysts.

Don't worry if you have never done any coding before. Learning to code is not as difficult as one might think. You may even find it fun. Back in 1944, when the first programmable computer in the United States was built, only highly trained mathematicians were able to code. At that time, coding required punching paper tape in specific sequences that the machine could read. (See a rendition of what this tape looked like in the margin.) Today, anyone with access to a computer, some spare time, and a little patience can learn how to code.

1.5 GETTING READY

To perform the analyses in this book, we first need to download and install the necessary files and programs. We should also familiarize ourselves with RStudio, which is the interface we use throughout.

❶ DOWNLOAD AND SAVE FILES

All the files we will use are in a folder named DSS, which is available at http://press.princeton.edu/dss. For easy access, we recommend saving the folder on your Desktop. This is where the code used throughout the book assumes the DSS folder is located. In case you choose to save the folder elsewhere, we also provide instructions for making the necessary changes to the code.

TIP: By default, your computer will likely save the DSS folder to your Downloads. To move it, you can copy and paste it or drag it to the new location.

❷ DOWNLOAD AND INSTALL R AND RSTUDIO

We will use two programs: R and RStudio. R is the statistical program, the engine if you will, that will perform the calculations and create the graphics for us. RStudio is the user-friendly interface we will use to communicate with R. While we could use R directly, going through RStudio makes writing and running code much easier.

Why do we use R as our statistical program? Because it is free, open-source (anyone can see the underlying code and improve it), powerful, and flexible. It is also widely used. Indeed, many jobs these days require knowledge of R.

Unfortunately, these programs are compatible only with Linux, Mac, and Windows operating systems. They cannot be used on tablets or phones. We provide instructions for using these two programs on a Mac or a Windows computer.

To download and install R, go to http://cran.r-project.org, select the link that matches your operating system, and follow the instructions.

To download and install RStudio, go to http://rstudio.com, select the link that matches your operating system, and follow the instructions.

❸ BECOME FAMILIAR WITH RSTUDIO

To analyze data, we always operate R through RStudio. Let's take a moment to become acquainted with RStudio's layout.

After installing both programs, go ahead and start RStudio. Then, from within RStudio, open a new R script, which is the type of file we use to store the code we write to analyze data. Instructions are shown in the margin.

TIP: How do we open a new R script? In the RStudio dropdown menu, click on File > New File > R Script. A new "Untitled" file will open. The extension of this type of file is ".R", which is why R scripts are also called R files.

After opening a new R script, RStudio's interface should look like figure 1.1.

- The upper-left window is the *R script*, which is where we write and run code, giving R commands to execute.
- The lower-left window is the *R console*, where R provides either the results of successfully executed code (known as outputs) or any error messages.
- The upper-right window is the *environment*, which is the storage room of the current R session. It lists all the objects we have created. (We will soon explain what objects are and provide examples showing how the environment works.)
- The lower-right window is where we find the *help* and *plots tabs*, which we will learn how to use later on.

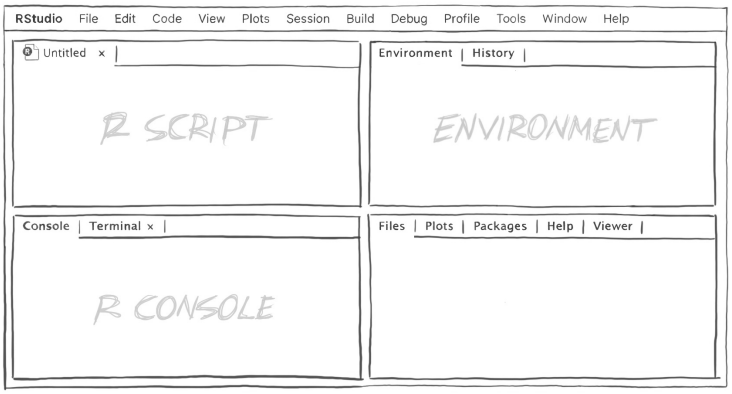


FIGURE 1.1. Layout of RStudio after opening a new R script. The upper-left window is the R script. The lower-left window is the R console. The upper-right window is the environment of the R session. The plots and help tabs appear in the lower-right window.

1.6 INTRODUCTION TO R

To use R, we need to learn the R programming language. (R is the name of both the statistical program and the programming language.) Learning a programming language is like learning a foreign language. It is not easy, and it takes a lot of practice and patience. The exercises in this book will help you learn to code in R, so be sure to follow along. Practice is everything!

Let's begin. R can be used to do many things. In our case, we will use R (i) as a calculator; (ii) to create objects, which is how R stores data; and (iii) to interact with data using functions.

WE WILL USE THE STATISTICAL PROGRAM R TO:

(i) do calculations
(ii) create objects
(iii) use functions.

1.6.1 DOING CALCULATIONS IN R

We can use R as a calculator. R can do summation (+), subtraction (−), multiplication (*), and division (/), as well as other more complicated mathematical operations. For example, the code to ask R to calculate 1 plus 3 is:

+, −, *, and / are some of the arithmetic operators recognized by R. Example: (4 − 1 + 3) * (2 / 3)

```
1 + 3
```

To run this or any other code, we first type it in the R script (the upper-left window of RStudio). Then, we highlight as much of it as we want to run and either (a) manually hit the run icon (shown in the margin) or (b) use the shortcut *command+enter* in Mac or *ctrl+enter* in Windows. The result, or output, of the executed code will show up in the R console (the lower-left window of RStudio). (Instead, we could type the code directly in the R console and hit enter, but we should avoid doing it that way. It is best to run code through an R script so that you can save it, re-run it, tweak it, expand it, and share it.)

After running the code above, we should see the following in the R console: first, the executed code shown in blue, indicating that R was able to run it without problems, and then the output shown in black. In this case, the output is:

```
4
```

Indeed, one plus three equals four.

Congratulations! You just wrote and ran your first line of code in R. Notice that now that you have written some code in the R script, RStudio shows the name of the file in red. This is to remind you that you have some unsaved changes. Once you save the file, the file name will return to black.

TIP: To save any changes you make to the R script, either (a) use the shortcut *command+S* in Mac or *ctrl+S* in Windows or (b) click on File > Save or Save As...

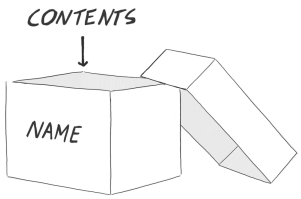
Throughout the book, we show the output that you should see in the R console right after the code that produces it. To distinguish the output from the code, we display the output with the symbol ## at the beginning of the line. For example, we display the code and output above as follows:

```
1 + 3
## [1] 4
```

TIP: Adding spaces around operators makes the code easier to read. R ignores these spaces. Example: 1+3 produces the same output as 1 + 3.

The first line, shown in cyan, is the code to be typed and run in the R script. The second line, which begins with ## and is shown in gray, is what should appear in the R console after running the code.

What does the number in brackets before the 4 mean? It indicates the position of the output immediately to its right. In this instance, [1] indicates that 4 is the first output of the code we ran. Later in the chapter, we will see examples of code that produce multiple outputs, which will clarify how this works.



1.6.2 CREATING OBJECTS IN R

In order to manipulate and analyze data, we need to load and store datasets. R stores information in what are known as objects, and so we need to learn how to create objects in R.

Think of an object as a box that can contain anything. All we need to do is give it a name, so that we know how to refer to it, and specify its contents.

To create an object in R, we use the assignment operator <-:

- To its left, we specify the name we want to give the object. This name can be anything as long as it does not begin with a number or contain spaces or special symbols like $ or % that are reserved for other purposes. Underscores _ are permitted and are good substitutes for spaces.
- To its right, we specify the contents of the object, that is, the data we want to store.

<- is the assignment operator. It creates new objects in R (unless one with the same name already exists, in which case R overwrites its contents). To its left, we specify the name of the object (without quotes). To its right, we specify the contents of the object. Example: four <- 4.

CREATING OBJECTS: To store data as an object in R, we run code using this format:

object_name <- *object_contents*

where:

- *object_name* is the name we want to give the object
- <- is the assignment operator, which creates an object by assigning contents to a name
- *object_contents* is the data we want to store in the object.

TIP: We would accomplish the same thing by running: four <- 4.

For example, if we want to create an object called *four* containing the output of the calculation 1+3, we run:

```
four <- 1 + 3
```

TIP: RStudio continues to work in the same R session until you quit the program. At that time, R will ask whether you want to save the workspace image, which contains all the objects you created during the R session. We recommend that you do not save it. If you need to continue to work with those objects, you can always re-create them by re-running your code.

Notice that after running the code above, the object will show up in the environment (the upper-right window in RStudio). As mentioned earlier, the environment is the storage room of the current R session. It shows the objects that we have created and that are available for us to use.

If we want to know the contents of the object *four*, we can type and run the name of the object in the R script. Its contents will appear in the R console. This is equivalent to asking R, what is inside the object named *four*?

```
four
## [1] 4
```

Not surprisingly, the object *four* contains the number 4.

Objects can contain text as well as numbers. For example, to create an object called *hello* containing the text "hi" we run:

```
hello <- "hi"
```

After running the code above, the environment should contain two objects: *four* and *hello*.

Let's stop here to learn something important about R. Look at the code above. Why did we use quotation marks around the content of the object "hi" but not around the name of the object hello? In other words, when do we use quotes " when coding in R? Here is the rule: When writing code, the names of objects, names of functions, and names of arguments as well as special values such as TRUE, FALSE, NA, and NULL should *not* be in quotes; all other text should be in quotes. (In the next subsection, we will see what we mean by functions and arguments. We will learn the meaning and usage of TRUE and FALSE in chapter 2 and of NA and NULL in chapter 3.)

" when writing code, the names of objects, names of functions, and names of arguments as well as special values such as TRUE, FALSE, NA, and NULL should not be in quotes; all other text should be in quotes. Examples: "this is just text", object_name. Never use quotes around a number unless you want R to treat it as text, in which case you will not be able to use it to perform arithmetic operations.

What would have happened had we tried to run the code above without quotes around *hi*? Go ahead and try it:

```
hello <- hi
## Error: object 'hi' not found
```

In the R console, you will see an error message (in red) that reads, "Error: object 'hi' not found". Indeed, by typing *hi* without quotes, you are telling R that *hi* is the name of an object. Because there is no object named *hi* in the environment, R gives you an error message. Encountering programming errors is part of the coding process. Try not to be discouraged by them.

TIP: If you have problems figuring out what a particular error means, Google it. Lots of data analysts participate in Q&A sites, such as Stack Overflow, which can be very helpful for this sort of thing.

A word of caution: R overwrites (replaces) old objects if we use the same name when creating a new object. For example, go ahead and run the following:

```
hello <- "hi, nice to meet you"
```

You should see that you still have only two objects in the environment: *four* and *hello*, but now *hello* contains the text "hi, nice to meet you" instead of simply "hi". To confirm this, we run:

```
hello
## [1] "hi, nice to meet you"
```

Note also that R is case-sensitive. It will treat *Hello* as a completely different object name than *hello*. If we run the name *Hello* by mistake, R will not be able to find the object because there is no object in the environment called *Hello* with an uppercase H at the beginning. To avoid this problem, we recommend using all lowercase letters when naming objects.