A DATA SCIENTIST'S GUIDE TO ACQUIRING, CLEANING, and MANAGING DATA in R

Samuel E. Buttrey and Lyn R. Whitaker

WILEY

A Data Scientist's Guide to Acquiring, Cleaning, and Managing Data in R

A Data Scientist's Guide to Acquiring, Cleaning, and Managing Data in R

Samuel E. Buttrey and Lyn R. Whitaker
Naval Postgraduate School, California, United States

WILEY

This edition first published 2018
© 2018 John Wiley & Sons Ltd

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by law. Advice on how to obtain permission to reuse material from this title is available at http://www.wiley.com/go/permissions.

The right of Samuel E. Buttrey and Lyn R. Whitaker to be identified as the authors of this work has been asserted in accordance with law.

Registered Offices
John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

Editorial Office
9600 Garsington Road, Oxford, OX4 2DQ, UK

For details of our global editorial offices, customer services, and more information about Wiley products visit us at www.wiley.com.

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Some content that appears in standard print versions of this book may not be available in other formats.

Limit of Liability/Disclaimer of Warranty
While the publisher and authors have used their best efforts in preparing this work, they make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives, written sales materials or promotional statements for this work. The fact that an organization, website, or product is referred to in this work as a citation and/or potential source of further information does not mean that the publisher and authors endorse the information or services the organization, website, or product may provide or recommendations it may make. This work is sold with the understanding that the publisher is not engaged in rendering professional services. The advice and strategies contained herein may not be suitable for your situation. You should consult with a specialist where appropriate. Further, readers should be aware that websites listed in this work may have changed or disappeared between when this work was written and when it is read. Neither the publisher nor authors shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

Library of Congress Cataloging-in-Publication Data applied for

Hardback ISBN: 9781119080022

Cover Design: Wiley
Cover Image: © Nongkran_ch/Gettyimages

Set in 10/12pt WarnockPro by SPi Global, Chennai, India

10 9 8 7 6 5 4 3 2 1

To Elinda and Mike

Contents

About the Authors

Samuel E. Buttrey received a bachelor's degree in statistics from Princeton University in 1983. After 8 years as a Wall Street computer systems analyst, he returned to graduate school and received MA and PhD degrees in statistics from the University of California at Berkeley, the latter in 1996. In that year, he joined the faculty of the Department of Operations Research at the Naval Postgraduate School in Monterey, California. He has published papers on nearest-neighbor and other classification methods and on applied problems ranging from numismatics and oceanography to human vision. He has also published papers describing his implementations of algorithms in software. His interests include classification, computationally intensive methods, and statistical graphics, and most recently, inter-point distance measures for mixed categorical and numeric data. He lives in Pacific Grove, California, with wife Elinda, son John, and some cats.

Lyn R. Whitaker received a bachelor's degree in genetics in 1978 and a PhD in statistics from the University of California, Davis, in 1985. She was an Assistant Professor in the Department of Statistics and Applied Probability at the University of California at Santa Barbara from 1985 to 1988, and joined the faculty of the Department of Operations Research at the Naval Postgraduate School in 1988. Her interests are applied statistics relevant to defense issues. These include unsupervised methods for large and messy data, the statistical aspects of reliability and survival analysis, and most recently, jointly with Buttrey, development and use of inter-point distances for mixed data types. She resides in Monterey, California, with husband Mike, father Fred, and, occasionally, children Alex, Lee, and Mary.

Preface

Statisticians use data to build models, and they use models to describe the world and to make predictions about what will happen next. There has been a large number of very good books that describe statistical modeling, but these modeling efforts usually start with a set of "clean," well-behaved data in which nothing is missing or anomalous.

In real life, data is messy. There will be missing values, impossible values, and typographical errors. Data is gathered from multiple sources, leading to both duplication and inconsistency. Data that should be categorical is coded as numeric; data that should be numeric can appear categorical; data can be hidden inside free-form text; and data can be in the form of dates in a wide number of possible formats. We estimate that 80% of the time taken in any data analysis problem is taken up just in reading and preparing the data. So, any analyst needs to know how to acquire data and how to prepare it for modeling, and the steps taken should be automatic, as far as possible, and reproducible.

This book describes how to handle data using the R software. R is the most widely used software in statistics, and it has the advantage of being free, open-source, and available on every major computing platform. Whatever software you use, you will find yourself facing the issues of acquiring, cleaning, and merging data, and documenting the steps you took. We hope this book will help you do these things efficiently.

Monterey, California, USA
November 30, 2016

Sam Buttrey and Lyn Whitaker

Acknowledgments

Our book is about how to use R to process data. We use R because it is powerful, versatile, and extensible. We thank the developers of R for their service to the statistical community for producing a high-quality open-source piece of software. We also thank the long list of colleagues and students who have helped frame our thinking about questions of statistics and data.

About the Companion Website

Don't forget to visit the companion website for this book:

www.wiley.com/go/buttrey/datascientistsguide 

There you will find valuable material designed to enhance your learning, including:

- A complete listing of all the R code in the Book
- Example datasets used in the Exercises

1

R

1.1 Introduction

This book focuses on one problem that is common to almost every statistical problem – indeed, to almost any problem involving any sort of analysis. That problem is acquiring and preparing the data. Across our many years of data analysis, we have learned that seemingly 80% of our time – maybe more – goes into the data preparation steps (a belief echoed by others such as Dasu and Johnson, 2003). Collectively, we call these actions *data cleaning*, although, as we will discuss later, we sometimes use that term for something a little more specific. Regardless of the name, almost any analysis requires that you (i) acquire that data, that is, read it into the computer program; (ii) clean the data, that is, identify entries that are duplicated or clearly erroneous or anomalous, and take other preparation steps (e.g., combining entries such as "Female," "female," and "F"); (iii) merge data from different sources; and (iv) prepare the data for modeling, which might involve dividing a set of numeric values into subsets, combining states into regions, and so on. This book discusses some approaches for accomplishing these four steps in the R language (R Core Team, 2013). A fifth problem, which receives less emphasis, is the problem of long-term curation of the data. Which parts of the data must be saved and in what way? We address that question by reference to the idea of *reproducible research*, which we discuss later in this chapter, and later in the book as well.

1.1.1 What Is R?

R is a computer program that lets you analyze data. By "analyze" we mean, first, read the data into the program and then operate on it – drawing graphs and charts, manipulating values, fitting statistical models, and so on. (Notice that we prefer to call data "it" rather than "them." We discuss this choice briefly toward the end of the chapter.) R is both a statistical "environment" and also

A Data Scientist's Guide to Acquiring, Cleaning, and Managing Data in R, First Edition.
Samuel E. Buttrey and Lyn R. Whitaker.
© 2018 John Wiley & Sons Ltd. Published 2018 by John Wiley & Sons Ltd.
Companion website: www.wiley.com/go/buttrey/datascientistsguide

a programming language, and it is very widely used both in commercial and academic settings. R is free and open-source and runs on Windows, Apple, and Linux operating systems. It is maintained by a group of volunteers who release bug fixes and new features regularly.

1.1.2 Who Uses R and Why?

R started as a tool for statisticians, evolving from a language called S that was created in the 1970s. Today, R remains the primary language of academic statisticians, and it also has a prominent place among analysts in business and government as well. It is used not only for building statistical models but also for handling and cleaning data, as in this book, and for developing new statistical methods, building simulations, for visualization, and generally for all the data-handling tools the statistician and the data scientist require. Because of the ease with which users can develop and distribute new methods, R has also become the tool of choice in certain fast-growing fields such as biostatistics and genetics. Articles on "surveys of the top tools used by data scientists" inevitably name R as one of the important tools with which data scientists, as well as statisticians, should be familiar. Moreover, R's popularity is such that there are extensions to R (see "packages" in Section 1.4.4) that allow you to connect to other programs such as the Python and Java languages, the H2O machine-learning system, the ArcGIS geographical information system, and many more.

1.1.3 Acquiring and Installing R

The primary way to acquire R is to download it from the Internet. The main R website for R is `www.r-project.org`, and the `www.cran.r-project.org` page ("CRAN" standing for "Comprehensive R Archive Network") is where you can download R itself. There are in fact dozens of "mirror" sites for CRAN – that is, websites that are essentially copies of the CRAN site – so as to reduce the load on the CRAN site. You can probably find a mirror near you on the "mirrors" page. After you download R, install it in the way you would normally install a program on your operating system.

At any one time, users around the world will be running slightly different versions of R, since new ones are released fairly frequently. For example, at this writing the current version of R was called 3.3.2, but many users are still using 3.2 or earlier versions. This will almost never cause problems, but it is a good idea to update your version of R from time to time.

There are also several slightly different versions of R distributed other than at CRAN. Microsoft R Open is a particular version of R that uses a different set of math libraries intended to make certain computations faster. Like "regular" R, Microsoft R Open is free, although it does not run on OS X. Other versions of R are intended to communicate with relational databases or with other

big-data platforms. For this book, we will assume you are running "regular" R – but in any case for our purposes all versions of R should behave exactly the same way.

1.1.4 Starting and Quitting R

The way you start R depends on your operating system. Normally double-clicking on an R icon will be enough to get R started. In the command-line interface of many Linux systems, or using the OS X terminal window, it may be enough just to type the upper-case letter `R` (or, for Windows command lines, `Rgui`). When R has started, you will see the command prompt `>`. This is the R *console*, the place where commands are entered. At this point, you can start typing commands to R. When it comes time to quit R, you can either "kill" the window in the usual way (for OS X, the red dot, the lightswitch in the top right, or via the File dialog; for Windows, the red X or File dialog) or you can type the `q()` command. In either case, R will then ask you if you want to "Save workspace image." If you answer "yes" to this question, R will save to the disk any changes you made during the current session, whereas if you answer "no," R will return its workspace to the condition it was in when R was last started. We almost always want to answer "yes" to this question!

1.2 Data

Data is information about the elements of whatever problem we are investigating. Data comes in many forms, but for our purposes it will always be presented in a set of computer-ready values. For example, a database concerning birds might include text about the habits of the birds, numbers giving lengths and weights of the individuals, maps showing migration patterns, images showing the birds themselves, sound recordings of the birds' calls, and so on. Although they look very different, all of these different pieces of information can be represented in the computer in digital form in one way or another. In this example, one of our primary tasks might be to ensure that each bird's description is correctly matched with the correct map, image, and song file. Our data analysis projects rarely include data quite so disparate, but in almost every case we need to acquire data, clean it (a process we start to describe in what follows and continue throughout the book), and prepare it for modeling, and in almost every case we expect our data to consist of both numeric and textual values.

1.2.1 Acquiring Data

The first step in a data analysis project, of course, is to get the data into R where it can be manipulated. We are old enough to remember the days when this involved typing all the data from the back of a book or journal paper into a

statistics package by hand, but happily this is not necessary today. On the other hand, data now comes in a variety of formats, few of which were created with the convenience of the data scientist in mind. In Chapter 6, we describe some of these common formats and how to use R to read data effectively.

1.2.2 Cleaning Data

We "clean" data when we detect (and, in many cases, remove) anomalies. Anomalies will very often be missing values, but they might also be absurd ones, as when people's ages are reported as 999 or −1. Sometimes, as in our earlier example, we might have genders reported as "Female," "female," and "F" and we want to combine these three values. In the cleaning process we might learn, for example, that one data source produced no data at all in August 2016; this sort of fact will need to be brought to the attention of the data provider. The data cleaning process also involves merging data from different sources, extracting subsets or reshaping the data in some way. All in all, data cleaning is the process of turning raw data, received from one or more providers, into a data set that can be used in visualization, modeling, and decision-making.

In practice these steps are iterative. Our cleaning process not only informs the modeling, but it sometimes leads us to re-acquire the data in a different, more usable form. Similarly, insights from modeling will often lead us to prepare the data in a new and more revealing way – because it is when we model that we often discover anomalies or other interesting attributes of the data.

1.2.3 The Goal of Data Cleaning

What a "clean" data set should look like depends on what your goals are. One useful perspective is given by Wickham (2014), who describes what he calls "tidy" data. A tidy data set is rectangular (or tabular); each row describes one unit of analysis (an observation), and each column gives one measurement (a variable). For example, in a data set giving measurements about people, each row would concern itself with a person, and the columns might give height, weight, age, blood type, and so on.

In some problems, it is not immediately clear what the unit of analysis is. For example, imagine data that describes the locations of boats over the course of a month, as recorded by GPS. For some purposes, a "tidy" data set would have one row per GPS ping, each row giving a ship identifier, a location, and a time. For other purposes, we might prefer a data set with one row per boat, each row giving the southernmost point that ship reaches, or perhaps giving a binary indicator of whether the ship did, or did not, spend time in international waters. Some data – images and sound, for example – do not lend themselves to this "tidy" approach.

The exact layout of your final data will depend on what you plan to do with it – and in some cases this won't be known until after you have operated on the data.

1.2.4 Making Your Work Reproducible

It is vital that other people be able to reproduce the actions you took on your data. Ideally, you or another analyst should be able to start with your raw data, run all the steps you applied to it, and emerge with exactly the same clean, prepared data sets. This will be useful to you when you encounter a situation similar to the one in the previous paragraph, where the form of the new data needs to be designed. But it is even more important for another analyst, since if you or another analyst can reproduce your results there will be no disagreement about the data. The act of making research reproducible has, in recent years, been rightfully recognized as a cornerstone of scientific progress. Record and document every step you take so that others can repeat them.

1.3 The Very Basics of R

This book is about handling data in R. It cannot teach you the very basics of R in detail – although, happily, there are many good books and online resources that can. (We give a few examples at the end of this chapter.) In this section, we list a few of the most basic facts about R, but, again, this book is not intended to teach you R. Rather, we focus on the details of R and of the way data is represented in R, in order to help you understand some of the ways to acquire, clean, and handle data inside R.

1.3.1 Top Ten Quick Facts You Need to Know about R

In this section, we give a few of the most important facts about R a beginner needs to know. There will be more detail on these facts later in the chapter and throughout the book.

1) The **prompt** is (by default) >. If you leave a command incomplete, maybe because there is an unclosed parenthesis or quotation mark, R gives you the continuation prompt, which is +. The Esc key (Windows) or control-C (other systems) produces the **break** command, which will take you back to the regular prompt. In this example, we show what a completed command looks like – in this case, R is computing the value of 3 divided by 2.

```
> 3 / 2
[1] 1.5
```

Here, R produced the prompt (>), and we typed `3 / 2` and pressed the Enter (or "Return") key. R then produced the output. We will talk about the `[1]` part in Chapter 2, but the computed value of `1.5` is shown. In the following example, we show what happens when we press Enter after typing the slash character:

```
> 3/
+ 2
[1] 1.5
```

Here, since the expression on the first line was incomplete, R produced the continuation prompt, +. When we typed `2` and hit Enter, the expression was complete and the result was shown. In case of confusion, press break until the original > prompt is showing.

In examples in this book where we want to show the R output, we also show the > prompt in front of our code. Remember, that > is produced by R; you don't need to type that yourself. (At the end of the chapter, we tell you where you can get all the code from the book in electronic form.)

2) R is **case-sensitive**, which means that upper- and lower-case letters are different in R. For example, the built-in R object `LETTERS` gives all 26 upper-case letters. A different item called `letters` contains the lower-case versions of the alphabet. There is no built-in object called `Letters`.
3) **Show** an object by typing its name. For example, if you type `ls` by itself, you see the contents of the function whose name is `ls`, the one that lists all the objects in your workspace (which we define later). To actually run the function and see the objects, you need to type the function's name together with parentheses. In this case, list your objects by typing `ls()`.
4) Get **help** for a function or object named `thing` with the command `help(thing)` or `?thing`. For example, to see the help for the `ls()` function, type `help(ls)`. If you don't know the name, try `help.search()` with a relevant word in quotation marks; for example, try `help.search ("matrices")` to see functions that handle matrices.
5) **Assign** a value or object to a name with the left-arrow (less-than plus hyphen): for example, the command `a <- 1` creates a new object named `a` with value 1. (You can also assign with a command such as `a = 1`, but we don't recommend it.) The assignment will over-write any existing object named `a` you might have had. Once you create an object, it is in your "workspace," and your workspace can be saved when you quit. So unless your computer crashes, when you create an object it will persist until you delete it. **Display** the set of objects in your workspace with `objects()` or `ls()`; remove an object with `remove()` or `rm()`. Not every character is permitted in the name of an R object. Start a name