The R Series

Reproducible Research with R and RStudio

Second Edition

Christopher Gandrud

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Reproducible Research with R and RStudio

Second Edition

Chapman & Hall/CRC
The R Series

Series Editors

John M. Chambers
Department of Statistics
Stanford University
Stanford, California, USA

Torsten Hothorn
Division of Biostatistics
University of Zurich
Switzerland

Duncan Temple Lang
Department of Statistics
University of California, Davis
Davis, California, USA

Hadley Wickham
RStudio
Boston, Massachusetts, USA

Aims and Scope

This book series reflects the recent rapid growth in the development and application of R, the programming language and software environment for statistical computing and graphics. R is now widely used in academic research, education, and industry. It is constantly growing, with new versions of the core software released regularly and more than 6,000 packages available. It is difficult for the documentation to keep pace with the expansion of the software, and this vital book series provides a forum for the publication of books covering many aspects of the development and application of R.

The scope of the series is wide, covering three main threads:

- Applications of R to specific disciplines such as biology, epidemiology, genetics, engineering, finance, and the social sciences.
- Using R for the study of topics of statistical methodology, such as linear and mixed modeling, time series, Bayesian methods, and missing data.
- The development of R, including programming, building packages, and graphics.

The books will appeal to programmers and developers of R software, as well as applied statisticians and data analysts in many fields. The books will feature detailed worked examples and R code fully integrated into the text, ensuring their usefulness to researchers, practitioners and students.

Published Titles

Stated Preference Methods Using R, *Hideo Aizaki, Tomoaki Nakatani, and Kazuo Sato*

Using R for Numerical Analysis in Science and Engineering, *Victor A. Bloomfield*

Event History Analysis with R, *Göran Broström*

Computational Actuarial Science with R, *Arthur Charpentier*

Statistical Computing in C++ and R, *Randall L. Eubank and Ana Kupresanin*

Reproducible Research with R and RStudio, Second Edition, *Christopher Gandrud*

Introduction to Scientific Programming and Simulation Using R, Second Edition, *Owen Jones, Robert Maillardet, and Andrew Robinson*

Nonparametric Statistical Methods Using R, *John Kloke and Joseph McKean*

Displaying Time Series, Spatial, and Space-Time Data with R, *Oscar Perpiñán Lamigueiro*

Programming Graphical User Interfaces with R, *Michael F. Lawrence and John Verzani*

Analyzing Sensory Data with R, *Sébastien Lê and Theirry Worch*

Parallel Computing for Data Science: With Examples in R, C++ and CUDA, *Norman Matloff*

Analyzing Baseball Data with R, *Max Marchi and Jim Albert*

Growth Curve Analysis and Visualization Using R, *Daniel Mirman*

R Graphics, Second Edition, *Paul Murrell*

Data Science in R: A Case Studies Approach to Computational Reasoning and Problem Solving, *Deborah Nolan and Duncan Temple Lang*

Multiple Factor Analysis by Example Using R, *Jérôme Pagès*

Customer and Business Analytics: Applied Data Mining for Business Decision Making Using R, *Daniel S. Putler and Robert E. Krider*

Implementing Reproducible Research, *Victoria Stodden, Friedrich Leisch, and Roger D. Peng*

Graphical Data Analysis with R, *Antony Unwin*

Using R for Introductory Statistics, Second Edition, *John Verzani*

Advanced R, *Hadley Wickham*

Dynamic Documents with R and knitr, Second Edition, *Yihui Xie*

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Reproducible Research with R and RStudio

Second Edition

Christopher Gandrud
Hertie School of Governance
Berlin, Germany

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

A CHAPMAN & HALL BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed on acid-free paper
Version Date: 20150505

International Standard Book Number-13: 978-1-4987-1537-9 (Paperback)

Contents

Preface

This book has its genesis in my PhD research at the London School of Economics. I started the degree with questions about the 2008/09 financial crisis and planned to spend most of my time researching capital adequacy requirements. But I quickly realized that I would actually spend a large proportion of my time learning the day-to-day tasks of data gathering, analysis, and results presentation. After plodding through for a while with Word, Excel, and Stata, my breaking point came while reentering results into a regression table after I had tweaked one of my statistical models, yet again. Surely there was a better way to *do* research that would allow me to spend more time answering my research questions. Making research reproducible for others also means making it better organized and efficient for yourself. My search for a better way led me straight to the tools for reproducible computational research.

The reproducible research community is very active, knowledgeable, and helpful. Nonetheless, I often encountered holes in this collective knowledge, or at least had no resource organize it all together as a whole. That is my intention for this book: to bring together the skills I have picked up for actually doing and presenting computational research. Hopefully, the book, along with making reproducible research more widely used, will save researchers hours of googling, so they can spend more time addressing their research questions.

Changes to the Second Edition

The tools of reproducible research have developed rapidly since the first edition of this book was published just two years ago. The second edition has been updated to incorporate the most important of these advancements, including discussions of:

- The *rmarkdown* package, which allows you to create reproducible research documents in PDF, HTML, and Microsoft Word formats using the simple and intuitive Markdown syntax.
- Improvements and changes to RStudio's interface and capabilities, such as its new tools for handling R Markdown documents.
- Expanded *knitr* R code chunk capabilities.
- The `kable` function in the *knitr* package and the *texreg* package for dynamically creating tables to present your data and statistical results.

- An improved discussion of file organization allowing you to take full advantage of relative file paths so that your documents are more easily reproducible across computers and systems.
- The *dplyr*, *magrittr*, and *tidyr* packages for fast data manipulation.
- Numerous changes to R syntax in user-created packages.
- Changes to GitHub's and Dropbox's interfaces.

Acknowledgements

I would not have been able to write this book without many people's advice and support. Foremost is John Kimmel, acquisitions editor at Chapman and Hall. He approached me in Spring 2012 with the general idea and opportunity for this book. Other editors at Chapman and Hall and Taylor and Francis have greatly contributed to this project, including Marcus Fontaine. I would also like to thank all of the book's reviewers whose helpful comments have greatly improved it. The first edition's reviewers include:

- Jeromy Anglim, Deakin University
- Karl Broman, University of Wisconsin, Madison
- Jake Bowers, University of Illinois, Urbana-Champaign
- Corey Chivers, McGill University
- Mark M. Fredrickson, University of Illinois, Urbana-Champaign
- Benjamin Lauderdale, London School of Economics
- Ramnath Vaidyanathan, McGill University

The developer and blogging community has also been incredibly important for making this book possible. Foremost among these people is Yihui Xie. He is the main developer behind the *knitr* package, co-developer of *rmarkdown*, and also an avid blog writer and commenter. Without him the ability to do reproducible research would be much harder and the blogging community that spreads knowledge about how to do these things would be poorer. Other great contributors to the reproducible research community include Carl Boettiger, Karl Broman, Markus Gesmann (who developed *googleVis*), Rob Hyndman, and Hadley Wickham (who has developed numerous very useful R packages). Thank you also to Victoria Stodden and Michael Malecki for helpful suggestions. And, of course, thank you to everyone at RStudio (especially JJ Allaire) for creating an increasingly useful program for reproducible research.

The second edition has benefited immensely from first edition readers' comments and suggestions. For a list of their valuable contributions, please see the book's GitHub Issues page `https://GitHub.com/christophergandrud/Rep-Res-Book/issues` and the first edition's Errata page `http://christophergandrud.GitHub.io/RepResR-RStudio/errata.htm`.

My students at Yonsei University were an important part of making the first edition. One of the reasons that I got interested in using many of the tools covered in this book, like using *knitr* in slideshows, was to improve a course I taught there: Introduction to Social Science Data Analysis. I tested many of the explanations and examples in this book on my students. Their feedback has been very helpful for making the book clearer and more useful. Their experience with using these tools on Microsoft Windows computers was also important for improving the book's Windows documentation. Similarly, my students at the Hertie School of Governance inspired and tested key sections of the second edition.

The vibrant community at Stack Overflow `http://stackoverflow.com/` and Stack Exchange `http://stackexchange.com/` are always very helpful for finding answers to problems that plague any computational researcher. Importantly, the sites make it easy for others to find the answers to questions that have already been asked.

My wife, Kristina Gandrud, has been immensely supportive and patient with me throughout the writing of this book (and pretty much my entire academic career). Certainly this is not the proper forum for musing about marital relations, but I'll do a musing anyways. Having a person who supports your interests, even if they don't completely share them, is immensely helpful for a researcher. It keeps you going.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Stylistic Conventions

I use the following conventions throughout this book:

- **Abstract variables**: Abstract variables, i.e. variables that do not represent specific objects in an example, are in `ALL CAPS TYPEWRITER TEXT`.
- **Clickable buttons**: Clickable Buttons are in `typewriter text`.
- **Code**: All code is in `typewriter text`.
- **Filenames and directories**: Filenames and directories more generally are printed in *italics*. I use CamelBack for file and directory names.
- **File extensions**: Like filenames, file extensions are *italicized*.
- **Individual variable values**: Individual variable values mentioned in the text are in *italics*.
- **Objects**: Objects are printed in *italics*. I use CamelBack for object names.
- **Object columns**: Data frame object columns are printed in *italics*.
- **Packages**: **R** packages are printed in *italics*.
- **Windows and RStudio panes**: Open windows and RStudio panes are written in *italics*.
- **Variable names**: Variable names are printed in **bold**. I use CamelBack for individual variable names.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Required R Packages

In this book I discuss how to use a number of user-written R packages for reproducible research. Many of these packages are not included in the default R installation. They need to be installed separately.

Note: in general you should aim to minimize the number of packages that your research depends on. Doing so will lessen the possibility that your code will "break" when a package is updated. This book depends on relatively many packages because of its special and unusual purpose of illustrating a variety of tools that you can use for reproducible research.

To install key user-written packages discussed in this book, copy the following code and paste it into your R console:

```
install.packages(c("brew", "countrycode",
                   "devtools", "dplyr",
                   "ggplot2", "googleVis",
                   "knitr", "MCMCpack",
                   "repmis", "RCurl",
                   "rmarkdown", "texreg",
                   "tidyr", "WDI",
                   "xtable", "Zelig"))
```

Once you enter this code, you may be asked to select a CRAN "mirror" to download the packages from.[1] Simply select the mirror closest to you.

In Chapter 9 we use the *Zelig* package (Owen et al., 2013) to create a simple Bayesian normal linear regression. For this to work properly you will need to install an additional package called *ZeligBayesian* (Owen, 2011). To do this, type the following code into your R console:

```
install.packages("ZeligBayesian",
                 repos = "http://r.iq.harvard.edu/",
                 type  = "source")
```

[1] CRAN stands for the Comprehensive R Archive Network.

Special issues for Windows and Linux Users

If you are using Windows, you will also need to install *Rtools* (Ripley and Murdoch, 2012). You can download *Rtools* from: `http://cran.r-project.org/bin/windows/Rtools/`. Please use the recommended installation to ensure that your system PATH is set up correctly. Otherwise your computer will not know where the tools are.

On Linux you will need to install the *RCurl* (Temple Lang, 2015) and *XML* (Temple Lang, 2013) packages separately. Use your Terminal to install these packages with the following code:

```
sudo apt-get update

sudo apt-get install libcurl4-gnutls-dev
sudo apt-get install libxml2-dev
sudo apt-get install r-cran-xml
sudo apt-get install r-cran-rjava
```

Additional Resources

Additional resources that supplement the examples in this book can be freely downloaded and experimented with. These resources include longer examples discussed in individual chapters and a complete short reproducible research project.

Chapter Examples

Longer examples discussed in individual chapters, including files to dynamically download data, code for creating figures, and markup files for creating presentation documents, can be accessed at: `https://GitHub.com/christophergandrud/Rep-Res-Examples`. Please see Chapter 5 for more information on downloading files from GitHub, where the examples are stored.

Short Example Project

To download a full (though very short) example of a reproducible research project created using the tools covered in this book go to: `https://GitHub.com/christophergandrud/Rep-Res-ExampleProject1`. Please follow the replication instructions in the main *README.md* file to fully replicate the project. It is probably a good idea to hold off looking at this complete example in detail until after you have become acquainted with the individual tools it uses. Become acquainted with the tools by reading through this book and working with the individual chapter examples.

The following two figures give you a sense of how the example's files are organized. Figure 1 shows how the files are organized in the file system. Figure 2 illustrates how the main files are dynamically tied together. In the *Data* directory we have files to gather raw data from the World Bank (2015) on fertilizer consumption and from Pemstein et al. (2010) on countries' levels of democracy. They are tied to the data through the `WDI` and `download.file` commands. A *Makefile* can run *Gather1.R* and *Gather2.R* to gather and clean the data. It runs *MergeData.R* to merge the data into one data file called *MainData.csv*. It also automatically generates a variable description file and a *README.md* recording the session info.

The *Analysis* folder contains two files that create figures presenting this data. They are tied to *MainData.csv* with the `read.csv` command. These files are run by the presentation documents when they are knitted. The presen-

tation documents tie to the analysis documents with *knitr* and the `source` command.

Though a simple example, hopefully these files will give you a complete sense of how a reproducible research project can be organized. Please feel free to experiment with different ways of organizing the files and tying them together to make your research really reproducible.

FIGURE 1
Short Example Project File Tree

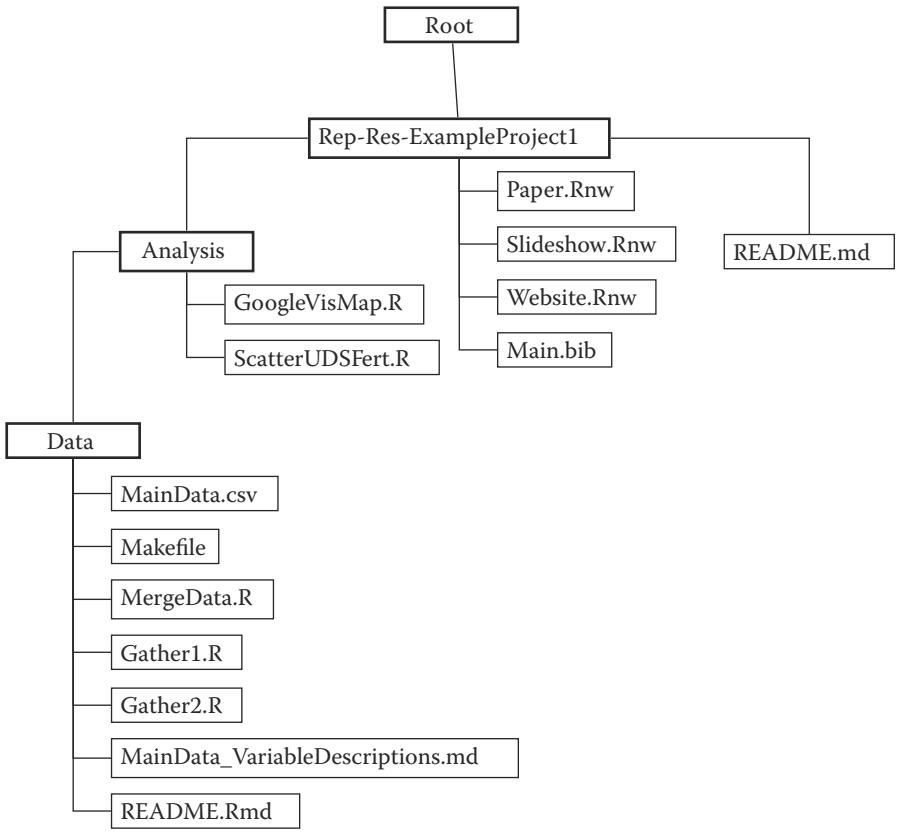
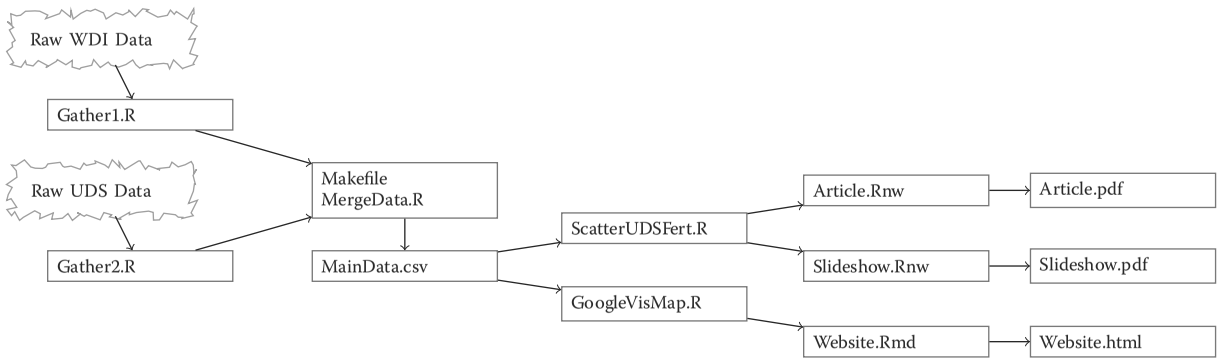


FIGURE 2
Short Example Main File Ties



`download.file` `read.csv` *`knitr`*
`Make` `source`
`merge`
`WDI`

Updates

Many of the reproducible research tools discussed in this book are improving rapidly. Because of this I will regularly post updates to the content covered in the book at: `https://GitHub.com/christophergandrud/Rep-Res-Book`.

Corrections

If you notice any corrections that should be made to fix typos, broken URLs, and so on, you can report them at: `https://GitHub.com/christophergandrud/Rep-Res-Book/issues`. I'll post notifications of changes to an Errata page at: `http://christophergandrud.GitHub.io/RepResR-RStudio/errata.htm`.

List of Figures

List of Tables

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Part I

Getting Started

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

1

Introducing Reproducible Research

Research is often presented in very selective containers: slideshows, journal articles, books, or maybe even websites. These presentation documents announce a project's findings and try to convince us that the results are correct (Mesirov, 2010). It's important to remember that these documents are not the research. Especially in the computational and statistical sciences, these documents are the "advertising". The research is the "full software environment, code, and data that produced the results" (Buckheit and Donoho, 1995; Donoho, 2010, 385). When we separate the research from its advertisement we are making it difficult for others to verify the findings by reproducing them.

This book gives you the tools to dynamically combine your research with the presentation of your findings. The first tool is a workflow for reproducible research that weaves the principles of reproducibility throughout your entire research project, from data gathering to the statistical analysis, and the presentation of results. You will also learn how to use a number of computer tools that make this workflow possible. These tools include:

- the **R** statistical language that will allow you to gather data and analyze it;
- the **LaTeX** and **Markdown** markup languages that you can use to create documents–slideshows, articles, books, and webpages–for presenting your findings;
- the *knitr* and *rmarkdown* **packages** for R and other tools, including **command-line shell programs** like GNU Make and Git version control, for dynamically tying your data gathering, analysis, and presentation documents together so that they can be easily reproduced;
- **RStudio**, a program that brings all of these tools together in one place.

1.1 What Is Reproducible Research?

Though there is some debate over what are the necessary and sufficient conditions for a replication (Makel and Plucker, 2014, 2), research results are generally considered *replicable* if there is sufficient information available for independent researchers to make the same findings using the same procedures

with new data.[1] For research that relies on experiments, this can mean a researcher not involved in the original research being able to rerun the experiment, including sampling, and validate that the new results are comparable to the original ones. In computational and quantitative empirical sciences, results are replicable if independent researchers can recreate findings by following the procedures originally used to gather the data and run the computer code. Of course, it is sometimes difficult to replicate the original data set because of issues such as limited resources to gather new data or because the original study already sampled the full universe of cases. So as a next-best standard we can aim for "*really reproducible research*" (Peng, 2011, 1226).[2] In computational sciences[3] this means:

> the data and code used to make a finding are available and they are sufficient for an independent researcher to recreate the finding.

In practice, research needs to be *easy* for independent researchers to reproduce (Ball and Medeiros, 2011). If a study is difficult to reproduce it's more likely that no one will reproduce it. If someone does attempt to reproduce this research, it will be difficult for them to tell if any errors they find were in the original research or problems they introduced during the reproduction. In this book you will learn how to avoid these problems.

In particular you will learn tools for dynamically "*knitting*"[4] the data and the source code together with your presentation documents. Combined with well-organized source files and clearly and completely commented code, independent researchers will be able to understand how you obtained your results. This will make your computational research easily reproducible.

[1] This is close to what Lykken (1968) calls "operational replication".

[2] The idea of really reproducible computational research was originally thought of and implemented by Jon Claerbout and the Stanford Exploration Project beginning in the 1980s and early 1990s (Fomel and Claerbout, 2009; Donoho et al., 2009). Further seminal advances were made by Jonathan B. Buckheit and David L. Donoho who created the Wavelab library of MATLAB routines for their research on wavelets in the mid-1990s (Buckheit and Donoho, 1995).

[3] Reproducibility is important for both quantitative and qualitative research (King et al., 1994). Nonetheless, we will focus mainly on on methods for reproducibility in quantitative computational research.

[4] Much of the reproducible computational research and literate programming literatures have traditionally used the term "weave" to describe the process of combining source code and presentation documents (see Knuth, 1992, 101). In the R community weave is usually used to describe the combination of source code and LaTeX documents. The term "knit" reflects the vocabulary of the *knitr* R package (knit + R). It is used more generally to describe weaving with a variety of markup languages. The term is used by RStudio if you are using the *rmarkdown* package, which is similar to *knitr*. We also cover the *rmarkdown* package in this book. Because of this, I use the term knit rather than weave in this book.

1.2 Why Should Research Be Reproducible?

Reproducible research is one of the main components of science. If that's not enough reason for you to make your research reproducible, consider that the tools of reproducible research also have direct benefits for you as a researcher.

1.2.1 For science

Replicability has been a key part of scientific inquiry from perhaps the 1200s (Bacon, 1859; Nosek et al., 2012). It has even been called the "demarcation between science and non-science" (Braude, 1979, 2). Why is replication so important for scientific inquiry?

Standard to judge scientific claims

Replication opens claims to scrutiny, allowing us to keep what works and discard what doesn't. Science, according to the American Physical Society, "is the systematic enterprise of gathering knowledge ... organizing and condensing that knowledge into testable laws and theories". The "ultimate standard" for evaluating scientific claims is whether or not the claims can be replicated (Peng, 2011; Kelly, 2006). Research findings cannot even really be considered "genuine contribution[s] to human knowledge" until they have been verified through replication (Stodden, 2009b, 38). Replication "requires the complete and open exchange of data, procedures, and materials". Scientific conclusions that are not replicable should be abandoned or modified "when confronted with more complete or reliable ... evidence".[5]

Reproducibility enhances replicability. If other researchers are able to clearly understand how a finding was originally made, then they will be better able to conduct comparable research in meaningful attempts to replicate the original findings. Sometimes strict replicability is not feasible, for example, when it is only possible to gather one data set on a population of interest. In these cases reproducibility is a "minimum standard" for judging scientific claims (Peng, 2011).

It is important to note that though reproducibility is a minimum standard for judging scientific claims, "a study can be reproducible and still be wrong" (Peng, 2014). For example, a statistically significant finding in one study may remain statistically significant when reproduced using the original data/code, but when researchers try to replicate it using new data and even methods, they are unable to find a similar result. The original finding could simply have been noise, even though it is fully reproducible.

[5]See the American Physical Society's website at `http://www.aps.org/policy/statements/99_6.cfm`. See also Fomel and Claerbout (2009).

Avoiding effort duplication & encouraging cumulative knowledge development

Not only is reproducibility important for evaluating scientific claims, it can also contribute to the cumulative growth of scientific knowledge (Kelly, 2006; King, 1995). Reproducible research cuts down on the amount of time scientists have to spend gathering data or developing procedures that have already been collected or figured out. Because researchers do not have to discover on their own things that have already been done, they can more quickly build on established findings and develop new knowledge.

1.2.2 For you

Working to make your research reproducible does require extra upfront effort. For example, you need to put effort into learning the tools of reproducible research by doing things such as reading this book. But beyond the clear benefits for science, why should you make this effort? Using reproducible research tools can make your research process more effective and (hopefully) ultimately easier.

Better work habits

Making a project reproducible from the start encourages you to use better work habits. It can spur you to more effectively plan and organize your research. It should push you to bring your data and source code up to a higher level of quality than you might if you "thought 'no one was looking' " (Donoho, 2010, 386). This forces you to root out errors–a ubiquitous part of computational research–earlier in the research process (Donoho, 2010, 385). Clear documentation also makes it easier to find errors.[6]

Reproducible research needs to be stored so that other researchers can actually access the data and source code. By taking steps to make your research accessible for others you are also making it easier for yourself to find your data and methods when you revise your work or begin new a project. You are avoiding personal effort duplication, allowing you to cumulatively build on your own work more effectively.

Better teamwork

The steps you take to make sure an independent researcher can figure out what you have done also make it easier for your collaborators to understand your work and build on it. This applies not only to current collaborators, but also future collaborators. Bringing new members of a research team up to speed on a cumulatively growing research project is faster if they can easily understand what has been done already (Donoho, 2010, 386).

[6] Of course, it's important to keep in mind that reproducibility is "neither necessary nor sufficient to prevent mistakes" (Stodden, 2009a).

Changes are easier

A third person may or may not actually reproduce your research even if you make it easy for them to do so. But, *you will almost certainly reproduce parts or even all of your own research.* No actual research process is completely linear. You almost never gather data, run analyses, and present your results without going backwards to add variables, make changes to your statistical models, create new graphs, alter results tables in light of new findings, and so on. You will probably try to make these changes long after you last worked on the project and long since you remembered the details of how you did it. Whether your changes are because of journal reviewers' and conference participants' comments or you discover that new and better data has been made available since beginning the project, designing your research to be reproducible from the start makes it much easier to change things later on.

Dynamic reproducible documents in particular can make changing things much easier. Changes made to one part of a research project have a way of cascading through the other parts. For example, adding a new variable to a largely completed analysis requires gathering new data and merging it with existing data sets. If you used data imputation or matching methods you may need to rerun these models. You then have to update your main statistical analyses, and recreate the tables and graphs you used to present the results. Adding a new variable essentially forces you to reproduce large portions of your research. If when you started the project you used tools that make it easier for others to reproduce your research, you also made it easier to reproduce the work yourself. You will have taken steps to have a "better relationship with [your] future [self]" (Bowers, 2011, 2).

Higher research impact

Reproducible research is more likely to be useful for other researchers than non-reproducible research. Useful research is cited more frequently (Donoho, 2002; Piwowar et al., 2007; Vandewalle, 2012). Research that is fully reproducible contains more information, i.e. more reasons to use and cite it, than presentation documents merely showing findings. Independent researchers may use the reproducible data or code to look at other, often unanticipated, questions. When they use your work for a new purpose they will (should) cite your work. Because of this, Vandewalle et al. even argue that "the goal of reproducible research is to have more impact with our research" (2007, 1253).

A reason researchers often avoid making their research fully reproducible is that they are afraid other people will use their data and code to compete with them. I'll let Donoho et al. address this one:

> *True. But competition means that strangers will read your papers, try to learn from them, cite them, and try to do even better. If you prefer obscurity, why are you publishing?* (2009, 16)

1.3 Who Should Read This Book?

This book is intended primarily for researchers who want to use a systematic workflow that encourages reproducibility as well as practical state-of-the-art computational tools to put this workflow into practice. These people include professional researchers, upper-level undergraduate, and graduate students working on computational data-driven projects. Hopefully, editors at academic publishers will also find the book useful for improving their ability to evaluate and edit reproducible research.

The more researchers that use the tools of reproducibility the better. So I include enough information in the book for people who have very limited experience with these tools, including limited experience with R, LaTeX, and Markdown. They will be able to start incorporating reproducible research tools into their workflow right away. The book will also be helpful for people who already have general experience using technologies such as R and LaTeX, but would like to know how to tie them together for reproducible research.

1.3.1 Academic researchers

Hopefully so far in this chapter I've convinced you that reproducible research has benefits for you as a member of the scientific community and personally as a computational researcher. This book is intended to be a practical guide for how to actually make your research reproducible. Even if you already use tools such as R and LaTeX you may not be leveraging their full potential. This book will teach you useful ways to get the most out of them as part of a reproducible research workflow.

1.3.2 Students

Upper-level undergraduate and graduate students conducting original computational research should make their research reproducible for the same reasons that professional researchers should. Forcing yourself to clearly document the steps you took will also encourage you to think more clearly about what you are doing and reinforce what you are learning. It will hopefully give you a greater appreciation of research accountability and integrity early in your career (Barr, 2012; Ball and Medeiros, 2011, 183).

Even if you don't have extensive experience with computer languages, this book will teach you specific habits and tools that you can use throughout your student research and hopefully your careers. Learning these things earlier will save you considerable time and effort later.

1.3.3 Instructors

When instructors incorporate the tools of reproducible research into their assignments they not only build students' understanding of research best prac-

tice, but are also better able to evaluate and provide meaningful feedback on students' work (Ball and Medeiros, 2011, 183). This book provides a resource that you can use with students to put reproducibility into practice.

If you are teaching computational courses, you may also benefit from making your lecture material dynamically reproducible. Your slides will be easier to update for the same reasons that it is easier to update research. Making the methods you used to create the material available to students will give them more information. Clearly documenting how you created lecture material can also pass information on to future instructors.

1.3.4 Editors

Beyond a lack of reproducible research skills among researchers, an impediment to actually creating reproducible research is a lack of infrastructure to publish it (Peng, 2011). Hopefully, this book will be useful for editors at academic publishers who want to be better at evaluating reproducible research, editing it, and developing systems to make it more widely available. The journal *Biostatistics* is a good example of a publication that is encouraging (actually requiring) reproducible research. From 2009 the journal has had an editor for reproducibility that ensures replication files are available and that results can be replicated using these files (Peng, 2009). The more editors there are with the skills to work with reproducible research the more likely it is that researchers will do it.

1.3.5 Private sector researchers

Researchers in the private sector may or may not want to make their work easily reproducible outside of their organization. However, that does not mean that significant benefits cannot be gained from using the methods of reproducible research. First, even if public reproducibility is ruled out to guard proprietary information,[7] making your research reproducible to members of your organization can spread valuable information about how analyses were done and data was collected. This will help build your organization's knowledge and avoid effort duplication. Just as a lack of reproducibility hinders the spread of information in the scientific community, it can hinder it inside of a private organization. Using the sort of dynamic automated processes run with clearly documented source code we will learn in this book can also help create robust data analysis methods that help your organization avoid errors that may come from cutting-and-pasting data across spreadsheets.[8]

[7]There are ways to enable some public reproducibility without revealing confidential information. See Vandewalle et al. (2007) for a discussion of one approach.

[8]See this post by David Smith about how the J.P. Morgan "London Whale" problem may have been prevented with the type of processes covered in this book: `http://blog.revolutionanalytics.com/2013/02/did-an-excel-error-bring-down-the-london-whale.html` (posted 11 February 2013).

Also, the tools of reproducible research covered in this book enable you to create professional standardized reports that can be easily updated or changed when new information is available. In particular, you will learn how to create batch reports based on quantitative data.

1.4 The Tools of Reproducible Research

This book will teach you the tools you need to make your research highly reproducible. Reproducible research involves two broad sets of tools. The first is a **reproducible research environment** that includes the statistical tools you need to run your analyses as well as "the ability to automatically track the provenance of data, analyses, and results and to package them (or pointers to persistent versions of them) for redistribution". The second set of tools is a **reproducible research publisher**, which prepares dynamic documents for presenting results and is easily linked to the reproducible research environment (Mesirov, 2010, 415).

In this book we will focus on learning how to use the widely available and highly flexible reproducible research environment–R/RStudio (R Core Team, 2014; RStudio, Inc., 2015).[9] R/RStudio can be linked to numerous reproducible research publishers such as LaTeX and Markdown with Yihui Xie's *knitr* package (2015b) or the related *rmarkdown* package (Allaire et al., 2015a). The main tools covered in this book include:

- **R**: a programming language primarily for statistics and graphics. It can also be useful for data gathering and creating presentation documents.

- ***knitr* and *rmarkdown***: related R packages for literate programming. They allow you to combine your statistical analysis and the presentation of the results into one document. They work with R and a number of other languages such as Bash, Python, and Ruby.

- **Markup languages**: instructions for how to format a presentation document. In this book we cover LaTeX, Markdown, and a little HTML.

- **RStudio**: an integrated developer environment (IDE) for R that tightly combines R, *knitr*, *rmarkdown*, and markup languages.

- **Cloud storage & versioning**: Services such as Dropbox and Git/GitHub that can store data, code, and presentation files, save previous versions of these files, and make this information widely available.

- **Unix-like shell programs**: These tools are useful for working with large

[9] The book was created with R version 3.2.0 and developer builds of RStudio version 0.99.370.

research projects.[10] They also allow us to use command-line tools including GNU Make for compiling projects and Pandoc, a program useful for converting documents from one markup language to another.

1.5 Why Use R, *knitr*/*rmarkdown*, and RStudio for Reproducible Research?

Why R?

Why use a statistical programming language like R for reproducible research? R has a very active development community that is constantly expanding what it is capable of. As we will see in this book, R enables researchers across a wide range of disciplines to gather data and run statistical analyses. Using the *knitr* or *rmarkdown* package, you can connect your R-based analyses to presentation documents created with markup languages such as LaTeX and Markdown. This allows you to dynamically and reproducibly present results in articles, slideshows, and webpages.

The way you interact with R has benefits for reproducible research. In general you interact with R (or any other programming and markup language) by explicitly writing down your steps as source code. This promotes reproducibility more than your typical interactions with Graphical User Interface (GUI) programs like SPSS[11] and Microsoft Word. When you write R code and embed it in presentation documents created using markup languages, you are forced to explicitly state the steps you took to do your research. When you do research by clicking through drop-down menus in GUI programs, your steps are lost, or at least documenting them requires considerable extra effort. Also it is generally more difficult to dynamically embed your analysis in presentation documents created by GUI word processing programs in a way that will be accessible to other researchers both now and in the future. I'll come back to these points in Chapter 2.

Why knitr *and* rmarkdown*?*

Literate programming is a crucial part of reproducible quantitative research.[12] Being able to directly link your analyses, your results, and the code you used to produce the results makes tracing your steps much easier. There are many different literate programming tools for a number of different programming

[10] In this book I cover the Bash shell for Linux and Mac as well as Windows PowerShell.

[11] I know you can write scripts in statistical programs like SPSS, but doing so is not encouraged by the program's interface and you often have to learn multiple languages for writing scripts that run analyses, create graphics, and deal with matrices.

[12] Donald Knuth coined the term literate programming in the 1970s to refer to a source file that could be both run by a computer and "woven" with a formatted presentation document (Knuth, 1992).

languages.[13] Previously, one of the most common tools for researchers using R and the LaTeX markup language was *Sweave* (Leisch, 2002). The packages I am going to focus on in this book are newer and have more capabilities. They are called *knitr* and *rmarkdown*. Why are we going to use these tools in this book and not *Sweave* or some other tool?

The simple answer is that they are more capable than *Sweave*. Both *knitr* and *rmarkdown* can work with markup languages other than LaTeX including Markdown and HTML. *rmarkdown* can even output Microsoft Word documents. They can work with programming languages other than R. They highlight R code in presentation documents making it easier for your readers to follow.[14] They give you better control over the inclusion of graphics and can cache code chunks, i.e. save the output for later. *knitr* has the ability to understand *Sweave*-like syntax, so it will be easy to convert backwards to *Sweave* if you want to.[15] You also have the choice to use much simpler and more straightforward syntax with *knitr* and *rmarkdown*.

knitr and *rmarkdown* have broadly similar capabilities and syntax. They both are literate programming tools that can produce presentation documents from multiple markup languages. They have almost identical syntax when used in Markdown. Their main difference is that they take different approaches to creating presentation documents. *knitr* documents must be written using the markup language associated with the desired output. For example, with *knitr*, LaTeX must be used to create PDF output documents and Markdown or HTML must be used to create webpages. *rmarkdown* builds directly on *knitr*, the key difference being that it uses the straightforward Markdown markup language to generate PDF, HTML, and MS Word documents.[16]

Because you write with the simple Markdown syntax, *rmarkdown* is generally easier to use. It has the advantage of being able to take the same markup document and output multiple types of presentation documents. Nonetheless, for complex documents like books and long articles or work that requires custom formatting, *knitr* LaTeX is often preferable and extremely flexible, though the syntax is more complicated.

Why RStudio?

Why use the RStudio integrated development environment for reproducible research? R by itself has the capabilities necessary to gather data, analyze it, and, with a little help from *knitr*/*rmarkdown* and markup languages, present results in a way that is highly reproducible. RStudio allows you to do all of

[13] A very interesting tool that is worth taking a look at for the Python programming language is HTML Notebooks created with IPython. For more details see `http://ipython.org/ipython-doc/dev/notebook/index.html`.

[14] Syntax highlighting uses different colors and fonts to distinguish different types of text.

[15] Note that the Sweave-style syntax is not identical to actual *Sweave* syntax. See Yihui Xie's discussion of the differences between the two at: `http://yihui.name/knitr/demo/sweave/`. *knitr* has a function (`Sweave2knitr`) for converting *Sweave* to *knitr* syntax.

[16] It does this by relying on a tool called Pandoc (MacFarlane, 2014).

these things, but simplifies many of them and allows you to navigate through them more easily. It also is a happy medium between R's text-based interface and a pure GUI.

Not only does RStudio do many of the things that R can do but more easily, it is also a very good standalone editor for writing documents with LaTeX and Markdown. For LaTeX documents it can, for example, insert frequently used commands like `\section{}` for numbered sections (see Chapter 11).[17] There are many LaTeX editors available, both open source and paid. But RStudio is currently the best program for creating reproducible LaTeX and Markdown documents. It has full syntax highlighting. Its syntax highlighting can even distinguish between R code and markup commands in the same document. It can spell check LaTeX and Markdown documents. It handles *knitr*/*rmarkdown* code chunks beautifully (see Chapter 3).

Finally, RStudio not only has tight integration with various markup languages, it also has capabilities for using other tools such as C++, CSS, JavaScript, and a few other programming languages. It is closely integrated with the version control programs Git and SVN. Both of these programs allow you to keep track of the changes you make to your documents (see Chapter 5). This is important for reproducible research since version control programs can document many of your research steps. It also has a built-in ability to make HTML slideshows from *knitr*/*rmarkdown* documents. Basically, RStudio makes it easy to create and navigate through complex reproducible research documents.

1.5.1 Installing the main software

Before you read this book you should install the main software. All of the software programs covered in this book are open source and can be easily downloaded for free. They are available for Windows, Mac, and Linux operating systems. They should run well on most modern computers.

You should install R before installing RStudio. You can download the programs from the following websites:

- **R**: `http://www.r-project.org/`,
- **RStudio**: `http://www.rstudio.com/products/rstudio/download/`.

The download webpages for these programs have comprehensive information on how to install them, so please refer to those pages for more information.

After installing R and RStudio you will probably also want to install a number of user-written packages that are covered in this book. To install all of these user-written packages, please see page xix.

[17] If you are more comfortable with a what-you-see-is-what-you-get (WYSIWYG) word processor like Microsoft Word, you might be interested in exploring Lyx. It is a WYSIWYG-like LaTeX editor that works with *knitr*. It doesn't work with the other markup languages covered in this book. For more information see: `http://www.lyx.org/`. I give some brief information on using Lyx with *knitr* in Chapter 3's Appendix.

Installing markup languages

If you are planning to create LaTeX documents you need to install a TeX distribution.[18] They are available for Windows, Mac, and Linux systems. They can be found at: `http://www.latex-project.org/ftp.html`. Please refer to that site for more installation information.

If you want to create Markdown documents you can separately install the *markdown* package in R. You can do this the same way that you install any package in R, with the `install.packages` command.[19]

GNU Make

If you are using a Linux computer you already have GNU Make installed.[20] Mac users will need to install the command-line developer tools. There are two ways to do this. One is go to the App Store and download Xcode (it's free). Once Xcode is installed, install command-line tools, which you will find by opening Xcode then clicking on `Preference` → `Downloads`. However, Xcode is a very large download and you only need the command-line tools for Make. To install just the command-line tools, open the Terminal and try to run Make by typing `make` and hitting return. A box should appear asking you if you want to install the command-line developer tools. Click `Install`. Windows users will have Make installed if they have already installed Rtools (see page xx). Mac and Windows users will need to install this software not only so that GNU Make runs properly, but also so that other command-line tools work well.

Other Tools

We will discuss other tools such as Git that can be a useful part of a reproducible research workflow. Installation instructions for these tools will be discussed below.

1.6 Book Overview

The purpose of this book is to give you the tools that you will need to do reproducible research with R and RStudio. This book describes a workflow for reproducible research primarily using R and RStudio. It is designed to give you the necessary tools to use this workflow for your own research. It is not designed to be a complete reference for R, RStudio, *knitr*/*rmarkdown*, Git, or any other program that is a part of this workflow. Instead it shows you how

[18] LaTeX is is really a set of macros for the TeX typesetting system. It is included in all major TeX distributions.

[19] The exact command is: `install.packages("markdown")`.

[20] To verify this, open the Terminal and type: `make -version` (I used version 3.81 for this book). This should output details about the current version of Make installed on your computer.

these tools can fit together to make your research more reproducible. To get the most out of these individual programs I will along the way point you to other resources that cover these programs in more detail.

To that end, I can recommend a number of resources that cover more of the nitty-gritty:

- Michael J. Crawley's (2013) encyclopaedic R book, appropriately titled ***The R Book***, published by Wiley.

- Hadley Whickham (2014a) has a great new book out from Chapman and Hall on ***Advanced R***.

- Yihui Xie's (2013) book ***Dynamic Documents with R and knitr***, published by Chapman and Hall, provides a comprehensive look at how to create documents with *knitr*. It's a good complement to this book's generally more research project–level focus.

- Norman Matloff's (2011) tour through the programming language aspects of R called ***The Art of R Programming: A Tour of Statistical Design Software***, published by No Starch Press.

- Cathy O'Neil and Rachel Schutt (2013) give a great introduction the field of data science generally in ***Doing Data Science***, published by O'Reilly Media Inc.

- For an excellent introduction to the command-line in Linux and Mac, see William E. Shotts Jr.'s (2012) book ***The Linux Command-line: A Complete Introduction*** also published by No Starch Press. It is also helpful for Windows users running PowerShell (see Chapter 4).

- The RStudio website (`http://www.rstudio.com/ide/docs/`) has a number of useful tutorials on how to use *knitr* with LaTeX and Markdown. They also have very good documentation for *rmarkdown* at `http://rmarkdown.rstudio.com/`.

That being said, my goal is for this book to be *self-sufficient*. A reader without a detailed understanding of these programs will be able to understand and use the commands and procedures I cover in this book. While learning how to use R and the other programs I personally often encountered illustrative examples that included commands, variables, and other things that were not well explained in the texts that I was reading. This caused me to waste many hours trying to figure out, for example, what the `$` is used for (preview: it's the component selector, see Section 3.1.2). I hope to save you from this wasted time by either providing a brief explanation of possibly frustrating and mysterious things and/or pointing you in the direction of good explanations.

1.6.1 How to read this book

This book gives you a workflow. It has a beginning, middle, and end. So, unlike a reference book, it can and should be read linearly as it takes you through an empirical research processes from an empty folder to a completed set of documents that reproducibly showcase your findings.

That being said, readers with more experience using tools like R or LaTeX may want to skip over the nitty-gritty parts of the book that describe how to manipulate data frames or compile LaTeX documents into PDFs. Please feel free to skip these sections.

More-experienced R users

If you are an experienced R user you may want to skip over the first section of Chapter 3: Getting Started with R, RStudio, and *knitr*/*rmarkdown*. But don't skip over the whole chapter. The latter parts contain important information on the *knitr*/*rmarkdown* packages. If you are experienced with R data manipulation you may also want to skip all of Chapter 7.

More-experienced LaTeX users

If you are familiar with LaTeX you might want to skip the first part of Chapter 11. The second part may be useful as it includes information on how to dynamically create BibTeX bibliographies with *knitr* and how to include *knitr* output in a Beamer slideshow.

Less-experienced LaTeX/Markdown users

If you do not have experience with LaTeX or Markdown you may benefit from reading, or at least skimming, the introductory chapters on these top topics (chapters 11 and 13) before reading Part III.

1.6.2 Reproduce this book

This book practices what it preaches. It can be reproduced. I wrote the book using the programs and methods that I describe. Full documentation and source files can be found at the book's GitHub repository. Feel free to read and even use (within reason and with attribution, of course) the book's source code. You can find it at: `https://GitHub.com/christophergandrud/Rep-Res-Book`. This is especially useful if you want to know how to do something in the book that I don't directly cover in the text.

If you notice any errors or places where the book can be improved please report them on the book's GitHub Issues page: `https://GitHub.com/christophergandrud/Rep-Res-Book/issues`. Corrections will be posted at: `http://christophergandrud.GitHub.io/RepResR-RStudio/errata.htm`.

1.6.3 Contents overview

The book is broken into four parts. The first part (chapters 2, 3, and 4) gives an overview of the reproducible research workflow as well as the general computer skills that you'll need to use this workflow. Each of the next three parts of the book guides you through the specific skills you will need for each part of the reproducible research process. Part two (chapters 5, 6, and 7) covers the data gathering and file storage process. The third part (chapters 8, 9, and 10) teaches you how to dynamically incorporate your statistical analysis, results figures, and tables into your presentation documents. The final part (chapters 11, 12, and 13) covers how to create reproducible presentation documents including LaTeX articles, books, slideshows, and batch reports as well as Markdown webpages and slideshows.

References

Allaire, J. , Cheng, J. , Xie, Y. , McPherson, J. , Chang, W. , Allen, J. , Wickham, H. , and Hyndman, R. (2015a). rmarkdown: Dynamic Documents for R. R package version 0.5.1.

Allaire, J. , Horner, J. , Marti, V. , and Porte, N. (2015b). markdown: 'Markdown' Rendering for R. R package version 0.7.7.

Altman, M. and King, G. (2007). A proposed standard for the scholarly citation of quantitative data. D-Lib Magazine, 13(3/4).

Arel-Bundock, V. (2013). WDI: World Development Indicators (World Bank). R package version 2.4.

Arel-Bundock, V. (2014). countrycode: Convert Country Names and Country Codes. R package version 0.18.

Bååth, R. (2012). The state of naming conventions in R. The R Journal, 4(2):74–75.

Bache, S. M. and Wickham, H. (2014). magrittr: A Forward-Pipe Operator for R. R package version 1.5.

Bacon, F. R. (1267/1859). Opera quaedam hactenus inedita. Vol. I. containing I.–Opus tertium. II.–Opus minus. III.–Compendium philosophiae. Google eBook. Retrieved from http://books.google.com/books?id=wMUKAAAAYAAJ.

Ball, R. and Medeiros, N. (2011). Teaching integrity in empirical research: A protcol for documenting data management and analysis. The Journal of Economic Education, 43(2):182–189.

Barr, C. D. (2012). Establishing a culture of reproducibility and openness in medical research with an emphasis on the training years. Chance, 25(3):8–10.

Boettiger, C. and Temple Lang, D. (2012). Treebase: An R package for discovery, access and manipulation of online phylogenies. Methods in Ecology and Evolution, 3(6):1060–1066.

Bowers, J. (2011). Six steps to a better relationship with your future self. The Political Methodologist, 18(2):2–8.

Box, G. E. and Cox, D. R. (1964). An analysis of transformations. Journal of the Royal Statistical Society. Series B (Methodological), 26:211–252.

Braude, S. (1979). ESP and Psychokinesis. A Philosophical Examination. Temple University Press, Philadelphia, PA.

Buckheit, J. B. and Donoho, D. L. (1995). Wavelab and reproducible research. In Antoniadis, A. , editor, Wavelets and Statistics, pages 55–81. Springer, New York.

Burbidge, J. B. and Robb, L. (1988). Alternative transformations to handle extreme values of the dependent variable. Journal of the American Statistical Association, 83(401):123–127.

Carslaw, D. and Ropkins, K. (2015). openair: Tools for the Analysis of Air Pollution Data. R package version 1.5.

Chang, W. (2012). R Graphics Cookbook: Practical Recipes for Visualizing Data. O'Reilly Media, Inc., Sebastopol, CA.

Chang, W. , Cheng, J. , Allaire, J. , Xie, Y. , and McPherson, J. (2015). shiny: Web Application Framework for R. R package version 0.11.1.

Couture-Beil, A. (2014). rjson: JSON for R. R package version 0.2.15.

Crawley, M. J. (2005). Statistics: An Introduction Using R. John Wiley and Sons Ltd., Chichester.

Crawley, M. J. (2013). The R Book. John Wiley and Sons Ltd., Chichester, 2nd edition.

Creative Commons (2012). Data. http://wiki.creativecommons.org/Data.

Donoho, D. L. (2002). How to be a highly cited author in mathematical sciences. in-cites. http://www.in-cites.com/scientists/DrDavidDonoho.html.

Donoho, D. L. (2010). An invitation to reproducible computational research. Biostatistics, 11(3):385–388.

Donoho, D. L. , Maleki, A. , Shahram, M. , Rahman, I. U. , and Stodden, V. (2009). Reproducible research in computational harmonic analysis. Computing in Science & Engineering, 11(1):8–18.

Dowle, M. , Short, T. , Lianoglou, S. , with contributions from R Saporta, A. S. , and Antonyan, E. (2014). data.table: Extension of data.frame. R package version 1.9.4.

Ehrenberg, A. S. C. (1977). Rudiments of numeracy. Journal of the Royal Statistical Society. Series A General, 140(3):277–297.

Fomel, S. and Claerbout, J. F. (2009). Reproducible Research. Computing in Science & Engineering, 11(1):5–7.
Frazier, M. (2008). Bash parameter expansion. The Linux Journal. Available at: http://www.linuxjournal.com/content/bash-parameter-expansion.
Gandrud, C. (2013a). Github: A tool for social data set development and verification in the cloud. The Political Methodologist, 20(2):2–7.
Gandrud, C. (2013b). The diffusion of financial supervisory governance ideas. Review of International Political Economy, 20(4):881–916.
Gandrud, C. (2015). repmis: Miscellaneous Tools for Reproducible Research. R package version 0.4.2.
Gandrud, C. and Grafström, C. (2015). Inflated expectations: How government partisanship shapes bureaucrats' inflation forecasts. Political Science Research and Methods. Available at: http://dx.doi.org/10.1017/psrm. 2014.34.
Gelman, A. (2011). Tables as graphs: The Ramanujan principle. Significance, 8(4):183.
Gentry, J. (2015). twitteR: R Based Twitter Client. R package version 1.1.8.
Gesmann, M. and de Castillo, D. (2015). googleVis: R Interface to Google Charts. R package version 0.5.8.
Goodrich, B. and Lu, Y. (2007). normal.bayes: Bayesian normal linear regression. Zelig Everyone's Statistical Software. Available at: http://gking.harvard.edu/zelig.
Herndon, T. , Ash, M. , and Pollin, R. (2014). Does high public debt consistently stifle economic growth? a critique of reinhart and rogoff. Cambridge Journal of Economics, 38(2):257–279.
Hlavac, M. (2014). stargazer: LaTeX/HTML code and ASCII text for well-formatted regression and summary statistics tables. R package version 5.1.
Horner, J. (2011). brew: Templating Framework for Report Generation. R package version 1.0-6.
Howe, B. (2012). Virtual appliances, cloud computing, and reproducible research. Computing in Science & Engineering, 14(4):36–41.
Hyndman, R. J. (2010). Transforming data with zeros. Available at: http://robjhyndman.com/hyndsight/transformations/. Accessed March 2015.
Kelly, C. D. (2006). Replicating empirical research in behavioral ecology: How and why it should be done but rarely ever is. The Quarterly Review of Biology, 81(3):221–236.
King, G. (1995). Replication, replication. PS: Political Science and Politics, 28(3):444–452.
King, G. (2007). An introduction to the dataverse network as an infrastructure for data sharing. Sociological Methods & Research, 36(2):173–199.
King, G. , Keohane, R. , and Verba, S. (1994). Designing Social Inquiry. Princeton University Press, Princeton.
Knuth, D. E. (1990). The future of tex and metafont. NTG: Maps, 5:145.
Knuth, D. E. (1992). Literate Programming. CSLI Lecture Notes. Center for the Study of Language and Information, Stanford, CA.
Leifeld, P. (2015). texreg: Conversion of R Regression Output to LaTeX or HTML Tables. R package version 1.35.
Leisch, F. (2002). Sweave: Dynamic generation of statistical reports using literate data analysis. In Härdle, W. and Rönz, B. , editors, Compstat 2002: Proceedings in Computational Statistics, pages 575–580. Physica Verlag, Heidelberg. http://www.stat.uni-muenchen.de/~leisch/Sweave.
Lykken, D. T. (1968). Statistical significance in psychological research. Psychologial Bulletin, 70:151–159.
MacFarlane, J. (2014). Pandoc: A Universal Document Converter. Version 1.13.0.1.
Makel, M. C. and Plucker, J. A. (2014). Facts are more important than novelty: Replication in the education sciences. Educational Researcher, 43(6):304–316.
Matloff, N. (2011). The Art of Programming in R: A Tour of Statistical Programming Design. No Starch Press, San Francisco.
Mesirov, J. P. (2010). Accessible reproducible research. Science, 327(5964):415–416.
Meyer, A. (2006). Repeating patterns of mimicry. PLoS Biol, 4(10).
Munzert, S. , Rubba, C. , Meißner, P. , and Nyhuis, D. (2015). Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining. Wiley, Chichester.
Murrell, P. (2011). R Graphics. Chapman and Hall/CRC Press, Boca Raton, FL, 2nd edition.

Nagler, J. (1995). Coding style and good computing practices. PS: Political Science and Politics, 28(3):488–492.
Nosek, B. A. , Spies, J. R. , and Motyl, M. (2012). Scientific utopia: II. Restructring incentives and practices to promote truth over publishability. Perspectives on Psychological Science, 7(6):615–631.
O'Neal, C. and Schutt, R. (2013). Doing Data Science: Straight Talk from the Frontline. O'Reilly Media Inc., Sebastopol, CA.
Owen, M. (2011). ZeligBayesian: A Zelig Model. R package version 0.1.
Owen, M. , Imai, K. , King, G. , and Lau, O. (2013). Zelig: Everyone's Statistical Software. R package version 4.2-1.
Pemstein, D. , Meserve, S. A. , and Melton, J. (2010). Democratic compromise: A latent variable analysis of ten measures of regime type. Political Analysis, 18(4):426–449.
Peng, R. D. (2009). Reproducible research and biostatistics. Biostatistics, 10(3):405–408.
Peng, R. D. (2011). Reproducible research in computational science. Science, 334:1226–1227.
Peng, R. D. (2014). The real reason reproducible research is important. Simply Statistics. http://simplystatistics.org/2014/06/06/the-real-reason-reproducible-research-is-important/.
Piwowar, H. A. , Day, R. S. , and Fridsma, D. B. (2007). Sharing detailed research data is associated with increased citation rate. PLoS ONE, 2(3):1–5.
R Core Team (2014). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org/.
Ramsey, N . (2011). Noweb: A simple, extensible tool for literate programming. http://www.cs.tufts.edu/~nr/noweb/.
Reinhart, C. and Rogoff, K. (2010). Growth in a time of debt. American Economic Review: Papers & Proceedings, 100.
Ripley, B. and Murdoch, D. (2012). Rtools: Building R for Windows. http://cran.r-project.org/bin/windows/Rtools/.
RStudio, Inc. (2015). RStudio: Integrated development environment for R. Boston, MA. Version 0.99.
Ryan, J. A. (2015). quantmod: Quantitative Financial Modelling Framework. R package version 0.4-4.
Shotts Jr., W. E. (2012). The Linux Command-line: A Complete Introduction. No Starch Press, San Francisco.
Stodden, V. (2009a). The reproducible research standard: Reducing legal barriers to scientific knowledge and innovation. In Communia: Global Science & Economics of Knowledge-Sharing Institutions Torino, Italy June 30.
http://www.stanford.edu/~vcs/talks/VictoriaStoddenCommuniaJune2009-2.pdf.
Stodden, V. (2009b). The legal framework for reproducible scientific research. Computing in Science & Engineering, 11(1):35–40.
Temple Lang, D. (2013). XML: Tools for parsing and generating XML within R and S-Plus. R package version 3.98-1.1.
Temple Lang, D. (2014). RJSONIO: Serialize R objects to JSON, JavaScript Object Notation. R package version 1.3-0.
Temple Lang, D. (2015). RCurl: General network (HTTP/FTP/…) client interface for R. R package version 1.95-4.6.
Therneau, T. M. (2015). survival: Survival Analysis. R package version 2.38-1.
Tufte, E. R. (2001). The Visual Display of Quantitative Information. Graphics Press, Cheshire, CT, 2nd edition.
Ushey, K. , McPherson, J. , Cheng, J. , and Allaire, J. (2015). packrat: A Dependency Management System for Projects and their R Package Dependencies. R package version 0.4.3.
Vaidyanathan, R. , Cheng, J. , Allaire, J. , Xie, Y. , and Russell, K. (2014). htmlwidgets: HTML Widgets for R. R package version 0.3.2.
van Belle, G. (2008). Statistical Rules of Thumb. John Wiley and Sons, Hoboken, NJ, 2nd edition.
Vandewalle, P. (2012). Code sharing is associated with research impact in image processing. Computing in Science & Engineering, 14(4):42–47.
Vandewalle, P. , Barrenetxea, G. , Jovanovic, I. , Ridolfi, A. , and Vetterli, M. (2007). Experiences with reproducible research in various facets of signal processing research. Acoustics, Speech and Signal Processing, 4:1253–1256.

Wickham, H. (2009). ggplot2: Elegant Graphics for Data Analysis. Springer, New York, 2nd edition.
Wickham, H. (2010). A layered grammar of graphics. Journal of Computational and Graphical Statistics, 19(1):3–28.
Wickham, H. (2014a). Advanced R. Chapman and Hall/CRC Press, Boca Raton, FL.
Wickham, H. (2014b). Tidy Data. Journal of Statistical Software, 59(10):1–23.
Wickham, H. (2014c). tidyr: Easily Tidy Data with spread() and gather() Functions. R package version 0.2.0.
Wickham, H. (2015a). httr: Tools for Working with URLs and HTTP. R package version 0.6.1.
Wickham, H. (2015b). rvest: Easily Harvest (Scrape) Web Pages. R package version 0.2.0.
Wickham, H. and Chang, W. (2015a). devtools: Tools to Make Developing R Packages Easier. R package version 1.7.0.
Wickham, H. and Chang, W. (2015b). ggplot2: An Implementation of the Grammar of Graphics. R package version 1.0.1.
Wickham, H. and Francois, R. (2015). dplyr: A Grammar of Data Manipulation. R package version 0.4.1.
Wilson, G. , Aruliah, D. A. , Brown, C. T. , Hong, N. P. C. , Davis, M. , Guy, R. T. , Haddock, S. H. D. , Huff, K. , Mitchell, I. M. , Plumbley, M. D. , Waugh, B. , White, E. P. , and Wilson, P. (2012). Best practices for scientific computing. arXiv, 29 November 2012:1–6. Available at: http://arxiv.org/pdf/1210.0530v3.
World Bank (2015). World development indicators. http://data.worldbank.org/data-catalog/world-development-indicators.
Xie, Y. (2013). Dynamic Documents with R and knitr. Chapman and Hall/CRC Press, Boca Raton, FL.
Xie, Y. (2014). animation: A gallery of animations in statistics and utilities to create animations. R package version 2.3.
Xie, Y. (2015a). formatR: Format R Code Automatically. R package version 1.2.
Xie, Y. (2015b). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.10.