The R Series

Reproducible Research with R and RStudio

Third Edition

Christopher Gandrud

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Reproducible Research with R and RStudio

Third Edition

Chapman & Hall/CRC
The R Series

Series Editors

John M. Chambers, Department of Statistics, Stanford University, California, USA
Torsten Hothorn, Division of Biostatistics, University of Zurich, Switzerland
Duncan Temple Lang, Department of Statistics, University of California, Davis, USA
Hadley Wickham, RStudio, Boston, Massachusetts, USA

Recently Published Titles

The Essentials of Data Science: Knowledge Discovery Using R
Graham J. Williams

blogdown: Creating Websites with R Markdown
Yihui Xie, Alison Presmanes Hill, Amber Thomas

Handbook of Educational Measurement and Psychometrics Using R
Christopher D. Desjardins, Okan Bulut

Displaying Time Series, Spatial, and Space-Time Data with R, Second Edition
Oscar Perpinan Lamigueiro

Reproducible Finance with R
Jonathan K. Regenstein, Jr

R Markdown
The Definitive Guide
Yihui Xie, J.J. Allaire, Garrett Grolemund

Practical R for Mass Communication and Journalism
Sharon Machlis

Analyzing Baseball Data with R, Second Edition
Max Marchi, Jim Albert, Benjamin S. Baumer

Spatio-Temporal Statistics with R
Christopher K. Wikle, Andrew Zammit-Mangion, and Noel Cressie

Statistical Computing with R, Second Edition
Maria L. Rizzo

Geocomputation with R
Robin Lovelace, Jakub Nowosad, Jannes Muenchow

Advanced R, Second Edition
Hadley Wickham

Dose Response Analysis Using R
Christian Ritz, Signe Marie Jensen, Daniel Gerhard, Jens Carl Streibig

Distributions for Modelling Location, Scale, and Shape
Using GAMLSS in R
Robert A. Rigby , Mikis D. Stasinopoulos, Gillian Z. Heller and Fernanda De Bastiani

Hands-On Machine Learning with R
Bradley Boehmke and Brandon Greenwell

Statistical Inference via Data Science
A ModernDive into R and the Tidyverse
Chester Ismay and Albert Y. Kim

Reproducible Research with R and RStudio, Third Edition
Christopher Gandrud

For more information about this series, please visit: https://www.crcpress.com/go/the-r-series

Reproducible Research with R and RStudio

Third Edition

Christopher Gandrud

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

International Standard Book Number-13: 978-0-367-14398-5 (Paperback)
International Standard Book Number-13: 978-0-367-14402-9 (Hardback)

Library of Congress Cataloging-in-Publication Data

Names: Gandrud, Christopher, author.
Title: Reproducible research with R and RStudio / by Christopher Gandrud.
Description: Third edition. | Boca Raton, FL : CRC Press, [2020] | Series: The R series | Includes bibliographical references and index. | Summary: "Brings together the skills and tools needed for doing and presenting computational research. Using straightforward examples, the book takes you through an entire reproducible research workflow"-- Provided by publisher.
Identifiers: LCCN 2019046298 (print) | LCCN 2019046299 (ebook) | ISBN 9780367143985 (paperback) | ISBN 9780367144029 (hardback) | ISBN 9780429031854 (ebook)
Subjects: LCSH: Research--Statistical methods. | R (Computer program language)
Classification: LCC Q180.55.S7 G36 2020 (print) | LCC Q180.55.S7 (ebook) | DDC 001.4/2202855133--dc23
LC record available at https://lccn.loc.gov/2019046298
LC ebook record available at https://lccn.loc.gov/2019046299

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Contents

Preface

Motivation

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 organizing 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 Third Edition

- Spring cleaning: updated package recommendations, examples, and URLs. Removed technologies no longer in regular use.
- More advanced R Markdown and less LaTeX in discussions of markup languages and examples.
- Stronger focus on reproducible working directory tools.

- Updated discussion of cloud storage services and persistently citing reproducible material.
- Added discussion of Jupyter notebooks and reproducible practices in industry.
- Examples of data manipulation with Tidyverse tibbles (in addition to standard data frames) and `pivot_longer()` and `pivot_wider()` functions for pivoting data.
- Naming conventions are in current R-Tidyverse best practice.

A detailed list of changes for the third edition is available at `https://github.com/christophergandrud/Rep-Res-Book/issues/57#issuecomment-421739971`.

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.

Acknowledgments

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 & Hall. He approached me in Spring 2012 with the general idea and opportunity for this book. Other editors at Chapman & Hall and Taylor & 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

Many other anonymous reviewers also gave great feedback over the years.

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.

The library at the University of California, San Francisco was a great home for writing the third edition.

Kristina Gandrud has been immensely supportive and patient with me throughout the writing of this book (and my entire career).

About the Author

Christopher Gandrud is Head of Economics and Experimentation at Zalando SE. He leads teams of social data scientists and software engineers building and evaluating large-scale automated decision-making systems. He was previously a research fellow at the Institute for Quantitative Social Science, Harvard University developing statistical software for the social and physical sciences. He has held posts at City, University of London, the Hertie School of Governance, Yonsei University, and the London School of Economics where in 2012 he completed a PhD in quantitative political science.

Stylistic Conventions

I use the following conventions throughout the book:

- **Abstract variables**: Abstract variables, i.e. variables that do not reference specific objects, are in `ALL CAPS TYPEWRITER TEXT`.
- **Clickable buttons**: Clickable buttons are in `typewriter text`.
- **Code**: All code is in `typewriter text`.
- **File names and directories**: File names and directories more generally are printed in *italics*. Words are separated by em dashes—*kebab-case*.[1]
- **File extensions**: Like file names, 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 underscores (_) to separate words in object names.
- **Object columns**: Data frame object columns are printed in **bold**.
- **R Function names**: are followed by parentheses (e.g., `stats::lm()`)
- **Packages**: **R** packages are printed in *italics*. When a system, rather than the package that shares its name is referred to, it is not italicized, e.g. R Markdown (system) vs. *rmarkdown* (package).[2]
- **Windows and RStudio panes**: Open windows and RStudio panes are written in *italics*.
- **Variable names**: Variable names are printed in **bold**. Underscores (_) separate words in variable names.

[1] See `https://stackoverflow.com/a/17820138`. Posted 23 July 2013.

[2] See Yihui Xie's comment at: `https://andrewgelman.com/2016/01/14/rstanarm-and-more/#comment-259425`. Posted 14 January 2016.

Additional Resources

You can freely download additional resources supplementing examples in this book. 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-book-v3-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-book-v3-examples`. Please follow the replication instructions in the main *README.md*. It is 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 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 (2018) 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()`

functions. A *Makefile* can run *gather-1* and *gather-2.R* to gather and clean the data. It runs *merge-data.R* to merge the data into one data file called *main-data.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 *main-data.csv* with the `import()` function. These files are run by the presentation documents when they are knitted. The presentation documents tie to the analysis documents with *knitr* and the `source()` function.

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.

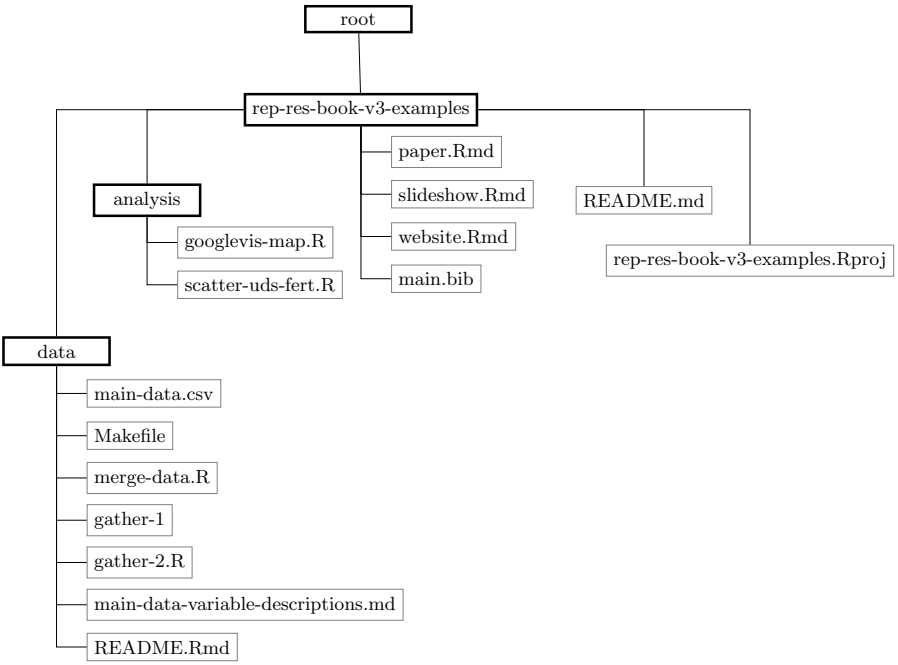

FIGURE 1: Short Example Project File Tree

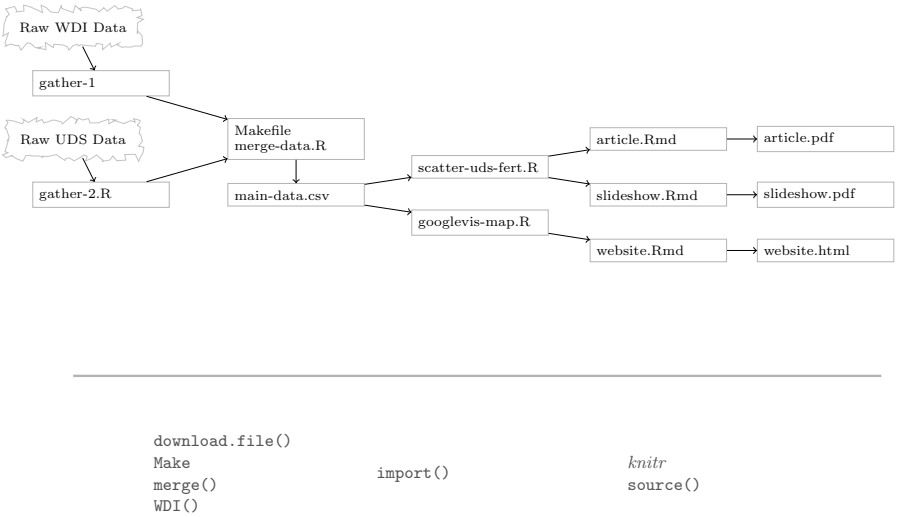

FIGURE 2: Short Example Main File Ties

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`.

Part I

Getting Started

1

Introducing Reproducible Research

Research is typically presented in very selective containers: slideshows, journal articles, books, or 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 easier and more robust. 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 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.

1.1 What Is Reproducible Research?

Though there is some debate over the necessary and sufficient conditions for a full replication (Makel and Plucker, 2014, 2), research results are generally considered[1] *replicable* if there is sufficient information available for independent researchers to make the same findings using the same procedures with new data.[2] 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 results. 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).[3] In computational sciences[4] 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

[1] Rokem et al. (2018, 3-4) note that some disciplines, e.g. computing machinery and meteorology, give "replicable" and "reproducible" the exact opposite meanings from the way they are used in this book and many other disciplines such as biology, economics, and epidemiology.

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

[3] The really reproducible computational research originates in the 1980s and early 1990s with Jon Claerbout and the Stanford Exploration Project (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).

[4] 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.

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*"[5] 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.

1.2 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 contributions 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

[5] 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, the term "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.

that are not replicable should be abandoned or modified "when confronted with more complete or reliable . . . evidence".[6]

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 have been noise, even though it is fully reproducible.

Avoiding effort duplication and 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.

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

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.[7]

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 a new 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 to 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).

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 make changes much easier. Changes made to

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