Packaging data analytical work reproducibly using R (and friends)

Ben Marwick
University of Washington, University of Wollongong
and
Carl Boettiger
University of California, Berkeley
and
Lincoln Mullen
George Mason University

March 20, 2018

Abstract

Computers are a central tool in the research process, enabling complex and large scale data analysis. As computer-based research has increased in complexity, so have the challenges of ensuring that this research is reproducible. To address this challenge, we review the concept of the research compendium as a solution for providing a standard and easily recognisable way for organising the digital materials of a research project to enable other researchers to inspect, reproduce, and extend the research. We investigate how the structure and tooling of software packages of the R programming language are being used to produce research compendia in a variety of disciplines. We also describe how software engineering tools and services are being used by researchers to streamline working with research compendia. Using real-world examples, we show how researchers can improve the reproducibility of their work using research compendia based on R packages and related tools.

Keywords: Reproducible research, Open source software, Computational science, Data science

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

1 Introduction

The purpose of this paper is to show how the R package can be a suitable template for organising files into a research compendium to enhance the reproducibility of research. We first establish a working definition of reproducibility – a term with diverse meanings across different domains – and then describe the 'what', 'why', and 'how' of using R packages as research compendia. We present real-world examples of research compendia of low, moderate, and high complexity to show how flexible the R package format is for this purpose. We conclude by reviewing some useful tools and templates that we have found to make it easy to create and share research compendia.

Long considered an axiom of science, the reproducibility of scientific research has recently come under scrutiny after some highly-publicised failures to reproduce biomedical studies (Begley & Ellis 2012) and psychological experiments (Open Science Collaboration et al. 2015). This has resulted in extensive discussion of how to define reproducibility, and how to enhance it across diverse research contexts (Stodden et al. 2016, Goodman et al. 2016, Marwick 2016). Here we use the definition proposed by Claerbout (1992) and the National Science Foundation Subcommittee on Replicability in Science (Bollen et al. 2015): reproducing research is the calculation of quantitative scientific results by independent researchers using the original data and methods. Stodden (2014) further divides reproducibility into three dimensions: empirical (e.g. details of reagents, cell lines, sample identities, instrument settings), statistical (e.g. details of statistical tests, model parameters, threshold values, etc.) and computational (e.g. details about code, software, hardware, and implementation). Our focus here is on the computational dimension, which is important because it transcends many of the domain-specific technical issues of empirical and statistical reproducibility. Indeed, the reproducibility of computational results has become an important consideration in fields such as digital history, digital humanities, and the social sciences (Graham et al. 2015, pp. 149, 157; Allison 2016; Goldstone 2017).

Virtually all researchers use computers as a central tool in their workflow. However, our formal education rarely includes any training in how to organise our computer files to make it easy to reproduce results and share our entire analysis pipeline with others. Without clear instructions, many researchers struggle to avoid chaos in their file structures, and so

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

are understandably reluctant to expose their workflow for others to see. This may be one of the reasons that so many requests for details about method, including requests for data and code, are turned down or go unanswered (Collberg & Proebsting 2016).

Scholarship will be strengthened if we are more open with research materials, yet we lack conventions and technical infrastructure for such openness. We see evidence of this need in the proliferation of commentary and how-to publications across different areas of science (Wilson et al. 2014, Stodden & Miguez 2014, Stanisic & Legrand 2014, LeVeque et al. 2012, Donoho 2010, Sandve et al. 2013, Rokem et al. 2017, Munafo et al. 2017). There are also some forward-looking journal editors attempting to shift the norms of their research community by encouraging or requiring authors to make available their data and code files when submitting their manuscript for publication. For example, *Biostatistics* invites authors to submit materials for a reproducibility review after the article has been accepted (Peng 2009). Authors submitting to *Political Analysis* and the *Quarterly Journal of Political Science* are required to provide their code for review, with editors reproducing the reported analyses prior to publication (Nosek et al. 2015). The *Journal of Cultural Analytics* requires authors to deposit code and dataset (Piper et al. 2017). A cross-discipline system of signalling open data and materials accompanying journal articles with badges is being used to incentivize openness in a dozen journals (Kidwell et al. 2016).

Our contribution to this drive to improve the openness and reproducibility of research is to show how R packages can be used as a research compendium for organising and sharing files. Although the R packaging system is traditionally a method for sharing statistical methods, we claim that R packages are suitable for use as research compendia that can help improve computational reproducibility. Our focus is on the R programming language because the R package structure is uniquely suitable to being easily adapted to solve problems of organising files and sharing them with other researchers. We describe how the conventional structure of R packages can be adapted for use as a research compendium, and illustrate this use with real-world examples.

We recognize that not all researchers are working in R, and that even those who are may not adopt our recommendations wholesale. Nevertheless, the principles of research compendia that we describe are generally applicable to projects that involve computation

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

using any language, and many of our recommendations about specific software can be adopted individually.

2 What is a research compendium? Three generic principles

The goal of a research compendium is to provide a standard and easily recognisable way for organising the digital materials of a project to enable others to inspect, reproduce, and extend the research. There are three generic principles that define research compendia, independent of particular software tools and disciplinary contexts.

1. A research compendium should organize its files according to the prevailing conventions of the scholarly community, whether that be an academic discipline or a lab group. Following these conventions will help other people recognize the structure of the project, and also support tool building which takes advantage of the shared structure.

2. A research compendium should maintain a clear separation of data, method, and output, while unambiguously expressing the relationship between those three. In practice, this means data files must be separate from code files. This is important to let others easily identify how the original researcher operated on the data to generate the results. Keeping data and method separate treats the data as "read-only," so that the original data is untouched and all modifications are transparently documented in the code. The output files should be considered as disposable, with a mindset that one can always easily regenerate the output using the code and data. The relationship between which code operates on which data in which order to produce which outputs must be specified as well. In his advice to industry data scientists, Ben Baumer's paper in this collection similarly highlights the importance of keeping data separate from the presentation of data, or research outputs.

3. A research compendium should specify the computational environment that was used for the original analysis. At its most basic, this could be a plain text file that includes

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

a short list of the names and version numbers of the software and other critical tools used for the analysis. In more complex approaches, described below, the computational environment can be automatically preserved or reproduced as well.

Some of the earliest examples of these principles in action can be found in the work of Claerbout (1992) and his colleagues in the early 1990s. Claerbout introduced these practices in his Stanford geophysics research group (Buckheit & Donoho 1995). Other early examples of compendia appeared in econometrics (Koenker 1996), (Vinod 2001), epidemiology (Peng et al. 2006), and signal processing (Vandewalle et al. 2009). Two influential papers are Gentleman's (2005) article "Reproducible Research: A Bioinformatics Case Study" and Temple Lang's paper "Statistical Analyses and Reproducible Research" (Gentleman & Temple Lang 2004; 2012). The impact of these two papers is due to their clear description of an easy-to-use set of readily available tools based on the R programming language.

One of the most compelling tools described by Gentleman and Temple Lang is Sweave (Leisch 2002). This is an environment that facilitates writing "dynamic documents." A dynamic document is a single document that includes both the narrative text of a paper, and the R code that generates the plots and tables it includes. This document can be executed to run the R code and format the text to produce a PDF document that includes the output of the R code in the source document. By enabling authors to write text and code in a single document, Sweave enables Knuth's (1992) literate programming methods for R users.

Since those papers were published, there has been a substantial increase in the use of R as a research tool in many fields (Tippmann 2014). At the same time, there have been substantial improvements in the ease of making dynamic documents in R (especially knitr and rmarkdown; Xie 2015, Baumer & Udwin 2015, Allaire et al. 2016) and in making R packages (especially devtools and roxygen2; Wickham & Chang 2016, Gandrud 2013). This means that making a research compendium based on an R package is now a practical solution to the challenges of organising and communicating research results.

That said, there are related efforts to support making research compendia that do not use the R package structure. For example, Gandrud (2013) advocates a reproducible research project structure that is consistent with the principles above, but does not use

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

any elements of an R package. Likewise, ProjectTemplate (White 2014), recommended by Hilary Parker in her paper in this collection, is an R package that includes functions to generate a pre-determined directory structure to organise files, and auto-load packages, data and functions. Other similar packages for generating empty project directory structures include makeProject and prodigenr. A notable effort to promote reproducible research at the undergraduate level is Project TIER (Teaching Integrity in Empirical Research) for economics majors at Haverford College. Project TIER provides a detailed file-structure protocol for students working on their senior thesis, aiming to teach students to document their data management and analysis to enable an independent researcher to reproduce the student's data processing and analysis (Ball & Medeiros 2012). A third influential approach to organising research compendia can be found in lessons developed by the Reproducible-Research-Curriculum community (Curriculum 2016), the Data Carpentry (Teal et al. 2015) and Software Carpentry (Wilson 2013) organisations, Jenny Bryan's (2016) materials for her STAT545 course, and Karl Broman's tutorials (2016). These lessons are less prescriptive than Gandrud (2013) and ProjectTemplate, focusing more on high-level guidance similar to the generic principles we described above.

3 Why create a research compendium?

Using research compendia has benefits both for you as a researcher, and for the community that you work in. Among the most important of these benefits is that a research compendium is a convenient way to publicly share data and code (McKiernan et al. 2016). Papers with publicly available datasets receive a higher number of citations than similar studies without available data (Piwowar et al. 2007, Piwowar & Vision 2013, Henneken & Accomazzi 2011, Dorch 2012, Sears 2011, Gleditsch & Strand 2003). In addition to increased citations for data sharing, Pienta et al. (2010) found that data sharing is associated with higher publication productivity. They examined 7,040 NSF and NIH awards and concluded that the typical research grant award produces a median of five publications, but when data are archived a research grant award leads to a median of ten publications. Other benefits are documented elsewhere (Markowetz 2015, Donoho 2010).

In our own experience as researchers, we enjoy benefits of increased efficiency through

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

simplified file management and streamlined analytical workflows. These help us to work more transparently and efficiently. While reproducibility is no guarantee of correctness, making our results reproducible lets us inspect our own work for errors. Because we follow the same pattern in multiple projects, the startup and re-entry costs for each project are significantly reduced. The result is that we save time, and minimize the risk of errors that might result from results which cannot be reproduced. A compendium makes it easier to communicate our work with others, including collaborators and (not least of all) our future selves. As members of research communities, by including compendia with our published papers we benefit from increased impact and visibility of our research among our peers.

In the specific case of using an R package as a research compendium, you can benefit from the quality control mechanisms required to successfully build an R package. By organizing files into an R package, you follow conventions that save you time thinking about the best way to organise your project. Writing a function saves us from introducing errors that often result from repeated copying and pasting of code. This is because you can type one line of code (the name of your function) that calls hundreds of lines of code (your function). Writing functions for packages makes it easy to document the use of the code (particularly if you use roxygen2). This documentation can help you be productive more quickly when returning to work on a project after stepping away from it. When you iteratively develop a package, you can easily run comprehensive checks on your code with `R CMD check` at the terminal or `devtools::check()` at the R console. Regularly checking your package like this can help identify problems before they become very frustrating to solve.

4 How to make a research compendium?

At its simplest, a research compendium might consist of a single file of R code with in-line comments documenting the workflow. A slightly more complex approach might be a R markdown file with text and code in the same source document, and accompanying data files. In many cases, these simple approaches will be sufficient, and more elaborate organisation would add unnecessary complexity and points of failure. However, as a project grows you might start with a single file, then add subdirectories which have consistent and

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

readily meaningful names such as "analysis", "data", and so on. There are few strict rules here. The key principle is to organize the compendium so that another person can know what to expect from the plain meaning of the file and directory names. Using widely held conventions, such as the R package structure, will help other people to understand how your files relate to each other without having to ask you. Naming objects is difficult to do well, so it's worth to put some effort into a logical and systematic file naming convention if you have a complex project with many files and directories (for example, a multi-experiment study where each experiment has numerous data and code files).

The R package structure can help with providing a logical organisation of files, by providing a set of standard locations for certain types of files. An ideal package-based file organisation for a more complex project would look like this:

- A **README.md** file that describes the overall project and where to get started. It can be helpful to include graphical summary of the interlocking pieces of the project.
- Script files with reusable functions go in the `R/` directory. If these functions are documented using Roxygen, then the documentation will be automatically generated in a `man/` directory.
- Raw data files are kept in the `data/` directory. If your data are very large, or streaming, an alternative is to include a small sample dataset so that people can try out the techniques without having to run very expensive computations.
- Analysis scripts and reports files go in the `analysis/` directory. In many cases it can be useful to give the analysis scripts ascending names, e.g. 001-load.R, 002-clean.R etc. This kind of file-naming helps with organisation, but it doesn't capture the full tree of dependencies in the way a `Makefile` or an R Markdown file does. To manage more complex workflows, the `analysis/` directory could include either an R markdown file, a `Makefile` or a `Makefile.R` file. These files are important because they control the order of the code execution. In more complex projects careful use of caching or a `Makefile` can save time by only running code that hasn't changed since it was last run.
- A **DESCRIPTION** file in the project root provides formally structured, machine- and human-readable information about the authors, the project license, the software

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

dependencies and other metadata of the compendium. When a `DESCRIPTION` file is included along with the other items above, then the compendium is also a formal, installable R package. When your compendium is an R package, you can take advantage of many time-saving tools for package development, testing, and sharing (for example, the `devtools` package that we noted above). R's built-in `citation()` function can use that metadata, along with references to any publications that result from the project, to provide users with the necessary information to cite your work.

We have developed the rrtools package to bootstrap the creation of research compendia packages that conform to the minimum ideal requirements listed above (Marwick et al. 2017).

5 Examples of real-world research compendia using R packages

In this section we describe a selection of real-world package-based research compendia of varying complexity. Our hope is that these examples will give practical guidance about how you can organise your projects into package-based compendia.

5.1 Small compendia

A simple example might look like this:

Duffy's (2015) parasite study provides an excellent real-world example of this simple research compendium format. Her compendium is archived on zenodo (`http://dx.doi.org/10.5281/zenodo.17804`), with the development repository on GitHub (`https://github.com/duffymeg/BroodParasiteDescription`). This format fulfills the three generic principles described above: it organizes the files according to a convention, separates data and code, and specifies the computational environment. The `DESCRIPTION` file shows that R version 3.2.0 or higher is required for this project, and the packages that the project used are also listed there.

Although Duffy's compendium is an excellent template for many research projects, it

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

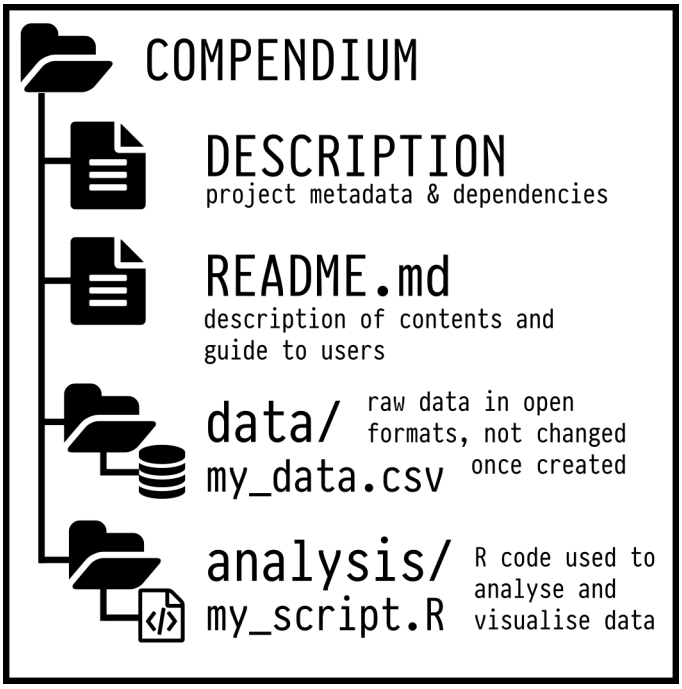

Figure 1:

might not qualify as a compendium as defined by Gentleman and Temple Lang because it does not contain a dynamic document. An R script file, however, can be converted to an R Markdown document which includes both text and code. The diagram below shows an intermediate-level compendium. Compared to the simple compendium described above, the `my_scripts.R` file has become `my_report.Rmd`. Another change is the addition of a `LICENSE` file that specifies how the contents of the compendium are allowed to be reused (more on this below in the section 'How to share a research compendium?').

The example above is based on the real-world compendium of the zooarchaeological study by Conrad et al. (2016). Their compendium is archived at the University of New Mexico (`http://repository.unm.edu/handle/1928/26730`) and the development version is on GitHub (`https://github.com/cylerc/AP_SC`). The R Markdown file generates all the data visualisations and statistical contents of the publication, with some basic commentary on the methods. However, it does not contain the complete manuscript as submitted to the publisher.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

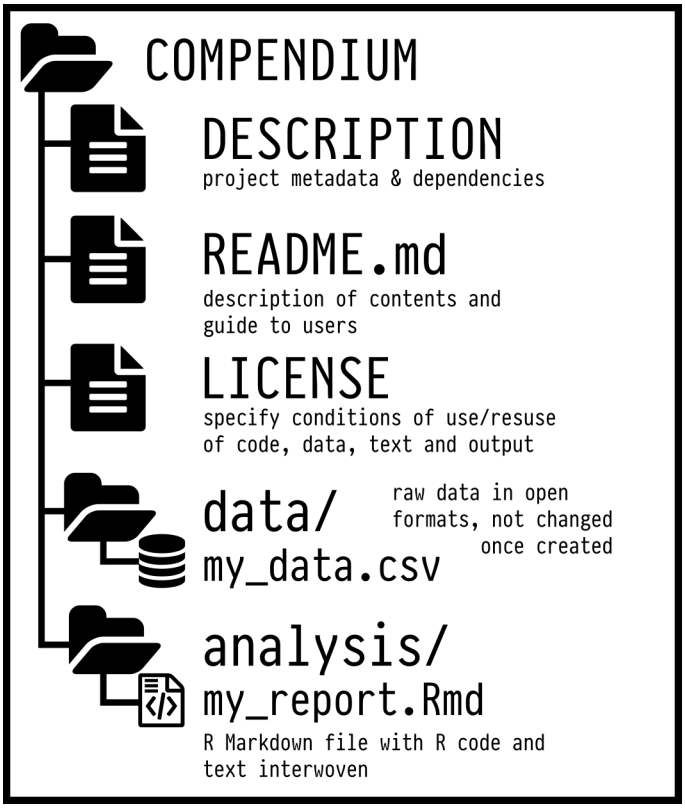

Figure 2:

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

5.2 Medium compendia

While the two examples above qualify as minimal R packages, our next example shows a more complete use of the R package structure. This includes the `R/` and `man/` directories. The `R/` directory contains custom functions that are used repeatedly throughout the project. The `man/` directory contains the manual (i.e., documentation) for the use of the functions. The `NAMESPACE` file is a feature of R packages that exports the functions in a package so that they can be used in the current project and by other packages.

A real-world compendium demonstrating this more complete use of the R package structure is Hollister et al. (2016). Their archived compendium is online at zenodo (`http://dx.doi.org/10.5281/zenodo.40271`) and the development repository is at GitHub (`https://github.com/USEPA/LakeTrophicModelling`). It differs slightly from the diagram above because Hollister et al. have their main manuscript in the `vignettes/` directory. The vignette is a traditional component of an R package for providing detailed examples of how package functions can be used. By including the manuscript as a vignette, it can be automatically generated from the R Markdown document when the package is installed. Their vignette document includes the full text of the manuscript, as well as the figures and tables generated by R, along with the usual scholarly apparatus such as cross-references, citations and a reference list. A similar example of this use of an R package as a research compendium can be seen in Negre et al. (2016).

5.3 Large compendia

The diagram below outlines one possible structure of a compendium at the more complex end of the spectrum. This example includes continuous integration; `.travis.yml` is a configuration file for the Travis continuous integration service. The `Dockerfile` is a file that instructs the Docker software how to create a virtual environment for running the R code in an isolated and portable context. The `Makefile` is a file with instructions for executing R code in the compendium in a way that avoids unnecessary repetition. These three files contain organizing metadata that are intended to be both machine- and human-readable, but are not written in the R language. Unit tests, contained in the `tests/` directory, are R code scripts to test that specific functions in the compendium produce their

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

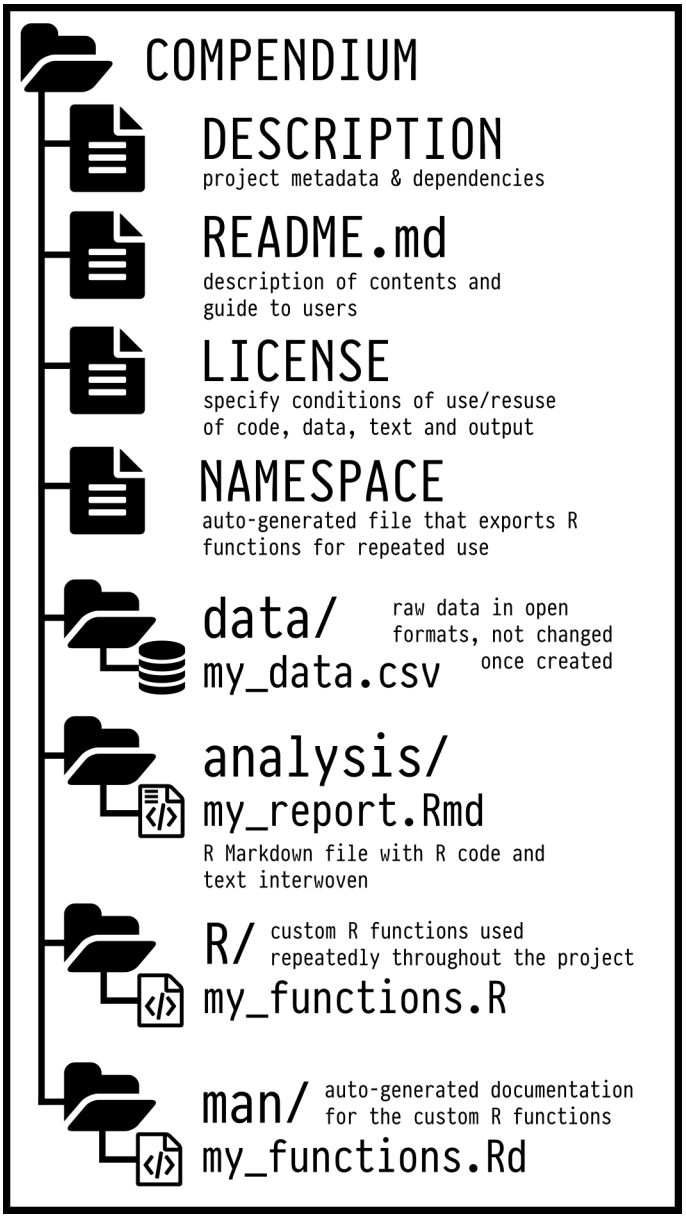

Figure 3:

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

expected outputs, given known inputs. The addition of these tools, initially developed for software engineering, solves the problems of specifying the computational environment and the relationships between data, code, and output.

A real-world example of this more complex type of research compendium can be seen in Boettiger, et al. (2015). Their archived compendium is online at zenodo (`http://dx.doi.org/10.5281/zenodo.12669`) and the development repository is at GitHub (`https://github.com/cboettig/nonparametric-bayes`). Boettiger et al. have a top-level `manuscripts/` directory that holds the files that generate the journal article and supplementary documents, as well as a `Makefile` and a `Dockerfile`. Other notable items at the top level of this compendium are the `.drone.yml`, `.zenodo.json` and `.travis.yml` files. The drone file contains configuration details for the Drone continuous integration service that operates in Docker containers. In this case, each time a commit is made to the repository, the Drone web service automatically renders the manuscript files (including running the R code in the manuscript) to PDF in a Docker container defined by the `Dockerfile` in `manuscripts/`. This provides an automatic check to see if a PDF can be generated from the manuscript R markdown files after the latest commit. The `.travis.yml` file performs a similar purpose, but is focused on whether or not the R package can successfully build in a generic Linux environment. The `.zenodo.json` file provides machine-readable metadata about the compendium. This is useful for automatically archiving the compendium at zenodo. Additional examples of similarly complex compendium can be found in Clarkson et al. (2015), Marwick et al. (2016), Marwick, Van Vlack, Conrad, Shoocongdej, Thongcharoenchaikit & Kwak (2017), Marwick, Hayes, Clarkson & Fullagar (2017) and Eglen (2016). These examples indicate that this use of the R package is a sustainable and efficient approach to packaging research reproducibly.

5.4 Makefiles

The `Makefile` in the example above and in Boettiger's compendia uses the `make` language (rather than R) to concisely specify the relationship between data, the output, and the code that generates it. A `Makefile` defines outputs (called "targets") in terms of inputs (called "dependencies") and the code necessary to produce them (called "recipes"). The

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

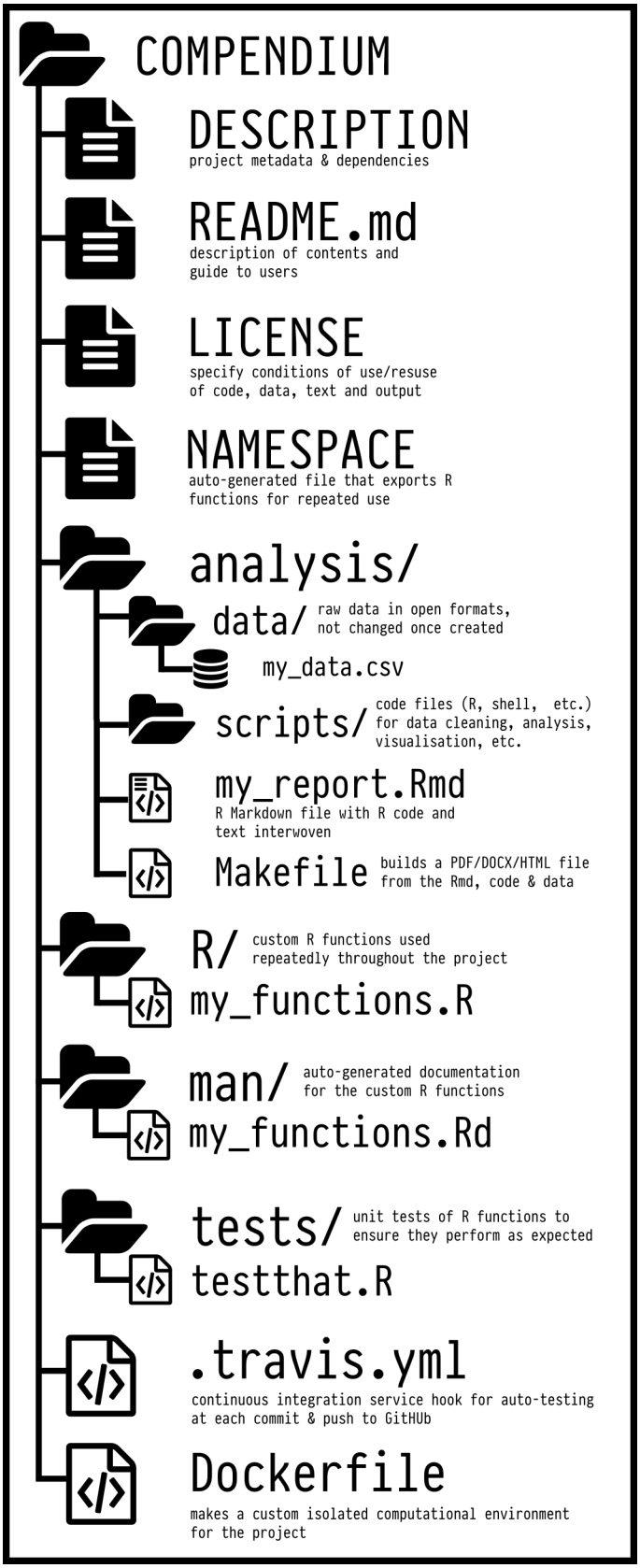

Figure 4:
15

program GNU Make then creates the outputs from that specification, see Broman (2013) for more details. In addition to specifying the relationships between outputs and inputs unambiguously, Make also lets you re-build only those parts of a project that are out of date, saving time for lenghty computations. In addition to GNU Make, the remake package (FitzJohn 2016) allows you to write Make-like instructions entirely in R.

5.5 Isolating the computational environment

The `Dockerfile` in these more complex examples helps to specify an isolated computational environment that includes all the dependencies necessary for your analysis to run (Boettiger 2015). Using a lightweight virtualisation system such as Docker minimizes problems with changes in R packages that might break your code, since Docker will preserve the exact versions of R and the packages that you used for your analysis. Docker makes it easy to share your project because when you share your Docker container with collaborators, you don't need to worry about differences between operating systems. This is important if your analysis requires software outside of the R programming environment.

While Docker will virtualize R, its packages, and the entire operating system, there are other solutions that are focused exclusively on the R environment. The packrat package, for example, downloads the source code of the exact versions of the packages you have used in your project, and stores them with your project (Ushey et al. n.d.). Another approach is provided by checkpoint (Microsoft Corporation 2016) which installs specific previous versions of R packages from the Microsoft R Archived Network. The devtools package (Wickham & Chang 2016) similarly enables installation of previous versions of packages from CRAN (the Comprehensive R Archive Network, the default R package repository). These packages can protect you from updates to packages that might stop your code from working, and help you and your collaborators work with the same package versions.

5.6 Continuous integration

With continuous integration, each Git commit you make to your repository on GitHub (or similar service) triggers a script that builds your R package, and reports to you if the build succeeded or not. This is convenient because it saves you from having to manually build

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

your package after each update to your code. The complex research compendia examples cited above make use of Travis CI, Drone.io and CircleCI services. These provide free remote continuous integration services for public GitHub projects. Another advantage of these services is that they provide badges for your repository webpage that signal the current state of your compendium, based on the last build attempt (e.g. 'build passing' or 'build failing').

6 How to share a research compendium?

To share a research compendium you need to think about four separate but related issues: 1) licensing, 2) version control, 3) persistence, and 4) metadata.

A license file indicates to others how your work may be re-used. You should of course seek specific intellectual property advice relevant to your work from your organisation, as well as the policies of any publisher distributing material from the compendium (such as an associated academic article). While research compendia typically use open source software licenses (see `http://opensource.org`), authors should be aware that such licenses are designed for software and may not always be well suited for other elements of a compendium. In many jurisdictions, data are considered facts rather than a creative work, so they are not protected by copyright law. Stodden (2009) recommends using the Creative Commons Public Domain declaration (CC-0) for data. A compendium may also include documentation such as a scientific article and figures that are more appropriately licensed under a Creative Commons Attribution (CC-BY) license. Note that some data repositories (such as `DataDryad.org` or figshare.com, discussed below) will require a CC-BY or CC-0 license for all submitted content.

A version control system such as Git is the best way to preserve the history of changes to the research compendium. Version control facilitates both private collaboration among colleagues on the project and the distribution and maintenance of the compendium in the future (Ram 2013, Jones 2013, Loeliger & McCullough 2012). Bryan's paper in this collection is a detailed and accessible introduction to using Git in research contexts.

Once you are ready to share your compendium, the best way is to archive a specific commit at a repository that issues persistent URLs, such as a Digital Object Identifier

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

(DOI), for file archives (e.g. osf.io, figshare.com or zenodo.org). DOIs are designed to be far more persistent than other URLs, which often break as web pages change over time (Klein et al. 2014). There are many discipline-specific DOI-issuing repositories, listed at re3data.org. You should survey the literature of your research community to see what repositories are preferred.

DOIs provided by data repositories have another advantage over other web links: when a data repository provides an author with a DOI, it must also provide basic metadata information to a central registry which oversees the creation of DOIs: either CrossRef (primarily journal articles) or DataCite (data and other products). Once your compendium is hosted at a repository, you include the DOI to the compendium in your reports and publications. This means you have a publicly available snapshot of the code that matches results in the paper. A DOI also simplifies citation of the compendium, so you can cite it in your paper (and others can cite it in their work).

Code development can continue after the paper is published, but with a DOI that links to a specific commit, you and other users of your code can be confident that they have the version that matches the paper. Putting the compendium on Dropbox, Google Drive, or similar services is another way to make it easily available. However, these services don't offer the same persistence as a DOI-issuing repository, and are suitable only for short-term, temporary storage during development.

While CRAN is one of the biggest and best-known systems for archiving and distributing R packages, we do not recommend it for research compendium packages. The main reason that CRAN is not suitable is that it is very strict about the directory structure and contents for the R packages that it accepts. For example, the top-level `analysis/` directory in several of the examples above would not be allowed in a package hosted on CRAN. There are ways around this (e.g. use the `inst/` or `vignettes/` directory, cf. Hollister's compendium), but this makes the package structure less intuitive. This has the disadvantage of making the compendium harder to navigate and so less accessible to others.

A second reason why CRAN is not suitable for research compendium packages is that CRAN has a 5 MB size limit for package data and documentation. Hollister's package is 6 MB after installation (from GitHub), and many research compendia exceed this size

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

because of moderate to large raw data files, image files, cached computation results, etc., so these could not be hosted on CRAN. The DOI-issuing repositories that we describe above are more suitable than CRAN for research compendium packages because they have more generous file size limits, and no constraints on directory organization.

7 Useful tools and templates for making research compendia

Probably the most useful set of tools for making research compendia as R packages is the `devtools` package (Wickham & Chang 2016), combined with the RStudio integrated development environment. Both of these offer useful shortcuts for meeting the standards of an R package. Our rrtools package extends functions in devtools and provides instructions, templates, and functions for making a basic compendium-package suitable for doing reproducible research with R (Marwick et al. 2017). Wickham's (2015) book *R Packages* is a useful in-depth guide to the format of R packages. The manual *Writing R Extensions* is included with every R installation and describes the process of creating R packages.

More specifically, several people have developed templates for using R packages as research compendia. These templates are mostly for their personal use, and are works-in-progress, but are freely available for others to adapt and learn from:

- Jeff Hollister's manuscriptPackage
- Carl Boettiger's template
- Francisco Rodriguez-Sanchez's template
- Ben Marwick's researchcompendium
- Lincoln Mullen's example Makefiles in R-package like repositories

For writing research papers in R Markdown, a useful package is bookdown, which enables figure and table captions and cross-referencing. There are many R Markdown templates for journal articles from certain publishers in the rticles package. These packages and templates, together with RStudio add-ins such as the wordcountaddin for character, word and sentence counts, and citr for inserting formatted citations, provide a rich scholarly

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

writing environment for working with R packages in RStudio. See also rOpenSci's Reproducibility Guide (rOpenSci et al. 2017) for additional community-contributed templates and resources.

8 Conclusion

In summary, we have defined the general principles of research compendia, and described a number of different ways that researchers have organized their work to meet those criteria. For a researcher working primarily in R, a compendium based on the R package standards, along with additional tools drawn from the software engineering toolkit, is a natural, efficient, and reliable way of organizing one's work. Researchers working in different languages can draw ideas from the general principles and example projects presented here. We encourage researchers to strive to organize their code in reproducible ways and treat their code and data as much a product of their scholarship worthy of distribution as the articles themselves.

9 Acknowledgements

This essay originated in discussions at the 2015 rOpenSci unconference (cf. `https://github.com/ropensci/unconf/issues/11` and `https://github.com/ropensci/unconf/issues/31`). Thanks to those who were part of these discussions and helped to broaden our understanding of compendia and R packages. BM was also supported by an Australian Research Council Future Fellowship (FT140100101).

References

Allaire, J. et al. (2016), *rmarkdown: Dynamic Documents for R*. R package version 0.9.6. **URL:** *https://CRAN.R-project.org/package=rmarkdown*

Allison, S. (2016), 'Other people's data: Humanities edition', *Journal of Cultural Analytics*

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

.

URL: *http://culturalanalytics.org/2016/12/other-peoples-data-humanities-edition/*

Ball, R. & Medeiros, N. (2012), 'Teaching integrity in empirical research: A protocol for documenting data management and analysis', *The Journal of Economic Education* **43**(2), 182–189.

Baumer, B. & Udwin, D. (2015), 'R markdown', *Wiley Interdisciplinary Reviews: Computational Statistics* **7**(3), 167–177.

Begley, C. G. & Ellis, L. M. (2012), 'Drug development: Raise standards for preclinical cancer research', *Nature* **483**(7391), 531–533. 10.1038/483531a.

URL: *http://dx.doi.org/10.1038/483531a*

Boettiger, C. (2015), 'An introduction to Docker for reproducible research', *ACM SIGOPS Operating Systems Review* **49**(1), 71–79.

Boettiger, C., Mangel, M. & Munch, S. (2015), 'Avoiding tipping points in fisheries management through gaussian process dynamic programming', *Proceedings of the Royal Society of London B: Biological Sciences* **282**(1801), 20141631.

Bollen, K., Cacioppo, J., Kaplan, R., Krosnick, J. & Olds, J. (2015), 'Social, behavioral, and economic sciences perspectives on robust and reliable science: Report of the subcommittee on replicability in science, advisory committee to the national science foundation directorate for social, behavioral, and economic sciences', *Retrieved from the National Science Foundation Web site: www. nsf. gov/sbe/AC_Materials/SBE_Robust_and_Reliable_Research_Report. pdf* .

Broman, K. (2013), 'minimal make: A minimal tutorial on make'.

URL: *http://kbroman.org/minimal_make*

Broman, K. (2016), 'Organize your data and code'.

URL: *http://kbroman.org/steps2rr/pages/organize.html*

Bryan, J. (2016), 'Stat545: Data wrangling, exploration, and analysis with R'.

URL: *http://stat545.com/*

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

Buckheit, J. B. & Donoho, D. L. (1995), 'Wavelab and reproducible research', *Wavelets and Statistics. Lecture Notes in Statistics* **103**, 55–81.

Claerbout, J. F. & Karrenfach, M. (1992), 'Electronic documents give reproducible research a new meaning'.

Clarkson, C., Smith, M., Marwick, B., Fullagar, R., Wallis, L. A., Faulkner, P., Manne, T., Hayes, E., Roberts, R. G. & Jacobs, Z. (2015), 'The archaeology, chronology and stratigraphy of Madjedbebe (Malakunanja ii): A site in northern Australia with early occupation', *Journal of human evolution* **83**, 46–64.

Collberg, C. & Proebsting, T. A. (2016), 'Repeatability in computer systems research', *Communications of the ACM* **59**(3), 62–69.

Conrad, C., Higham, C., Eda, M. & Marwick, B. (2016), 'Palaeoecology and forager subsistence strategies during the pleistocene holocene transition: A reinvestigation of the zooarchaeological assemblage from Spirit Cave, Mae Hong Son province, Thailand', *Asian Perspectives* **55**(1), 2–27.

Curriculum, R. S. (2016), 'The organization lesson for the reproducible science curriculum'.
URL: *https://github.com/Reproducible-Science-Curriculum/rr-organization1*

Donoho, D. L. (2010), 'An invitation to reproducible computational research', *Biostatistics* **11**(3), 385–388.
URL: *http://biostatistics.oxfordjournals.org/content/11/3/385.short*

Dorch, S. (2012), 'On the citation advantage of linking to data: Astrophysics'.
URL: *https://halshs.archives-ouvertes.fr/hprints-00714715/*

Duffy, M. A., James, T. Y. & Longworth, A. (2015), 'Ecology, virulence, and phylogeny of Blastulidium paedophthorum, a widespread brood parasite of Daphnia', *Applied and Environmental Microbiology* .
URL: *http://aem.asm.org/content/early/2015/06/02/AEM.01369-15.abstract*

Eglen, S. J. (2016), 'Bivariate spatial point patterns in the retina: a reproducible review', *Journal de la Société Française de Statistique* **157**(1), 33–48.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

FitzJohn, R. (2016), *remake: Make-like build management*. R package version 0.2.0.
URL: *https://github.com/richfitz/remake*

Gandrud, C. (2013), *Reproducible Research with R and RStudio*, CRC Press.

Gentleman, R. & Temple Lang, D. (2004), 'Statistical analyses and reproducible research', *Bioconductor Project Working Papers* p. 2.

Gentleman, R. & Temple Lang, D. (2012), 'Statistical analyses and reproducible research', *Journal of Computational and Graphical Statistics* .

Gentleman, R. et al. (2005), 'Reproducible research: A bioinformatics case study', *Statistical applications in genetics and molecular biology* **4**(1), 1034.

Gleditsch, N. P. & Strand, H. (2003), 'Posting your data: Will you be scooped or will you be famous?', *International Studies Perspectives* **4**(1), 72–107.
URL: *http://onlinelibrary.wiley.com/doi/10.1111/1528-3577.04105*

Goldstone, A. (2017), 'From reproducible to productive', *Journal of Cultural Analytics* .
URL: *http://culturalanalytics.org/2017/02/from-reproducible-to-productive/*

Goodman, S. N., Fanelli, D. & Ioannidis, J. P. A. (2016), 'What does research reproducibility mean?', *Science Translational Medicine* **8**(341), 341ps12–341ps12.
URL: *http://stm.sciencemag.org/content/8/341/341ps12*

Graham, S., Milligan, I. & Weingart, S. (2015), *Exploring Big Historical Data: The Historian's Macroscope*, Imperial College Press, Hackensack, NJ.

Henneken, E. A. & Accomazzi, A. (2011), 'Linking to data - effect on citation rates in astronomy', *CoRR* **abs/1111.3618**.
URL: *http://arxiv.org/abs/1111.3618*

Hollister, J. W., Milstead, W. B. & Kreakie, B. J. (2016), 'Modeling lake trophic state: a random forest approach', *Ecosphere* **7**(3), n/a–n/a. e01321.
URL: *http://dx.doi.org/10.1002/ecs2.1321*

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

Jones, Z. M. (2013), 'Git/GitHub, transparency, and legitimacy in quantitative research', *The Political Methodologist* **21**(1), 6–7.
URL: *http://zmjones.com/static/papers/git.pdf*

Kidwell, M. C., Lazarevi, L. B., Baranski, E., Hardwicke, T. E., Piechowski, S., Falkenberg, L.-S., Kennett, C., Slowik, A., Sonnleitner, C., Hess-Holden, C., Errington, T. M., Fiedler, S. & Nosek, B. A. (2016), 'Badges to acknowledge open practices: A simple, low-cost, effective method for increasing transparency', *PLoS Biol* **14**(5), 1–15.
URL: *http://dx.doi.org/10.1371/journal.pbio.1002456*

Klein, M., Van de Sompel, H., Sanderson, R., Shankar, H., Balakireva, L., Zhou, K. & Tobin, R. (2014), 'Scholarly context not found: One in five articles suffers from reference rot', *PLOS ONE* **9**(12), 1–39.

Knuth, D. E. (1992), 'Literate programming', *CSLI Lecture Notes, Stanford, CA: Center for the Study of Language and Information (CSLI), 1992* **1**.

Koenker, R. (1996), Reproducible econometric research, department of econometrics, university of illinois, urbana-champaign, Technical report, IL, Tech. Rep.
URL: *http://www.econ.uiuc.edu/ roger/research/repro/repro.html*

Leisch, F. (2002), Sweave: Dynamic generation of statistical reports using literate data analysis, *in* 'Compstat', Springer, pp. 575–580.

LeVeque, R. J., Mitchell, I. M. & Stodden, V. (2012), 'Reproducible research for scientific computing: Tools and strategies for changing the culture', *Computing in Science & Engineering* **14**(4), 13–17.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6171147*

Loeliger, J. & McCullough, M. (2012), *Version Control with Git: Powerful Tools and Techniques for Collaborative Software Development*, "O'Reilly Media, Inc.".

Markowetz, F. (2015), 'Five selfish reasons to work reproducibly', *Genome Biology* **16**.

Marwick, B. (2016), 'Computational reproducibility in archaeological research: Basic principles and a case study of their implementation', *Journal of Archaeological Method and*

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

Theory pp. 1–27.
URL: *http://dx.doi.org/10.1007/s10816-015-9272-9*

Marwick, B., Clarkson, C., O'Connor, S. & Collins, S. (2016), 'Early modern human lithic technology from jerimalai, east timor', *Journal of Human Evolution* **101**, 45–64.

Marwick, B., Hayes, E., Clarkson, C. & Fullagar, R. (2017), 'Movement of lithics by trampling: An experiment in the madjedbebe sediments, northern australia', *Journal of Archaeological Science* **79**, 73–85.
URL: *http://www.sciencedirect.com/science/article/pii/S0305440317300080*

Marwick, B., Van Vlack, H., Conrad, C., Shoocongdej, R., Thongcharoenchaikit, C. & Kwak, S. (2017), 'Adaptations to sea level change and transitions to agriculture at Khao Toh Chong rockshelter, peninsular Thailand', *Journal of Archaeological Science* **77**, 94–108.
URL: *http://www.sciencedirect.com/science/article/pii/S0305440316301637*

Marwick, B. et al. (2017), *rrtools: Tools for Writing Reproducible Research in R*. R package version 0.1.0.
URL: *https://github.com/benmarwick/rrtools*

McKiernan, E. C., Bourne, P. E., Brown, C. T., Buck, S., Kenall, A., Lin, J., McDougall, D., Nosek, B. A., Ram, K., Soderberg, C. K., Spies, J. R., Thaney, K., Updegrove, A., Woo, K. H. & Yarkoni, T. (2016), 'How open science helps researchers succeed', *eLife* **5**, e16800.
URL: *https://dx.doi.org/10.7554/eLife.16800*

Microsoft Corporation (2016), *checkpoint: Install Packages from Snapshots on the Checkpoint Server for Reproducibility*. R package version 0.3.16.
URL: *https://CRAN.R-project.org/package=checkpoint*

Munafo, M. R., Nosek, B. A., Bishop, D. V. M., Button, K. S., Chambers, C. D., Percie du Sert, N., Simonsohn, U., Wagenmakers, E.-J., Ware, J. J. & Ioannidis, J. P. A. (2017), 'A manifesto for reproducible science', *Nature Human Behaviour* **1**, 0021.
URL: *http://dx.doi.org/10.1038/s41562-016-0021*

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

Negre, J., Munoz, F. & Lancelotti, C. (2016), 'Geostatistical modelling of chemical residues on archaeological floors in the presence of barriers', *Journal of Archaeological Science* **70**, 91–101.
URL: *http://www.sciencedirect.com/science/article/pii/S0305440316300255*

Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., Buck, S., Chambers, C. D., Chin, G., Christensen, G., Contestabile, M., Dafoe, A., Eich, E., Freese, J., Glennerster, R., Goroff, D., Green, D. P., Hesse, B., Humphreys, M., Ishiyama, J., Karlan, D., Kraut, A., Lupia, A., Mabry, P., Madon, T., Malhotra, N., Mayo-Wilson, E., McNutt, M., Miguel, E., Levy Paluck, E., Simonsohn, U., Soderberg, C., Spellman, B. A., Turitto, J., VandenBos, G., Vazire, S., Wagenmakers, E. J., Wilson, R. & Yarkoni, T. (2015), 'Promoting an open research culture: Author guidelines for journals could help to promote transparency, openness, and reproducibility', *Science* **348**(6242), 1422–5.

Open Science Collaboration et al. (2015), 'Estimating the reproducibility of psychological science', *Science* **349**(6251), aac4716.

Peng, R. D. (2009), 'Reproducible research and biostatistics', *Biostatistics* **10**(3), 405–408.
URL: *http://biostatistics.oxfordjournals.org/content/10/3/405*

Peng, R. D., Dominici, F. & Zeger, S. L. (2006), 'Reproducible epidemiologic research', *American journal of epidemiology* **163**(9), 783–789.
URL: *http://aje.oxfordjournals.org/content/163/9/783.abstract*

Pienta, A. M., Alter, G. C. & Lyle, J. A. (2010), 'The enduring value of social science research: the use and reuse of primary research data'.
URL: *https://deepblue.lib.umich.edu/handle/2027.42/78307*

Piper, A. et al. (2017), 'Data sharing policy', *Journal of Cultural Analytics* .
URL: *http://culturalanalytics.org/about/about-ca/*

Piwowar, H. A., Day, R. S. & Fridsma, D. B. (2007), 'Sharing detailed research data is associated with increased citation rate', *PLoS ONE* **2**(3), e308.
URL: *http://dx.plos.org/10.1371/journal.pone.0000308*

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

Piwowar, H. A. & Vision, T. J. (2013), 'Data reuse and the open data citation advantage', *PeerJ* **1**, e175.

Ram, K. (2013), 'Git can facilitate greater reproducibility and increased transparency in science.', *Source code for biology and medicine* **8**(1), 7.

Rokem, A., Marwick, B. & Staneva, V. (2017), Assessing reproducibility, *in* J. Kitzes, D. Turek & F. Deniz, eds, 'The Practice of Reproducible Research: Case Studies and Lessons from the Data-Intensive Sciences.', University of California Press, CA.

rOpenSci et al. (2017), 'Reproducibility guide'.
URL: *http://ropensci.github.io/reproducibility-guide/*

Sandve, G. K., Nekrutenko, A., Taylor, J. & Hovig, E. (2013), 'Ten simple rules for reproducible computational research', *PLoS Comput Biol* **9**(10), e1003285.
URL: *http://dx.doi.org/10.1371/journal.pcbi.1003285*

Sears, J. (2011), Data sharing effect on article citation rate in paleoceanography, *in* 'AGU Fall Meeting Abstracts', Vol. 1, p. 1628.
URL: *http://adsabs.harvard.edu/abs/2011AGUFMIN53B1628S*

Stanisic, L. & Legrand, A. (2014), Effective reproducible research with Org-Mode and Git, *in* 'Euro-Par 2014: Parallel Processing Workshops', Springer, pp. 475–486.
URL: *http://link.springer.com/chapter/10.1007/978-3-319-14325-5_41*

Stodden, V. (2009), 'The legal framework for reproducible scientific research: Licensing and copyright', *Computing in Science & Engineering* **11**(1), 35–40.

Stodden, V. (2014), 'What scientific idea is ready for retirement? reproducibility.', *Edge* .
URL: *https://www.edge.org/response-detail/25340*

Stodden, V., McNutt, M., Bailey, D. H., Deelman, E., Gil, Y., Hanson, B., Heroux, M., Ioannidis, J. P. & Taufer, M. (2016), 'Enhancing reproducibility for computational methods', *Science* **354**(6317), 1240–1241.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

Stodden, V. & Miguez, S. (2014), 'Best practices for computational science: Software infrastructure and environments for reproducible and extensible research', *Journal of Open Research Software* **2**(1).
URL: *http://openresearchsoftware.metajnl.com/articles/10.5334/jors.ay/*

Teal, T. K., Cranston, K. A., Lapp, H., White, E., Wilson, G., Ram, K. & Pawlik, A. (2015), 'Data carpentry: workshops to increase data literacy for researchers', *International Journal of Digital Curation* **10**(1), 135–143.

Tippmann, S. (2014), 'Programming tools: Adventures with R', *Nature* **517**(7532), 109–110.

Ushey, K., McPherson, J., Cheng, J., Atkins, A. & Allaire, J. (n.d.), *packrat: A Dependency Management System for Projects and their R Package Dependencies*. R package version 0.4.7-12.
URL: *https://github.com/rstudio/packrat/*

Vandewalle, P., Kovacevic, J. & Vetterli, M. (2009), 'Reproducible research in signal processing', *IEEE Signal Processing Magazine* **26**(3), 37–47.

Vinod, H. (2001), 'Care and feeding of reproducible econometrics', *Journal of Econometrics* **100**(1), 87–88.

White, J. M. (2014), *ProjectTemplate: Automates the creation of new statistical analysis projects*. R package version 0.6.
URL: *https://CRAN.R-project.org/package=ProjectTemplate*

Wickham, H. (2015), *R packages*, 1 edition edn, " O'Reilly Media, Inc.".

Wickham, H. & Chang, W. (2016), *devtools: Tools to Make Developing R Packages Easier*. R package version 1.12.0.
URL: *https://CRAN.R-project.org/package=devtools*

Wilson, G. (2013), 'Software carpentry: Lessons learned', *CoRR* **abs/1307.5448**.
URL: *http://arxiv.org/abs/1307.5448*

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3192v2 | CC BY 4.0 Open Access | rec: 20 Mar 2018, publ: 20 Mar 2018

Wilson, G., Aruliah, D. A., Brown, C. T., Chue Hong, N. P., Davis, M., Guy, R. T., Haddock, S. H. D., Huff, K. D., Mitchell, I. M., Plumbley, M. D., Waugh, B., White, E. P. & Wilson, P. (2014), 'Best practices for scientific computing', *PLoS Biology* **12**(1), e1001745.
URL: *http://dx.plos.org/10.1371/journal.pbio.1001745*

Xie, Y. (2015), *Dynamic Documents with R and knitr*, 2nd edn, Chapman and Hall/CRC, Boca Raton, Florida. ISBN 978-1498716963.
URL: *http://yihui.name/knitr/*