Sharing and organizing research products as R packages

Matti Vuorre[1] & Matthew J. C. Crump[2]

[1] Department of Psychology, Columbia University, New York, USA
[2] Department of Psychology, Brooklyn College of CUNY, New York USA

A consensus on the importance of open data and reproducible code is emerging. How should data and code be shared to maximize the key desiderata of reproducibility, permanence, and accessibility? Research assets should be stored persistently in formats that are not software restrictive, and documented so that others can reproduce and extend the required computations. The sharing method should be easy to adopt by already busy researchers. We suggest the R package standard as a solution for creating, curating, and communicating research assets. The R package standard, with extensions discussed herein, provides a format for assets and metadata that satisfies the above desiderata, facilitates reproducibility, open access, and sharing of materials through online platforms like GitHub and Open Science Framework. We discuss a stack of R resources that help users create reproducible collections of research assets, from experiments to manuscripts, in the RStudio interface. We created an R package, *vertical*, to help researchers incorporate these tools into their workflows, and discuss its functionality at length in an online supplement. Together, these tools may increase the reproducibility and openness of psychological science.

Keywords: reproducibility; research methods; R; open data; open science
Word count: 4787

Introduction

Research projects produce many assets, such as experiments, data, analyses, manuscripts, posters, slides, stimulus sets and materials, computational models, and more. Although creating and communicating research assets are mainstay activities in science, there is a trend (Vanpaemel, Vermorgen, Deriemaecker, & Storms, 2015; Wicherts, Borsboom, Kats, & Molenaar, 2006) encouraging more open and transparent sharing practices (Houtkoop et al., 2018; Lindsay, 2017). However, the value of shared assets is often limited by poor documentation, incompatible file formats, or lack of organization, resulting in low rates of reproducibility (Hardwicke et al., 2018). Some data sharing guidelines and good practices have been suggested (e.g., Rouder, Haaf, & Snyder, 2019; Rouder, 2016); along with related issues of ethics and privacy, ownership of data, attitudes about sharing, and where to share data (Klein et al., 2018; Martone, Garcia-Castro, & VandenBos, 2018). Nevertheless, curating standards for research assets have not emerged, possibly due to the variance of research in psychology. Instead of developing another standard, we suggest borrowing existing standards and practices from software engineering. Specifically, the R package standard, with additional R authoring tools, provides a robust framework for organizing and sharing reproducible research products.

Some advances in data-sharing standards have emerged: In much of psychological science it is now customary to share data on Open Science Framework (OSF). However, those materials often contain idiosyncratic file organization and minimal or missing documentation for raw data. In specific areas, organization and documentation standards have emerged, (e.g., the BIDS framework in neuroscience, Gorgolewski et al., 2016), but they usually only consider data and code instead of the project as a whole. More comprehensive proposals are described in the Transparency and Openness Promotion (Nosek et al., 2015), and Peer Reviewers' Openness initiative guidelines (Morey et al., 2016), but these fall short of describing detailed standards for organization and metadata. Additionally, these standards (if they exist) may not be widely known, recognized, agreed upon, and/or adopted at large.

We sought for a standard for organization and sharing that would adhere to the FAIR (Findable, Accessible, Interoperable, Reusable) guidelines to maximize the reuse potential of data and support "discovery through good data management" (Wilkinson et al., 2016, see also general discussion). Additionally, we recognized the added value of including other research outputs ("products"; e.g., manuscripts) in a reproducible collection of materials that could be shared with the data. We identified the R package standard as a solution that doesn't present overwhelming overhead for already busy researchers. Here, we discuss the R package standard for creating reproducible research material containers for data, analysis code, and metadata. Then, we introduce a collection

The authors declare equal authorship.

Correspondence concerning this article should be addressed to Matti Vuorre, 406 Schermerhorn Hall, 1190 Amsterdam Avenue MC 5501, New York, NY 10027. E-mail: mv2521@columbia.edu

of additional R packages for incorporating other research products into the project. These additional tools are easily accessible through our *vertical* package, thus named because an entire research project can be completed from top to bottom as a stack of R processes from within the RStudio software platform.

R Packages

R is a programming language commonly used for statistical analyses (R Core Team, 2019), and is available at https://www.r-project.org. RStudio is an integrated development environment (IDE; RStudio Team, 2016), and is available at https://www.rstudio.com/. Both are free, open source and work on Windows, Mac, and Linux operating systems. We chose these tools because they are already widely used among psychology researchers: In one crowd-sourced analysis, 16 of 29 analysis teams used R (Silberzahn et al., 2018; see also Yee & Debbie, 2017, and http://r4stats.com/articles/popularity/). Further, many psychologists have already developed R packages that share analysis tools and data.

A cornerstone of R are user-created packages, which contain R functions, data, metadata, and documentation written in a standardized format that allows seamless sharing across users and operating systems. When data and functions are wrapped in R packages, they are immediately available to other R users through the R console, and their documentation can be easily viewed in R or online. Next, we outline how data and functions are included in R packages, and then turn to including other assets, such as manuscripts and posters. We present a brief outline, a complete guide to creating R packages is outside the scope of this manuscript, and we refer readers to Wickham (2015), which remains an authoritative and in-depth guide on creating R packages.

Creating a package through RStudio is straightforward (see Figure 1). When a new package is created with the method shown in Figure 1, the required files and directories are automatically created. First, the "DESCRIPTION" file includes basic information about the package (e.g., authors, description) in a machine readable format. To edit it, users can click on the file in the RStudio File browser. The "blank.Rproj", "man" and "NAMESPACE" files/directories shown in Figure 1 should not be edited by users, and we thus turn to R functions.

Functions

Because R packages are primarily intended for R functions, an "R" directory is automatically created to store source code for defining functions. Researchers often develop and use custom functions in their analysis. Although it is possible to declare them directly in the analysis scripts, we suggest declaring functions using the R package standard, by placing the function's source code into a file in the "R" directory. Functions in R packages are portable, such that others can install the package from their R console, load it, and start using the functions immediately. Packages can also depend on other packages (and be depended on), such that R will automatically include any other packages required for your functions to run appropriately. Functions within R packages are documented in a standardized manner, and the documentation for a function can be viewed within R or online.

Additionally, learning and following R conventions for declaring functions has a pedagogical benefit to the researcher and may improve their practices. There is also a reuse benefit to the researcher: Functions can be difficult to find in old scripts, but easy to find and load if they are already packaged and called from an existing package. Thus, formally including one's functions in R packages facilitates reproducibility and sharing.

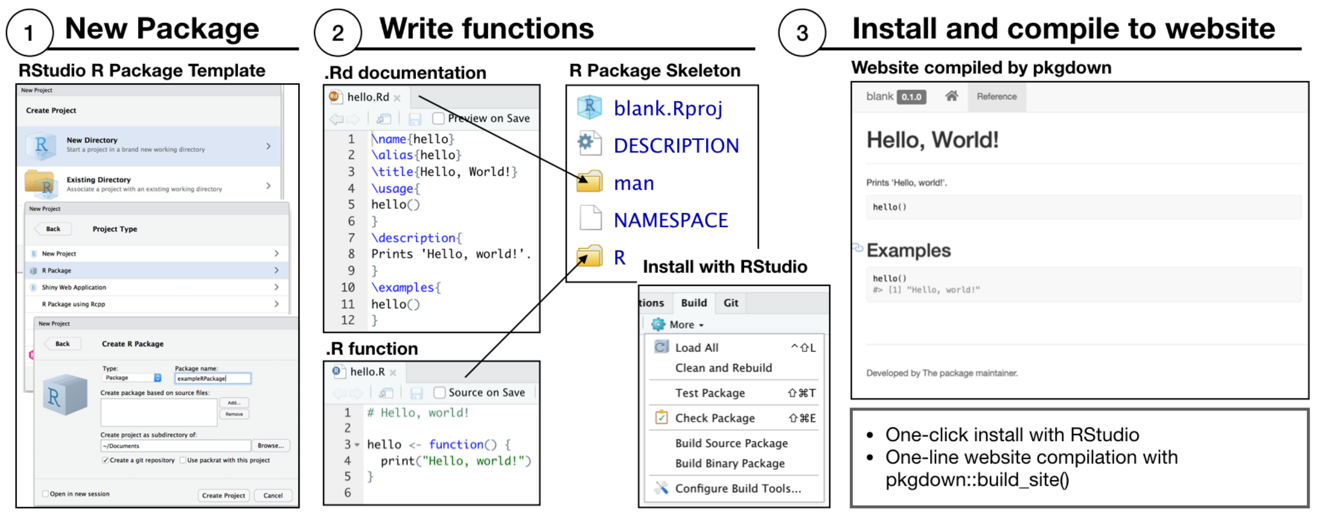
Data

Broadly, there are 3 steps to including data in an R package: 1. Placing raw data in the "data-raw" directory, 2. creating an R script that processes the raw data and creates an R data object into the "data" directory, and 3. documenting the final data object.

First, raw data in any format is stored in the "data-raw" directory to ensure that the method of sharing remains software-agnostic. Importantly, the raw data files should never be modified themselves. Second, instructions for converting and cleaning the raw data into analyzable format should also be included in this directory. These pre-processing steps should be declared so that the entire analysis is transparent and reproducible, preferably in an R script in the "data-raw" directory. That script should end with creating an R data object in the project's "data" directory. Placing the R data object there ensures that it is included in the resulting R package, which makes using the data effortless for any R user, and eschews the need to download additional files. Loading the R package makes the included data immediately available in the R console.

Finally, the resulting data object in "data" should be documented in a standardized format by placing a `data.R` file in the "R" directory, a process discussed in more detail in our complete online supplementary tutorial, and in Wickham (2015). That standard format of documentation is especially useful because it allows datasets' documentation to be viewed in R (e.g., type `?attitude` in the R console) and online (see below).

To recap, R package standards offer a systematic method for creating and curating reproducible research assets. Researchers could use the R package infrastructure to store and document their data and functions, and, as we will see, analyses, manuscripts, posters, and presentations as well. The entire research project is then easily shared online, and show-



```
\name{hello}
\alias{hello}
\title{Hello, World!}
\usage{
hello()
}
\description{
Prints 'Hello, world!'.
}
\examples{
hello()
}
```

```
# Hello, world!

hello <- function() {
  print("Hello, world!")
}
```

Hello, World!

Prints 'Hello, world!'.

```
hello()
```

Examples

```
hello()
#> [1] "Hello, world!"
```

Developed by The package maintainer.

- One-click install with RStudio
- One-line website compilation with pkgdown::build_site()

Figure 1. Creating an R package using the RStudio graphical user interface.

cased as a website, as we will show below. To enable these additional features, many R packages have been developed (e.g., writing manuscripts within R). To facilitate their use, we have created an R package, *vertical*, that, when installed, also installs these other packages, and makes it easier to create projects that use them.

Reproducible research projects with *vertical*

vertical extends the R package template with a set of additional files and directories for products such as manuscripts and posters. These assets are organized such that they can then be easily shared online on GitHub or OSF, installed to R as an R package, and showcased online as a website. Figure 2 illustrates the workflow of creating a *vertical* research project.

Additional benefits of packaging everything together in this manner are that every computation supporting manuscripts, supplementary analyses, data preprocessing, etc. is reproducible and uses the same data in the same computational environment. Because sharing of materials is built-in, no additional work is needed after-the-fact in organizing the materials for sharing them. We begin with an overview of this process, but a more exhaustive step-by-step tutorial is provided online (Crump & Vuorre, n.d.)[1]. Before using *vertical*, or any packages contained therein, users must first install it in R (the *devtools* (Wickham et al., 2019) package is necessary for installing R packages from GitHub):

```r
install.packages("devtools")
devtools::install_github("CrumpLab/vertical")
```

After installing the package, users should restart RStudio to make all its features available. First, similar to creating an R package as shown above, a new *vertical* research project is created through the RStudio GUI (Figure 3.1). Figure 3.2 shows the resulting file and directory structure, which is an extension of the R package structure shown in Figure 1. Users may enable or disable components (as indicated by directories like “experiments”) as needed. Next, we describe the components, and then how to share the resulting research projects and showcase them as websites.

In addition to data and functions, almost any kind of content can be included within the project directory structure. Most importantly, to create reproducible assets, such as data analyses or manuscripts, these materials should be created with the R Markdown language. R Markdown is a simple plain text format that allows mixing prose with code (R, Python, etc.), and thus enables including one’s text and analyses in a single document. Because of its importance, we describe R Markdown first.

Reproducible documents with R Markdown

R Markdown is a plain text file format and an R package that allows executable code snippets to be embedded alongside regular text (Xie, Allaire, & Grolemund, 2018). For example, an R Markdown file can include the typical prose found in manuscripts, the analysis code (R, Python, etc.), and code snippets that display numerical results and figures within the text. R Markdown documents are transparent and reproducible. In principle, recipients can see the analysis scripts, and reproduce them by compiling the document on their own machine. R Markdown documents separate content and style, and can be compiled to multiple output formats, such as PDF, HTML, and Microsoft Word. Importantly, R Markdown is basically plain text, and thus effortless to learn. Nevertheless, a tutorial on R Markdown is beyond our scope (see Xie et al.,

[1]https://crumplab.github.io/vertical

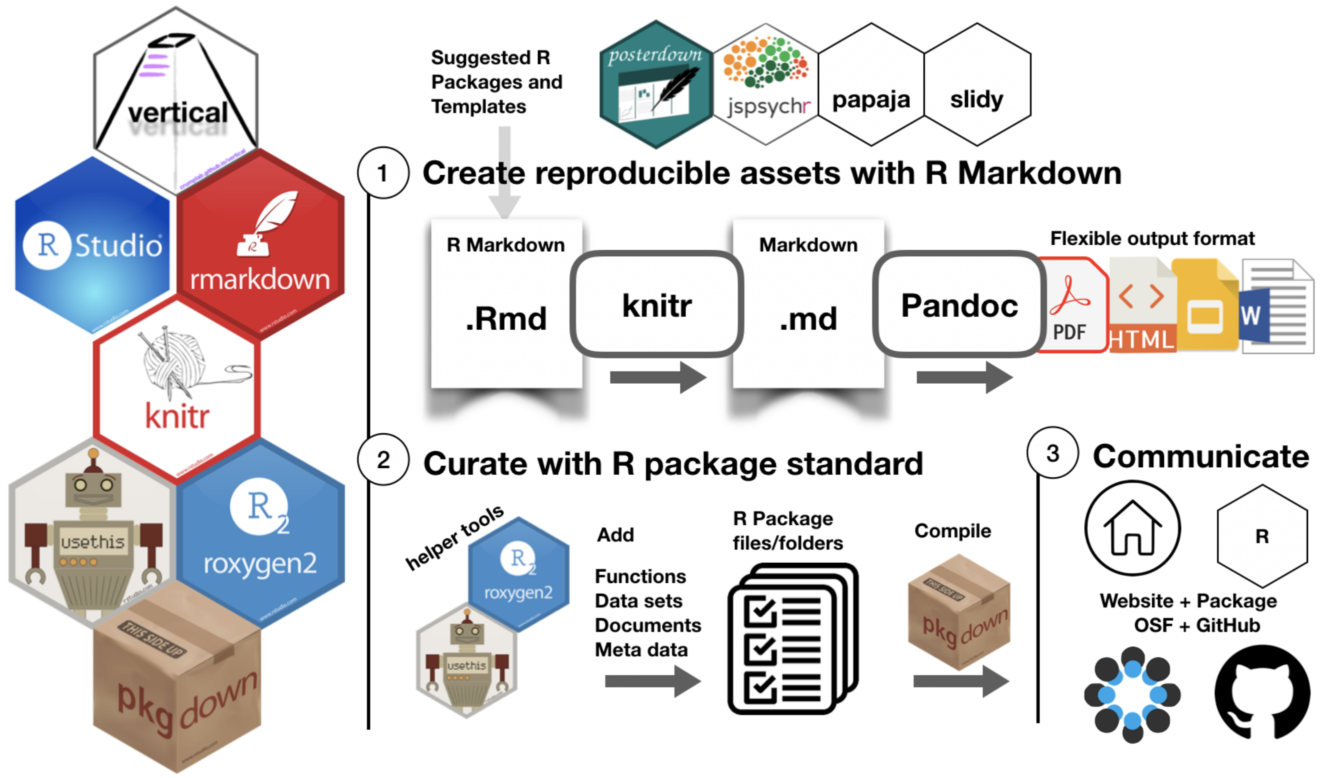


Figure 2. An overview of the vertical workflow. The project template suggests R Markdown modules for asset creation (manuscripts, slides, posters), that are compiled to multiple formats. Content is curated follow R package standards, then compiled and communicated in the form of an online repository (GitHub/OSF), project website, and R package.

2018)[2].

Data analysis

Data analysis can be made fully transparent and reproducible with R Markdown. Above, we suggested placing any preprocessing steps in a separate file in the "data-raw" directory, and creating a documented R data object in "data". If the data object is called `mydata`, once the package is installed and loaded, simply calling `mydata` in an R script will call the data object, ready to be used in analyses. Our favored approach is to write any analyses relevant to a product (e.g., manuscript or a poster) directly into the product's R Markdown source file. Typically, we would include data analysis code in the R Markdown source of the APA style manuscript (see below).

Nevertheless, sometimes supplementary analyses are conducted that are not included in any other document. These are most conveniently placed in R Markdown files into the "vignettes" directory. Then, when the project is compiled into a website (see below), those analyses are viewable as part of the resulting website. When a *vertical* project is created, the "vignettes" directory is automatically created with an example data analysis R Markdown script.

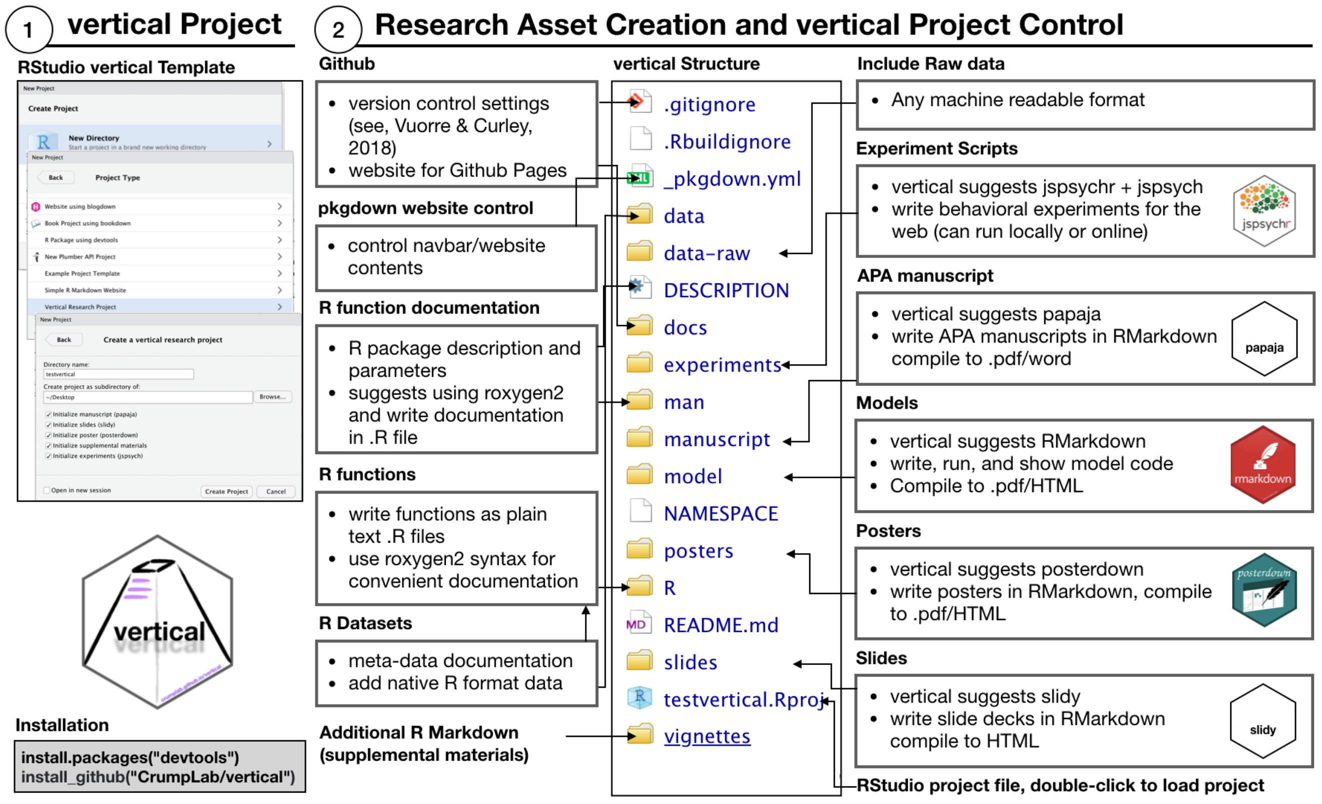
APA manuscripts

Reproducible APA formatted manuscripts can easily be created with the *papaja* R package, that provides an R Markdown template file, and additional helper functions for creating tables and figures (Aust & Barth, 2018). For example, the R Markdown file can contain the text and code to generate the manuscript and all results. *papaja* provides several functions for reporting results, especially from common statistical analyses such as ANOVAs, t-tests, and regressions. For example, *papaja*'s `apa_print()` function embeds statistical results directly into the manuscript, obviating the need to copy them by hand and removing human error in reporting. References can be automatically populated from Zotero using the `citr` plugin (Aust, 2019). We refer readers to the *papaja* documentation website for more information.[3]

When a new *vertical* project is created, a *papaja* manuscript template is automatically included in the "manuscript" directory. Then, once the project is compiled, the manuscript PDF is copied to the "docs" folder, from where it can be viewed as

[2]See also https://rmarkdown.rstudio.com/authoring_quick_tour.html for a quick overview of R Markdown.

[3]https://crsh.github.io/papaja_man/introduction.html


1 vertical Project
RStudio vertical Template
New Project
Create Project
New Directory
Start a project in a brand new working directory
Back
Project Type
Website using blogdown
Book Project using bookdown
R Package using devtools
New Plumber API Project
Example Project Template
Simple R Markdown Website
Vertical Research Project
Create a vertical research project
Directory name:
testvertical
Create project as subdirectory of:
~/Desktop
Browse...
Initialize manuscript (papaja)
Initialize slides (slidy)
Initialize poster (posterdown)
Initialize supplemental materials
Initialize experiments (jspsych)
Open in new session
Create Project
Cancel
vertical
Installation
2 Research Asset Creation and vertical Project Control
Github
version control settings (see, Vuorre & Curley, 2018)
website for Github Pages
pkgdown website control
control navbar/website contents
R function documentation
R package description and parameters
suggests using roxygen2 and write documentation in .R file
R functions
write functions as plain text .R files
use roxygen2 syntax for convenient documentation
R Datasets
meta-data documentation
add native R format data
Additional R Markdown (supplemental materials)
vertical Structure
.gitignore
.Rbuildignore
_pkgdown.yml
data
data-raw
DESCRIPTION
docs
experiments
man
manuscript
model
NAMESPACE
posters
R
README.md
slides
testvertical.Rproj
vignettes
Include Raw data
Any machine readable format
Experiment Scripts
vertical suggests jspsychr + jspsych
write behavioral experiments for the web (can run locally or online)
jspsychr
APA manuscript
vertical suggests papaja
write APA manuscripts in RMarkdown compile to .pdf/word
papaja
Models
vertical suggests RMarkdown
write, run, and show model code
Compile to .pdf/HTML
rmarkdown
Posters
vertical suggests posterdown
write posters in RMarkdown, compile to .pdf/HTML
posterdown
Slides
vertical suggests slidy
write slide decks in RMarkdown compile to HTML
slidy
RStudio project file, double-click to load project

```
install.packages("devtools")
install_github("CrumpLab/vertical")
```

Figure 3. Creating a *vertical* project with the RStudio *vertical* project template, and description of the file-organization and structure of a *vertical* project.

part of the resulting website (see below).

Posters

vertical includes a posters folder where poster documents can be deposited and shared on the website. Additionally, *vertical* suggests using the *posterdown* package (Thorne, 2019) for poster creation through an R Markdown document. When the provided template is used, the resulting reproducible document is viewable as part of the website.

Slides

vertical includes a slides directory for slide decks in any format, which can then be included for downloading or viewing on the resulting website. There are several R Markdown options for creating reproducible slide decks in various formats. When a new *vertical* project is created, a *slidy* template is included to create web-ready presentations.[4] Slidy presentations are viewable in a browser like typical PowerPoint or keynote presentations, but have some additional features like long scroll-able slides, interactive figures and animations, and rendering full websites within individual slides.

Other materials

Other research assets may include supplementary materials, stimulus sets, code for a computational model, additional analyses, or other documents, like a public-facing blog. *vertical* is flexible with respect to the inclusion of additional assets not specified by the template. For example, arbitrary R Markdown documents can be included in the vignettes directory. When the project is compiled (see below), any R Markdown files in that directory are compiled and served on the website.

Experiments

Finally, *vertical* projects can include experiments in the experiments directory. These materials could be packaged as a compressed .zip file, which would then be accessible from an online repository. *vertical* suggests using *jsPsych* (De Leeuw, 2015), a JavaScript library for building browser-based experiments that can be served to the web or run locally. Selecting the *jsPsych* module during project initialization downloads the most recent version of *jsPsych* to the experiments folder. A basic html template is also provided, which loads the jsPsych library and presents some basic word stimuli. To further edit

[4]https://bookdown.org/yihui/rmarkdown/slidy-presentation.html

the experiment, users can simply edit the html file as they would when creating any other jsPsych experiment. When the *vertical* project is compiled, this experiment is immediately available from the website navigation bar.

For users who wish to work entirely within R Markdown, we suggest *jspsychr* (Crump, 2019),[5] an R package that helps creating jsPsych experiments with R Markdown in RStudio. This option allows using RStudio's ability to pass objects between R and JavaScript when constructing experiment materials. The template creates a sample *jsPsych* experiment produced by *jspsychr*, which contains the R Markdown script to generate the experiment file, the html file to run the experiment, and a folder to save the data and document the experiment.

Browser-friendly experiments have several benefits over local-only experiments. First, scripts are open-source and easily reusable (no installation is required to run an experiment). Second, scripts can be shared in a self-contained manner such that the experiment is reproducible and the methods are transparent and verifiable. Third, because the experiment is an html file it can be shared and run as part of the project website, facilitating understanding of the experimental procedures. Sharing experiment source code with a demonstration is not yet common, but could benefit the review process and readers' understanding of the procedures by making methods more transparent.

Sharing and communicating

Above, we discussed using R Markdown, R package standards, and other R resources for creating research products. Then next step is sharing and communicating these products. When the products are created as suggested above, *vertical* facilitates sharing the entire project as an R package for easy access to the data and functions, as a version controlled repository for access to the complete source code for reproducing every component of the project, and as a website, where the components are easily viewed by others.

Sharing the source code with version control

An organizing principle of our approach is that all project materials are stored in a single directory on the user's local filesystem. When version control, such as Git, is enabled in this directory, the entire project can be seamlessly shared with others through online platforms such as GitHub and GitLab. Version control systems provide filesystems with additional functioning for keeping track of files' versions, improved options for collaborating, and seamless integration with online collaboration platforms (GitHub and GitLab); however, the details are outside the scope of this manuscript and we refer readers to Vuorre & Curley (2018). By default, *vertical* projects enable Git for new projects. Further, projects developed with Git (on a sharing platform such as GitHub or GitLab) can be seamlessly integrated with the Open Science Framework for long-term curation and, for example, to enable digital object identifiers (DOI).

Sharing the project as a website

An additional benefit of the R package organizing principles, and using the *vertical* helper tools, is that the entire project can be showcased as a website (see Figure 4. The website is compiled by *pkgdown* (Wickham & Hesselberth, 2019) in one command (`vertical::build_vertical()`), or by clicking Addins -> Build Vertical in RStudio, and expertise with web standards is not required. For instructions on customizing the resulting website, see the *pkgdown* instructions (https://usethis.r-lib.org/).

Building a *vertical* project as shown above creates the website in the docs folder, from where it can be viewed. During this process, all the R Markdown files within the project are evaluated while the documents are compiled, thus ensuring that all analyses, and thus products, are up to date. Importantly, the website can be easily made available on the internet. To do so, users should integrate their project with a GitHub repository (Vuorre & Curley, 2018), and then enable GitHub pages under the repository's settings. An example of the resulting website can be viewed at https://mvuorre.github.io/rparp/.

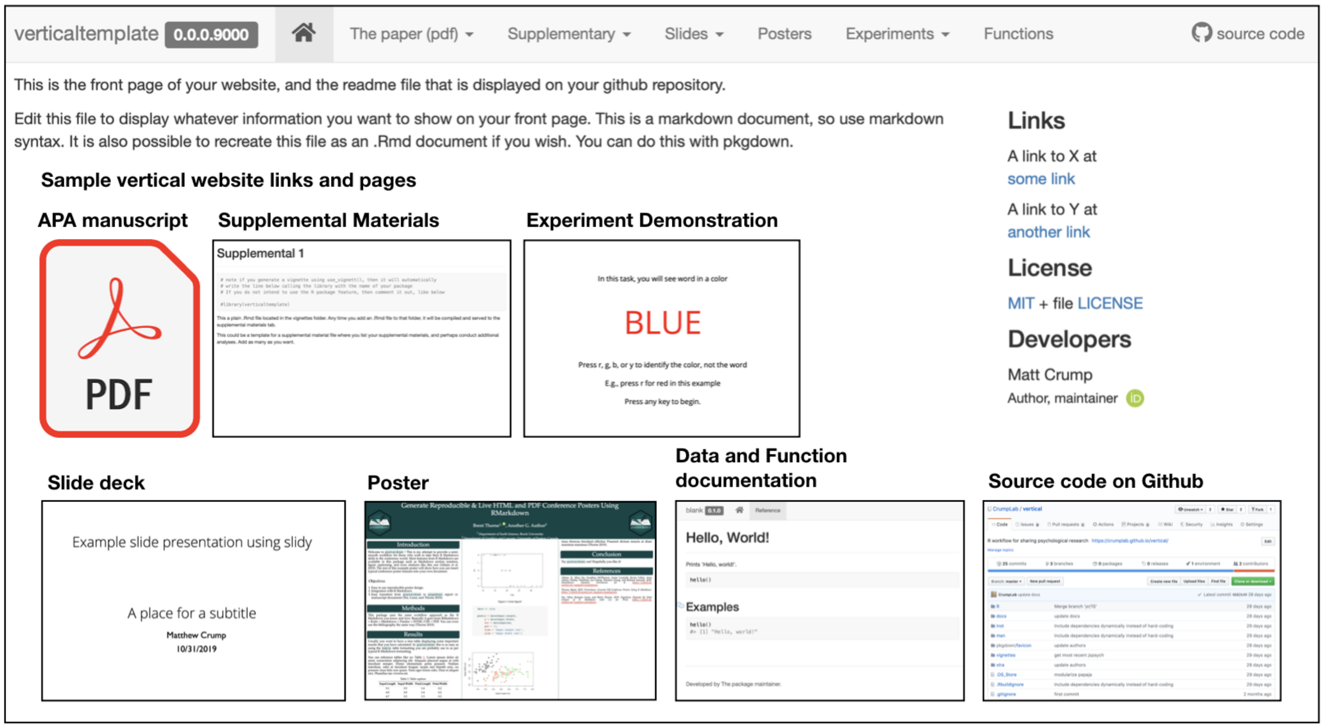
Sharing as an R Package

If the project contains R functions, data, and documentation following the R package standard (as discussed above), then others can install the package on their computers for fast access to the data and functions from within R. Again, integrating the project with GitHub is beneficial, because then others can install the package from GitHub using the *devtools* package in R (`devtools::install_github("account/repository")`). Sharing as an R package is very convenient for other R users, as any packaged data and functions are immediately available for use, natively in R, after package installation. We expand on the extended benefits of using R packages for transparency and reproducibility in the general discussion.

General Discussion

We presented *vertical*, a stack of R tools for a single-platform solution for creating, curating, and communicating psychological research projects in a reproducible manner. It might seem that the benefits of this approach apply only to R users. We believe this is not the case; the workflow creates a single package that includes all the research products related to a product, which can then be viewed in a public (or private, if users so choose) website and source code repository. Of course, to get the most out of the proposal, some R use is required (from both the creator and subsequent users.) However, any

[5]https://crumplab.github.io/jspsychr/



verticaltemplate 0.0.0.9000 | The paper (pdf) | Supplementary | Slides | Posters | Experiments | Functions | source code

This is the front page of your website, and the readme file that is displayed on your github repository.

Edit this file to display whatever information you want to show on your front page. This is a markdown document, so use markdown syntax. It is also possible to recreate this file as an .Rmd document if you wish. You can do this with pkgdown.

Sample vertical website links and pages

APA manuscript

Supplemental Materials

Supplemental 1

```
# note if you generate a vignette using use_vignett(), then it will automatically
# write the line below calling the library with the name of your package
# If you do not intend to use the R package feature, then comment it out, like below

#library(verticaltemplate)
```

This a plain .Rmd file located in the vignettes folder. Any time you add an .Rmd file to that folder, it will be compiled and served to the supplemental materials tab.

This could be a template for a supplemental material file where you list your supplemental materials, and perhaps conduct additional analyses. Add as many as you want.

Experiment Demonstration

In this task, you will see word in a color

BLUE

Press r, g, b, or y to identify the color, not the word

E.g., press r for red in this example

Press any key to begin.

Links

A link to X at some link

A link to Y at another link

License

MIT + file LICENSE

Developers

Matt Crump

Author, maintainer

Slide deck

Example slide presentation using slidy

A place for a subtitle

Matthew Crump

10/31/2019

Poster

Generate Reproducible & Live HTML and PDF Conference Posters Using RMarkdown

Data and Function documentation

blank 0.1.0 | Reference

Hello, World!

Prints 'Hello, world!'.

```
hello()
```

Examples

```
hello()
#> [1] "Hello, world!"
```

Developed by The package maintainer

Source code on Github

Figure 4. Research assets from a *vertical* project displayed on a website. The navigation tab bar on the top links to the content pages depicted in the body of the webpage.

standardized computational procedure assumes some computational environment, our choice of R is natural due to its already widespread adoption in the psychological researcher community. Furthermore, by requiring raw data inclusion, our proposal does not strictly require using R. And, the software we have discussed, RStudio, works well with other languages, including JavaScript, Python, and C++.

Validating research assets

Our suggested approach has wider implications for making research practices in psychology more transparent and reproducible. Consider the implied truth claims associated with any peer-reviewed manuscript. The act of publishing can be seen as a validity assertion: that the truth claims made by the authors are verifiably true, or more weakly that the research assets are verifiable. Unlike a set of conclusions from premises that may be tested for logical validity, conclusions and claims from a research project are often not as easily scrutinized and tested. Reviewers extend trust that methods and results were as written when assets can not be verified directly. Sharing research assets in the manner discussed here would greatly facilitate scrutinizing and testing all components of the research project, and thus potentially increase the reliability of the resulting scientific products.

Furthermore, organizing research assets in the manner discussed here allow its components to be formally tested, a process referred to as unit testing in software engineering. It refers to the practice of verifying that functions perform as intended in a variety of cases. We make a loose analogy between building a *vertical* project and unit testing. Building is a call to compile individual components of the project. If a user adheres to the *vertical* workflow, then R must be able to perform all of the scripted operations. Successfully building a *vertical* project is loosely analogous to performing unit tests that confirm and validate the process of asset creation. So, compiling a vertical project allows a researcher to unit test their own research project and verify its assets. By sharing a *vertical* project, other researchers can perform the same verification. We note that *vertical* does not conduct proper unit tests on functions included in an R package. However, because these projects are R packages, proper unit tests can be easily included (see Wickham, 2011).

Collaborating with vertical

Software developers routinely use version control platforms, such as GitHub and GitLab to coordinate software development across a team of collaborators (see, Vuorre & Curley, 2018). We recognize that scientific research is *not* software development. However, some components of the research workflow readily conform to the software development model—e.g., developing experimental programs, data analysis—and as such stand to benefit from borrowing best practices and tools from that domain.

Research projects can be hosted on GitHub throughout the lifespan of the project to make use of tools for collaboration. With this in mind, *vertical* strongly suggests using Git (along with its online platforms). For example, we used GitHub as a collaboration tool to create *vertical* and to write this manuscript (using *papaja*). Among other features, Git provides functionality similar to track changes in Microsoft Word that allow contributions to be reviewed, accepted, and modified. It makes this process robust even when many collaborators are involved. We also made extensive use of the GitHub issues tab, which is normally used to report bugs or feature requests to a software developer. Many users may find this feature very useful. In our case, we opened project related issues for discussion and then closed the issues once they were addressed. A side-effect is that the GitHub issues tab, and the Git history, may double as a project development narrative, or a lab journaling system preserving issues and solutions for posterity.

Teaching R with vertical

vertical may be most beneficial to users who are already well-versed in R; and, although the website tutorial is intended for R beginners, it is not a tutorial on the R language. Nevertheless, *vertical* could be used for pedagogical purposes, like a museum guide offering an orientation to collections in a large museum. For example, an R programming class for psychologists (or statistics/methods course using R) could be structured around the components of a vertical project, such as how to program functions, import and analyse data, write manuscripts, posters, slide decks, and websites in R. Moreover, *vertical* provides a convenient and well-organized workflow for students to save their work as they learn each component.

Extending vertical

vertical is a modular open-source tool that can be extended by anyone. We added content modules that we think are common in many psychology research projects, and can incorporate new modules as they are identified and suggested by a user base. Users can fork the *vertical* source code on GitHub, make suggested changes, and submit pull requests for review and inclusion.

FAIR

To maximize the reuse potential of open data and support "discovery through good data management", Wilkinson et al. (2016) proposed the FAIR (Findable, Accessible, Interoperable, Reusable) guiding principles. Below, we describe how a *vertical* workflow targets FAIR principles and facilitates accessibility and data reanalyses (FAIR principles paraphrased from, Wilkinson et al., 2016; and Martone et al., 2018)

Findable data are discoverable through persistent identifiers (e.g., permanent URLs / DOIs):

- Projects are accessible through the internet at OSF or GitHub
- DOIs for a *vertical* project can be minted through OSF or Zenodo

Accessible data is available to approved researchers (e.g., anyone or lab members only) through a standardized communications protocol (e.g., Internet):

- Projects on GitHub and OSF can be made accessible to a specific set of users, or completely private, if desired
- Most data sharing guides emphasize manually uploading heterogeneous data files to a repository (e.g., OSF). R packages allow that, but also facilitate programmatic access to the data, and therefore are more accessible than methods relying on manual transfer of files

Interoperable data is described (via metadata and file organization) in a common, openly available, non-proprietary language:

- R packages describe a strict organization of source files, resulting in a software product that is directly usable in a programming environment
- R packages describe a common language for documenting the data, its use, and dependencies
- R package (meta)data is human- and machine-readable and in a non-proprietary format
- Data wrapped in an R package is accessible in other programming environments due to standardized formatting

Reusable data are richly described and licensed to facilitate reuse:

- R packages are described in a standard format
- R packages facilitate use of appropriate licenses

Conclusion

Researchers may improve the reproducibility of their work by borrowing standards from software development, such as the R package standard, discussed here for sharing research products. By organizing research assets in a standardized format, their reuse will be less time consuming and error-prone. Standardized formats also facilitate large-scale meta-scientific investigations (Gorgolewski et al., 2016), and inclusion of data sets for meta-analyses.

More generally, in addition to the convenience of a single-platform *vertical* workflow, adopting R standards during asset creation has the side effect of demanding some amount of attention on how research products are curated, stored, documented, and shared. Drawing attention to these details may serve to increase the reproducibility of our work.

References

Aust, F. (2019). *Citr: 'RStudio' add-in to insert markdown citations*. Retrieved from https://CRAN.R-project.org/package=citr

Aust, F., & Barth, M. (2018). *papaja: Create APA manuscripts with R Markdown*. Retrieved from https://github.com/crsh/papaja

Crump, M. J. C. (2019). *Jspsychr: Templates and functions for writing and running jspsych experiments from r-studio*. Retrieved from https://github.com/CrumpLab/jspsychr

Crump, M. J. C., & Vuorre, M. (n.d.). *Vertical: Reproducible worfklow for psychological science research asset creation and communication*. Retrieved from https://github.com/CrumpLab/vertical

De Leeuw, J. R. (2015). JsPsych: A javascript library for creating behavioral experiments in a web browser. *Behavior Research Methods*, *47*(1), 1–12.

Gorgolewski, K. J., Auer, T., Calhoun, V. D., Craddock, R. C., Das, S., Duff, E. P., ... Poldrack, R. A. (2016). The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Scientific Data*, *3*, 160044. https://doi.org/10.1038/sdata.2016.44

Hardwicke, T. E., Mathur, M. B., MacDonald, K., Nilsonne, G., Banks, G. C., Kidwell, M. C., ... Frank, M. C. (2018). Data availability, reusability, and analytic reproducibility: Evaluating the impact of a mandatory open data policy at the journal Cognition. *Royal Society Open Science*, *5*(8), 180448. https://doi.org/10.1098/rsos.180448

Houtkoop, B. L., Chambers, C., Macleod, M., Bishop, D. V. M., Nichols, T. E., & Wagenmakers, E.-J. (2018). Data Sharing in Psychology: A Survey on Barriers and Preconditions. *Advances in Methods and Practices in Psychological Science*, *1*(1), 70–85. https://doi.org/10.1177/2515245917751886

Klein, O., Hardwicke, T. E., Aust, F., Breuer, J., Danielsson, H., Mohr, A. H., ... Frank, M. C. (2018). A Practical Guide for Transparency in Psychological Science. *Collabra: Psychology*, *4*(1), 20. https://doi.org/10.1525/collabra.158

Lindsay, D. S. (2017). Sharing Data and Materials in Psychological Science. *Psychological Science*, *28*(6), 699–702. https://doi.org/10.1177/0956797617704015

Martone, M. E., Garcia-Castro, A., & VandenBos, G. R. (2018). Data sharing in psychology. *American Psychologist*, *73*(2), 111–125. https://doi.org/10.1037/amp0000242

Morey, R. D., Chambers, C. D., Etchells, P. J., Harris, C. R., Hoekstra, R., Lakens, D., ... Zwaan, R. A. (2016). The Peer Reviewers Openness Initiative: Incentivizing open research practices through peer review. *Royal Society Open Science*, *3*(1), 150547. https://doi.org/10.1098/rsos.150547

Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., ... Yarkoni, T. (2015). Promoting an open research culture. *Science*, *348*(6242), 1422–1425. https://doi.org/10.1126/science.aab2374

R Core Team. (2019). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Rouder, J. N. (2016). The what, why, and how of born-open data. *Behavior Research Methods*, *48*(3), 1062–1069. https://doi.org/10.3758/s13428-015-0630-z

Rouder, J. N., Haaf, J. M., & Snyder, H. K. (2019). Minimizing Mistakes in Psychological Science. *Advances in Methods and Practices in Psychological Science*, *2*(1), 3–11. https://doi.org/10.1177/2515245918801915

RStudio Team. (2016). *RStudio: Integrated Development Environment for R*. Boston, MA: RStudio, Inc. Retrieved from http://www.rstudio.com/

Silberzahn, R., Uhlmann, E. L., Martin, D. P., Anselmi, P., Aust, F., Awtrey, E., ... Nosek, B. A. (2018). Many Analysts, One Data Set: Making Transparent How Variations in Analytic Choices Affect Results. *Advances in Methods and Practices in Psychological Science*, *1*(3), 337–356. https://doi.org/10.1177/2515245917747646

Thorne, W. B. (2019). *Posterdown: An r package built to generate reproducible conference posters for the academic and professional world where powerpoint and pages just won't cut it*. Retrieved from https://github.com/brentthorne/posterdown

Vanpaemel, W., Vermorgen, M., Deriemaecker, L., & Storms, G. (2015). Are We Wasting a Good Crisis? The Availability of Psychological Research Data after the Storm. *Collabra: Psychology*, *1*(1), Art. 3. https://doi.org/10.1525/collabra.13

Vuorre, M., & Curley, J. P. (2018). Curating Research Assets: A Tutorial on the Git Version Control System. *Advances in Methods and Practices in Psychological Science*, *1*(2), 219–236. https://doi.org/10.1177/2515245918754826

Wicherts, J. M., Borsboom, D., Kats, J., & Molenaar, D. (2006). The poor availability of psychological research data for reanalysis. *American Psychologist*, *61*(7), 726–728. https://doi.org/10.1037/0003-066X.61.7.726

Wickham, H. (2011). Testthat: Get started with testing. *The R Journal*, *3*, 5–10. Retrieved from https://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf

Wickham, H. (2015). *R Packages: Organize, Test, Document, and Share Your Code*. "O'Reilly Media, Inc.". Retrieved from http://r-pkgs.had.co.nz/

Wickham, H., & Hesselberth, J. (2019). *Pkgdown: Make static html documentation for a package*. Retrieved from https://CRAN.R-project.org/package=pkgdown

Wickham, H., Hester, J., & Chang, W. (2019). *Devtools: Tools to make developing r packages easier*. Retrieved from https://CRAN.R-project.org/package=devtools

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., … Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, *3*, 160018. https://doi.org/10.1038/sdata.2016.18

Xie, Y., Allaire, J. J., & Grolemund, G. (2018). *R markdown: The definitive guide*. Boca Raton, Florida: Chapman; Hall/CRC. Retrieved from https://bookdown.org/yihui/rmarkdown

Yee, S. J. W., & Debbie. (2017). Why You Should Become a UseR: A Brief Introduction to R. *APS Observer*, *30*(3). Retrieved from https://www.psychologicalscience.org/observer/why-you-should-become-a-user-a-brief-introduction-to-r

Aust, F. (2019). *Citr: 'RStudio' add-in to insert markdown citations*. Retrieved from https://CRAN.R-project.org/package=citr

Aust, F., & Barth, M. (2018). *papaja: Create APA manuscripts with R Markdown*. Retrieved from https://github.com/crsh/papaja

Crump, M. J. C. (2019). *Jspsychr: Templates and functions for writing and running jspsych experiments from r-studio*. Retrieved from https://github.com/CrumpLab/jspsychr

Crump, M. J. C., & Vuorre, M. (n.d.). *Vertical: Reproducible worfklow for psychological science research asset creation and communication*. Retrieved from https://github.com/CrumpLab/vertical

De Leeuw, J. R. (2015). JsPsych: A javascript library for creating behavioral experiments in a web browser. *Behavior Research Methods*, *47*(1), 1–12.

Gorgolewski, K. J., Auer, T., Calhoun, V. D., Craddock, R. C., Das, S., Duff, E. P., … Poldrack, R. A. (2016). The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Scientific Data*, *3*, 160044. https://doi.org/10.1038/sdata.2016.44

Hardwicke, T. E., Mathur, M. B., MacDonald, K., Nilsonne, G., Banks, G. C., Kidwell, M. C., … Frank, M. C. (2018). Data availability, reusability, and analytic reproducibility: Evaluating the impact of a mandatory open data policy at the journal Cognition. *Royal Society Open Science*, *5*(8), 180448. https://doi.org/10.1098/rsos.180448

Houtkoop, B. L., Chambers, C., Macleod, M., Bishop, D. V. M., Nichols, T. E., & Wagenmakers, E.-J. (2018). Data Sharing in Psychology: A Survey on Barriers and Preconditions. *Advances in Methods and Practices in Psychological Science*, *1*(1), 70–85. https://doi.org/10.1177/2515245917751886

Klein, O., Hardwicke, T. E., Aust, F., Breuer, J., Danielsson, H., Mohr, A. H., … Frank, M. C. (2018). A Practical Guide for Transparency in Psychological Science. *Collabra: Psychology*, *4*(1), 20. https://doi.org/10.1525/collabra.158

Lindsay, D. S. (2017). Sharing Data and Materials in Psychological Science. *Psychological Science*, *28*(6), 699–702. https://doi.org/10.1177/0956797617704015

Martone, M. E., Garcia-Castro, A., & VandenBos, G. R. (2018). Data sharing in psychology. *American Psychologist*, *73*(2), 111–125. https://doi.org/10.1037/amp0000242

Morey, R. D., Chambers, C. D., Etchells, P. J., Harris, C. R., Hoekstra, R., Lakens, D., … Zwaan, R. A. (2016). The Peer Reviewers Openness Initiative: Incentivizing open research practices through peer review. *Royal Society Open Science*, *3*(1), 150547. https://doi.org/10.1098/rsos.150547

Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., … Yarkoni, T. (2015). Promoting an open research culture. *Science*, *348*(6242), 1422–1425. https://doi.org/10.1126/science.aab2374

R Core Team. (2019). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Rouder, J. N. (2016). The what, why, and how of born-open data. *Behavior Research Methods*, *48*(3), 1062–1069. https://doi.org/10.3758/s13428-015-0630-z

Rouder, J. N., Haaf, J. M., & Snyder, H. K. (2019). Minimizing Mistakes in Psychological Science. *Advances in Methods and Practices in Psychological Science*, *2*(1), 3–11. https://doi.org/10.1177/2515245918801915

RStudio Team. (2016). *RStudio: Integrated Development Environment for R*. Boston, MA: RStudio, Inc. Retrieved from http://www.rstudio.com/

Silberzahn, R., Uhlmann, E. L., Martin, D. P., Anselmi, P., Aust, F., Awtrey, E., … Nosek, B. A. (2018). Many Analysts, One Data Set: Making Transparent How Variations in Analytic Choices Affect Results. *Advances in Methods and Practices in Psychological Science*, *1*(3), 337–356. https://doi.org/10.1177/2515245917747646

Thorne, W. B. (2019). *Posterdown: An r package built to generate reproducible conference posters for the academic and professional world where powerpoint and pages just won't cut it*. Retrieved from https://github.com/brentthorne/posterdown

Vanpaemel, W., Vermorgen, M., Deriemaecker, L., & Storms, G. (2015). Are We Wasting a Good Crisis? The Availability of Psychological Research Data after the Storm. *Collabra: Psychology*, *1*(1), Art. 3. https://doi.org/10.1525/collabra.13

Vuorre, M., & Curley, J. P. (2018). Curating Research Assets: A Tutorial on the Git Version Control System. *Advances in Methods and Practices in Psychological Science*, *1*(2), 219–236. https://doi.org/10.1177/2515245918754826

Wicherts, J. M., Borsboom, D., Kats, J., & Molenaar, D. (2006). The poor availability of psychological research data for reanalysis. *American Psychologist*, *61*(7), 726–728. https://doi.org/10.1037/0003-066X.61.7.726

Wickham, H. (2011). Testthat: Get started with testing. *The R Journal*, *3*, 5–10. Retrieved from https://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf

Wickham, H. (2015). *R Packages: Organize, Test, Document, and Share Your Code*. "O'Reilly Media, Inc.". Retrieved from http://r-pkgs.had.co.nz/

Wickham, H., & Hesselberth, J. (2019). *Pkgdown: Make static html documentation for a package*. Retrieved from https://CRAN.R-project.org/package=pkgdown

Wickham, H., Hester, J., & Chang, W. (2019). *Devtools: Tools to make developing r packages easier*. Retrieved from https://CRAN.R-project.org/package=devtools

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., … Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, *3*, 160018. https://doi.org/10.1038/sdata.2016.18

Xie, Y., Allaire, J. J., & Grolemund, G. (2018). *R markdown: The definitive guide*. Boca Raton, Florida: Chapman; Hall/CRC. Retrieved from https://bookdown.org/yihui/rmarkdown

Yee, S. J. W., & Debbie. (2017). Why You Should Become a UseR: A Brief Introduction to R. *APS Observer*, *30*(3). Retrieved from https://www.psychologicalscience.org/observer/why-you-should-become-a-user-a-brief-introduction-to-r