

# RESEARCH FILE ORGANIZATION

*The Life-Changing Magic of Tidying Up*

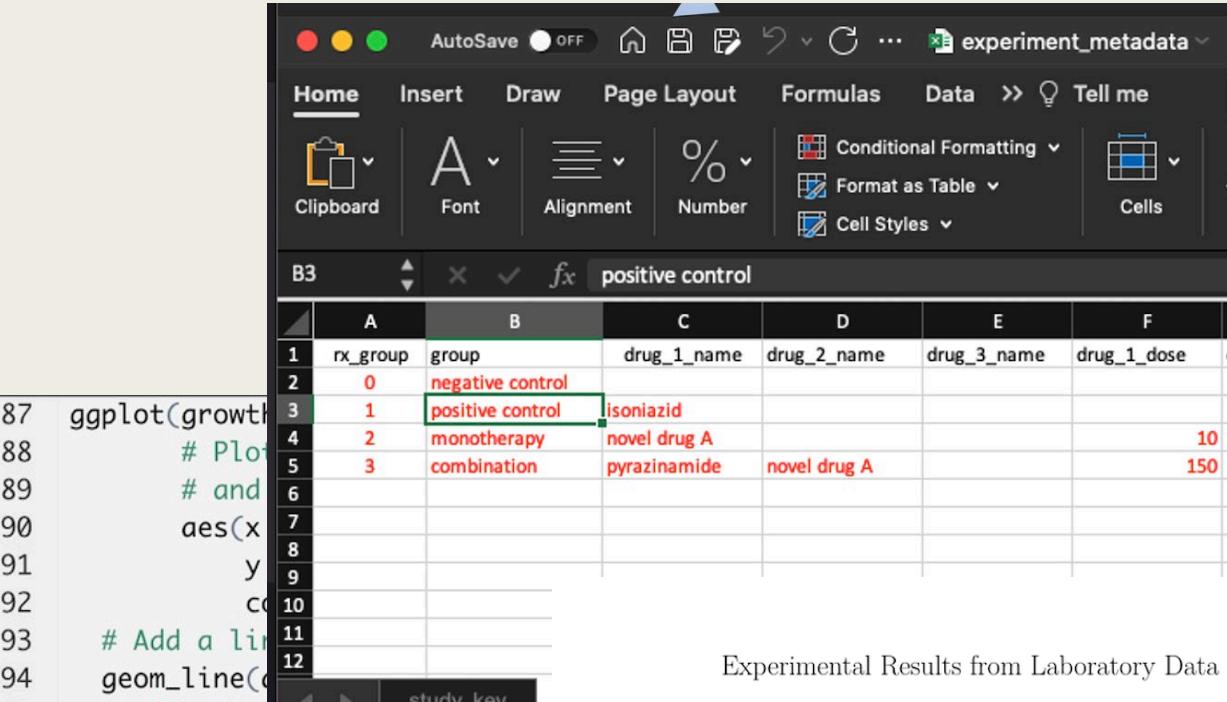
Brooke Anderson and Rachel Severson

# What we'll talk about

- Why should you organize your research files?
- How should you organize your research files?
- What can you do with well-organized files?
- Advanced topics: What can you do with a standardized directory structure? What can you do with Git?

# Research files

- Research datasets
- Code scripts for pre-processing and analyzing data
- Reports and presentations



Experimental Results from Laboratory Data

February 9, 2022

## Study information

Here is some information about this study, including the treatments that were investigated.

Table 1: Details of this study.

Study characteristic	Value in this study
Mouse strain	Balb/c
Route of administration	intrapulmonary aerosol
Treatments per week	3
Weeks of treatment	4
Measured inoculum of tuberculosis	3.55
Measured Mtb bacterial load one day after inoculation	2.15
Novel drug batch number	COMP-001-TR21

Table 2: Treatments tested in this study.

Type of treatment	Treatment
negative control	Untreated
	Isofurane (anesthetic)
	0.9% saline
monotherapy	Novel drug A, 10mg/kg by intrapulmonary aerosol
	Novel drug A, 25mg/kg by intrapulmonary aerosol
	Novel drug A, 50mg/kg by intrapulmonary aerosol

*Research Paper ■*

## Issues in Biomedical Research Data Management and Analysis: Needs and Barriers

NICHOLAS R. ANDERSON, MS, E. SALLY LEE, MS, J. SCOTT BROCKENBROUGH, PhD, MARK E. MINIE, PhD,  
SHERRILYNNE FULLER, PhD, JAMES BRINKLEY, MD, PhD, PETER TARCY-HORNOCH, MD

**Abstract** **Objectives:** A. Identify the current state of data management needs of academic biomedical researchers. B. Explore their anticipated data management and analysis needs. C. Identify barriers to addressing those needs.

**Design:** A multimodal needs analysis was conducted using a combination of an online survey and in-depth one-on-one semi-structured interviews. Subjects were recruited via an e-mail list representing a wide range of academic biomedical researchers in the Pacific Northwest.

**Measurements:** The results from 286 survey respondents were used to provide triangulation of the qualitative analysis of data gathered from 15 semi-structured in-depth interviews.

**Results:** Three major themes were identified: 1) there continues to be widespread use of basic general-purpose applications for core data management; 2) there is broad perceived need for additional support in managing and analyzing large datasets; and 3) the barriers to acquiring currently available tools are most commonly related to financial burdens on small labs and unmet expectations of institutional support.

**Conclusion:** Themes identified in this study suggest that at least some common data management needs will best be served by improving access to basic level tools such that researchers can solve their own problems. Additionally, institutions and informaticians should focus on three components: 1) facilitate and encourage the use of modern data exchange models and standards, enabling researchers to leverage a common layer of interoperability and analysis; 2) improve the ability of researchers to maintain provenance of data and models as they evolve over time through tools and the leveraging of standards; and 3) develop and support information management service cores that could assist in these previous components while providing researchers with unique data analysis and information design support within a spectrum of informatics capabilities.

■ *J Am Med Inform Assoc.* 2007;14:478–488. DOI 10.1197/jamia.M2114.

Downloaded from <https://academic.oup.com/jamia/article/14/4/478/788143>

## File organization in practice

One study surveyed over 250 biomedical researchers at the University of Washington.

Research Paper ■

## Issues in Biomedical Research Data Management and Analysis: Needs and Barriers

NICHOLAS R. ANDERSON, MS, E. SALLY LEE, MS, J. SCOTT BROCKENBROUGH, PhD, MARK E. MINIE, PhD,  
SHERRILYNNE FULLER, PhD, JAMES BRINKLEY, MD, PhD, PETER TARCY-HORNOCH, MD

**Abstract** Objectives: A. Identify the current state of data management needs of academic biomedical researchers. B. Explore their anticipated data management and analysis needs. C. Identify barriers to addressing those needs.

**Design:** A multimodal needs analysis was conducted using a combination of an online survey and in-depth one-on-one semi-structured interviews. Subjects were recruited via an e-mail list representing a wide range of academic biomedical researchers in the Pacific Northwest.

**Measurements:** The results from 286 survey respondents were used to provide triangulation of the qualitative analysis of data gathered from 15 semi-structured in-depth interviews.

**Results:** Three major themes were identified: 1) there continues to be widespread use of basic general-purpose applications for core data management; 2) there is broad perceived need for additional support in managing and analyzing large datasets; and 3) the barriers to acquiring currently available tools are most commonly related to financial burdens on small labs and unmet expectations of institutional support.

**Conclusion:** Themes identified in this study suggest that at least some common data management needs will best be served by improving access to basic level tools such that researchers can solve their own problems. Additionally, institutions and informaticians should focus on three components: 1) facilitate and encourage the use of modern data exchange models and standards, enabling researchers to leverage a common layer of interoperability and analysis; 2) improve the ability of researchers to maintain provenance of data and models as they evolve over time through tools and the leveraging of standards; and 3) develop and support information management service cores that could assist in these previous components while providing researchers with unique data analysis and information design support within a spectrum of informatics capabilities.

■ *J Am Med Inform Assoc.* 2007;14:478–488. DOI 10.1197/jamia.M2114.

Downloaded from <https://academic.oup.com/jamia/article/14/4/478/788143>

## File organization in practice

*They noted that, “a common theme surrounding data management and analysis was that many researchers prefer to utility their own individual methods to organize data. ... Some researchers admitted to having no organizational method at all, while others used whatever method best suited their individual needs.”*

*Research Paper ■*

## Issues in Biomedical Research Data Management and Analysis: Needs and Barriers

NICHOLAS R. ANDERSON, MS, E. SALLY LEE, MS, J. SCOTT BROCKENBROUGH, PhD, MARK E. MINIE, PhD,  
SHERRILYNNE FULLER, PhD, JAMES BRINKLEY, MD, PhD, PETER TARCY-HORNOCH, MD

**Abstract** **Objectives:** A. Identify the current state of data management needs of academic biomedical researchers. B. Explore their anticipated data management and analysis needs. C. Identify barriers to addressing those needs.

**Design:** A multimodal needs analysis was conducted using a combination of an online survey and in-depth one-on-one semi-structured interviews. Subjects were recruited via an e-mail list representing a wide range of academic biomedical researchers in the Pacific Northwest.

**Measurements:** The results from 286 survey respondents were used to provide triangulation of the qualitative analysis of data gathered from 15 semi-structured in-depth interviews.

**Results:** Three major themes were identified: 1) there continues to be widespread use of basic general-purpose applications for core data management; 2) there is broad perceived need for additional support in managing and analyzing large datasets; and 3) the barriers to acquiring currently available tools are most commonly related to financial burdens on small labs and unmet expectations of institutional support.

**Conclusion:** Themes identified in this study suggest that at least some common data management needs will best be served by improving access to basic level tools such that researchers can solve their own problems.

Additionally, institutions and informaticians should focus on three components: 1) facilitate and encourage the use of modern data exchange models and standards, enabling researchers to leverage a common layer of interoperability and analysis; 2) improve the ability of researchers to maintain provenance of data and models as they evolve over time through tools and the leveraging of standards; and 3) develop and support information management service cores that could assist in these previous components while providing researchers with unique data analysis and information design support within a spectrum of informatics capabilities.

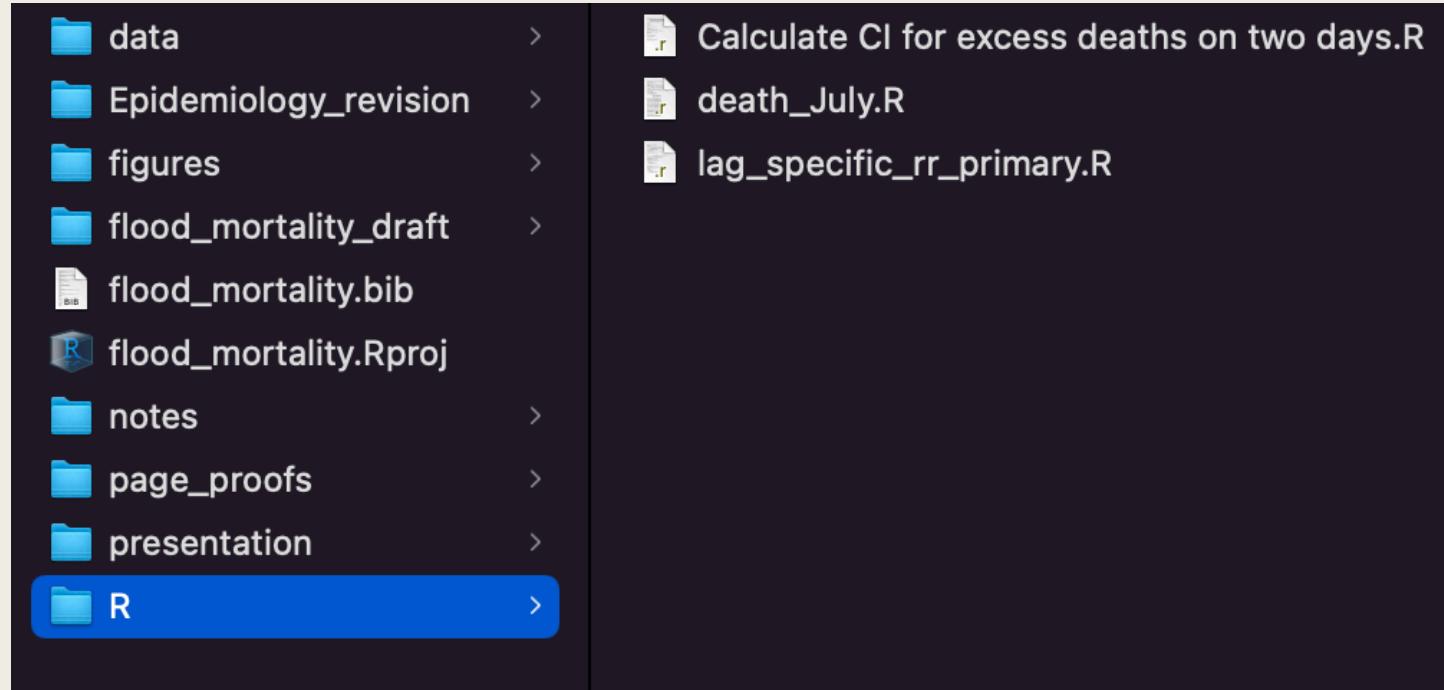
■ *J Am Med Inform Assoc.* 2007;14:478–488. DOI 10.1197/jamia.M2114.

Downloaded from <https://academic.oup.com/jamia/article/14/4/478/788143>

## Respondent answers

- One respondent answers, “They’re not organized in any way--they’re just thrown into files under different projects.”
- Another respondent: “I grab them when I need them, they’re not organized in any decent way.”

# WHY ORGANIZE FILES



## Efficiency

- With good organization, “**Methods and data sections in papers practically write themselves**, with no time wasted in frenzied hunting for missing information.” – Baker, 2016
- You can quickly get new people up to speed and efficiently pass along lab research projects as people come and go from the lab group
- You can write tools that work with the structure

# Efficiency



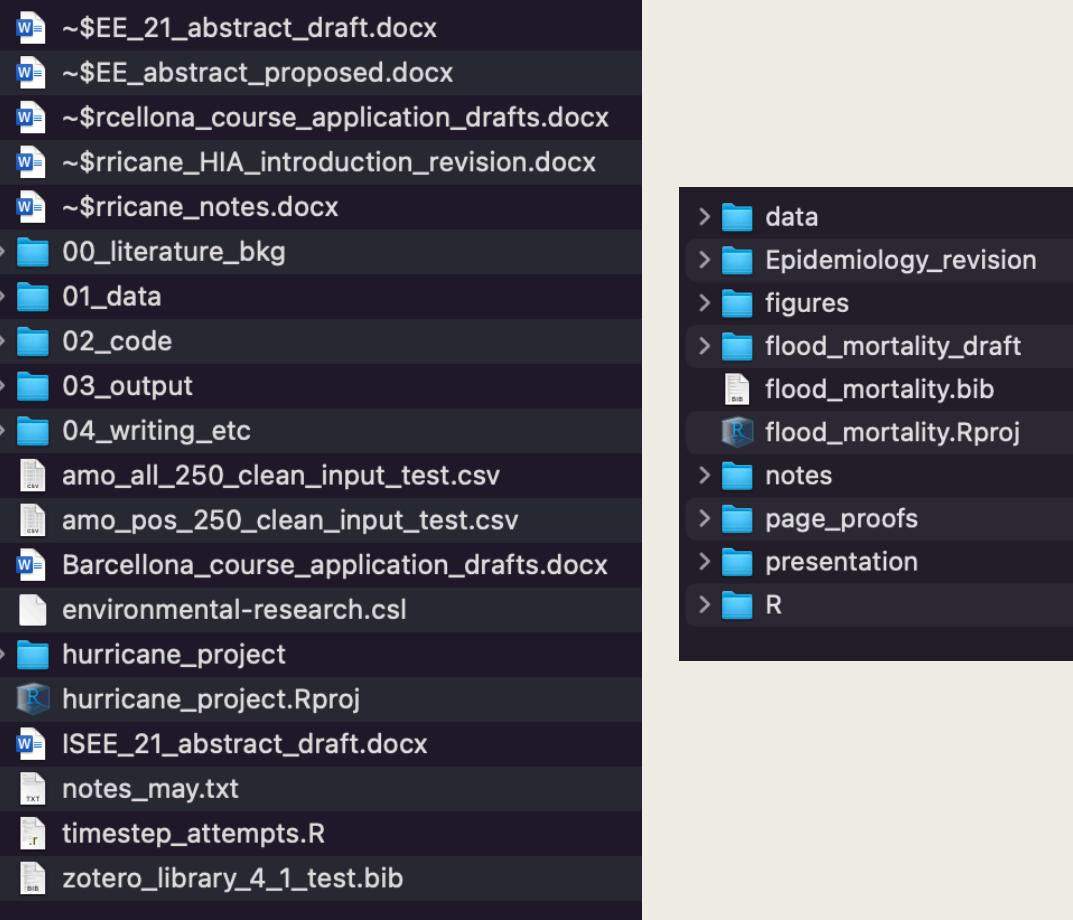
**“Everything you do, you will probably have to do over again. ... If you have organized and documented your work clearly, then repeating the experiment with the new data or the new parameterization will be much, much easier.” --Noble, 2009**



# Sharing research files

We're less likely to share things that aren't tidy.  
This is why we often blur our backgrounds  
for online meetings!

# Sharing research files

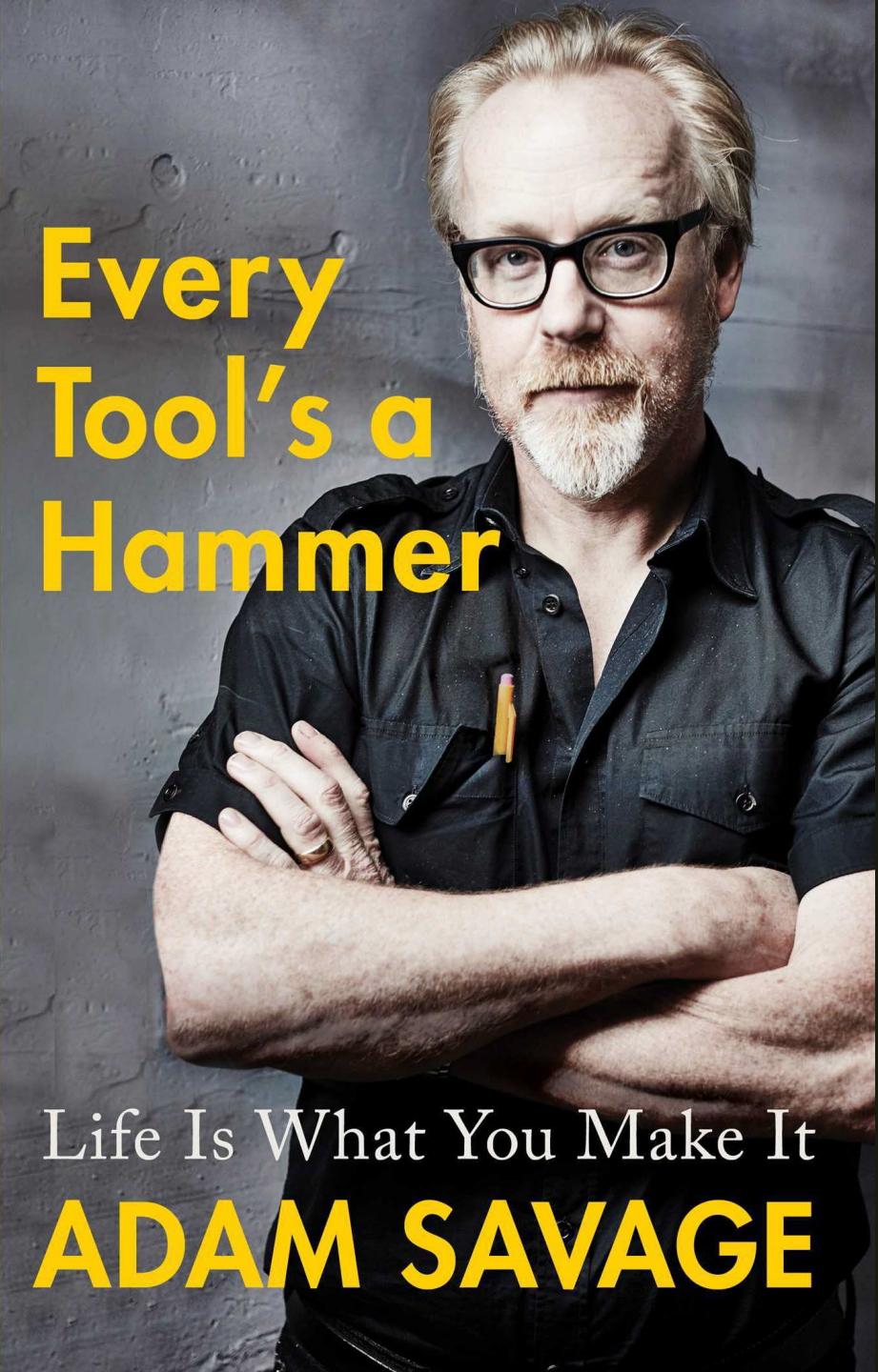


“Without clear instructions, many researchers struggle to avoid chaos in their file structures, and so are understandably reluctant to expose their workflow for others to see. This may be one of the reasons that so many requests for details about methods, including requests for data and code, are turned down or go unanswered.”

–Marwick et al., 2018



# ADAM SAVAGE



# Every Tool's a Hammer

Life Is What You Make It

# ADAM SAVAGE

## Knolling

"I started to clean up before heading upstairs at the end of the day, and lo and behold the shop became a far more efficient and well-oiled machine to work in.

The freed-up space in my mind and the open work space at my fingertips allowed me a lot of room, both mental and physical, to pursue a wide variety of projects, and I finally started to understand how much benefit was to be gained by taking the time to clean."



# KNOLLING LEGOS

# Why you should organize research files

- You can do your research more efficiently
- You can introduce new people to the project quickly
- You are more likely to share well-organized files, which is critical for research reproducibility
- You have more mental space to focus on substantive questions as you work

# HOW TO ORGANIZE FILES

# Using a single directory

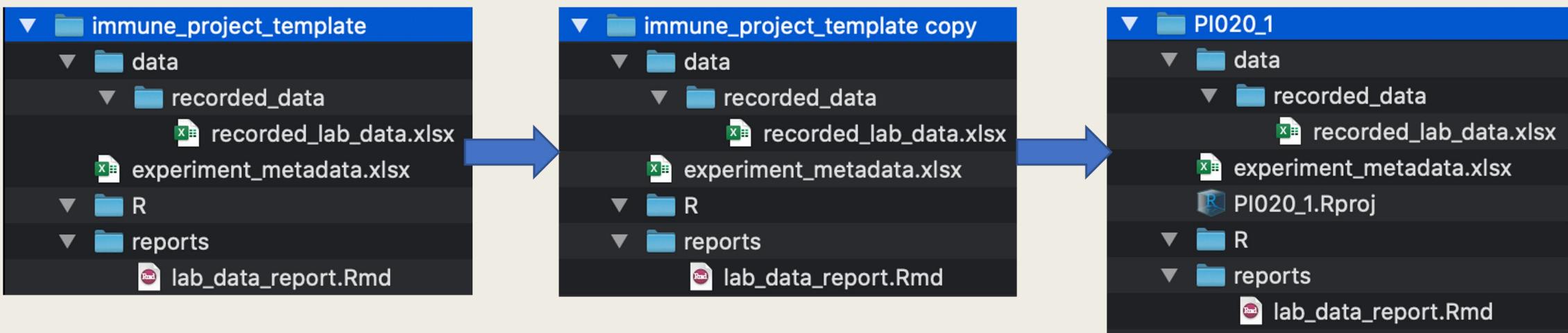
- Easy to set up as an R Project
- Easy to set up with version control
- Portable
- Shareable



# Blueprint versus physical template

1

Find the project directory template in the file finder program on your computer. Copy the entire directory, paste the copy where you want to store the project directory for your new study, and rename the directory to the name of your new study.



The physical implementation of this is very simple—create an empty example directory and copy and rename for new projects. The harder part is designing a good structure for the directory.

# Designing your system



Think about use



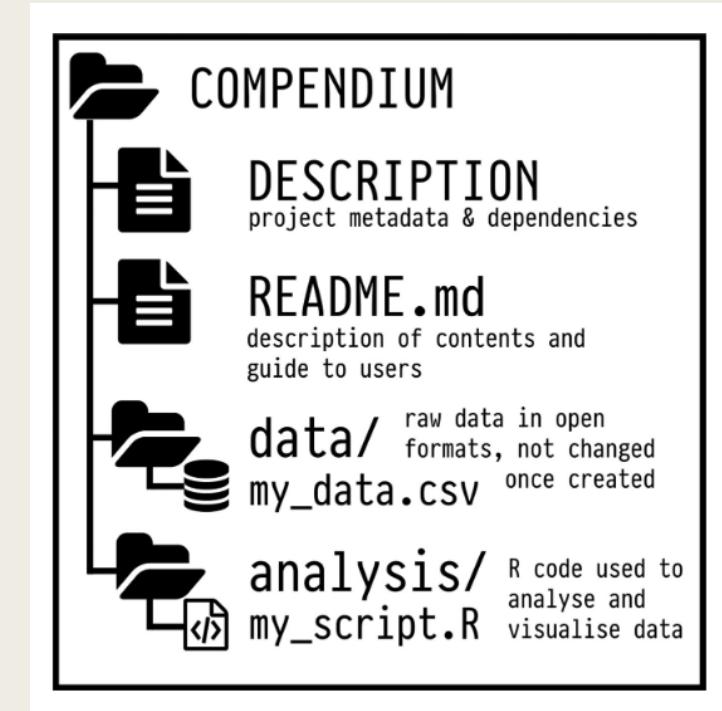
Think about reuse



Think about discoverability

# Use and Reuse

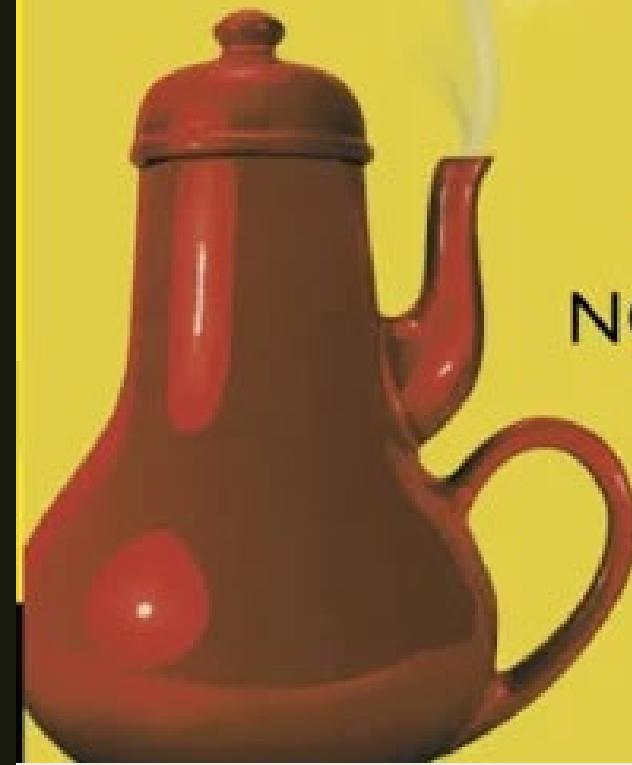
- Group similar types of files together
- Name subdirectories generically



# Discoverability

Can a user figure out how to use something quickly, easily, and correctly?

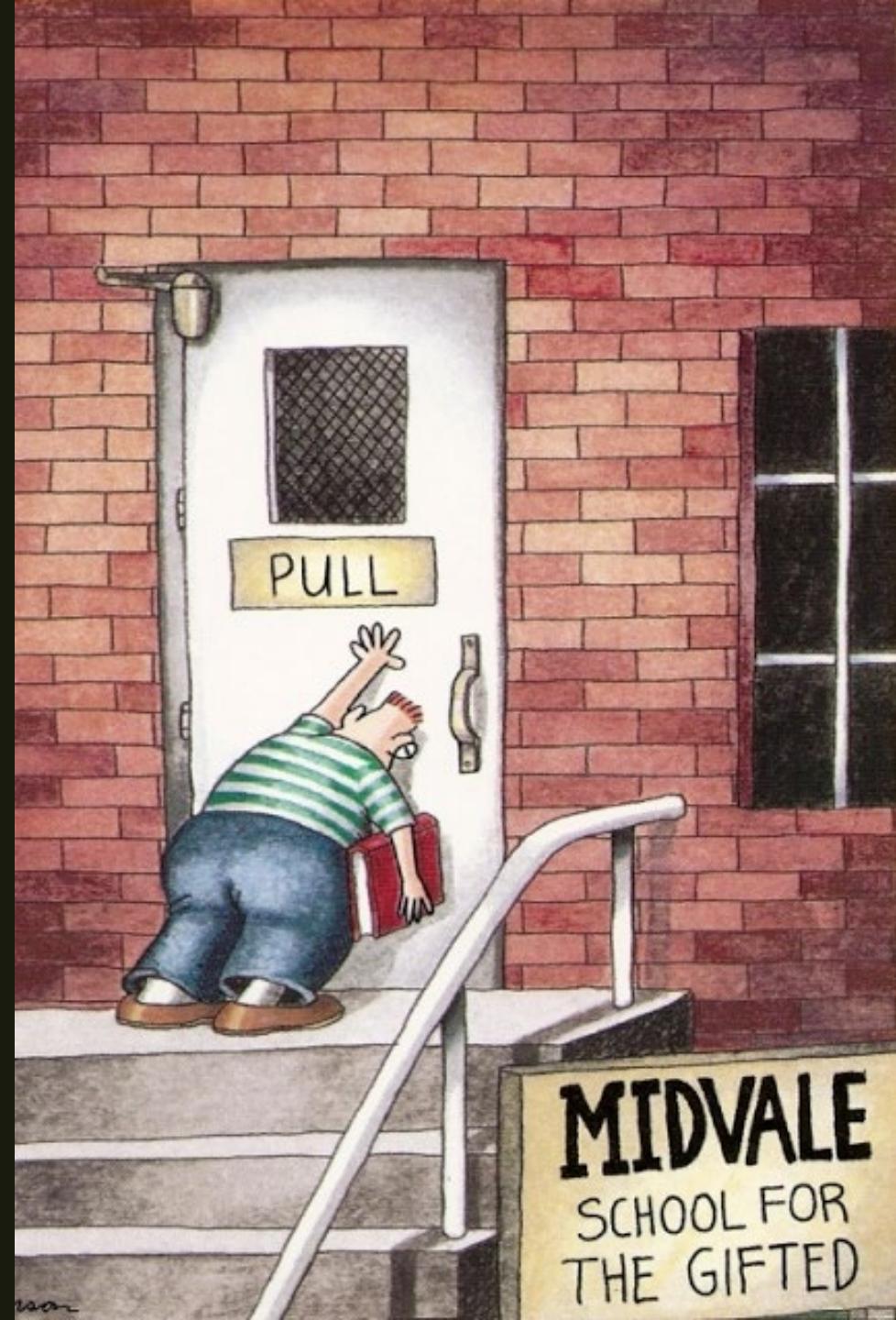
*The DESIGN  
of EVERYDAY  
THINGS*



DON  
NORMAN

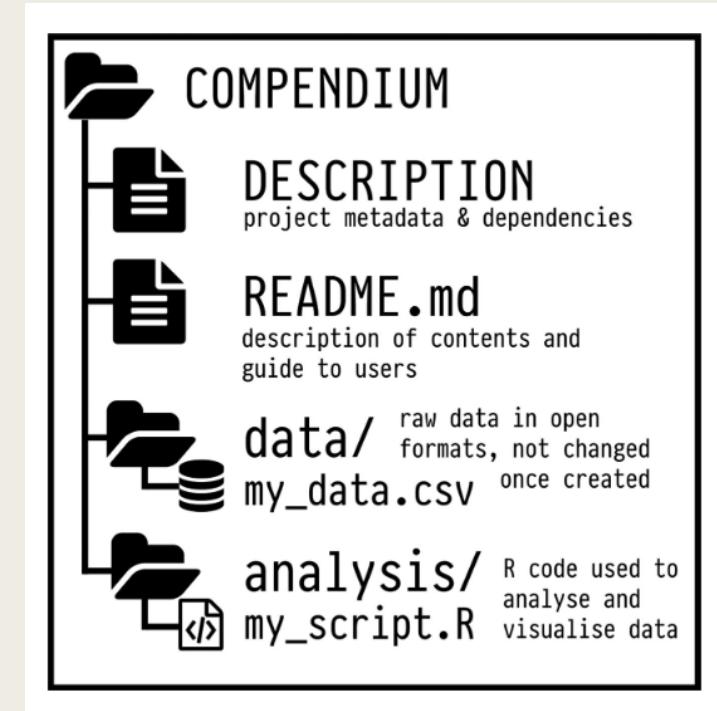
# Discoverability

Can a user figure out how to use something quickly, easily, and correctly?



# Discoverability

- “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.” –Marwick et al., 2018
- “The core guiding principle is simple: Someone unfamiliar with your project should be able to look at your computer files and understand in detail what you did and why.” –Noble, 2009

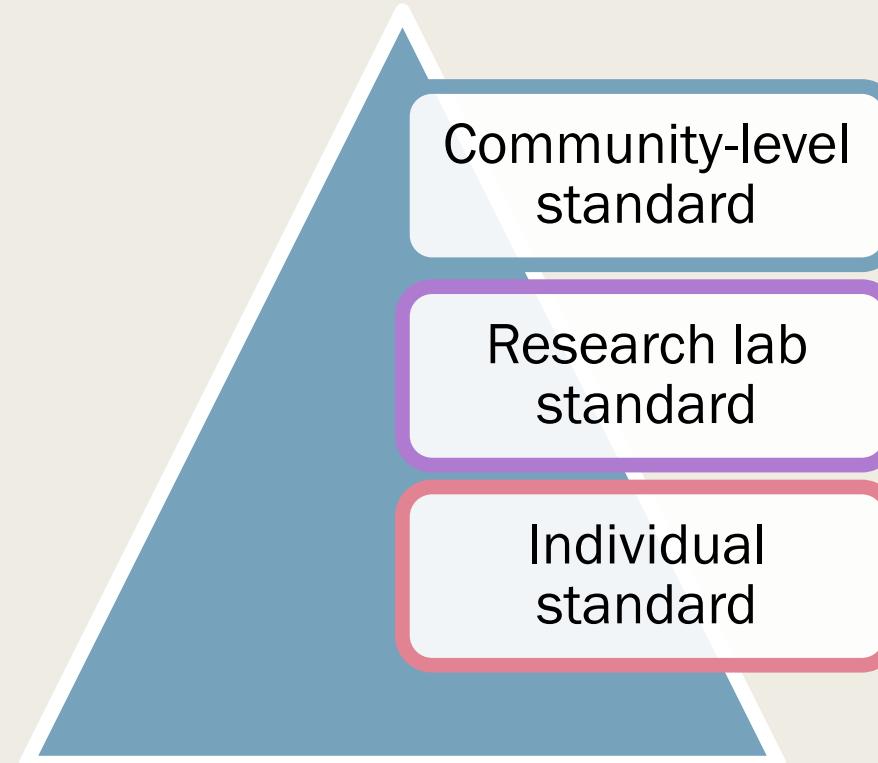




# Standards aid discoverability

If you follow a standard of your broader academic community, it makes project files easy for someone else from that community to navigate (Marwick et al., 2018).

# Use a standard system to organize files



# Marwick—Compendium based on R Package structure

THE AMERICAN STATISTICIAN  
2018, VOL. 72, NO. 1, 80–88  
<https://doi.org/10.1080/00031305.2017.1375986>



Check for updates

## Packaging Data Analytical Work Reproducibly Using R (and Friends)

Ben Marwick<sup>a</sup>, Carl Boettiger<sup>b</sup>, and Lincoln Mullen<sup>c</sup>

<sup>a</sup>University of Washington, Seattle, WA; <sup>b</sup>University of Wollongong, Wollongong, New South Wales; <sup>c</sup>University of California, Berkeley, CA; <sup>d</sup>George Mason University, Fairfax, VA

### 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 recognizable way for organizing 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.

### ARTICLE HISTORY

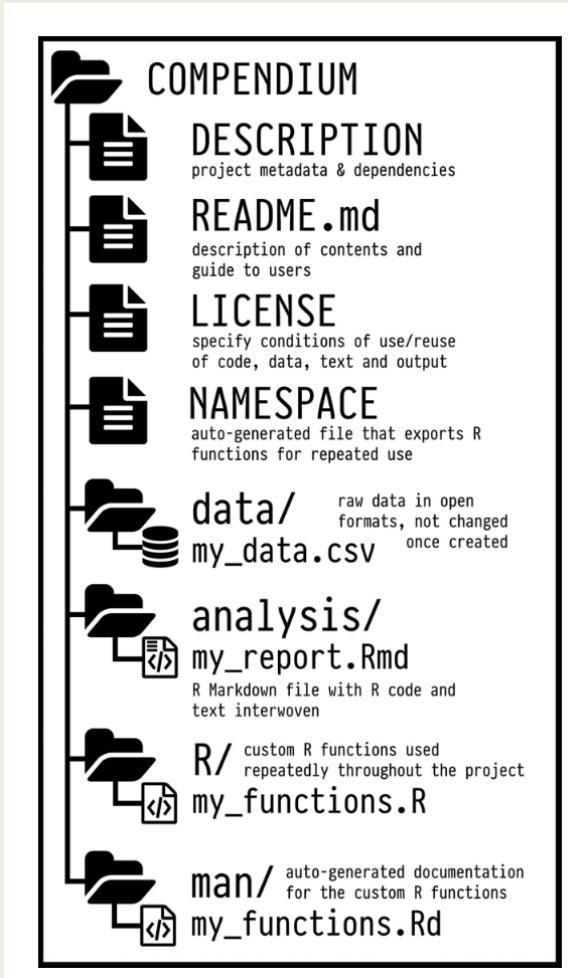
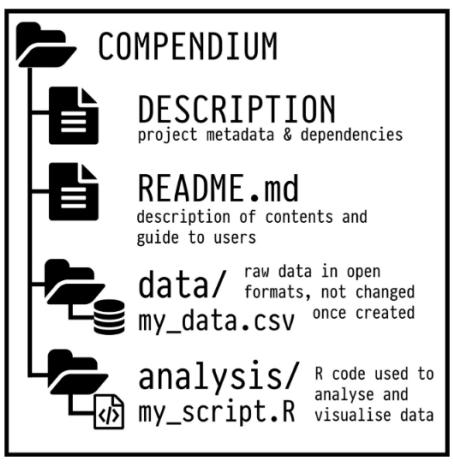
Received May 2017  
Revised August 2017

### KEYWORDS

Computational science; Data science; Open source software; Reproducible research

Marwick et al. suggest a series of directory structures in their 2018 paper that is based on the structure of R package directories

# Marwick Compendium



Their structures range from very simple to more complex to accommodate projects of different complexity

# How to organize your research files

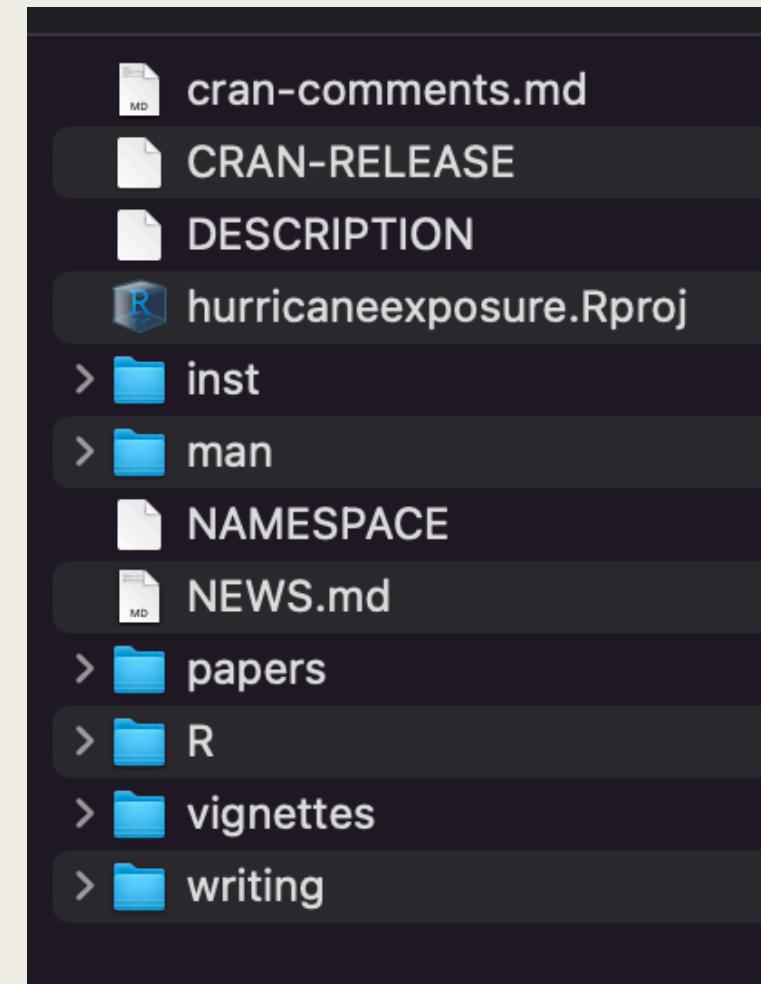
- Organize files for a project within a single computer directory
- Design a system that is easy to use, easy to reuse, and easy to figure out
- Use a standard system to organize your files

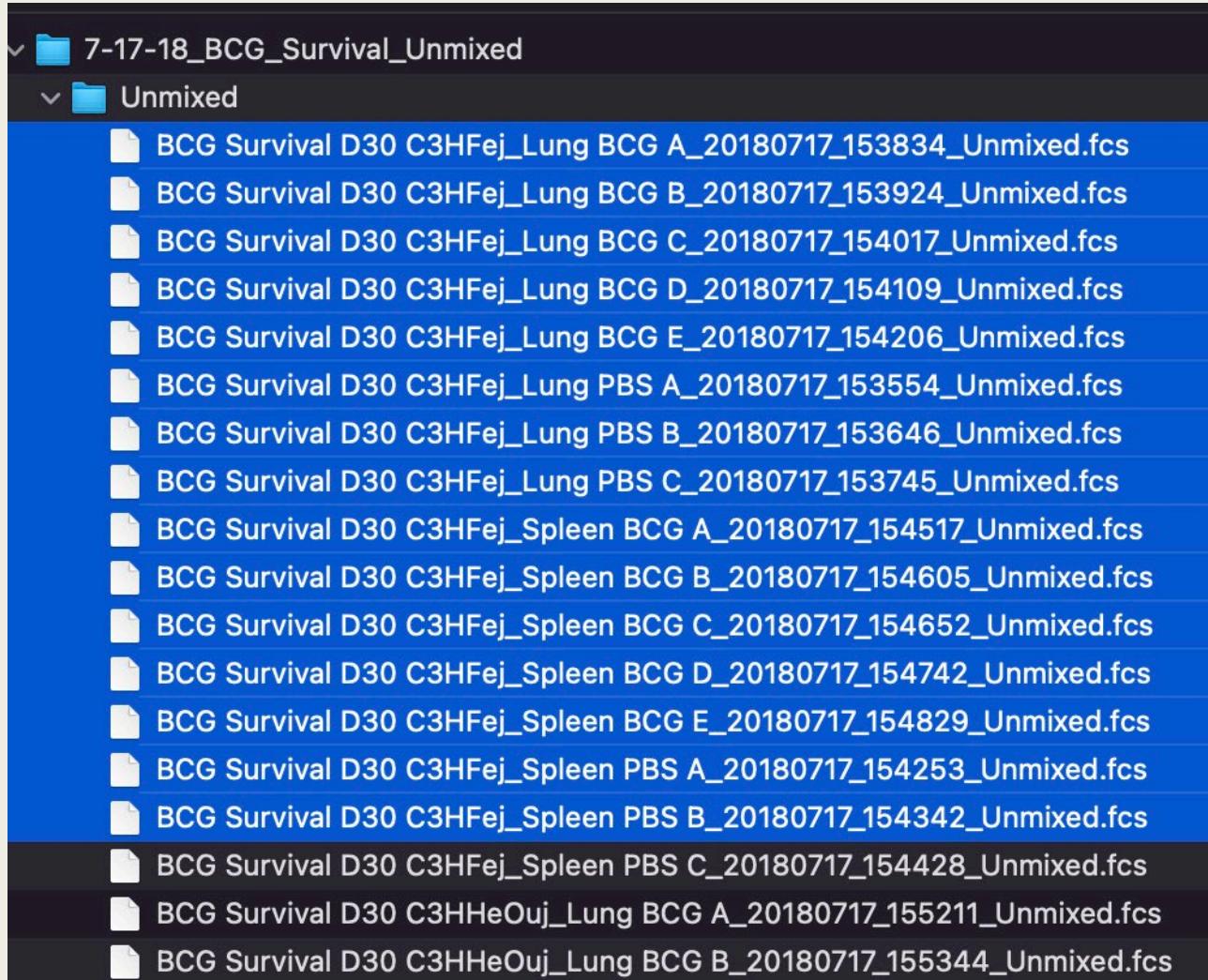
# ADVANCED TOPICS



# Coding to a standard directory structure

“ 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.**” –Marwick et al., 2018

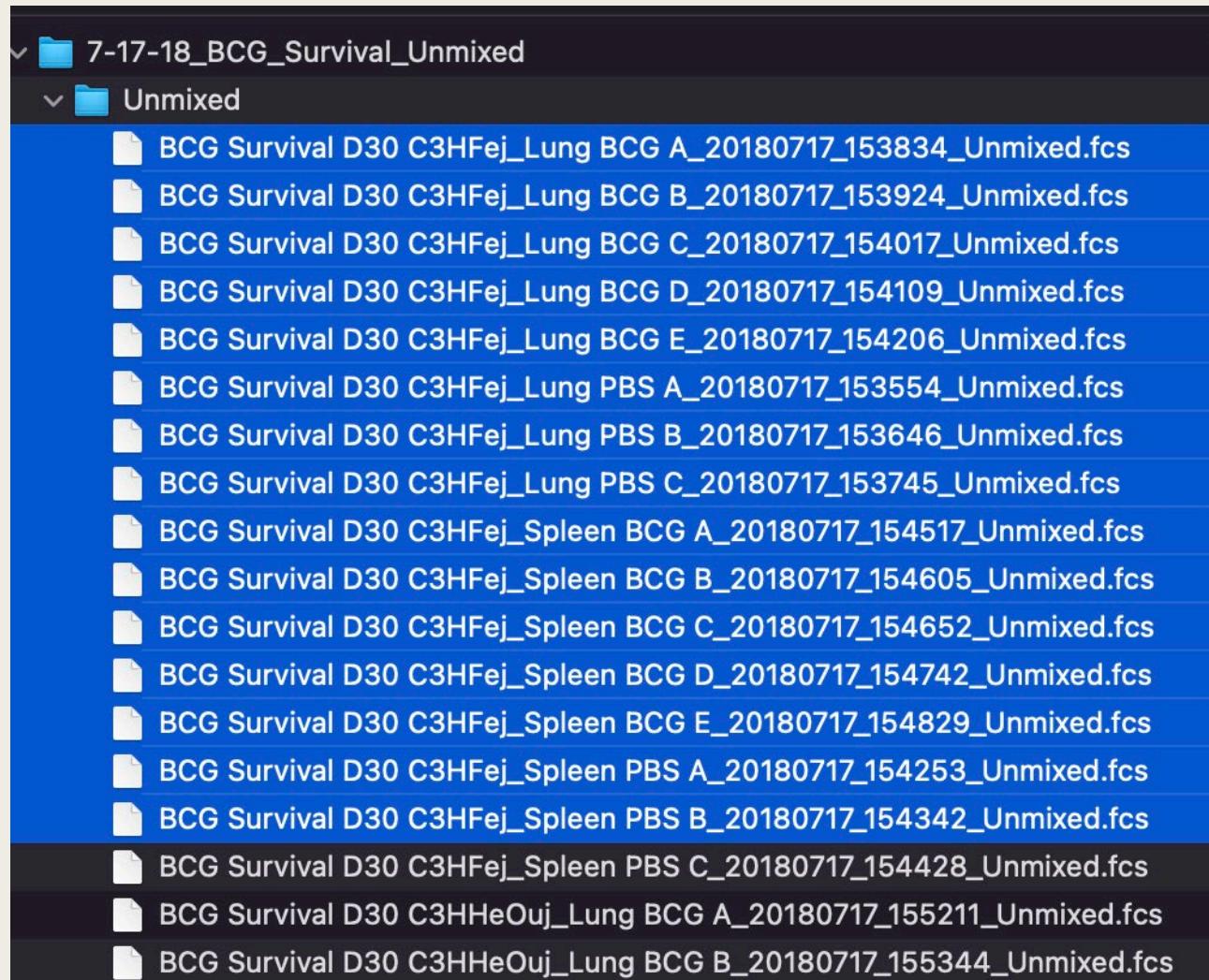




## Coding to a standard directory structure

“Organizing data files into a single directory with consistent filenames prepares us to iterate over *all* of our data...

Think of it this way: remember when you discovered you could select many files with your mouse cursor? With this trick, you could move 60 files as easily as six files... By using consistent file naming and directory organization, you can do the same programmatically suing the Unix shell and other programming languages.” – Buffalo, 2015

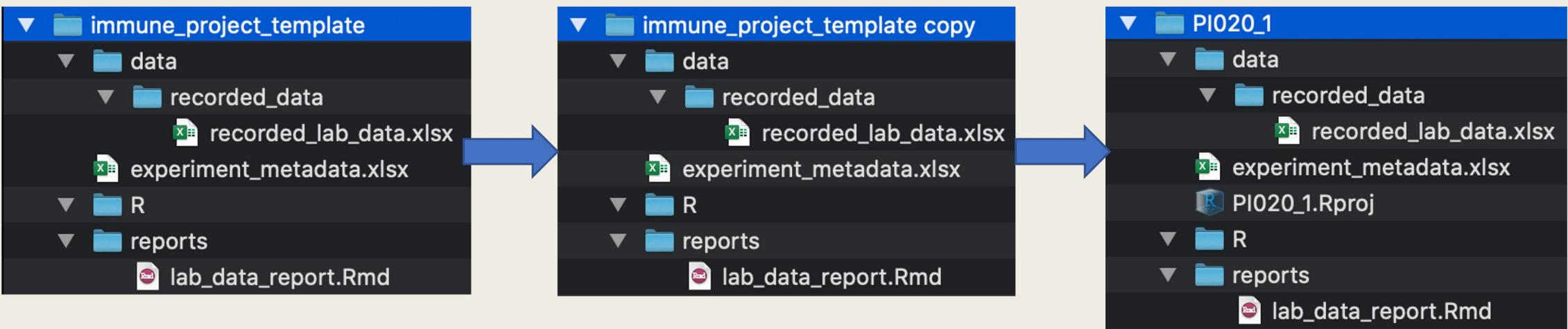


# Regular expressions and filenames

- R can list all files in a directory with a system call
- Map-apply programming lets you do the same thing to a list of files
- Regular expressions help you pull information out of character strings, including filepaths

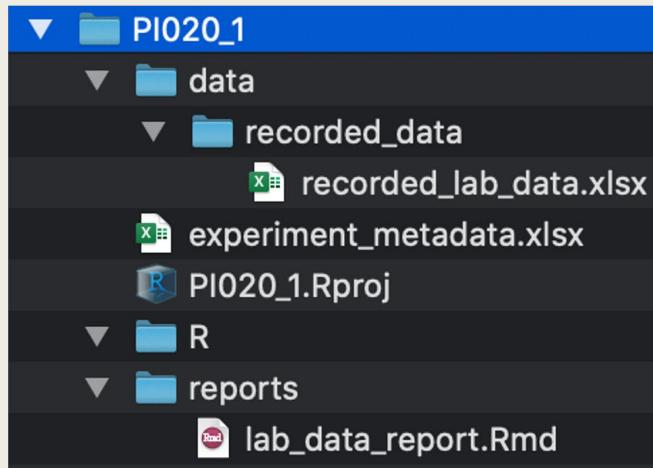
**1**

Find the project directory template in the file finder program on your computer. Copy the entire directory, paste the copy where you want to store the project directory for your new study, and rename the directory to the name of your new study.



2

Open data recording templates and replace the placeholder data (saved in red font to indicate that it's placeholder data) with data from the real project. Change the font color to black to show that these are data from the project, rather than placeholder data.

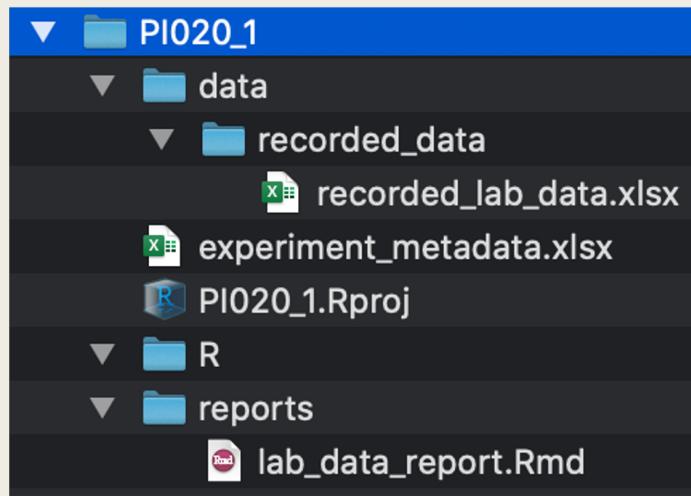


rx_group	week	weight_g
0	0	22.3
0	1	22.6
0	2	23.4
0	3	24.7
0	4	24.7
0	5	24.9
1	0	19.5
1	1	20.3
1	2	20.3
1	3	22.2
1	4	21.3
1	5	22
2	0	20.5
2	1	20.8
2	2	21.2
2	3	20.8

rx_group	week	weight_g
0	0	38.2
0	1	38.4
0	2	38.5
0	3	38.8
1	0	33.7
1	1	33
1	2	32.8
1	3	32.3
2	0	29.2
2	1	28.8
2	2	28.7
2	3	27.4
3	0	35.8
3	1	35.3
3	2	34.8
3	3	35.1
6	0	31.2

## 3

Open the project report template. Render it to PDF to create the report. If you'd like, you can make changes to the template Rmarkdown report file to customize it for this project.



The RStudio interface displays the 'lab\_data\_report.Rmd' file. The code editor shows the following RMarkdown code:

```
1 ---  
2 title: "Experimental Results from Laboratory Data"  
3 date: "r format(Sys.Date(), '%e %B %Y')"  
4 output:  
5   pdf_document:  
6     keep_tex: false  
7 header-includes:  
8   - \usepackage{booktabs}  
9   - \usepackage{longtable}  
10  - \usepackage{array}  
11  - \usepackage{multirow}  
12  - \usepackage{wrapfig}  
13  - \usepackage{float}  
14  - \usepackage{color}  
15  - \usepackage{pdflscape}  
16  - \usepackage{tabu}  
17  - \usepackage{threeparttable}  
18  - \usepackage{threeparttablex}  
19  - \usepackage[normalem]{ulem}  
20  - \usepackage{makecell}  
21  - \usepackage{xcolor}  
22 ---  
23  
24 ````{r setup, include=FALSE, message = FALSE, warning = FALSE}  
25 knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning = FALSE)  
26 library(tidyverse)  
27 library(readxl)  
28 library(knitr)  
29 library(kableExtra)  
30 library(scales)  
31 library(ggbeeswarm)  
32 library(broom)  
33 library(multcomp) # For Dunnett's test  
34 ````  
35
```

The rendered output on the right side of the interface includes:

- Study information**:  
Experimental Results from Laboratory Data  
February 25, 2022
- Table 1: Details of this study**:  

Study characteristic	Value in this study
Mouse strain	Balb/c
Route of administration	intrapulmonary aerosol
Treatments per week	3
Weeks of treatment	4
Measured inoculum of tuberculosis	3.55
Measured Mtb bacterial load one day after inoculation	2.15
Novel drug batch number	COMP-001-TR21
- Table 2: Treatments tested in this study**:  

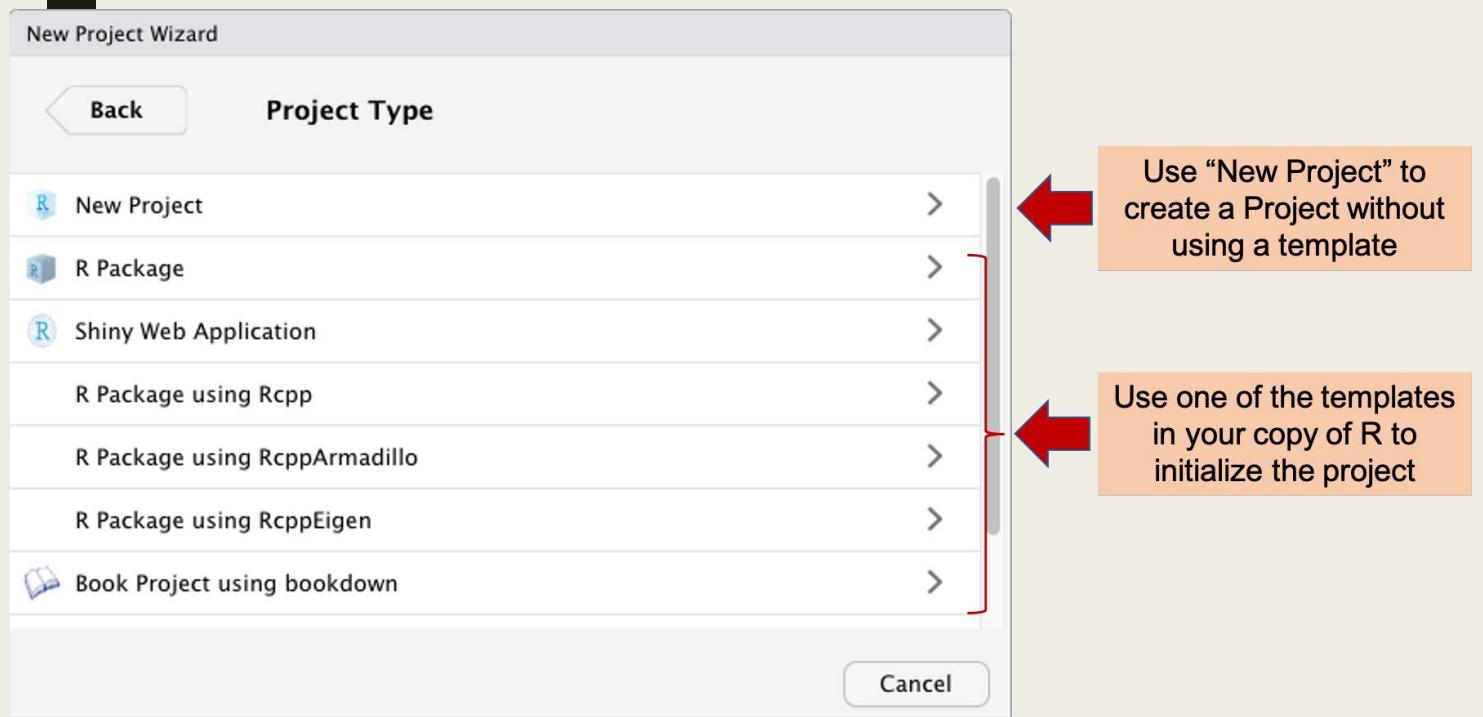
Type of treatment	Treatment
negative control	Untreated
	Isoflurane (anesthetic) 0.9% saline

monotherapy	Novel drug A, 10mg/kg by intrapulmonary aerosol
	Novel drug A, 25mg/kg by intrapulmonary aerosol
	Novel drug A, 50mg/kg by intrapulmonary aerosol

1

# Creating an R Project template



R Project templates create more customized R projects. When you begin a new project with one, it will self-populate with a directory structure and potentially template files.

You get new templates by installing R packages.

# Creating an R Project template

New Project Wizard

Back      Project Type

- R Package using RcppArmadillo >
- R Package using RcppEigen >
- Book project using bookdown >
- R Package using devtools >
- Immunology Drug-Testing Project >**
- Example Project Template >
- Simple R Markdown Website >

Record and analyze drug testing data for Dr. Gonzalez-Juarrero's research group

Cancel

→ New Project Wizard

Back

**Record and analyze drug testing data for Dr. Gonzalez-Juarrero's research group**

Directory name: PI020 Add the name of this study

Create project as subdirectory of: ~/Documents/my\_books/improve\_repro\_set\_of\_studies Browse...

Team members on this experiment: Mercedes Gonzalez-Juarrero Add the team members for this study

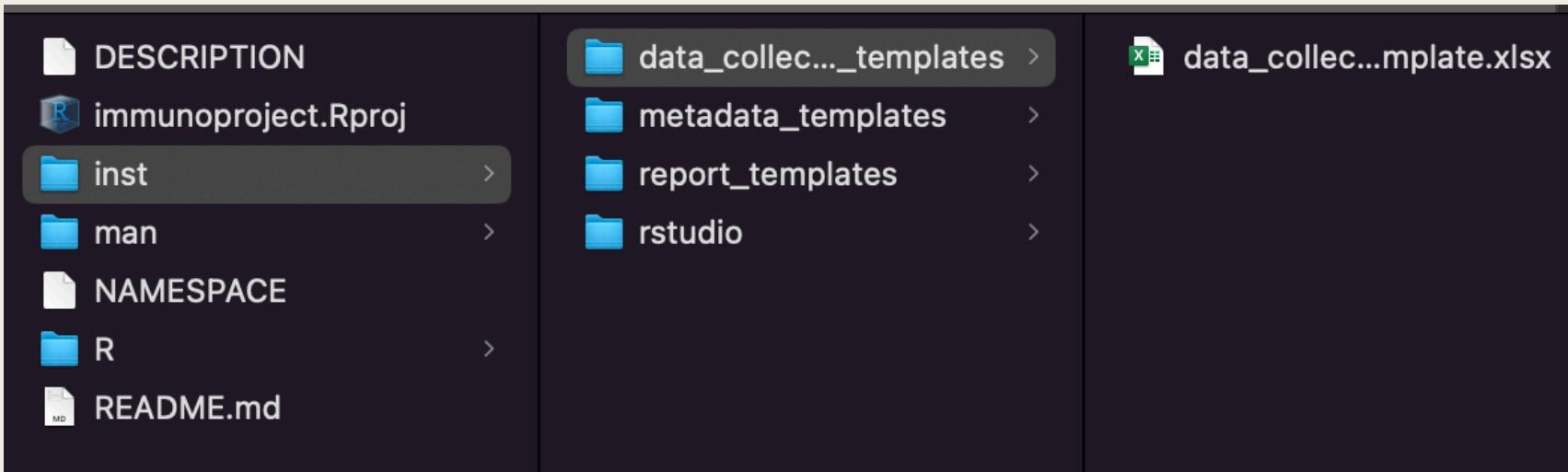
Includes flow cytometry data

Includes single cell RNA-seq data Indicate if the study also includes flow cytometry or single cell RNA-seq data

Open in new session

Create Project Cancel

# Creating an R Project template

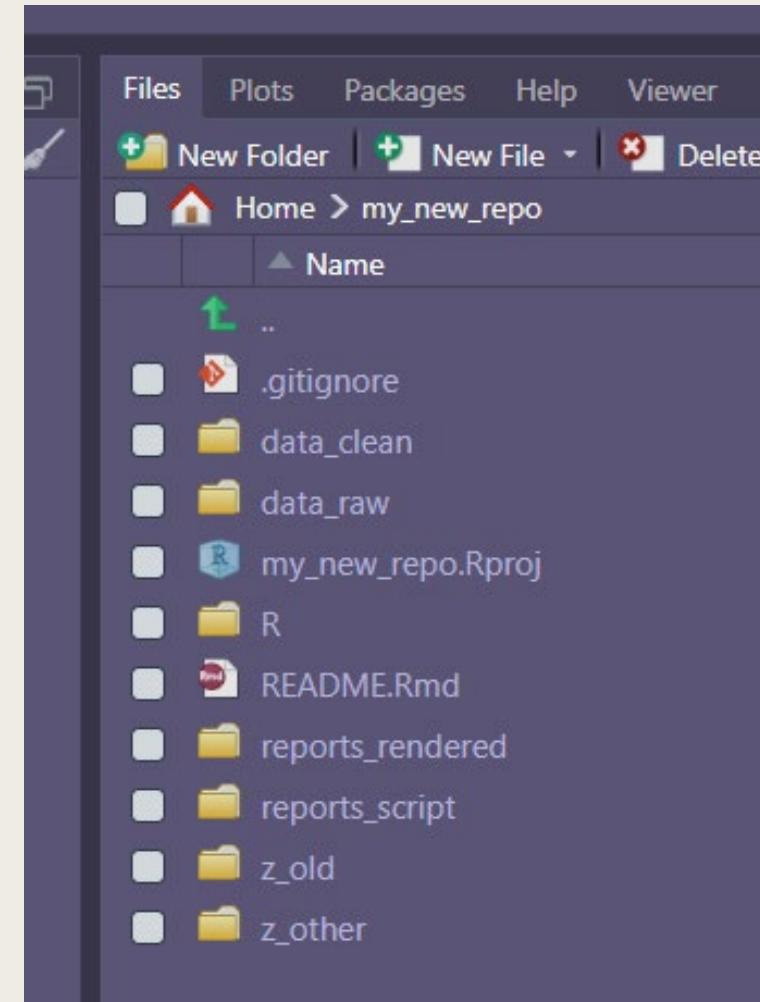
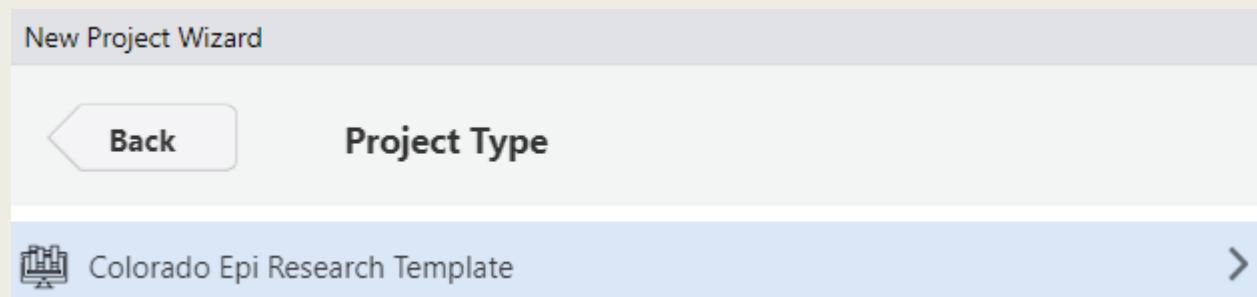


- You create an R Project template as an R package, which means it can be shared and loaded in the same way as other R packages
- Most template files will be saved in the “inst” directory
- You can use code in the R directory to set up the project directory structure and copy in template files
- You can also use code in the R directory to customize the project set-up based on selections the user makes in the R Project Wizard

# Creating an R Project template

- [ColoradoEpiResearch](#) and [ColoradoEpiApp](#) are two examples of custom R project templates.
- You can check out the code and make adjustments, or install and use as-is:

```
devtools::install_github("ColoradoEpiProject/Colorado  
EpiResearch")
```



# Git



- Git is a version control system.
- Git manages the evolution of a set of files (a repository) in a sane, highly structured way.
- Git is like the “Track Features” features from Microsoft Word on steroids.
- Git’s original purpose was to help groups of developers work collaboratively on big software projects. The data science community has re-purposed Git to use it to work on all of the different types of files that make up a typical project (data, figures, reports) in addition to source code.

# Should I Git...?



- The initial installation process and change in workflow is all worth it for the benefit of version control and ease of communicating and collaborating with other people.

## 1.3 Is it going to hurt?

Yes.

You have to install Git, get local Git talking to GitHub, and make sure RStudio can talk to local Git (and, therefore, GitHub). This is one-time or once-per-computer pain.

# Benefits of Git



- Exposure: if someone needs to see your work or if you want people to try out your code, they can clone or fork your repository with Git, or browse your project on GitHub.
- You can track the development of projects (like R packages) on GitHub. You can modify your fork to add features or fix bugs and send them back to the owner as a proposed change.
- If you need to collaborate on analysis or code development... use Git and GitHub! This is more analogous to Google Docs than it is to the edit, save, attach workflow.

# Git Basics



- Once you have successfully installed Git on your computer, for each individual project:
  - *Dedicate a folder to it.*
  - *Make it an RStudio Project.*
  - *Make it a Git repository.*
- Repository: a directory of files that Git manages holistically.
- Commit: functions like a snapshot of all the files in the repository at a specific moment.
- Diff: a set of differences between two commits.

# Git Basics



- In your normal workflow, instead of only saving individual files, periodically you make a **commit**, which snapshots all of the files in the project. You can inspect that snapshot or revert to it later.
- Every time you make a commit, you must also write a **commit message**. Messages should convey the *why* behind a change. The ‘diff’ will show the content of a change.
- The **history** includes commit messages and diffs.
- Git automatically assigns each commit with a unique identifying name (a Secure Hash Algorithm, or SHA), which is a string of 40 letters and numbers. You can use a commit’s SHA to revert to that commit.

# Git Basics



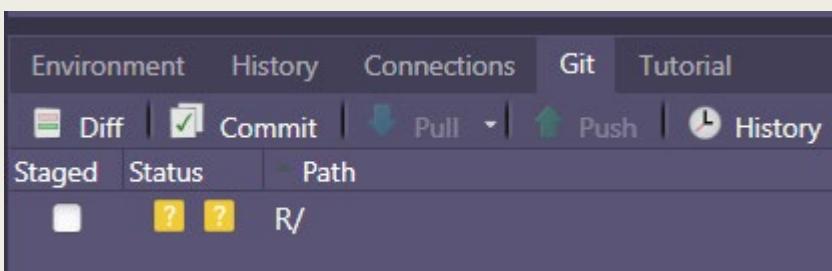
- You can work with Git from the command line.

```
rseverso@PHDC5CD2495BC0 MINGW64 ~/Documents/my_new_repo (master)
$ git add .
warning: LF will be replaced by CRLF in README.Rmd.
The file will have its original line endings in your working directory

rseverso@PHDC5CD2495BC0 MINGW64 ~/Documents/my_new_repo (master)
$ git commit --message "Initial commit"
[master (root-commit) a3b0ddb] Initial commit
 3 files changed, 100 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 README.Rmd
 create mode 100644 my_new_repo.Rproj

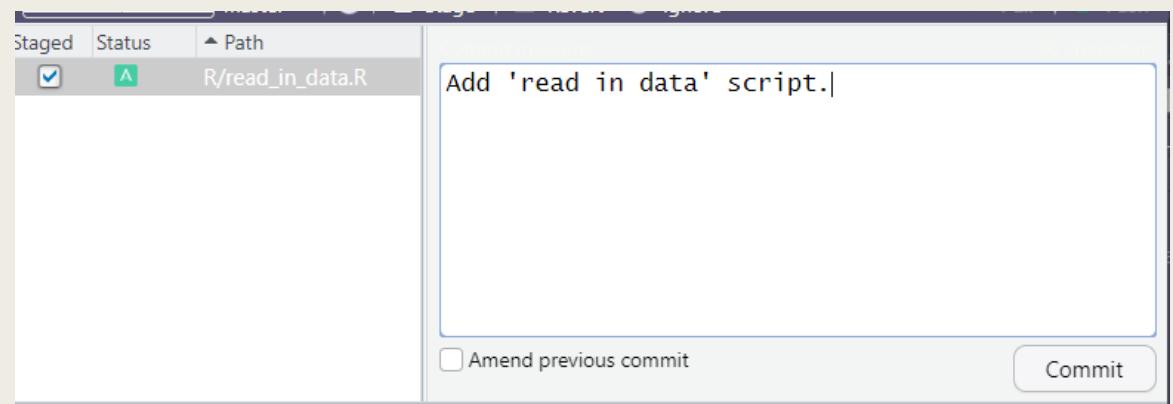
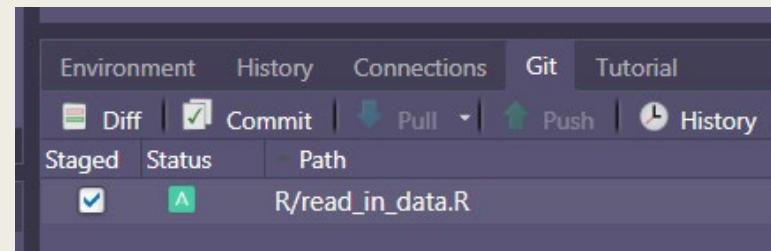
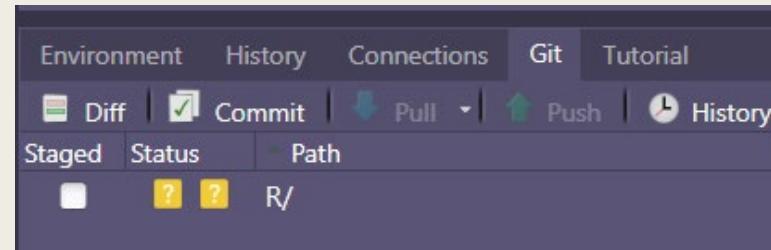
rseverso@PHDC5CD2495BC0 MINGW64 ~/Documents/my_new_repo (master)
$ █
```

- RStudio's GUI has a “Git” pane:



# Git Basics

- With RStudio's GUI,
  - *Files are automatically tracked.*
  - *You can stage a file, and*
  - *Commit with a commit message.*
  - *You can also see a log of your commits, and*
  - *Revert to previous commits.*



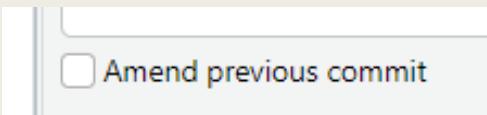
# Git Basics



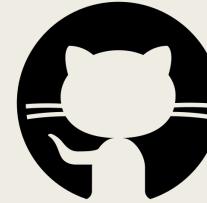
Using a Git commit is like using anchors and other protection when climbing. If you're crossing a dangerous rock face you want to make sure you've used protection to catch you if you fall. Commits play a similar role: if you make a mistake, you can't fall past the previous commit. Coding without commits is like free-climbing: you can travel much faster in the short-term, but in the long-term the chances of catastrophic failure are high! Like rock climbing protection, you want to be judicious in your use of commits. Committing too frequently will slow your progress; use more commits when you're in uncertain or dangerous territory. Commits are also helpful to others, because they show your journey, not just the destination.

R Packages, Hadley Wickham (Wickham (2015))

- As you are making progress in a project, you might end up with a lot of small commits, especially when you are in “uncertain or dangerous territory.”
- An option down the road can be to use a “Repeated Amend” workflow – instead of cluttering up your history with tiny commits, you build up a “good” commit gradually, by amending.



# Git and GitHub



- GitHub is one example of a hosting service that provides a home for your Git-based projects on the internet.
- GitHub is hyperlink-rich, and allows you to visualize your repository and its history. Exploring your repository's history is more convenient and safer (remote and read-only) in GitHub compared to the corresponding local Git workflow.
- GitHub and is a nice way to collaborate if you want other people to be able to see your repository and to make changes.

# Further resources

- Marwick, B., Boettiger, C., and Mullen, L. (2018). Packaging data analytical work reproducibly using R (and friends). *The American Statistician*, 72(1):80–88.
- Savage, A. (2020). *Every Tool's a Hammer: Life Is What You Make It*. Atria Books.
- Making R Project templates: [https://rstudio.github.io/rstudio-extensions/rstudio\\_project\\_templates.html](https://rstudio.github.io/rstudio-extensions/rstudio_project_templates.html)
- Jenny Bryan's Happy Git and Github for the user: <https://happygitwithr.com/>

# Git Basics: Going back to the past

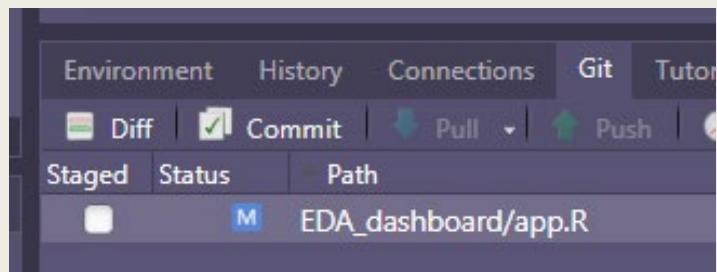


- There are a few ways to “undo” and go back in time with Git.
- There are two main scenarios that might come up:
  1. *You have not committed problematic code*
  2. *You have committed problematic code*

# Git Basics: Going back to the past



- If you have not committed problematic code, you can click the ‘Diff’ or ‘Commit’ buttons in the Git tab:



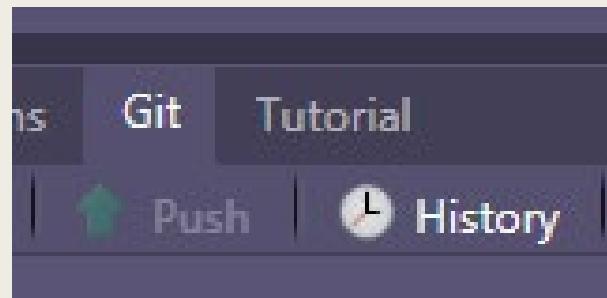
- From there, you can discard individual chunks, or choose to “Discard all” of the changes that you’ve made since your last commit.

```
@@ -27,10 +27,12 @@ ui <- fluidPage(
27 | 27
28 | 28     # Show a plot of the generated distribution
29 | 29     mainPanel(
30 | 30         plotOutput("distPlot")
31 | 31     )
32 |
33     missing_comma
32 | 34     )
33 | 35   )
34 |
35 | 37 # Define server logic required to draw a histogram
36 | 38 server <- function(input, output) {
```

# Git Basics: Going back to the past



- If you **have** committed some problematic code, you can use your commit SHAs to revert to a previous commit.
- To see your commit history, you can click the ‘History’ button in the Git tab:



Subject	Author	Date (UTC)	SHA
HEAD -> refs/heads/master updates to shiny app	Rachel Severson <rseverson6@gmail.com>	2025-06-23	53faf771
add shiny app	Rachel Severson <rseverson6@gmail.com>	2025-06-17	c2a64a0f
add initial EDA	Rachel Severson <rseverson6@gmail.com>	2025-06-17	4c08d006
save 'read in data' script	Rachel Severson <rseverson6@gmail.com>	2025-06-17	62a622c3
Initial commit	Rachel Severson <rseverson6@gmail.com>	2025-06-17	3a62814c



Commits 1-5 of 5

**Author**

Rachel Severson &lt;rseverson6@gmail.com&gt;

**Date (UTC)**

2025-06-23T19:00:00Z

**Subject**

updates to shiny app

**Parent**

HEAD -&gt; refs/heads/master

EDA\_dashboard/app.R

EDA\_dashboard/app.R

@@ -29,6 +29,8 @@ ui &lt;- fluidPage(

29 29 mainPanel(

30 30 plotOutput("distPlot")

31 31 )

32 32 missing\_comma

33 33 )

34 34 )

35 35 )

36 36 )

In the upper right corner, you can see the SHAs for all of your commits in this repo. **53faf771** is my problematic commit that is breaking my Shiny App. **c2a64a0f** is the commit that I would like to revert to.

[View file @ 53faf77](#)

# Git Basics: Going back to the past



- In the Terminal pane, you can use the `git reset --hard [SHA]` command to return your repository to its state as of your selected commit.

```
rseverso@PHDC5CD2495BC0 MINGW64 ~/Documents/my_new_repo (master)
$ git reset --hard c2a64a0f
```

- After running `git reset --hard`, your selected commit is the most recent shown in the history, and your repo is reverted to its state as of that commit:

```
rseverso@PHDC5CD2495BC0 MINGW64 ~/Documents/my_new_repo (master)
$ git reset --hard c2a64a0f
HEAD is now at c2a64a0 add shiny app

rseverso@PHDC5CD2495BC0 MINGW64 ~/Documents/my_new_repo (master)
$
```

Subject	Author	Date (UTC)	SHA
HEAD -> refs/heads/master add shiny app	Rachel Severson <rseverso6@gmail.com>	2025-06-17	c2a64a0f
add initial EDA	Rachel Severson <rseverso6@gmail.com>	2025-06-17	4c08d006
save 'read in data' script	Rachel Severson <rseverso6@gmail.com>	2025-06-17	62a622c3
Initial commit	Rachel Severson <rseverso6@gmail.com>	2025-06-17	3a62814c

# Git Basics: Going back to the past



- You can go back to your original “bad” commit by using that commit’s SHA. To see all of your git actions, you can run `git reflog`:

```
rseverso@PHDC5CD2495BC0 MINGw64 ~/Documents/my_new_repo (master)
$ git reflog
c2a64a0 (HEAD -> master) HEAD@{0}: reset: moving to c2a64a0f
53faf77 HEAD@{1}: commit: updates to shiny app
c2a64a0 (HEAD -> master) HEAD@{2}: reset: moving to c2a64a0f
71bba0c HEAD@{3}: commit: shiny edits
c2a64a0 (HEAD -> master) HEAD@{4}: commit: add shiny app
4c08d00 HEAD@{5}: commit: add initial EDA
62a622c HEAD@{6}: commit (amend): save 'read in data' script
72b7a8a HEAD@{7}: commit: save 'read in data' script
3a62814 HEAD@{8}: commit (initial): Initial commit
```

- This log shows the **53faf77** “updates to shiny app” commit. You can run `git reset --hard 53faf77` to go back to that commit state.