Reproducible Research Practices for Economists

Mindy L. Mallory

October 24, 2017

Questions for the Audience

How many of your research folders look like this?

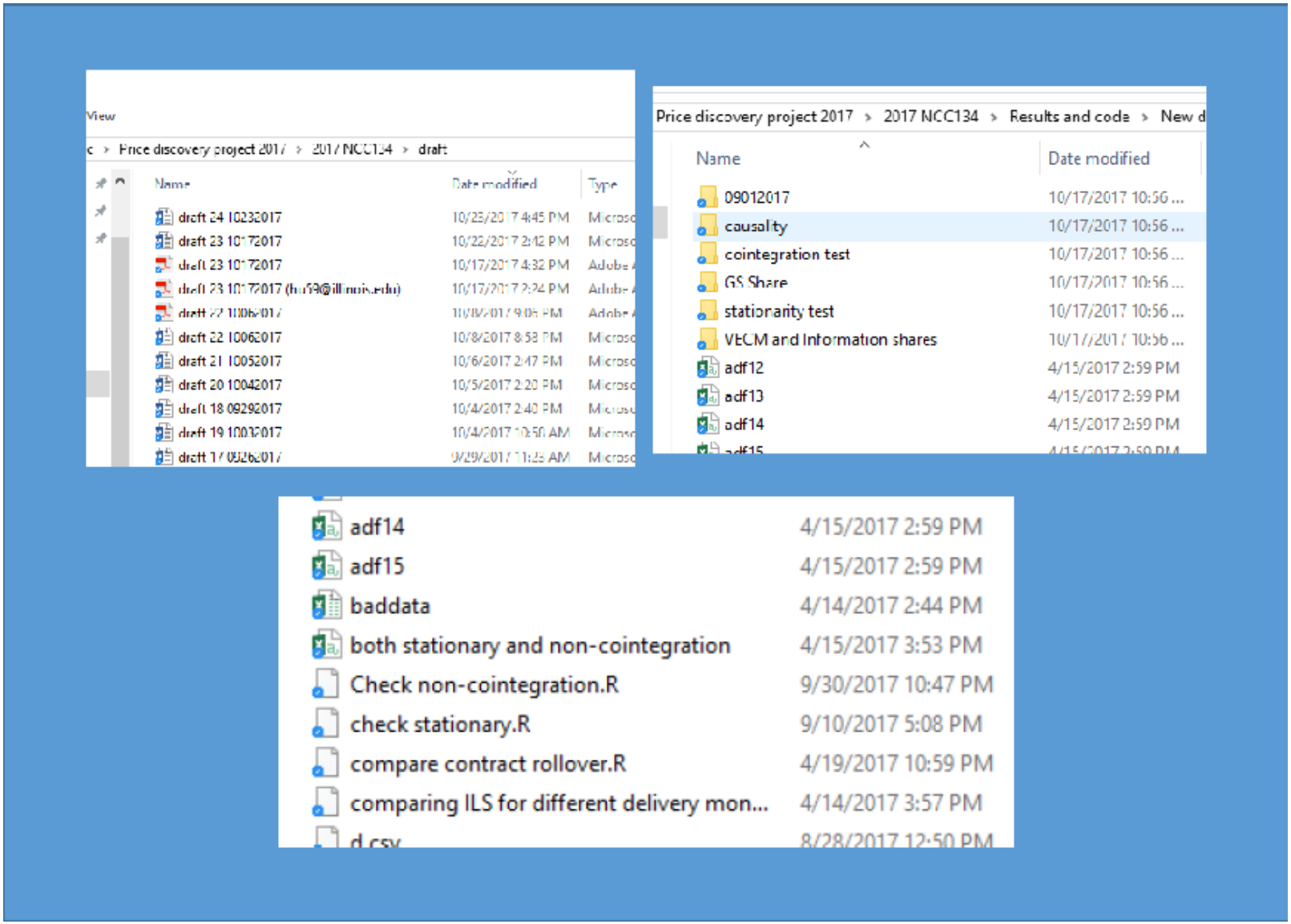
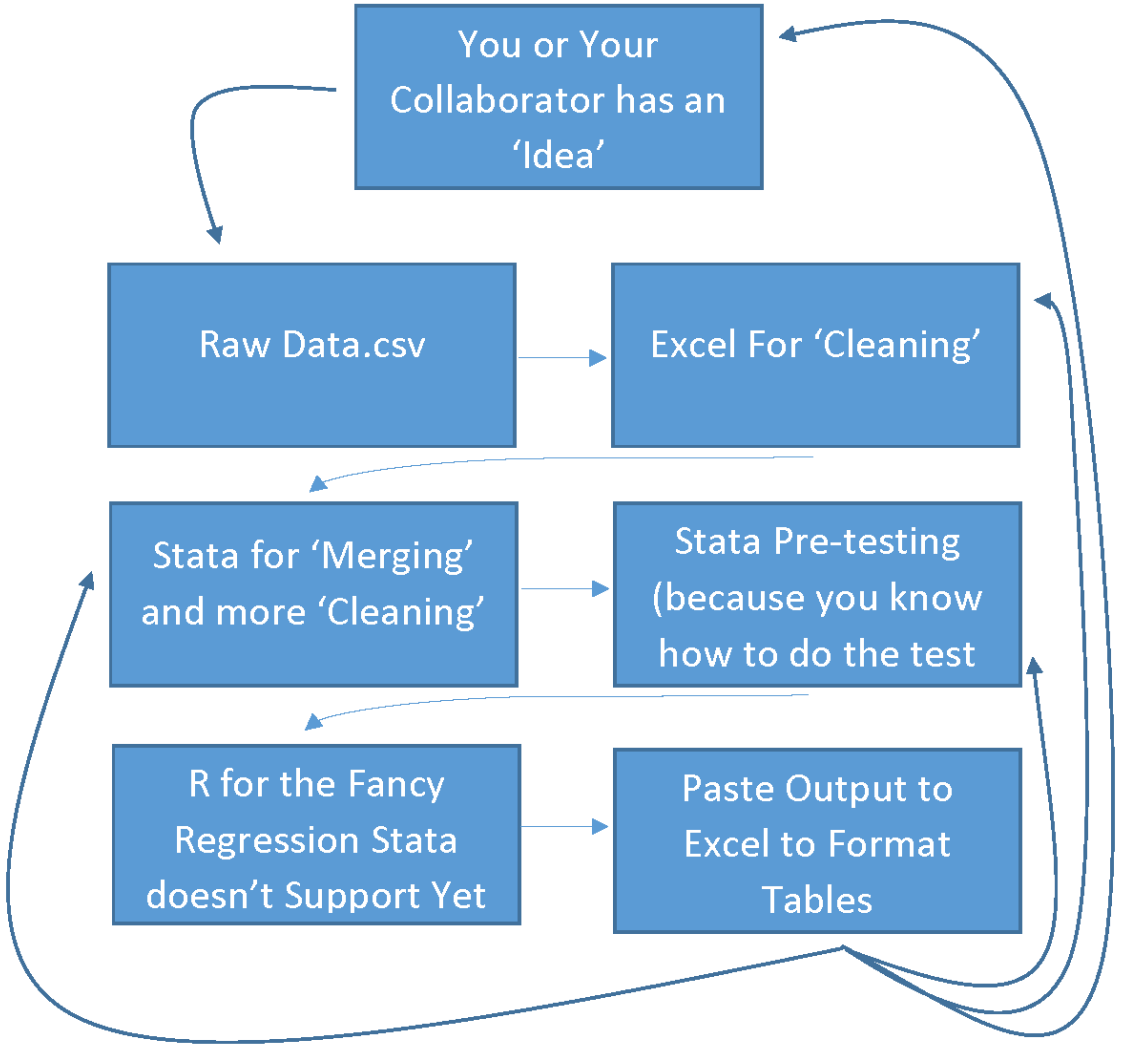


Figure 1: Picking on Zhepeng

How many of you have a research work flow that looks like this?



Questions for the Audience

How many of you would rather die than have to reproduce a table from a paper you published 2 years ago?

Questions for the Audience

Do you wake up in a cold sweat dreaming that Reviewer number 2 asked you to update your data-set (perform robustness test, etc) and you couldn't even reproduce your original results?

Questions for the Audience

Students, have you ever purposely obfuscated your code figuring if your professor can't follow it they can't criticize it?

Questions for the Audience

Have you ever lost data between submission and being asked to revise and resubmit and then you had to go and REPURCHASE!!! said data?

Questions for the Audience

Have you ever lost an entire paper due to the Word file becoming corrupted then you thought you salvaged the paper through document recovery but then it got rejected because you missed some weird characters from the file corruption and reviewer number 2 recommended rejecting your paper because the authors were 'careless' to allow the weird characters to remain the document?

I can say yes to all of these questions!

But I got tired of being nervous all the time!

There is a better way!

Reproducible research with R, RStudio, RMarkdown, Knitr, and Github

- R - is awesome statistical computing software (open source and free!)
- Rstudio - is an awesome integrated development environment (program making it convenient to work with R); also open source and free

Reproducible research with R, RStudio, RMarkdown, Knitr, and Github

- RMarkdown is a kind of markup language supported by RStudio that uses **Knitr** to weave statistical analysis and results into beautifully formatted documents.
  - Written in plaintext, it understands latex code and documents can be rendered into many different output formats
    - ★ PDF
    - ★ Beamer
    - ★ HTML
    - ★ Word*

Reproducible research with R, RStudio, RMarkdown, Knitr, and Github

- Github - is a cloud-based repository that is great at versioning (it was designed by and for software developers)

The Basics - Before Getting Started

- Install R, RStudio, Git, and Gitkraken
- Install the following packages in R by executing the following commands in the RStudio console:

```
install.packages("xts")
install.packages("tseries")
install.packages("tsDyn")
install.packages("broom")
install.packages("vars")
```

The Basics - Set up a clean, reproducible project repository

- Create a new repository on Github.com
- Choose a meaningful repository name
- Be sure to initialize with a Readme file by clicking the checkbox (somehow it helps RStudio and GitHub set an initial connection)
- After creating the repository

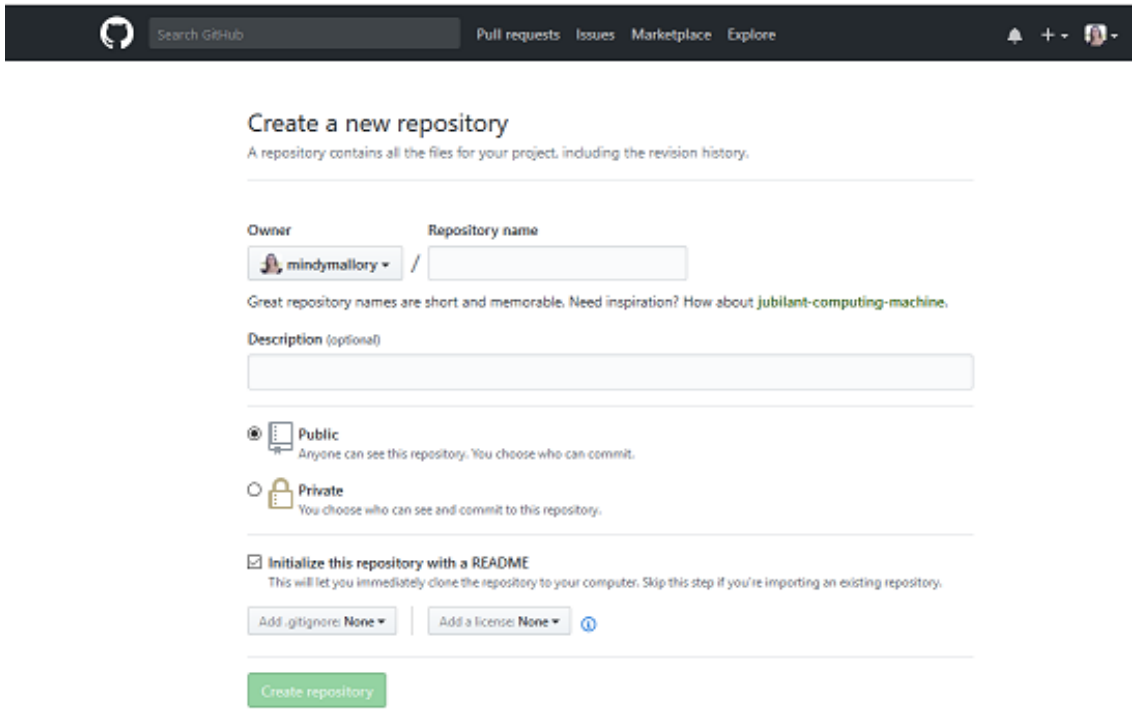


Figure 3:

The Basics - Set up a clean, reproducible project repository

- After creating the repository, click 'Clone or Download' and copy the link to the repository.

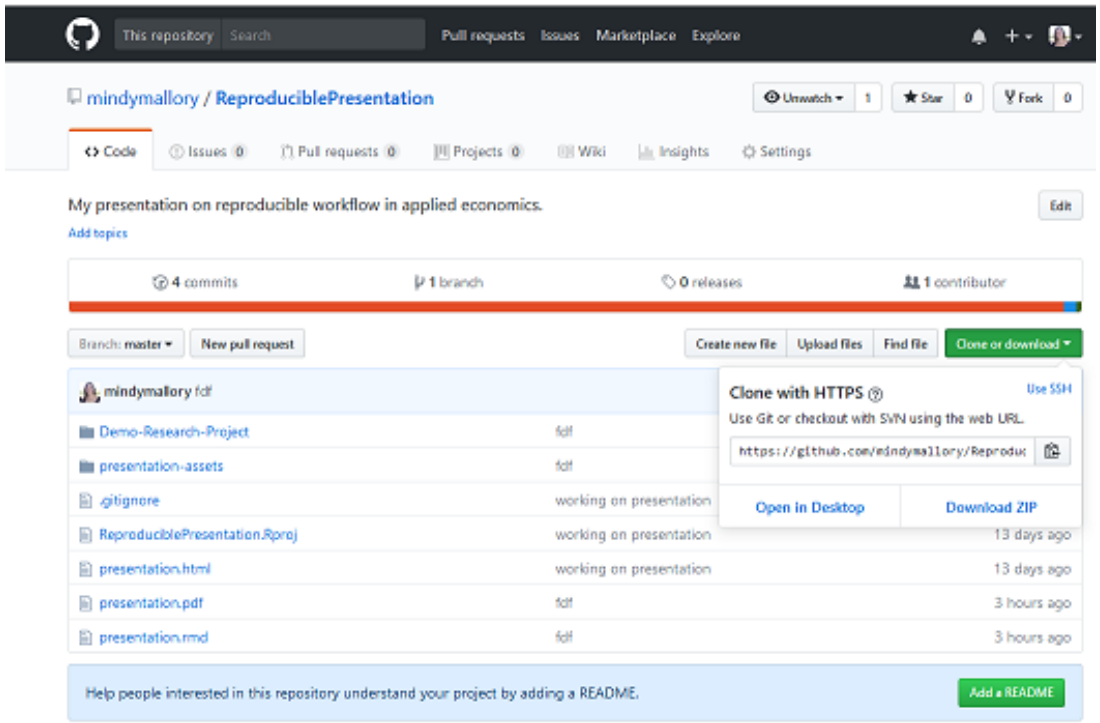
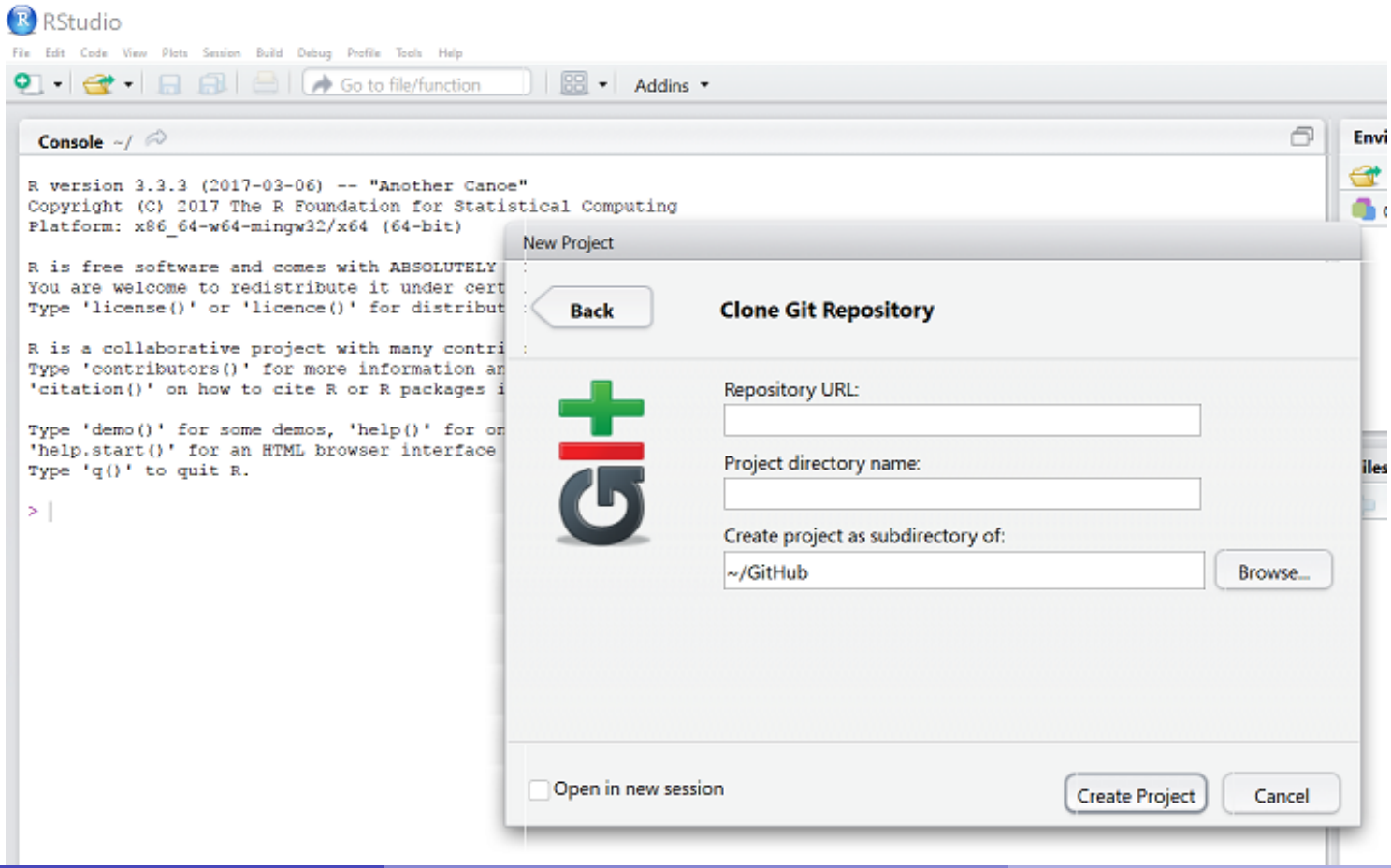


Figure 4:

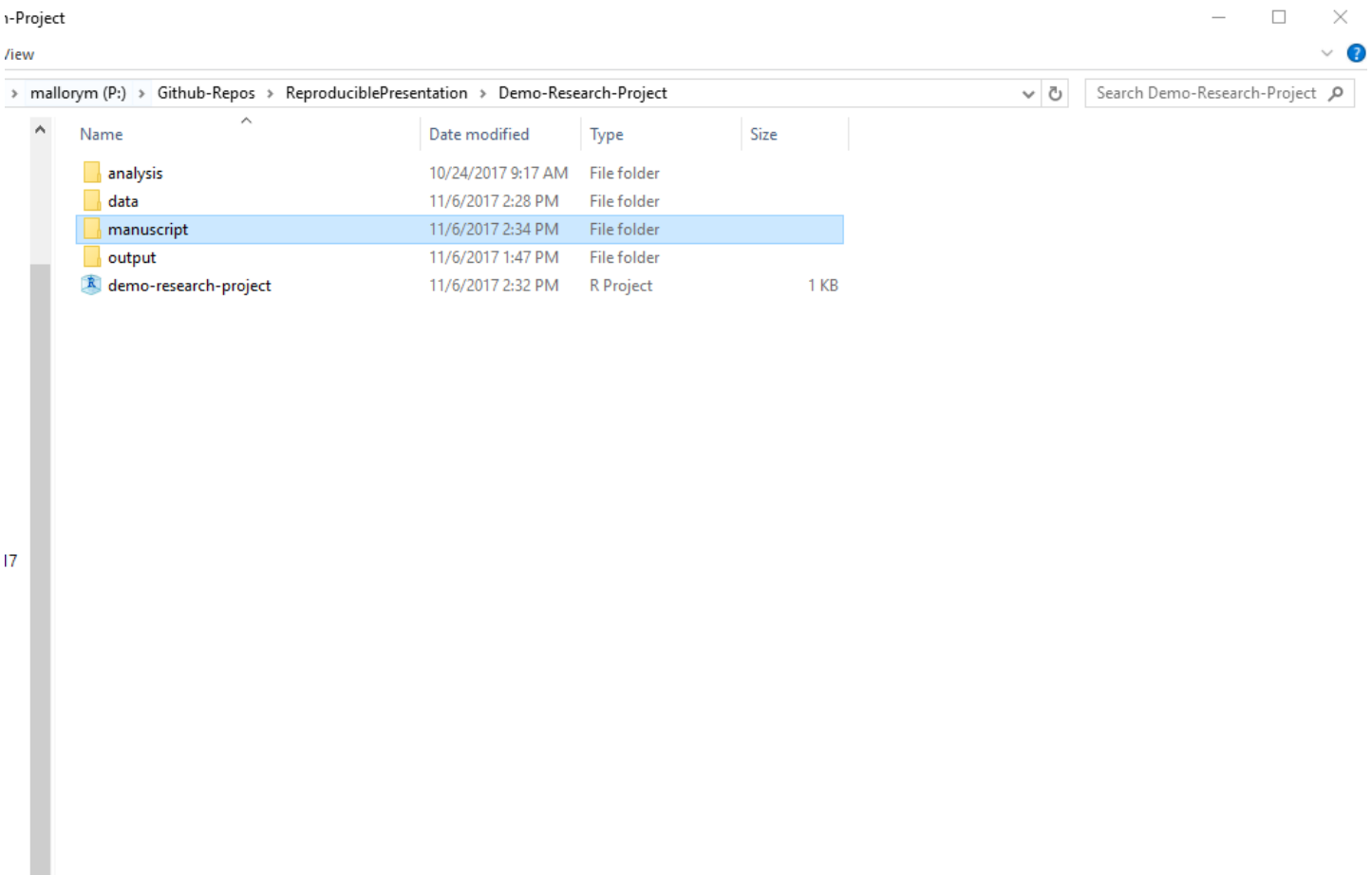
The Basics - Set up a clean, reproducible project repository

- Open up RStudio and navigate through 'File' -> 'New Project'
- Choose 'Version Control' -> 'Git'
- Then paste the link you copied from github.com into 'Repository URL' and click 'Create Project'

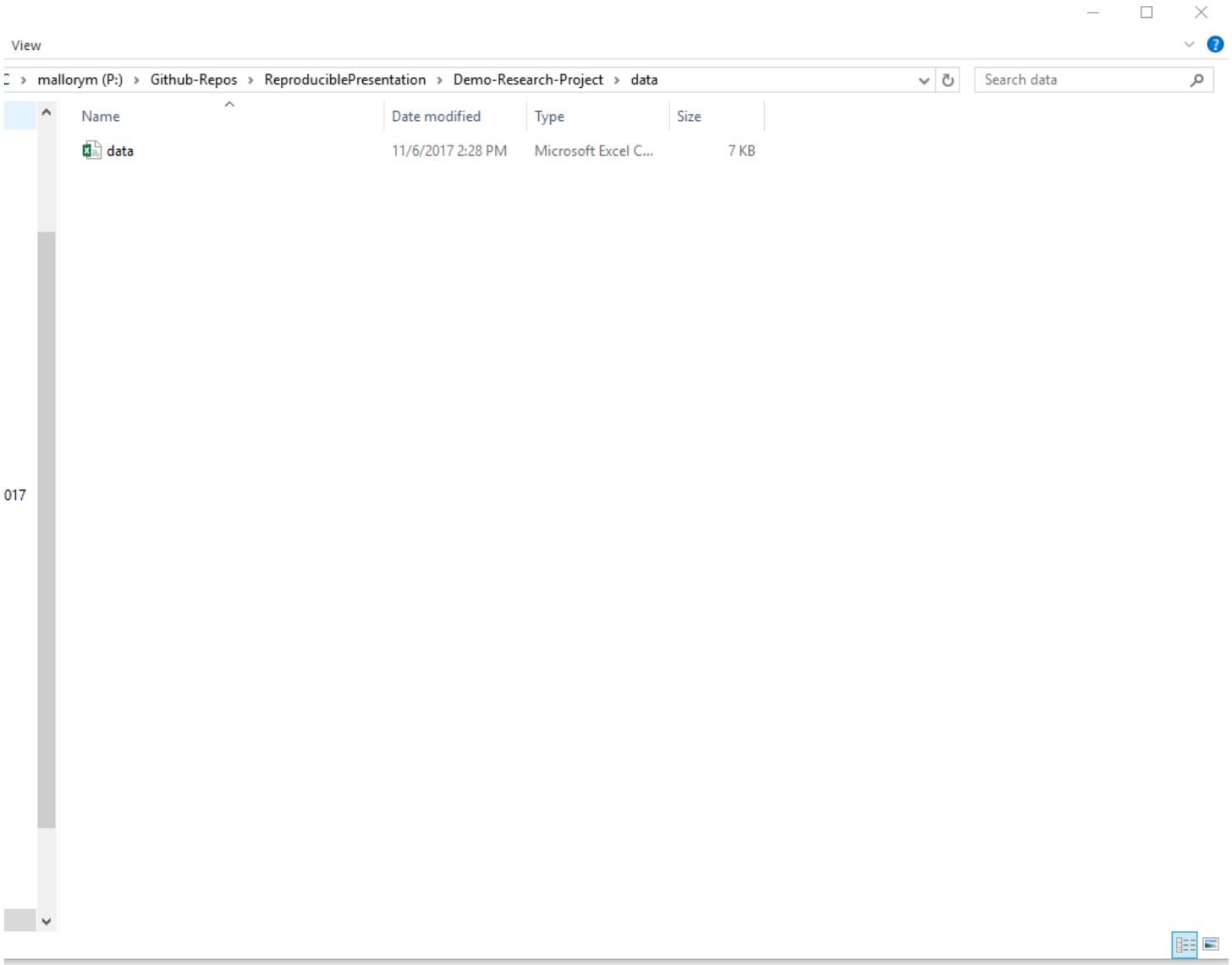


The Basics - Set up a clean, reproducible project repository

- RStudio Rule #1 - use projects!
- Never change the working directory
- Once you have created a project, the working directory is automatically set to this file path

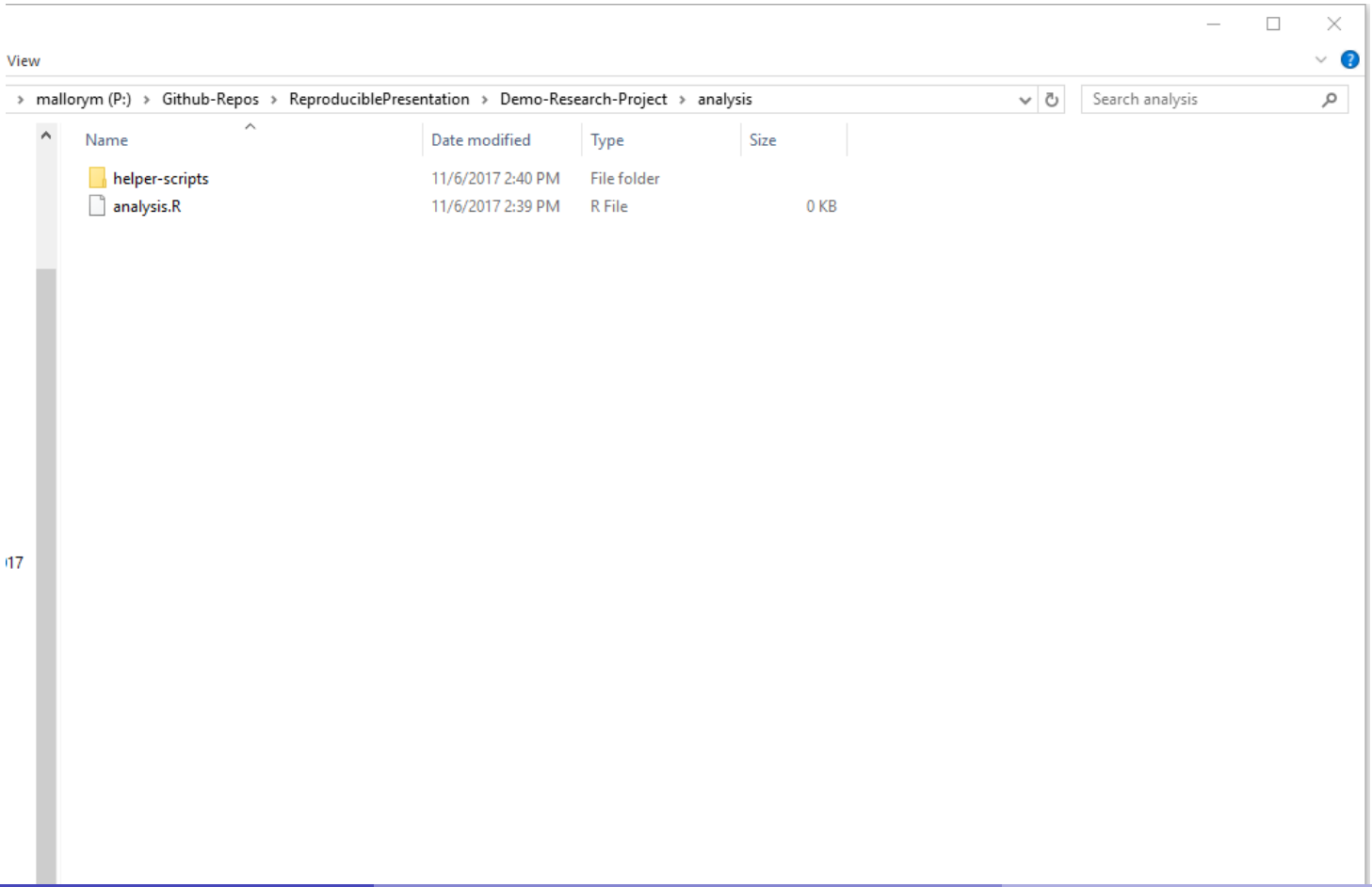


The Basics - Put your raw data in the 'data' folder and never touch again



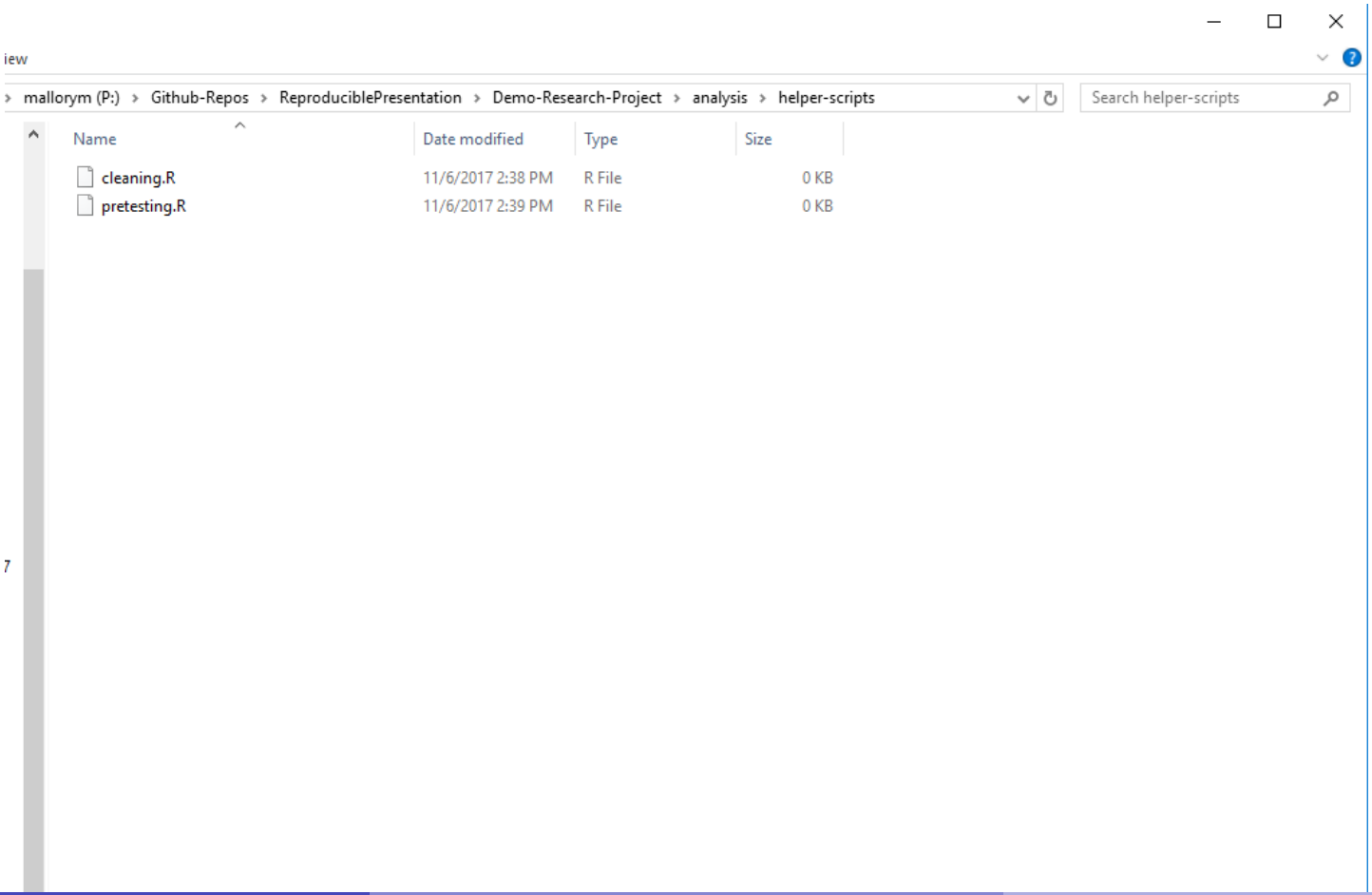
The Basics - Organize Scripts

- Document what each script does
- If your project requires an elaborate 'readme.txt' with instructions about which scripts to run and in what order, you probably need to automate this.



The Basics - Organize Scripts

- Document what each script does
- If your project requires an elaborate 'readme.txt' with instructions about which scripts to run and in what order, you probably need to automate this.



Data Analysis - Cleaning

Your analysis may involve 'cleaning' raw data.

- May be aggregating many individual files
- Dealing with missing data
- Merging two or many large datasets

This type of activity should be done by the `cleaning.R` script that takes raw data files and makes them useful.

If at all possibly, do not save intermediate cleaned data. Run scripts that build from raw data everytime so you know it is reproducible.

Look at cleaning.R

Data Analysis - Pretesting

Similarly, you may need to check for stationarity or do other common diagnostic tests that inform model choice.

This file will take cleaned data from `cleaning.R` and perform diagnostics. The tests will create R objects that can be called an inserted into manuscript results.

Look at pretesting.R

Data Analysis - Fit Main Model

Then, your main analysis can be performed in analysis.R. This script will fit model and the output will be R objects that can be inserted to display results directly into tables and text of your manuscript.

Look at analysis.R

Write Paper in RMarkdown

RMarkdown is an easy to use way to create reproducible reports that can be rendered to many formats.

- Accepts Latex commands for math equations and other formatting
- Supports reference management with bibtex
- Excecute R scripts right in the document and incorporate the results into your document

Look at manuscript.Rmd