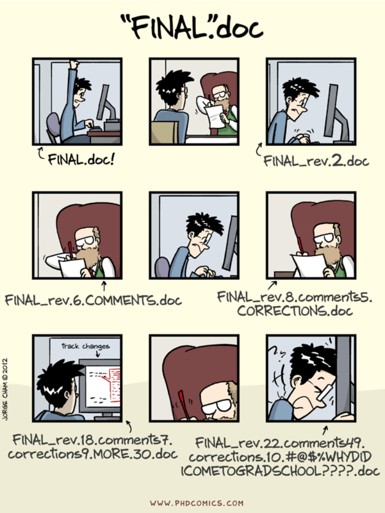
Developing a Workflow to Maximize Reproducibility and Research Impact: Managing Data, Computer Code, and Projects for Success

John R. Fieberg & Althea A. ArchMiller

4/11/2017

Why worry about reproducibility?

Working towards future reproducibility makes my code easier for my collaborators (and me) to read, run, and debug today,
and that's why I think reproducibility is a **win-win for all researchers**.”
-Althea



Why worry about reproducibility?

“[Reproducibility] provides security, saves time, and forces me to be more thoughtful about my workflow.” - Ethan Young

- make your life easier!
- collaborations
- broader research impact
- increased citations
- transparency
- grant and journal requirements

Is my research reproducible?

- Are your research documents stored in these formats?
  - .csv
  - .txt
  - .pdf
  - .html
  - .R
    - YES!
  - .doc/.docx
  - .sas
  - .xls/.xlsx
  - any other proprietary file format
    - NO!

Is my research reproducible?

- Is your code linear?
  - Clear environment often and at beginning of script
  - Don't save .Rdata or history
  - Each program should focus on one main task or analysis
  - Don't rely on manual commenting/uncommenting

```r
# What variables are significant?
lm.out <- lm(weight ~ height, data = trial.data)
remove(lm.out) # clear previous lm.out for each
               #   new lm() definition above

# Is the relationship significant?
#     (If not, clear and try a new regressor)
summary(lm.out)
```

Is my research reproducible?

- Are your files easily shared with others?
  - Organized directory structure
  - Files relatively linked
  - Well-documented & commented
  - Consistency in coding practices

“The point of having style guidelines is to have a common vocabulary of coding so people can concentrate on *what* you are saying, rather than on *how* you are saying it.” - Google’s R Style Guide

Workshop Outline

The goal for this workshop is to help you develop the tools to develop a workflow to maximize reproducibility, collaborations, and research impact.

1. RStudio Projects for organizing data, code, and output
2. R-Markdown and R-Oxygen for documenting your code
3. GitHub for version-control, collaborating and archiving

1. RStudio Projects

Think about a typical data analysis project, maybe a dissertation chapter or an experiment that you've managed from data collection through publication. What are typical **folders** that you've used?

- Raw data
- Processed data
- Analysis scripts
- Paper/Manuscript-related documents
- Sharing documents ("transmittals")
- Metadata
- Maps or other deliverables

RStudio Projects provide an opportunity for you to organize and manage all of these types of folders in **one place** in a way that **relatively links** everything together and **eases sharing**.

1. RStudio Projects

Tips

- Treat data as read-only
    - Don't use Excel, etc, to manipulate raw data
    - Use a single R program for all manipulation
    - Save "cleaned" or "procesed" data in easily loadable format

1. RStudio Projects

Other links

https://swcarpentry.github.io/r-novice-gapminder/02-project-intro/

Tips

- Don't use github with large files :-(
- Create new projects in GitHub first, then sync them with RStudio

Why R-Markdown for manuscripts?

"I can do reproducible work in R (making me happy) and format the output report in Word (making my collaborators happy)" - Richard Layton `http://rmarkdown.rstudio.com/articles_docx.html`