Econ 211

Prof. Jeffrey Naecker

Wesleyan University

Tools for Transparent Research

Overview

- Today we will briefly go over a number of software tools used in transparent, reproducible research
- You will use *some* of these tools to do your final problem sets in the class and in your final project
- Other tools I am showing you just because they may be useful to you in the future
  - Can also use them for final projects if you want

Markdown

- Markdown is a lightweight markup language
- From plain text, can create Word documents, PDFs, slides, many other formats
- These slides are written in Markdown
- Demo: see Markdown Basics document
- How to view Markdown output
  - In RStudio: Hit the `Knit` button
  - Get a viewing program like Marked
  - From the command line: use `pandoc`

LaTeX

- Another, much more powerful markup language
- How lecture notes, exams, and problem sets in this class were created
- How most technical papers and books are created
- Massive amount of features for typesetting tables, formulas, figures

Markdown vs LaTeX

- Markdown is designed to be *lightweight* and *portable*
  - Limited functionality, but code very readable
  - Can output to many different file types
- LaTeX is designed to be *powerful* and *flexible*
  - Can tweak output extremely precisely, but code not as readable
  - Can only make PDFs

How to Compile LaTeX Output

- Must install LaTeX on your computer
- Create and open a `.tex` file in the TexShop editor that comes with LaTeX
  - Can get a basic template from 'Macros > Headings > 12pt > article"
- Compile to PDF by hitting "Typeset" button

R

- R is a **free** open-source programming language
- Runs on any operating system
- Optimized for statistical computing and graphics

Getting Started in R

- Install R
  - This will install the R language itself
  - Will also install program for writing and running R code (this is called an IDE for "Integrated Development Environment")
  - The standard IDE that gets installed is also called R, confusingly
- Recommend installing RStudio and writing/running all your code from there
  - The RStudio folks have also put together a nice getting started page

Working in R

- Open Rstudio
- Commands can be entered in the console (hit Enter to run)
- Commands can also be entered into a `.R` file
  - Hit "Run" button at upper right to run everything in the file
- Output will be printed to console
  - Unless saved using the assignment operator: `<-`
- Additional resources
  - Open Intro to R
  - Data Camp

R Packages

- In addition to base R can add *packages* to your code
- Packages give access to useful commands or data that other people have created
- The following packages are very useful:
  - ggplot2
  - dplyr
  - stringr
  - texreg
- Next lecture we will use a package for running power calculations as part of your final papers

Problem

- A common workflow in social science:
  - Write research paper in Markdown or LaTeX or Word
  - Copy statistical analysis results from R or Stata into paper document
- Issues with this
  - May copy wrong output
  - May change R code but not copy in new results
  - Don't know which version of code generated results in the paper

The Solution: R Markown

- We can solve this by running R (for the stats/figures) and Markdown (for the words) *in the same document*
- This is called R Markdown, or Rmd for short
- For example:

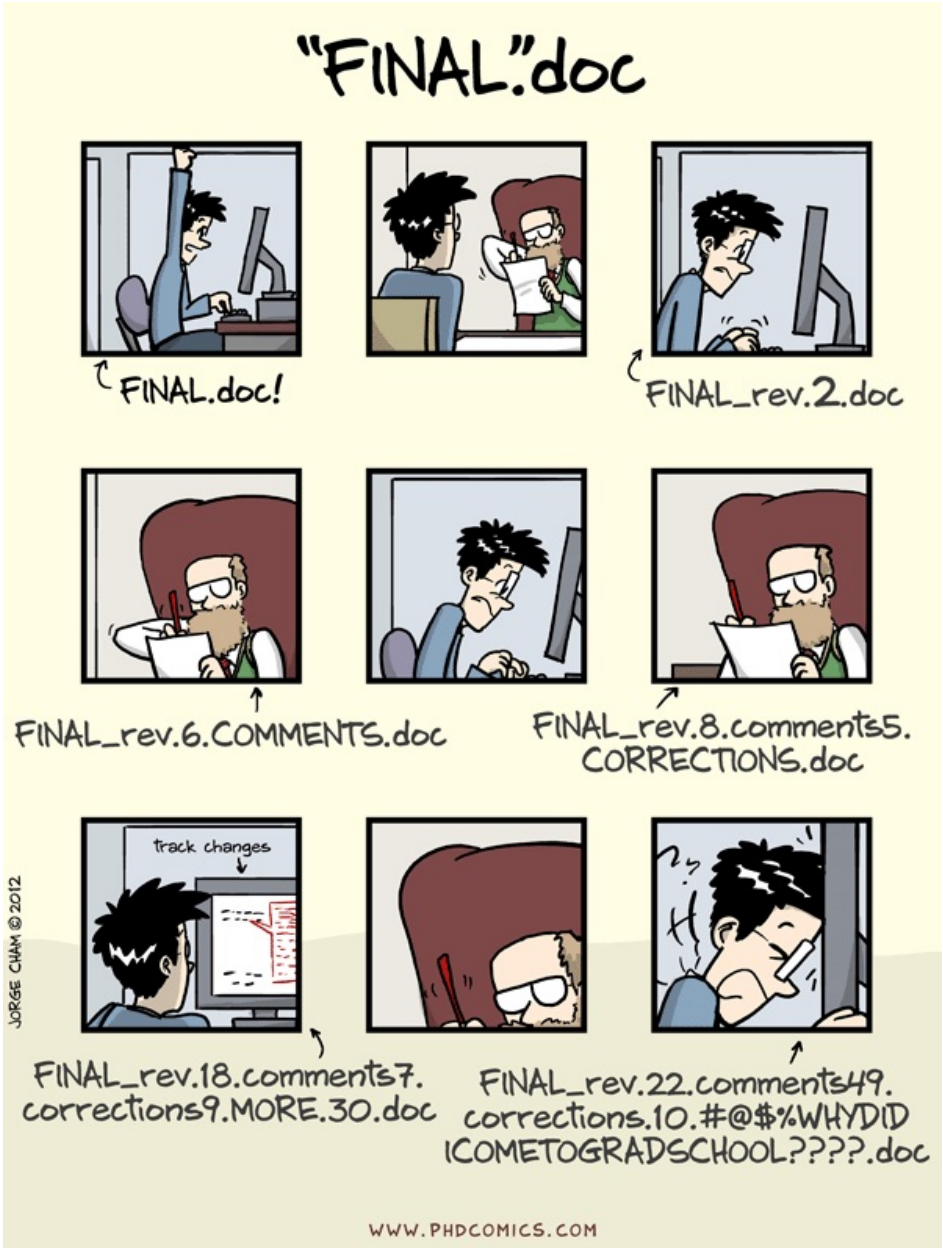
```
[] mean(c(1,2,3,4))
```

```
## [1] 2.5
```

Getting Started with R Markdown

- Install R and RStudio
- File > New File > RMarkdown
- Click "Insert" button to add code chunk
- Hit "Knit" button
- For more details, see R Markdown Cheat Sheet

Version Control



Getting Started with Git and Github

- Git is a distributed version control system (VCS)
- Used primarily for tracking and merging changes to plain text files

1. Install Git. If you have a Mac, this is done already. Just type "git version" in your Terminal program to confirm. If you have a Windows machine, download it here.
2. Make an account on GitHub. This is one of many places that you can use to sync your local git repository with one in the cloud so that your collaborators can easily see what you've done.
3. If you would like a third-party program to help visualize what is going on with Git, try SourceTree or GitKraken.

Setting up Git Project

- Create a folder for the project you are working on and put all the code in there
- Navigate to that folder in your command line
  - Terminal if you're on a Mac, Cygwin on Windows
  - Navigate around command line with `cd`, eg `cd ~/Desktop`
  - See where you are with `pwd`
  - List files with `ls`
- Run `git init` in the terminal to tell git to start tracking this folder
- Run `git add .` to stage all the files to git
- Run `git commit -m "First commit"` to commit the files

Adding Project on GitHub

- On GitHub, click the plus icon at the top right to add a new repository
- Copy the URL of the new repository you just created
- Back in your command line, run `git remote add origin url-you-just-copied-pasted-here`
- Run `git push -u origin master`

Downloading Existing Project from Github

- What should you do if your teammate has already put the directory up on GitHub?
- Go to project page on Github
- Click "Clone or Download" button and copy link
- On your command line, navigate to where you want project to go
- Enter `git clone url-you-just-copied`

Daily Git Workflow

- Navigate to the project folder in your command line
- If you are working on a project in a team, run `git pull origin master` to pull down any changes that others have made since the last time you pushed
- Make changes to the code and save your work
- Run `git add .` to stage your work
- Run git `commit -m "Useful reminder here about what I just did to commit your changes"`
  - This only updates the local repository, though
- It is fine to cycle through many add/commits, one for each major change you make to the code
- At the end of the day/work session, run `git push origin master` to update the web repository with your new commit(s)

Additional Git Resources

- More details on adding existing projects
- ProGit online book, chapter 1-3 and 5-7
- This git branching game is very useful
- A very nice one-page cheat sheet
- Tutorials from Atlassian
- Try Git walkthrough from Github