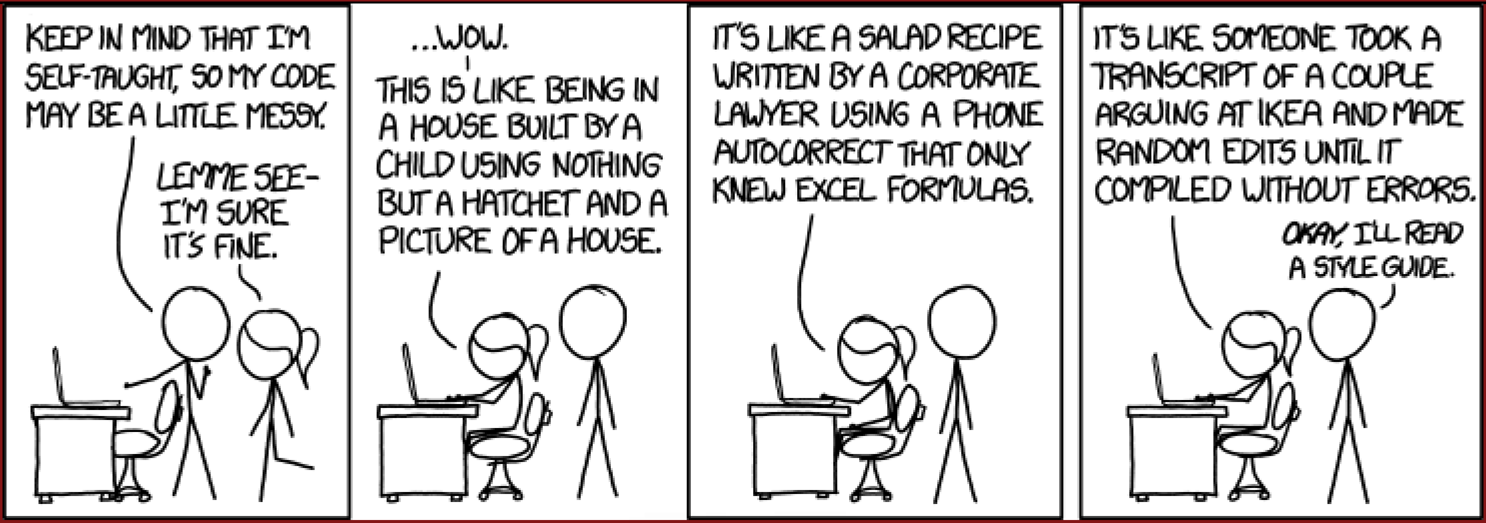
Big Data and Economics

The Empirical Workflow and Clean Code

Kyle Coombs (adapted from Tyler Ransom + Scott Cunningham)
Bates College | EC/DCS 368

Table of contents

Prologue


KEEP IN MIND THAT I'M SELF-TAUGHT, SO MY CODE MAY BE A LITTLE MESSY.
LEMME SEE— I'M SURE IT'S FINE.
...WOW.
THIS IS LIKE BEING IN A HOUSE BUILT BY A CHILD USING NOTHING BUT A HATCHET AND A PICTURE OF A HOUSE.
IT'S LIKE A SALAD RECIPE WRITTEN BY A CORPORATE LAWYER USING A PHONE AUTOCORRECT THAT ONLY KNEW EXCEL FORMULAS.
IT'S LIKE SOMEONE TOOK A TRANSCRIPT OF A COUPLE ARGUING AT IKEA AND MADE RANDOM EDITS UNTIL IT COMPILED WITHOUT ERRORS.
OKAY, I'LL READ A STYLE GUIDE.

Source: xkcd

Forgot to mention

- **Office Hours:**
    - My office hours are 9am-10am on Tuesdays and 3pm-4pm on Wednesdays
    - My office is 276 Pettengill
    - I'm also available by appointment on Zoom
- **Problem Set 0:** due on Sunday, September 17th at 11:59pm
- **Presentations:** Everyone does two, sign-up in the Presentations github repository
- **Problem Set 0:** due on Sunday, September 17th at 11:59pm
- **Problem Set 1:** due on Sunday, September 24th at 11:59pm
- **Project Proposal:** due on Sunday, September 24th at 11:59pm

Play along at home

- Sync your forks of the class repository
- Pull the latest changes from the class repository to your computer
- Open lectures/02-empirical-workflow.Rmd and you can follow along with the slides
    - Specifically, you can run the code live while I walk through it on the slides

Attribution

- Today's material comes from these sources:

1. Clean Code by Tyler Ransom

2. *Code and Data for the Social Sciences: A Practitioner's Guide*, by Gentzkow and Shapiro

3. Causal Inference and Research Design by Scott Cunningham

4. Jenny Bryan's UseR 2018 keynote address

Also a small contribution from here and other sundry internet pages

Jargon

- There is a jargon in this class that won't make sense at first, I'll try to flag it as it comes
    - If I don't flag a term, look it up on ChatGPT
    - If it still doesn't make sense, ask me -- could be I'm using it idiosyncratically
- Here's a few terms:
    - **Local machine:** Your personal (or any) computer that isn't a server accessed via the internet
    - **Version Control:** Keep track of different iterations of a project/code
    - **Repository:** The location on GitHub of all project files and (commented) file revision history
    - **GUI:** A Graphical User Interface -- what you're used to pointing and clicking to navigate a computer and execute programs
    - **Command line:** Removes the "graphical" from GUI, instead you type all commands to navigate a computer and execute programs
        - R operates via the Command line, RStudio is a GUI
        - On Mac, this is called Terminal
        - Windows has Powershell, but it Powershell uses quite user-unfriendly commands
        - If you installed Git for Windows, you got *Git Bash*, which uses Bash (Linux) commands
        - You can also install Windows Subsystem for Linux to run Linux on a Windows machine

Clean Code

Reducing empirical chaos

Sad story

- Once upon a time there was a boy who was writing a job market paper on unemployment insurance during the pandemic
- This boy presented the findings a half dozen times, spoke to the media some, and generally thought he had cool results
- Several people suggested he look at a handful of other outcome series and try changing his analysis unit frequency from monthly to weekly
- He also knew that he needed to restrict his sample to reduce noise

The horror!

- But then after making these changes and re-running his code that took two days, his new sample dropped by 50 percent!
- He was, understandably, terrified.
- The young boy spent a week looking for the fix weeding through six different versions of the .do, .R, .dta, .csv, .sh, .py files with suffixes like *_v1* and *_test* and *_test2* and *_final_I_swear* and *_okay_i_lied*
- Finally he discovered the phrase:

```
df %>% filter(insample_new==0)
```

instead of

```
df %>% filter(insample_new==1)
```

- The boy was very frustrated and decided to work on these slides while re-running his code.

What is Clean Code?

- **Clean Code:** Code that is easy to understand, easy to modify, and hence easy to debug
- Clean code saves you and your collaborators time

Why clean code matters: Scientific

- Good science is based on careful observations
- Science progresses through iteratively testing hypotheses and making predictions
- Scientific progress is impeded if
  - mistaken previous results are erroneously given authority
  - previous hypothesis tests are not reproducible
  - previous methods and results are not transparent
- Thus, for science that involves computer code, clean code is a must
- Minimizes (incompletely) the role of the influence of hidden researcher decisions" (Huntington-Klein et al. 2021)

- You will always make a mistake while coding
- What makes good programmers great is their ability to quickly identify and correct mistakes

Next lecture: Hidden Researcher Decisions