Causal Inference and Research Design

Scott Cunningham (Baylor)

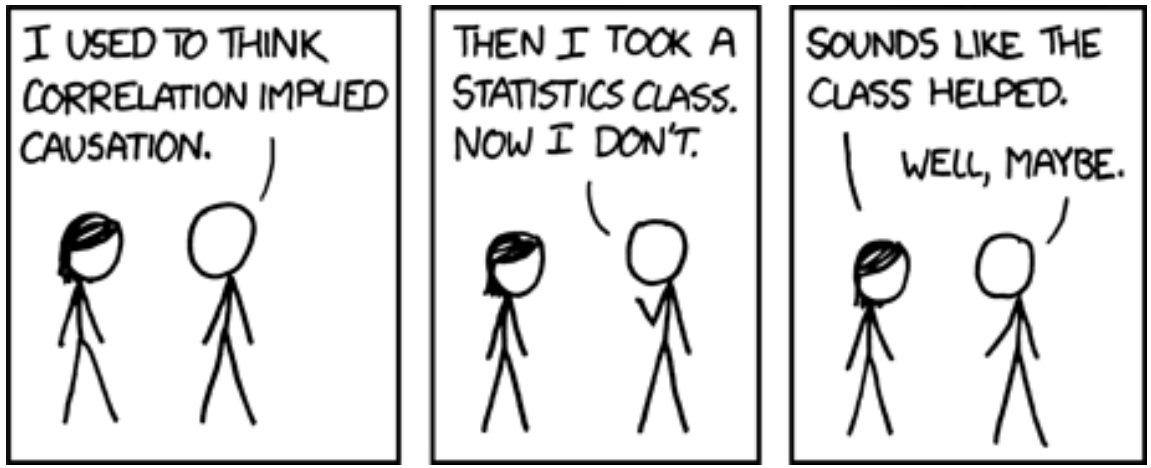


Figure: xkcd

Where to find this material

- A lot of this material is drawn from my book, Causal Inference: The Mixtape, which you can download from my website `www.scunning.com`

Structure and Assessment

The fundamental theme linking all lectures will be the estimation of *causal effects*

- Part 1 covers "the core" of applied econometrics, including hidden curriculum
- Part 2 covers causality foundations like potential outcomes and DAGs
- Part 3 covers contemporary research designs

Stata and R

A secondary goal of the workshop is to provide you with programming examples in Stata and R for implementing some but not all of the procedures we'll cover

- R and Stata code are provided many procedures (with more to come)
- I wrote the Stata and had the written by my RAs reviewed by a third exceptional student
- Programs and data can be downloaded from my GitHub repository (https://github.com/scunning1975/mixtape)

Textbooks

Helpful Textbooks imho

1. Cunningham (2018) (Mixtape) (under contract with Yale, but I can't share the new version yet – this deck is the closest thing to it)
2. Angrist and Pischke (2009) Mostly Harmless Econometrics (MHE)
3. Morgan and Winship (2014) Counterfactuals and Causal Inference (MW)

Readings

Readings:

- We will also discuss a number of papers in each lecture, each of which you will need to learn inside and out.
- Lecture slides and reading lists are available
- Key literature is contained in the shared dropbox folder which I'll distribute beforehand

About me

- Professor of economics at Baylor (Waco Texas),
- Graduated in 2007 from University of Georgia with a field in econometrics, IO, public, and labor field courses
- I knew I was going to be an empiricist, so I made econometrics my main field – passed field exam on second attempt
- Since graduating I've focused on topics in crime and risky sex such as sex work, drug policy, abortion, mental healthcare.
- I knew I couldn't achieve my goals without learning causal inference which I could tell I had only a vague understanding of
- This is because causal inference isn't taught historically in traditional econometrics

Sad story (to me!)

- Once upon a time there was a boy who wrote a job market paper using the NLSY97.
- This boy presented the findings a half dozen times, spoke to the media a few times, got 17 interviews at the ASSA, 7 flyouts, and an offer from Baylor
- He submitted the job market paper to the *Journal of Human Resources*, a top field journal in labor, and received a "revise and resubmit" request from the editor (woo hoo!)

The horror!

- But then digging into his one directory, he found countless versions of his do file and hundreds of files with random names
- And once he finally was able to get the code running again, he found a critical coding error that when corrected ("destroyed") his results
- The young boy was devastated and never resubmitted which he does not recommend (but he was sad!)

All competent empirical work is a mousetrap

"Happy families are all alike; every unhappy family is unhappy in its own way." - Leo Tolstoy, Anna Karenina

"Good empirical work is all alike; every bad empirical work is bad in its own way." - Scott Cunningham, This slide

Cunningham Empirical Workflow Conjecture

- The cause of most of your errors is **not** due to insufficient knowledge of syntax in your chosen programming language
- The cause of most of your errors is due to a poorly designed empirical workflow

Workflow

Wikipedia definition:

> *"A workflow consists of an orchestrated and repeatable pattern of activity, enabled by the systematic organization of resources into processes that transform materials, provide services, or process information."*

Dictionary definition:

> *"the sequence of industrial, administrative, or other processes through which a piece of work passes from initiation to completion."*

Empirical workflow

- Workflow is a fixed set of routines you bind yourself to which when followed identifies the most common errors
  - Think of it as your morning routine: alarm goes off, go to wash up, make your coffee, check Twitter, repeat *ad infinitum*
- Finding the outlier errors is a different task; empirical workflows catch typical and common errors created by the modal data generating processes

Why do we use checklists?

- Before going on a trip, you use a checklist to make sure you have everything you need
  - Charger (check), underwear (check), toothbrush (check), passport (oops), ...
- The empirical checklist is solely referring to the intermediate step between "getting the data" and "analyzing the data"
- It largely focuses on ensuring data quality for the most common, easiest to identify, situations you'll find yourself in

Simple checks

- Your checklist should be a few simple, yet non-negotiable, programming commands and exercises to check for coding errors
- Let's discuss a few

Time

- People often think empirical research is about "getting the data" and "analyzing the data"
- They have an "off to the races" mindset
- Just like running a marathon involves far far more time training than you ever spend running the marathon, doing empirical research involves far far more time doing tedious, repetitive tasks
- Since you do the tedious tasks repeatedly, they have the *most* potential for error which can be catastrophic
- How can we minimize these errors through a checklist?

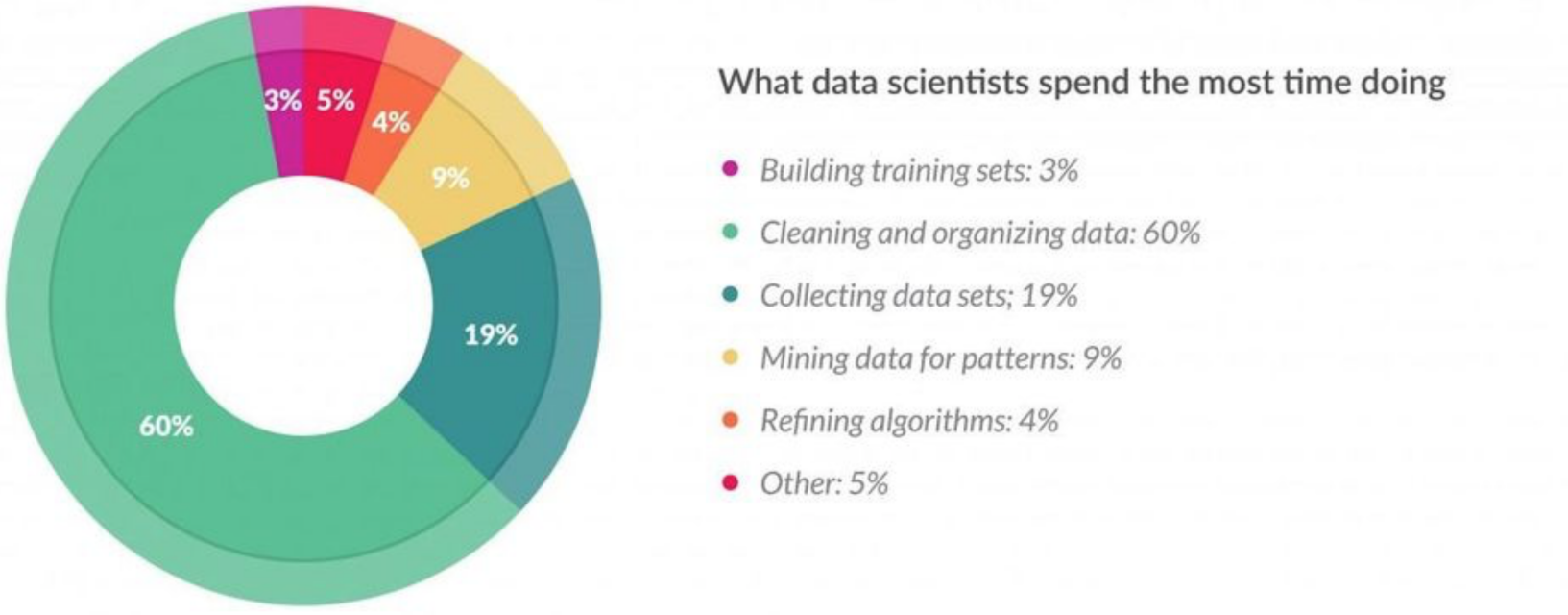


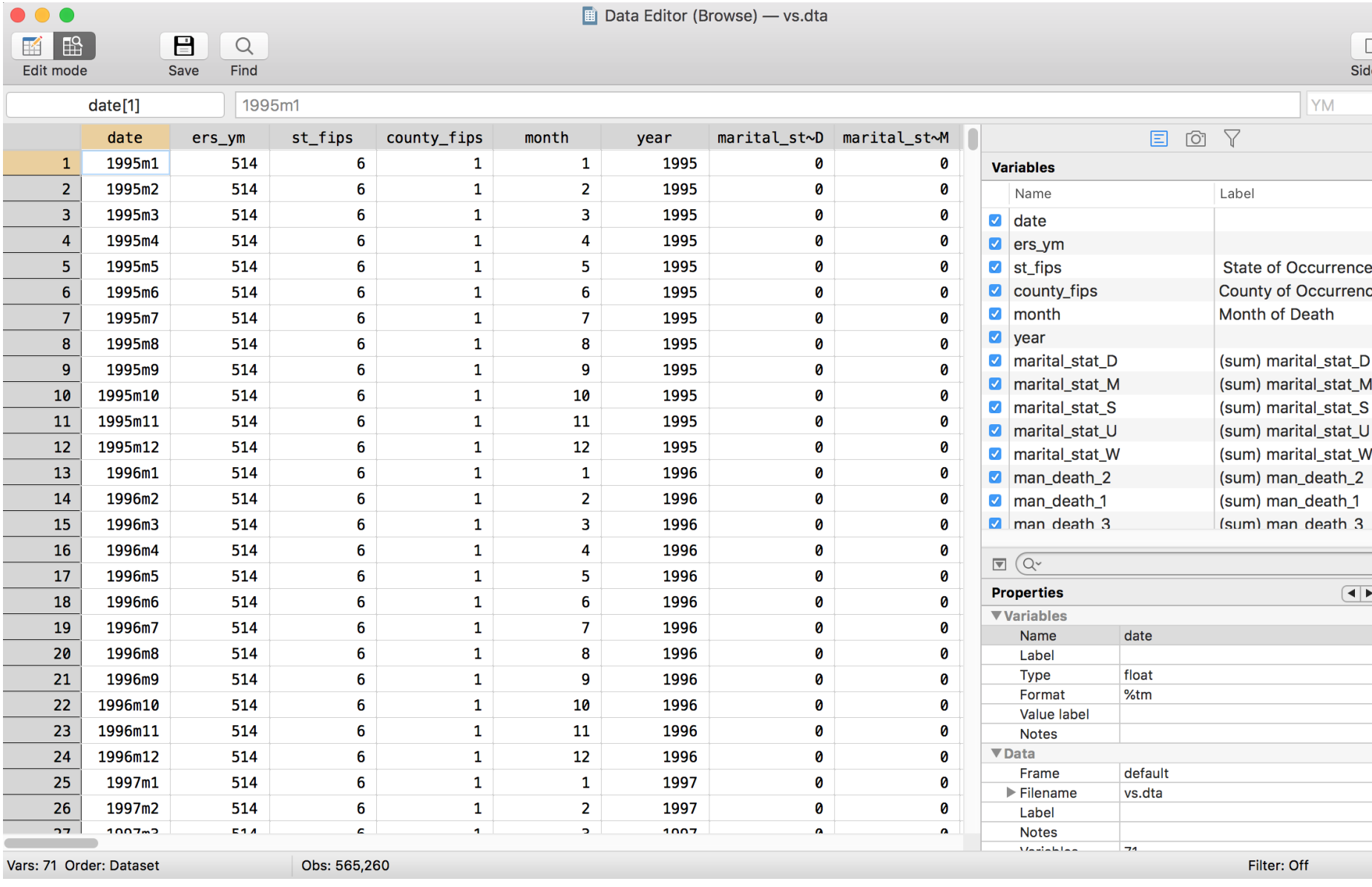
Figure: Image from Wenfei Xu at Columbia

Read the codebook

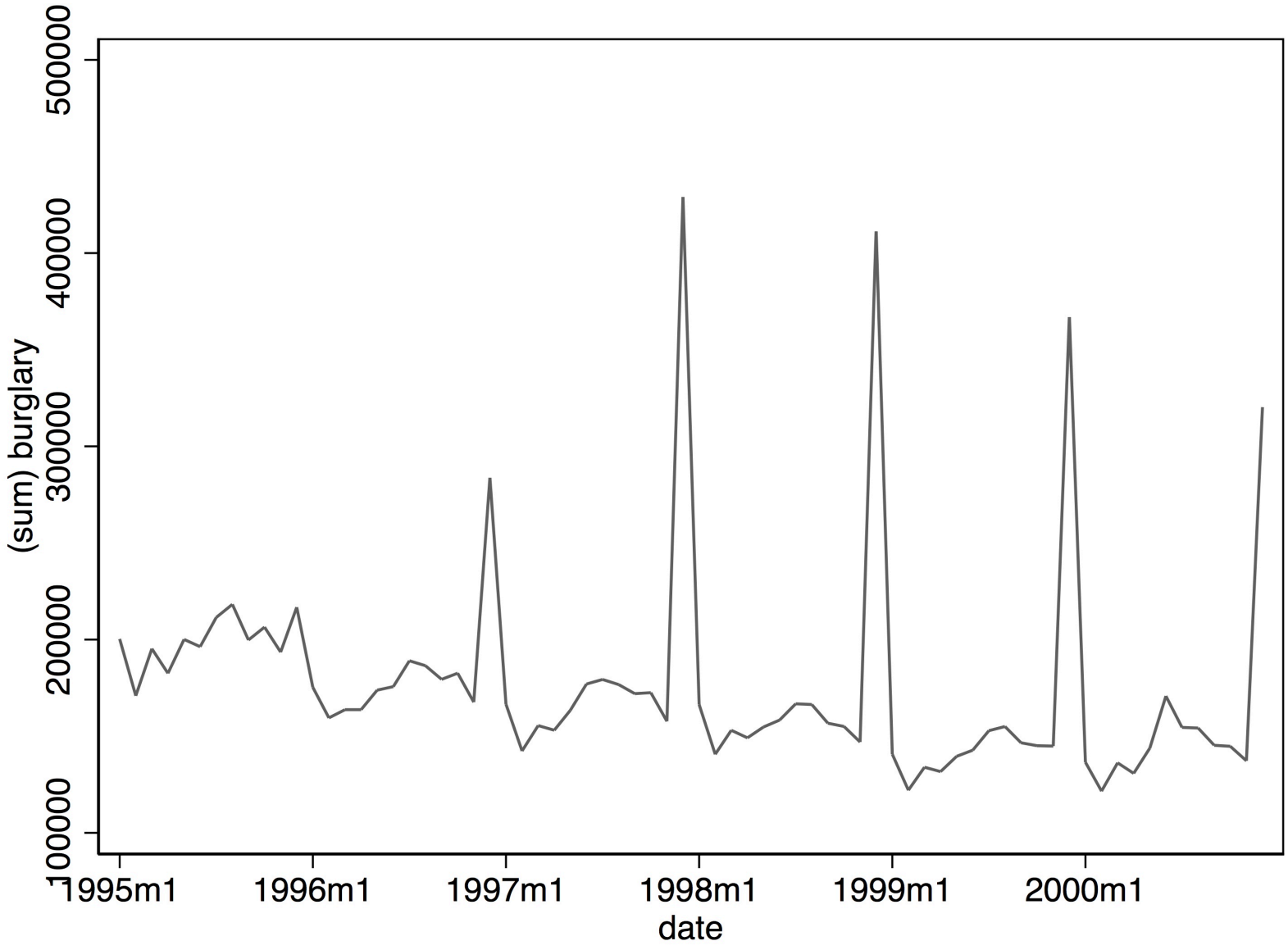
- We stand on the shoulders of giants
- Few like reading the codebook as it is not gripping literature
- But the codebook explains how to interpret the data you have acquired and it is not a step you can skip
- Set aside time to study it, and have it in a place where you can regularly return to it
- This goes for the `readme` that accompanies some datasets, too.

Look at the data

- The eyeball is not nearly appreciated enough for its ability to spot problems
- Use `browse` or excel to just read the spreadsheet with your eyes.
- Scroll through the variables and accompany yourself with what you've got visually



	date	ers_ym	st_fips	county_fips	month	year	marital_st~D	marital_st~M
1	1995m1	514	6	1	1	1995	0	0
2	1995m2	514	6	1	2	1995	0	0
3	1995m3	514	6	1	3	1995	0	0
4	1995m4	514	6	1	4	1995	0	0
5	1995m5	514	6	1	5	1995	0	0
6	1995m6	514	6	1	6	1995	0	0
7	1995m7	514	6	1	7	1995	0	0
8	1995m8	514	6	1	8	1995	0	0
9	1995m9	514	6	1	9	1995	0	0
10	1995m10	514	6	1	10	1995	0	0
11	1995m11	514	6	1	11	1995	0	0
12	1995m12	514	6	1	12	1995	0	0
13	1996m1	514	6	1	1	1996	0	0
14	1996m2	514	6	1	2	1996	0	0
15	1996m3	514	6	1	3	1996	0	0
16	1996m4	514	6	1	4	1996	0	0
17	1996m5	514	6	1	5	1996	0	0
18	1996m6	514	6	1	6	1996	0	0
19	1996m7	514	6	1	7	1996	0	0
20	1996m8	514	6	1	8	1996	0	0
21	1996m9	514	6	1	9	1996	0	0
22	1996m10	514	6	1	10	1996	0	0
23	1996m11	514	6	1	11	1996	0	0
24	1996m12	514	6	1	12	1996	0	0
25	1997m1	514	6	1	1	1997	0	0
26	1997m2	514	6	1	2	1997	0	0


(sum) burglary
100000
200000
300000
400000
500000
1995m1
1996m1
1997m1
1998m1
1999m1
2000m1
date

Missing observations

- Check the size of your dataset in Stata using `count`
- Check the number of observations per variable in Stata using `summarize`
  - String variables will always report zero observations under `summarize` so `count if X==""` will work
- Use `tabulate` also because oftentimes missing observations are recorded with a -9 or some other illogical negative value

Missing years

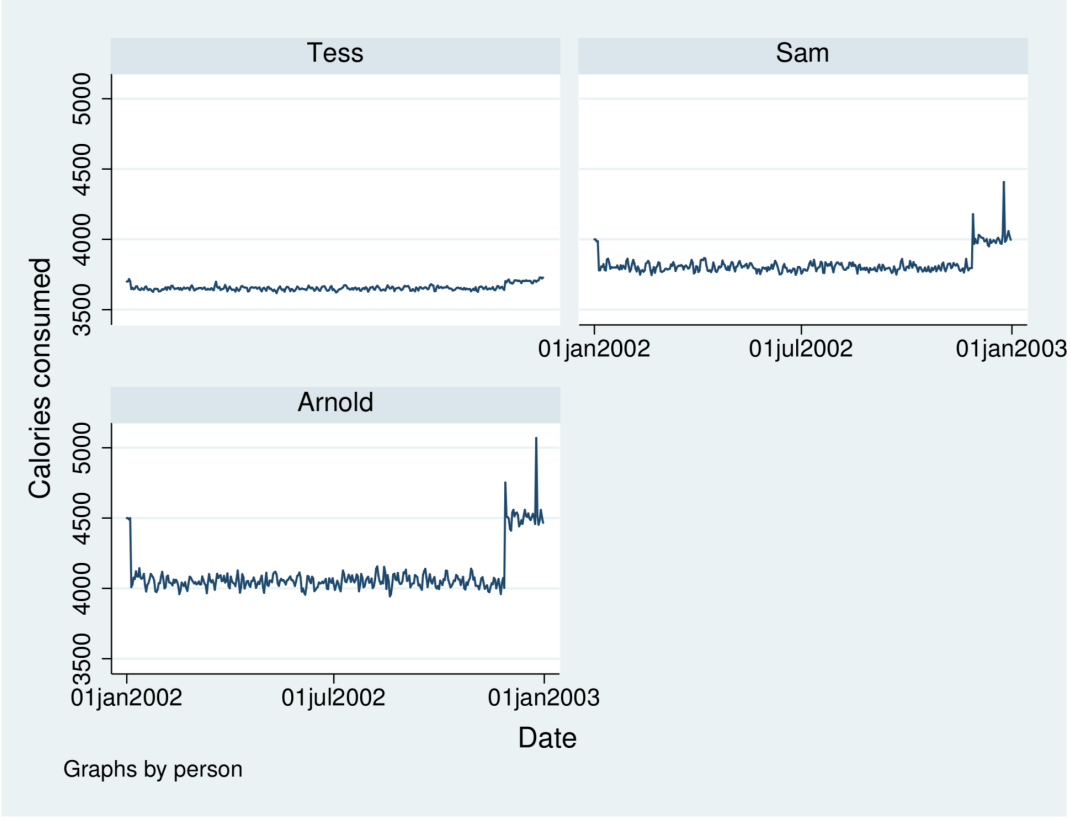
- Panel data can be overwhelming bc looking at each state/city/firm/county borders on the impossible
- Start with `collapse` to the national level by year and simply `list` to see if anything looks strange
  - What's "strange" look like?
  - Well wouldn't it be strange if national unemployment rates were zero in any year?
- You can use `xtline` to see time series for panel identifiers, with or without the subcommand of `overlay`

```
. collapse (sum) male_homicide female_homicide, by(year)

. list
```

	year	male_h~e	female~e
1.	1995	0	0
2.	1996	0	0
3.	1997	0	0
4.	1998	0	0
5.	1999	0	0
6.	2000	0	0
7.	2001	0	0
8.	2002	0	0
9.	2003	4474	910
10.	2004	4270	900
11.	2005	4450	895
12.	2006	4479	889
13.	2007	4480	895
14.	2008	4228	893
15.	2009	3857	866

```
. xtline calories, tlabel(#3)
```



```
. xtline calories, overlay
```

Panel observations are $N \times T$

- Say you have 51 state units (50 states plus DC) and 10 years
- $51 \times 10 = 510$ observations
- If you do not have 510 observations, then you have an unbalanced panel; if you have 510 observations you have a balanced panel
- Check the patterns using `xtdescribe` and simple counting tricks

Results

```
.
. tsset id date
       panel variable:  id (unbalanced)
        time variable:  date, 1995m1 to 2009m12
                delta:  1 month

.
.
end of do-file

. xtdescribe

      id:  4, 5, ..., 270                                    n =         69
    date:  1995m1, 1995m2, ..., 2009m12                      T =        180
           Delta(date) = 1 month
           Span(date)  = 180 periods
           (id*date uniquely identifies each observation)

Distribution of T_i:   min      5%     25%       50%       75%     95%     max
                        24      48     180       180       180     180     180

     Freq.  Percent    Cum. |  Pattern
 ---------------------------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
> _______________
       58     84.06   84.06 |  111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
> 111111111111111
        3      4.35   88.41 |  ............................................................111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
> 111111111111111
        2      2.90   91.30 |  ............................................................................................................................................................111111111
> 111111111111111
        2      2.90   94.20 |  ........................111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
> 111111111111111
        1      1.45   95.65 |  ................................................................................................................................................111111111111111111111
> 111111111111111
        1      1.45   97.10 |  ....................................................................................................................................111111111111111111111111111111111
> 111111111111111
        1      1.45   98.55 |  ....................................................................................111111111111111111111111111111111111111111111111111111111111111111111111111111111
> 111111111111111
        1      1.45  100.00 |  111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111.............................................
> ...............
 ---------------------------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
> _______________
       69    100.00         |  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
> XXXXXXXXXXXXXXX

.
```

log on (text)

```
. gen one = 1

. bysort county_group: egen count=sum(one)

. ta count
```

count	Freq.	Percent	Cum.
24	48	0.42	0.42
36	36	0.31	0.73
48	48	0.42	1.15
96	96	0.84	1.99
120	480	4.19	6.18
156	312	2.72	8.90
180	10,440	91.10	100.00
Total	11,460	100.00	

Merge

- During a stage of arranging datasets, you will likely merge – oftentimes a lot
- Make sure you count before and after you merge so you can figure out what went wrong, if anything
- Also make sure you're using the contemporary m:m syntax as many an excellent empiricists have been hurt by merge syntax errors

```
. count
  48,600

. do "/Users/scott_cunningham/Dropbox/Indy/Do/.tm-stata-55642.do"

. merge 1:1 id date using ../data/seer.dta
(note: variable month was byte, now float to accommodate using data's values)

    Result                           # of obs.
    -----------------------------------------
    not matched                       517,044
        from master                       384  (_merge==1)
        from using                    516,660  (_merge==2)

    matched                            48,216  (_merge==3)
    -----------------------------------------

. ta _merge

                _merge |      Freq.     Percent        Cum.
-----------------------+-----------------------------------
       master only (1) |        384        0.07        0.07
        using only (2) |    516,660       91.40       91.47
           matched (3) |     48,216        8.53      100.00
-----------------------+-----------------------------------
                 Total |    565,260      100.00

.
.
end of do-file

. count
  565,260
```

Don't forget the question

- "Exploring the data" is intoxicating to the point of distracting
- "All you can do is write the best paper on the question you're studying" – Mark Hoekstra
  - Note he didn't say "Write the best paper you're capable of writing"
  - He said **the best paper**
  - Important therefore to choose the right questions with real upside
- Slow down, think big picture, force yourself to figure out exactly what your question is, who is in your sample (and importantly who won't be) and what time periods you'll pull

Organize your directories

- After the coding error fiasco, I spent a lot of time wondering how this could happen
- I decided it was partly because of four problems related to
  1. organized subdirectories
  2. automation
  3. naming conventions
  4. version control
- I'll discuss each but I highly recommend that you just read Gentzkow and Shapiro's excellent resource "Code and Data for the Social Sciences: A Practitioner's Guide" `https://web.stanford.edu/~gentzkow/research/CodeAndData.pdf`

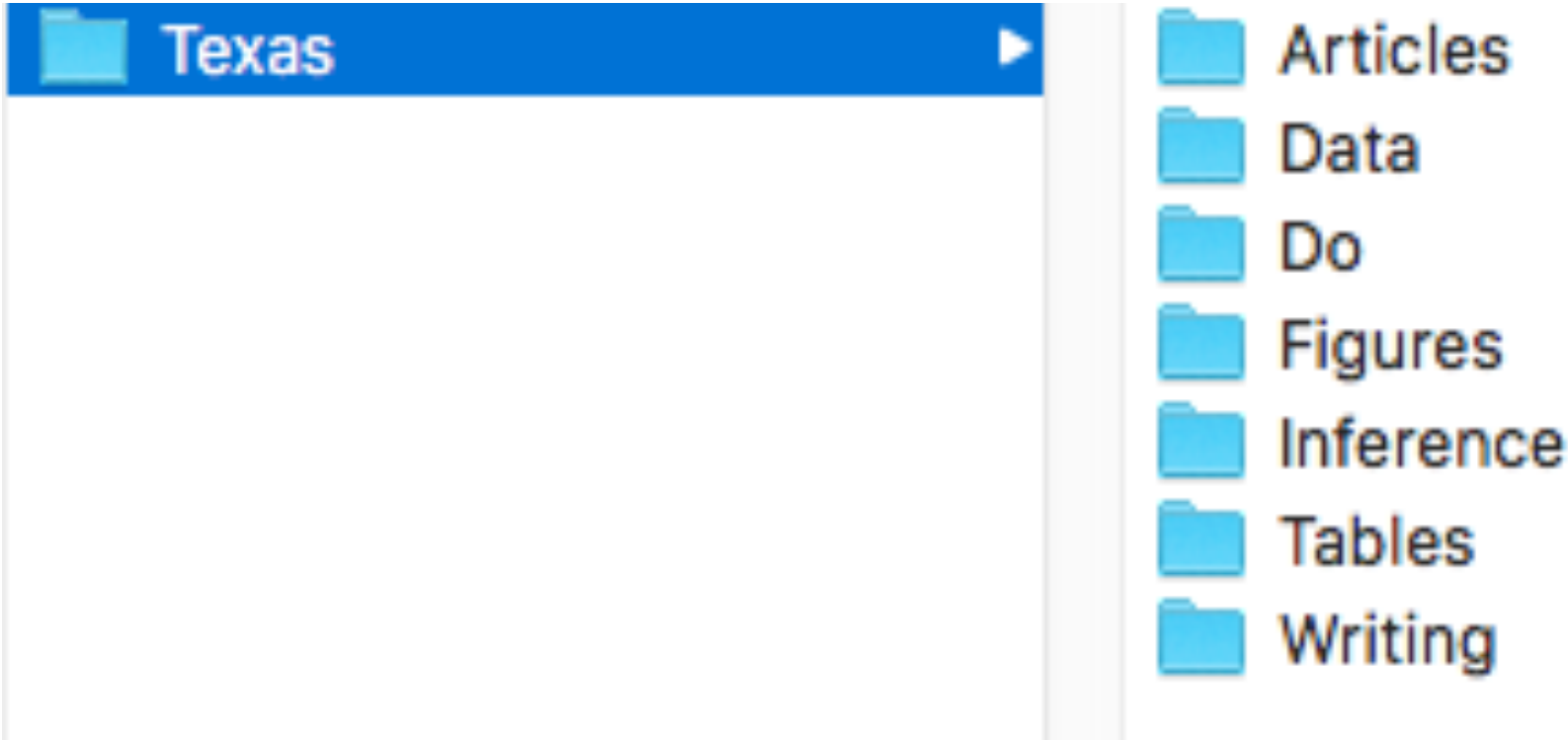
No correct organization

- There is no correct way to organize your directories,
- But all competent empiricists have adopted an intentional philosophy of how to organize their directories
- Why? Because you're writing for your future self, and your future self is lazy, distracted, disinterested and busy

Directories

- The typical applied micro project may have hundreds of files of various type and will take *years* just to finish not including time to publication
- So simply finding the files you need becomes more difficult if everything is stored in the same place
- When I start a new project, the first thing I do is create the following directories

Subdirectory organization

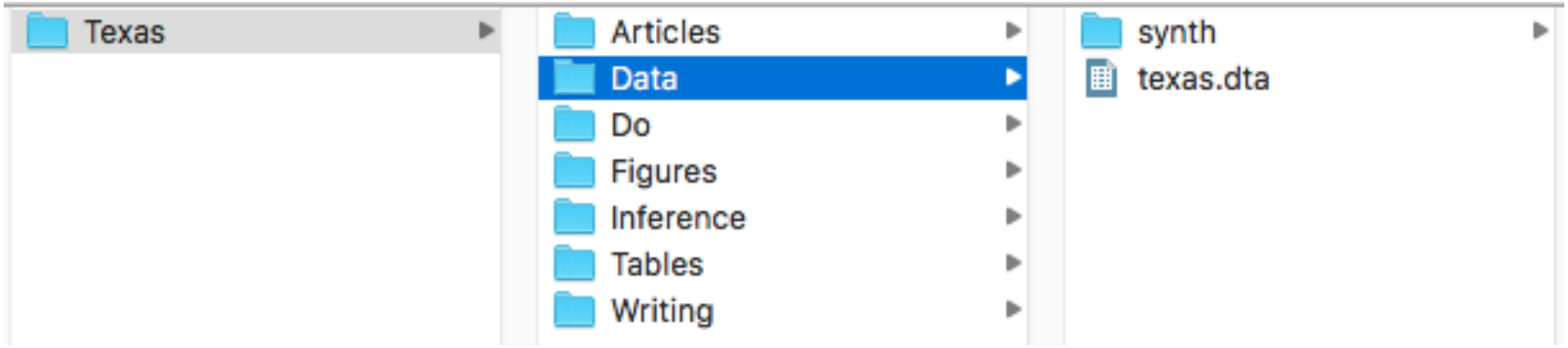


1) Name the project ("Texas")

Subdirectory organization

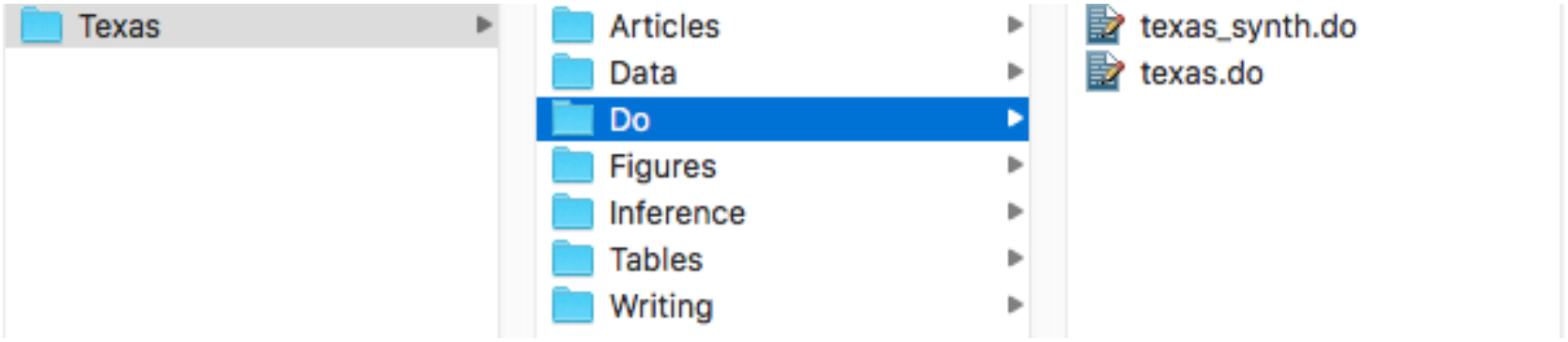
2) A subdirectory for all articles you cite in the paper

Subdirectory organization



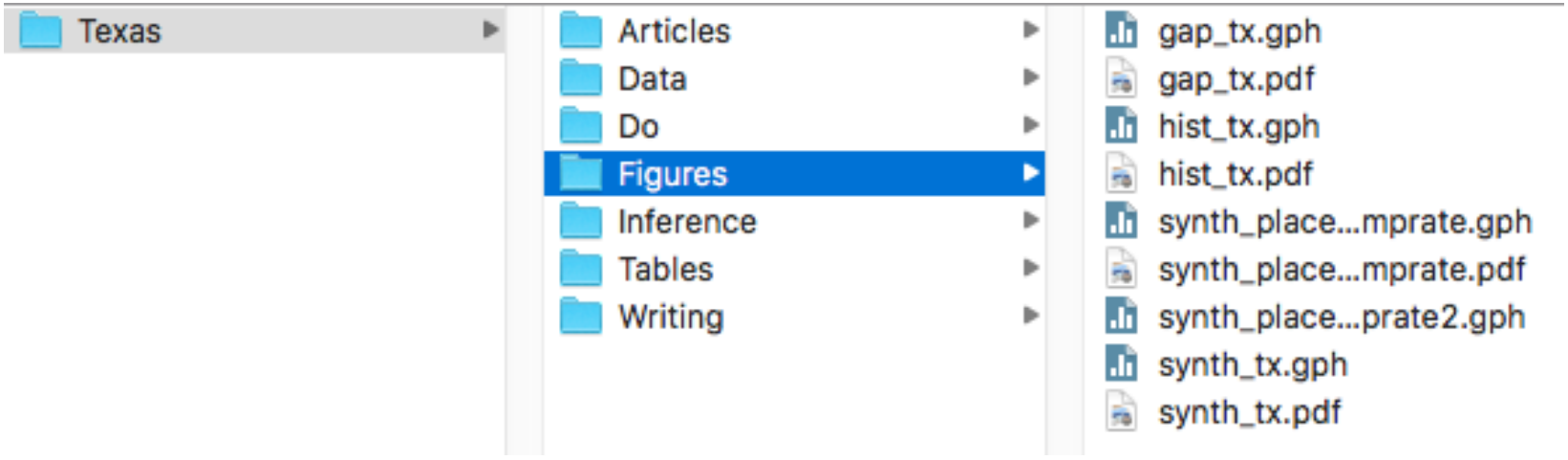
3) Data subdirectory containing all datasets

Subdirectory organization



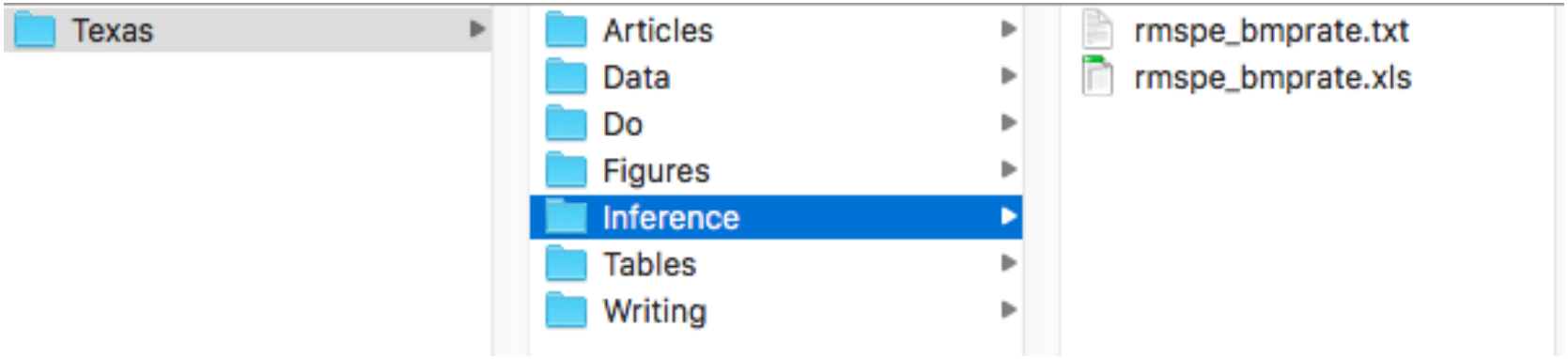
4) A subdirectory for all do files and log files

Subdirectory organization



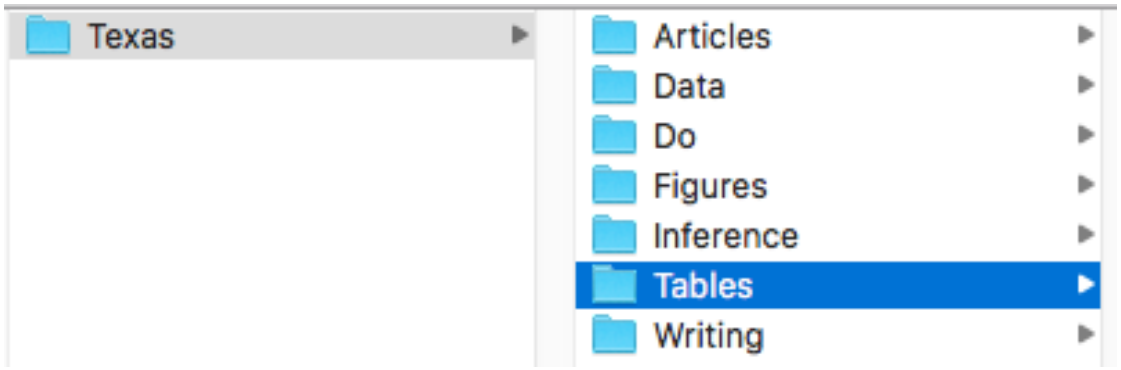
5) All figures produced by Stata or image files

Subdirectory organization



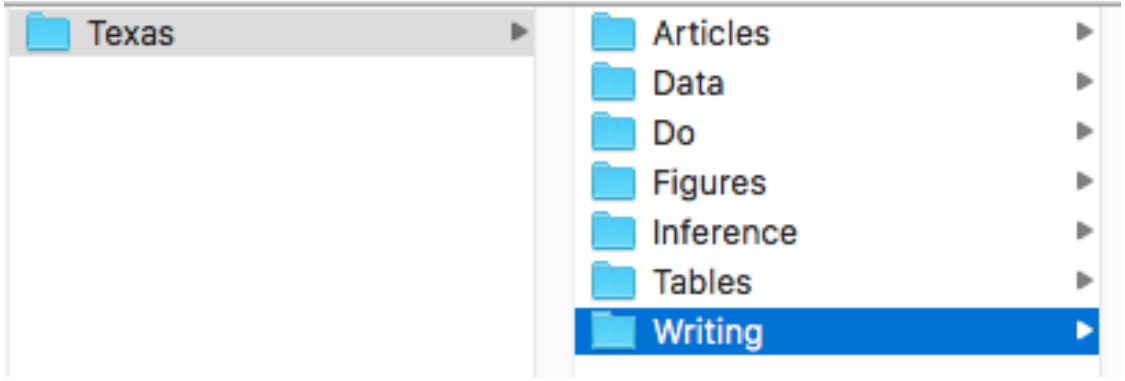
6) Project-specific heterogeneity (e.g., "Inference", "Grants", "Interview notes", "Presentations", "Misc")

Subdirectory organization



7) All tables generated by Stata (e.g., .tex tables produced by -estout-)

Subdirectory organization



8) A subdirectory reserved only for writing

Always use scripting programs NOT GUI

- Guess what - your future self doesn't even remember making do files, tables or figures, let alone typing into GUI command line
- Therefore throw her a bone, hold her hand and walk her exactly through everything
- Which means you've got to have replicable scripting files*
  - * Sure, sometimes use the the command line for messing around
  - But then put that messing around in the program

Good text editor

- Remember: the goal is to make beautiful programs
- Invest in a good text editor which has bundling capabilities that will integrate with Stata, R or LaTeX
- I use Textmate 2 because I use a Mac and in addition to a Stata and R bundle, it also allows for *column* editing
- PC users tend to love Sublime for the same reasons
- Stata and Rstudio also come with built-in text editors, which use slick colors for various types of programming commands

Headers

```
*************************************************************************
* name: texas.do
* author: scott cunningham (baylor university)
* description: estimates the causal effect of prison capacity
*              expansion on incarceration rates using synth
* date: march 19, 2018
*************************************************************************
```

Speak clearly

"Be conservative in what you do; be liberal in what you accept from others." - Jon Postel

- Smart sounding quote about both programming and relationships
- Your future self is time constrained, so explain *everything* to her as well as write clear code
- Optimally document your programs
- But speak your future self's love language so she understands

Automating Tables and Figures

- Your goal is to make "beautiful tables" that are never edited post-production as well as readable on their own
- Large fixed costs learning commands like -estout- or -outreg2-: incur them bc marginal costs are zero
- I use -estout- because Jann has written an excellent help file at `http://repec.org/bocode/e/estout/hlp_esttab.html` but many like -outreg2-
- Learn -twoway- and/or -ggplot2- and make "beautiful pictures" too

Different elements

- When I found my error, and after I regained my exposure, I eventually developed a system of naming
  1. variables,
  2. datasets, and
  3. do files
- As these are the three things you repeatedly use, you need to have a system, even if not mine

Naming conventions for variables

- Variables should be readable to a stranger
  - Say that you want to create the product of two variables. Name it the two variables with an underscore
  - `gen price_mpg = price * mpg`
- Otherwise name the variable exactly what it is
  - gen bmi = weight / (height^2 * 703)
- Avoid meaningless words (e.g., `lmb2`), dating (e.g., `temp05012020`) and numbering (e.g., `outcome25`) as your future self will be confused

Naming datasets and do files

- The overarching goal is always to name things so that a stranger seeing them can know what they are
- One day you will be the stranger on your own project! Make it easy on your future self!
- Choose some combination of simplicity and clarity but whatever you do, be consistent
- Avoid numbering datasets unless the numbers correspond to some meaningful thing, like randomization inference where each file is a set of coefficients and numbered according to FIPS index

Version control

- People swear by git, particularly Gentzkow and Shapiro
- I use Dropbox, and have for years. They have some version history for instance, though I'm not sure if it compares to git's capabilities.
- I'm slowly learning git and use git Tower, but many use the command line in Terminal
- Ideally your system allows you to revert to earlier versions without having ten billion files with names like `prison_03102019_sc.do`, etc.

Selling your work

- If you don't advocate for your work, *no one will*.
- Network, network, network
- You will need to become an expert in 1.5 areas, and you will need experts in those 1.5 areas to agree
- Study the effective of rhetoric of successful economists who expertly communicate their work to others both in their writing of the actual manuscript, as well as the presentation and promotion of their work

Find your mentors and sponsors

- Working with senior people at some point becomes necessary
- Good news: many senior people want to help you
- Bad news: they don't know who you are and can't find you
- It's a two sided matching problem
- Introduce yourself in socially appropriate ways!

Al Roth story

- I wrote Al Roth in 2007 and like Robert Browning to Elizabeth Barrett introduced myself by saying "I love your book on twosided matching with Sotomayor with all my heart."
- We became pen pals and then he won the Nobel Prize
- Scared, I wrote to congratulate him on the day he won and he immediately asked to help me
- "Interpersonal favors are meant to be paid forward not backwards" - Roth to me after a *second* favor!
- Nobody can help you if you don't know them bc help, sponsorship and mentoring is a two sided matching problem

More readings

- I've put several deck of slides and helpful articles for you in the dropbox folder
- Jesse Shapiro's "How to Present an Applied Micro Paper"
- Gentzkow and Shapiro's coding practices manual
- Rachael Meager on presenting as an academic
- Ljubica "LJ" Ristovska's language agnostic guide to programming for economists
- Grant McDermott on Version Control using Github `https://raw.githack.com/uo-ec607/lectures/master/02-git/02-Git.html#1`

Data Visualization

Every project should present compelling graphics summarizing the main results and main takeaway

- Study other people's pictures and get help from experts
  1. Kieran Healy's 2018 Visualization: A Practical Introduction (Princeton University Press); free version is `http://socviz.co/index.html#preface`.
  2. Ed Tufte's book Visual display of quantitative information is classic, but more a coffee table book plus no programming assistance.
- Learn Stata's -twoway- capabilities and/or R's -ggplot2-