Practical Data Analysis Tips Pt. 1

ECON 490

Taylor Mackay || Email: tmackay@fullerton.edu

Handling Data (and Avoiding Headaches)

General project workflow looks something like:

- Find some data set(s) online, download them – this is your ***raw data***
- Clean up/process your raw data to create your ***working data***

KEY POINT: You should always retain an ***unmodified*** version of your raw data

If you need to change your project in the future, and you can't recreate your working data set from raw data, you can wind up totally stuck

- If you're using Excel, don't just start editing data you've downloaded!
- Generally, this is less of a concern when using R/Python

Data Processing Workflow

Processing/cleaning your data means filtering rows, selecting columns, etc.

- You should have a clear record of how you're doing this
- Use .R script/code files – don't just run things in the console!
- If you're using Excel, use comments or a separate text file describing process

KEY QUESTION: *Ask yourself as you're working, "If my computer died/rebooted right now, how screwed would I be?"*

If you'd lose all your project progress if your computer died, something is wrong

- All code/data should be backed up remotely via Dropbox/iCloud/Drive, etc.

Where to Find Data

Easy places to start:
1) Capstone Data Resources page on ECON 490 Canvas page
2) IPUMS ACS/CPS websites for individual-level survey data
    - Use variable search feature to get a sense of what's available

As you look at new data, ask yourself, "What correlations/regressions are interesting?"
- If you can't think of something (relatively) quickly, move to another data source

WARNING: *ChatGPT is horrible with questions like, "What data sets should I use?"*
- 5.0 combines overconfidence with high error rate = lots of potential time lost
- Claude/Gemini are better (more conservative) but this one area where AI still struggles

Build the Plane While You're Flying

If a project idea isn't going to work, you want to know sooner rather than later

- Try to define and run a "minimum viable" regression to test your ideas
- Start by doing just enough data collection to run that test regression

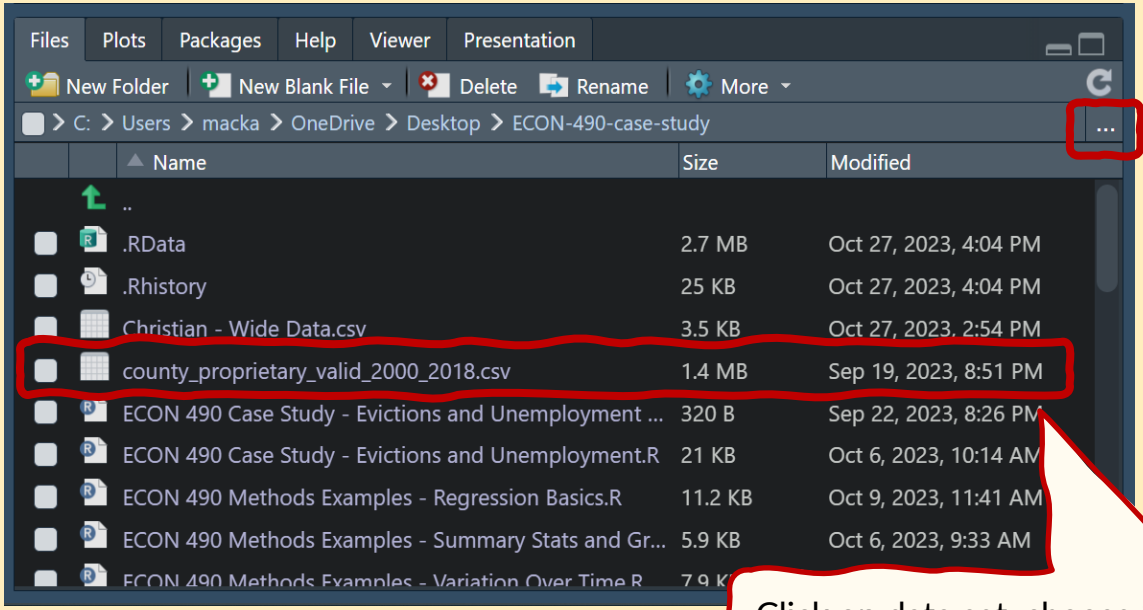
E.g., do you plan to collect 10 years of data? Try running a regression with 2 years first.

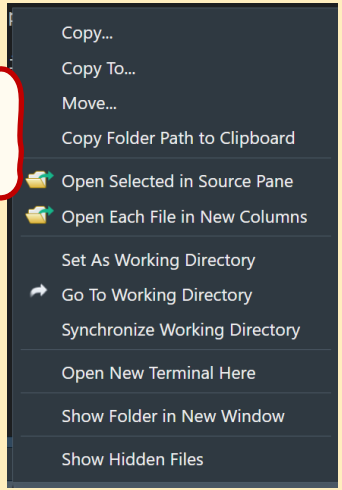
If test models don't work, you can adjust goals as you go

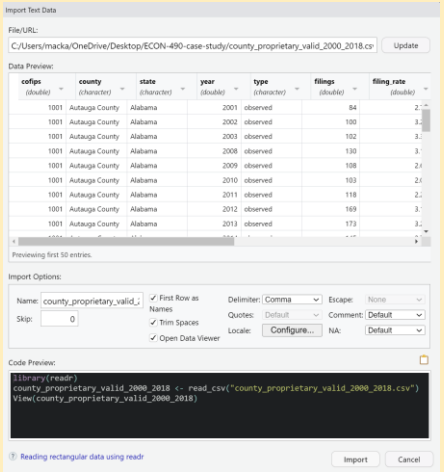
- Once you've test things, you can then collect more data
- ***Rule of thumb:*** *More data is better… once you have a clear plan/goal*

Loading Data into R

Click …, load folder with your data and set working directory







Click on data set, choose “Import Data,” to preview & load data

Wide vs. Long-Formatted Data

Two types of data structures:

- ***Wide*** data has same variable in multiple columns (here, GDP is split by year)
- ***Long*** data has each variable in one column

In almost all cases, data should be in long format for running regressions

- In R, use `pivot_longer()`

	state	year	GDP
1	AZ	2020	100
2	AZ	2021	105
3	AZ	2022	107
4	CA	2020	157
5	CA	2021	163
6	CA	2022	168
7	NM	2020	95
8	NM	2021	102
9	NM	2022	103

Long-formatted data

	state	gdp.2020	gdp.2021	gdp.2022
1	AZ	100	105	107
2	CA	157	163	168
3	NM	95	102	103

Wide-formatted data

Who's in my Regression Sample?

Key question before running any regression – who gets included?

- Sometimes, this answer is straightforward (esp. with state or county data)
- Matters a lot with individual data like the CPS (or business-level data, etc.)

Data sets like the CPS have kids/retired folks included

- You might not want them in a regression exploring union membership!
- Use the `filter()` function to restrict sample to observations you want

Describing sample restrictions is a ***key*** part of interpreting regression output

Outliers and “Weird” Observations

Always check your data with `summary()` and / or `table()`

- This lets you catch values that don’t look right
- Do you see outliers or repeated instances of the same random value?

Dealing with these situations requires background knowledge about your data

- Are extreme values feasible or plausible for a given variable?
- Do you have unusually coded missing values (e.g., -99, 999, 1000)?

As a general rule, be careful removing outliers if you think they’re real

- If you remove values or rows, how does this change regression interpretation?

The Next Slide is IMPORTANT

Is everyone paying attention?

Things You Should Absolutely Remember

Even if you forget everything else from tonight, remember the following:

1. Keep clean, separate, ***unedited*** versions of ***all*** raw data
2. Write code somewhere that you can save/back up (e.g., R scripts, Colab, etc.)
3. Check variable definitions for missing values
4. Always visually/manually check output when you combine/merge data

ASK YOURSELF: *What happens if my computer reboots right now? Am I okay?*

- Back all data and code up via Google Drive/Dropbox/OneDrive

Bonus Tips: Make sure your data is long-formatted before running regressions. Also, make sure your R code runs start-to-finish (don't just run chunks randomly)