data

adam okulicz-kozaryn

adam.okulicz.kozaryn@gmail.com

this version: Tuesday 24th January, 2023 09:15

outline

replication

data basics

merge

tips

outline

replication

data basics

merge

tips

replication, replication

- replication=write computer code that will do *everything*
  - from raw data (eg FED, IMF) to vis
- necessary for science
- otherwise we don't know what happened
- how was it calculated? is there a mistake? who knows?
- IT perspective `http://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.1001745`
- pol sci perspective

`http://gking.harvard.edu/files/gking/files/replication.pdf`

rules for everyday practice [revisit/stress later!!]

- once you have coded everything, double/triple-check it
  - leave it aside and check again
  - show it to other people, post on your website
- cross-check end output with raw data–e.g. are there the same numbers for randomly chosen data points– does it make sense?
- check with alt data ? they tell the same story?
  - i always google tables/graphs of what i study
- everything has been already studied by others
- like lit rev, its data rev

get code from others!

- the easiest way to do research in 21st century
- start with code others wrote, and build on their work
- this is the fastest, most efficient way to do research
- any research very close to yours, just email author and ask her to share code with you
- even if it sas or spss etc–you'll be able to figure it out quickly what is going on there and then implement something similar in stata
- don't reinvent the wheel: almost as if you were to start research without reading literature and had to come up with all theories and ideas on your own!

outline

replication

data basics

merge

tips

data basics

- dataset is a matrix
- cols are variables (var), rows are observations (obs; U/As)
- vars are characteristics of obs
- eg 'edu, 'age', and 'inc are vars and persons are obs
  - each row is a separate person
- have data clean! eg only one top row for var names
  - (xls is typically mess with unusable var names)

be super careful and clear

- define each var used in as much detail as possible
  - eg instead of “income” “median household income in current dollars”
- think about limitations, shortcomings
  - eg sampling error, missing data, etc
- always try to triangulate, ie measure the concept with multiple vars

data types

- there are dozens of data types/formats/files
- a basic distinction:
  - software-specific binary files (.dta, .sas7bdat, .sav)
  - generic text files (.txt, .dat, .csv, .tab)
- just google it! eg 'pandas read csv', 'pandas export spss' etc

APIs/databases

- but wait, we have databases and internet
- outside of academia, in the real world
- all data are in databases
- Oracle, MySQL, NoSQL, MsSQL, or even MsAccess
  - usually can use Python to pull directly from databases
- just google it, eg "Python World Bank API"
- but first check maybe Pandas has an interface
  - see notebook sec "API"

API=Application Programming Interface

- basically an interface to get internet data
- most major websites have API
- you can connect to API and get data
- websites run on top of databases, eg google, twitter
- API is a way to access that database
- pretty much any company and organization has a database, is online, and has API

API v web scraping

- web scrping is getting data from internet by hand
  - flexible–can do any website
  - but has to write the code to extract information
  - time consuming, better use API
  - we'll just do API
- if you want data from website without API
  - email them first! they probably have API! say research only, otherwhise need to pay
  - and if that doesnt work, let me know and we'll try to scrap it

outline

replication

data basics

merge

tips

the power of merge

- merging is one of the most useful things you'll learn here
- great value comes from simple fact of merging data
- recall from intro: there's a ton data of (and growing!)
- but these data are mostly useless unless in one file!
- somehow orgs (and researchers) in this persistent habit of having data chopped up in tiny multiple files
- hungry for knowledge want to use the data– this is where you come in! make $ just merging!
  - (and then fun: vis/graphs in 2wk, but merge first!)

easy to merge; difficult to do it right

- it depends on what kind of data (and luck) you have
- the challenge is to check what happened after the merge
- almost always it merges with issues
- thats where the work begins
- **always investigate carefully non-merges**
- **make sure that *ALL* nonmerges are as expected**
- **even matches can be wrong**
  - use a lot of des sta to investigate
  - always be skeptical, ask yourself whether it makes sense

after merge

- typically some obs did not merge due to diff coding
- say "Poland≠ "Republic of Poland"
- "CAMDEN"≠"Camden" etc
- then go back and fix it before merge:
- replace ctry="Poland" if ctry=="Republic of Poland"
- in many cases it wasn't supposed to merge
  - eg data A: 1995-2000, but B: 1990-1998
- have to be 100% sure that nonmerges are correct!

merging investigation

- very useful!!:
- investigate, eg tabulate time and geography
  - say year and state
- may also want to list part of datafile
  - especially if it is small
- can also sort on key vars
- it does take time to find out what happened

be clear about merging

- want to be clear about nonmerges in paper!
  - say how many nonmerges and what you did about it
  - eg dropped, fixed, etc

what to merge on?

- geography! usually have some!
- and can always aggregate up! say have city and state, so can merge m:1 on state
- time! say with weather–usually weather matters!
- occupation! there are occ codes eg `https://www.onetonline.org/find/descriptor/result/4.A.2.b.2`

outline

replication

data basics

merge

tips

datasets of the day

- climate/weather, down to county (easy access!)
  - `https://wonder.cdc.gov/EnvironmentalClimateData.html`
- religion!
  - `http://www.thearda.com/Archive/Files/Descriptions/RCMSCY10.asp`
  - `http://www.thearda.com/Archive/Files/Descriptions/CMS90CNT.asp`
- state level policy `https://www.statepolicyindex.com/data/`

make lots of comments in your code

- make comments in notebook in code cells, important!
  - explain to yourself what command does, what to look for, etc
- and use plenty of text cells
- if you do not make comments, you will forget...
- use very handy keywords like “TODO”, “KLUDGE”, “BUG”, “LATER”, “FIXME”
  - then can ctrl-f for them :)