data

adam okulicz-kozaryn

adam.okulicz.kozaryn@gmail.com

this version: Thursday 26th January, 2023 10:32

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 don't know what's up: how was it calculated? is there a mistake? who knows?
- IT perspective `http://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.1001745` [superb! read it!]
- 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 final output with raw data–eg are there the same numbers for randomly chosen data points; does it make sense?
- check with alt data: they tell the same story?
- ○ google tables/graphs of what i find
- ○ everything has been already studied by others
- ○ like lit rev, its data rev

get code from others! fastest, most efficient way

- the easiest way to do research in 21st century
- start with code others wrote, and build on their work
- any research very close to yours, try to find the code online, otherwhise just email author and ask her to share code
- even if it sas or spss etc–you'll be able to figure it out quickly what is going on there and then implement in Py
- don't reinvent the wheel: almost as if you were to start research without reading literature and had to come up with all theories, ideas, and setup 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), and 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 a mess with unusable var names)

be super careful and clear

- define key vars in as much detail as possible

◦ eg “income”$- >$“median hh income in current USD”

- think about limitations, shortcomings

◦ eg sampling error, missing data, etc

- always try to triangulate, ie measure the concept with multiple vars

data types

- dozens of data types/formats/files
- just google it! eg 'pandas read csv', 'pandas export spss'
- and see link to IO formats in notebook

APIs/online databases

- but wait, we have databases and internet
- Oracle, MySQL, NoSQL, etc
  - 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 link do documentation in notebook sec “API”
- also much easier to use API the web scraping

outline

replication

data basics

merge

tips

the power of merge

- do merge as much as possible! great value!
- merging is one of the most useful things you'll learn here
- there's a ton data of (and growing!)
- great value comes from simple fact of merging
- using just one data can only do so much
- by merging easily create dataset that nobody else
  - and produce insight nobody else had
- eg `https://www.amazon.com/gp/product/0063032376`
-

easy to merge; difficult to do it right

- the challenge is to check what happened after the merge
- **always investigate carefully non-merges**
- **make sure that *ALL* nonmerges are as expected**
- **even matches can be wrong**
  - use lots of des sta to investigate
  - be skeptical: does it makes sense?
- typically non-merges due to diff coding:
  “Poland≠ “Rep. of Poland”; “CAMDEN”≠”Camden” etc
- go back and fix it before merge:
- replace to “Poland” from “Republic of Poland”
- often wasn’t supposed to merge
  - eg data A: 1995-2000, but B: 1990-1998
- be 100% sure that nonmerges and merges are correct!

merging investigation

-
- tab _merge
- cross-tab _merge with geography and/or time
  - say year and state
- also want to list part of datafile!
  - _merge and key/id vars: geo, time, etc
- can also sort on key vars
- it does take time to find out what happened
- be clear about nonmerges in paper! (first as a comment in code!):
  - 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 collapse 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

data choice matters

- data management often takes 50-90% of time
- most of it is learning/figuring out data
- you'll spend 100+ hrs learning about specific datasets
- dont waste time! pick data that:
  - you're passionate about (eg sth you went to school to learn about, eg poverty, inequality, discrimination)
  - you'll use in other classes, possibly for thesis
  - advance your career after graduation, eg want to work for state–use data they produce or use a lot

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 :)

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/