ps1: I/O and descriptive stats; due in 2wks (feb11)

[version: Wednesday 19th February, 2025 19:53]

note: we will do brief presentations next class

1. read in at least 3 datasets in 3 different formats (eg html, sas, json)
2. produce several useful/interesting descriptive stats and interpret
3. do some manipulations such as subset/slice on condition, filter vars or obs using regexp, and groupby/agg

---

old ps comments

keep it simple especially when learning new things!

way easier to figure things out with a small and handy data

say keep 5 vars and 50 obs

so not only simplicity in code but also in data is good

later: large complex datasets and advanced code

but do try to simplify, esp when learning and figuring it out

if you have questions on my comments on your ps

do ask for clarification!!

i tend to be overly parsimonious

always cite data!

at a minimum say where exactly it come from, ie the url

if ambiguous say which year, wave, version etc

---

general directions (always the same):

- i will show your code in class and possibly repost, as per our core values–opensource and transparency, but if you'd like to keep it private, let me know, you just may get less feedback
- you must submit all the code that was executed from the very beginning starting with the very raw data as per replication principle; if data too big to fit online, then just start with eg "to fit data online took 10% random sample"
- ps are cumulative–can and should include much of previous code; can also use code you've written outside of this class (other classes, projects, etc)–but you have to clearly mark the code that has not been written for this class–otherwise, scholastic dishonesty!
- use your own dataset; again if you do not have a dataset, ask for help finding it
- you are only submitting code, so it must load data from Internet: `https://theaok.github.io/generic/howToPutDataOnline.html` (when you put data into any public space, try not to violate data copyrights... I haven't heard of anyone having problems with that, but be careful–for instance you may subset dataset to few vars and smaller sample); and it is also easier to learn on small datasets
- keep it simple! at the beginning of your notebook drop unnecessary vars; and even retain only fewer obs; keep it manageable; much easier to learn using simple data; can always complicate later!; much better to do it right using simple data than do it wrong using complex data!
- have nice structure in your file: sections, subsections, etc; may also have multiple files
- great to copy code from others; again, one of the rules for this class is 'be lazy': don't reinvent the wheel, whatever you are coding, has already been done, google things often; but of course you cannot submit 100% code by someone's else; if substantial/meaningful chunk of code by someone else (incl AI) cite!
- if you do something extra/fancy that is relevant and closely related to the assignment questions, it will be extra credit
- use coding rules that we've learned so far

- submit (only) the code into git repo; ps are due by the beginning of the next class unless indicated otherwise, eg "due in 2 weeks"; late ps not accepted
- we are on the way to developing the final project with these ps: as we progress, your ps should start resembling a coherent and logical project where you use learned techniques to answer interesting questions–say in few sentences (probably at the beginning) why are you doing what you are doing–that is, answer the "so what question": "ok, you're gonna run all that code, and so what?" what's the goal of all that, why are you doing this? you need a compelling justification for what you are doing: say what are those questions you want to answer; be brief, say couple sentences, typically 10-50 lines is enough; related: say why you use data you are using, is it best, does it serve the purpose?; and can ask us questions in comments
- be prepared do present your code in class (if time), just briefly, key points, couple minutes

- if you work in a group of say 2 people make it 2x better, eg if ps asks for joining 2 datasets, and there are 2 of you, then just join 4, etc, just do 2x more or better
- always have a brief description/interpretation of substantive output such as tables and graphs, say few sentences or a pargraph or max few; also may list problems you've encountered and ask questions
- always have exact links to all of the source data (so that i could create the map myself); note: exact links, eg do not say census.gov, but give full url to the data–i must be able o find it; sometimes there is no generic URL–then give steps: what I need to click to get the data!