basic organization and documentation

adam okulicz-kozaryn
`adam.okulicz.kozaryn@gmail.com`

this version: Wednesday 24th February, 2021 17:05

outline

misc

directory (folder) structure

code structure

naming, labeling

outline

datasets of the day

- climate! (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/RCMSCY.asp
  - http://www.thearda.com/Archive/Files/Descriptions/CMS90CNT.asp
  - http://www.thearda.com/Archive/Files/Descriptions/CMS52CNT.asp
  - more: http://www.thearda.com/Archive/Browse_s.asp?pg=Browse_s.asp&sr=0&m=31&t=Search%20Data%20Archive&searchterms=county&p=B&c=N
- state level policy https://www.statepolicyindex.com/data/

outline

replication: dofile: raw dat$-$ $>$final results

- always keep raw data intact
- then manipulate it and save, even several times
- will have few dats at different stages
- can begin stata session at any stage
- blackboard: draw workflow

files in general singularity rule

- **always one version of a dofile or datafile in one place**
- if you have 2 versions of the same file
  - sooner or later there will be problems!
  - you will update/change one, but forget the other one, etc
- exception is backup; but you never edit the backup!
- and you're all set because GIT does it all for you :)

code in general singularity rule

- just like with files, so with code:
- **have the same chunk of code only in one place**
- if you have code that does the same thing multiple times (in same or many dofiles)
  - then it is time to build some hierarchy and have
  - some parent and some child dofiles
  - typically, a parent will do something basic and generic
- and then different children will pick up the data from parent and each will be doing something differently
- blackboard: draw diagram/flow chart

hierarchy of dofiles / branching

- we often use same data for many projects (eg GSS)
- need one dofile that makes data ready for multiple projects
  - it processes raw data and saves it in usable format (recode, label, calculate new vars, etc)
  - and then always start from there for each new project
  - and do your project specific analysis

datafiles: hierarchy / branching, too

- never overwrite the original datafile, and a good idea to keep datafiles at different stage of advancement
  - especially if data are complex:
  - rawFile$-$ >file1$-$ >file2 –and those are produced by: dofile0$-$ >dofile1$-$ >dofile2 (or subsequent sections in one dofile!)
- and again dofile0 will be common for all projects
- but there may be for project A abd B: dofile1A and dofile1B
- in other words one parent dofile0 will have 2 children: dofile1A and dofile1B
- likewise, rawFile will have 2 different children file1A and file1B

directory structure automated

- ok, let's have a look at code
- for simplicity, i just posted one dofile, but
- i actually mean multiple dofiles: i used triple horizontal line to simulate having a separate file :) dofile

backup

- **backup all files at least once a week**–computers break regularly; flash drives break really often
- have automatic system for backups (i use cron)
  - otherwise you'll forget
- just keep copy of everything in the cloud, goog, amzn, etc

outline

sections, subsections

- dofile should have a multi-layerd structure
  - like chapters, sections, sub-sect in book
- for different levels, use different kinds of comments: box, block, one line, horizontal line, etc

  type them in dofiles and scroll down to already existing
  - now i just use '***', '**', '*', '//'
  - i used to use —— (still in dofile)
- definitely use "FIXME" "LATER" "KLUDGE" etc

outline

general

- naming and labeling looks like waste of time
- but at the end saves time
- labels are like “postit” notes
- importantly, it prevents mistakes/misinterpretations
  - especially, if a project is big and/or you share it with others
  - or if it takes long time

labeling dataset

- labeling dataset is not as useful as labeling variables/values
- it is useful if you have really many datasets and/or problems in these datasets

variable names, labels, and value labels

- variable name is...a variable name, eg educ
- var lab describes var, eg "highest degree completed"
- note is like label, except it can be>80 chars
  - eg put there full svy question: "how would you describe highest level of your education?"
- value label describes values that a variable takes on
  - (output of codebook, or tab and tab,nola), eg:
  - "primary school" 1
  - "high school" 2
  - "college or university"3
- dofile

labels tips

- give variables short names, eg inc
- labels, on the other hand should be descriptive, eg "2004 hh income"
- labels prevent confusion later and for others
- they automatically appear on graphs, regressions, etc.
- use lookfor, especially if you have many variables
- be lazy (remember it's our core value)
  - only label what is necessary
  - indeed, only keep data and variables that are necessary
  - you have the code, so you can always add back in later

more tips on var names

- i dont like '_' anymore
- i just use Caps to denote words, eg
- hhInc as opposed to hh_inc; i guess it's cleaner
- and typicaly i have 3 letter var namees 'swb'
- or 6 letter that combine 2 words: say menHea for mental health
- but do whatever is natural to you!
  - and is simple clean and consistent