macros and loops

adam okulicz-kozaryn

adam.okulicz.kozaryn@gmail.com

this version: Wednesday 24th March, 2021 15:06

outline

misc

macros

loops

wrap up

outline

misc

macros

loops

wrap up

outline

misc

macros

loops

wrap up

macros

- stata forgets local macros after a run from the dofile
- but if you copy-paste, it remembers them...

general idea

- macro='a single instruction that expands automatically into a set of instructions to perform a particular task'
- macro is just a variable or an object
- ○ under which you can assign something
- i guess stata uses term “macro” because there are already variables in dataset

intuition

- macro is like a variable that can take on some values
- value, string, or actually anything including spaces
- hence, a macro can even contain a chunk of code
- macro is like an object in Python or R

local/global

- there are local and global macros
- typically stata users use local
- global always persists during the whole stata session
- on windows local macros persist within one run (you run a block of code),
- ○ globals persist the whole session
- dofile

outline

misc

macros

loops

wrap up

general idea

- repeats the code within the loop for different elements
  - (vars, values, items, items in macro, etc)
- loops are essential for automation
- loops save time and are fun!

intuition

- think of a loop as a simple program:
  provide items for which the job is to be done (e.g. all vars)
  and define the job in a loop (e.g. recode -9 to .)
- again, you can loop over a list of any items, eg numbers or strings

branching

- branching simply means “if else”:

  if something ... do something... else do something
- it makes your code more general
- but adds lines
- dofile

outline

misc

macros

loops

wrap up

use

- macros useful if you need to **repeat the same thing**
  - e.g. path, list of variables, even chunk of code
- loops useful if you need to **do/execute the same thing**
  - eg replace -9 with ., merge data, run regressions, make graphs
- make **repetitive strings** (paths,var lists,etc) into macros
- make **repetitive tasks** (graphs, exporting results, etc) into programs
- any data management job or really any coding requires macros and loops

tips/tricks

- try to define macros (and ados) at the beginning of the dofile/section
  so that it is easy to find
- don't overdo with macros and loops
- again, do not create fancy code for the sake of fanciness
- avoid too long loops, say over 50 lines, difficult to debug, better have it in small steps (though some people disagree)
- i don't do it but it is nice to have indentation

use them for descriptive stats!

- now that you have merged so much data
- it's time for descriptive stats
- you have a dream dataset, time to get to know it
- and macros and loops are excellent for descriptive stats
- ○ perhaps especially for graphs
- ○ there will be a dedicated class to graphs
- ○ but use macros/loops to crank out bunch of graphs and then study them and interpret

CSIKSZENTMIHALYI, M. (1991): Flow: The Psychology of Optimal Experience, Harper Perennial.

——— (2010): "Creativity, fulfillment and flow," Ted Talk,
http://blog.ted.com/2008/10/23/creativity_fulf.