actual programming: adofiles

adam okulicz-kozaryn

adam.okulicz.kozaryn@gmail.com

this version: Tuesday 7th November, 2017 09:19

outline

misc

[*] .ado

outline

misc

[*] .ado

ps3

- cleanliness, simplicity !
- don't be afraid to delete stuff !
- you have git so you can always go back
- so delete reshape, infix, etc unless it is doing something useful; ask me questions
- i know that you can use these because that was last assignment
- but don't keep any stuff that we don't need any more
- same with files, delete everything that we don't need now
- you can always go back ...

ps3

- make you code organized in a logical order...
- renaming in one block, merging in the other block....
- as you write it, things are all over the place, naturally
- like in a paper...
- rewrite your code and reorganize...
- maybe even print it out, mark, reorgainze,
- use scissors and glue if it helps

looking in the future...

- from now on, assignments are less of an exercise and more of a draft of a final project...
- so instead of practicing let's apply knowledge and do something more useful
- it should make sense...so don't do things that do something that is not really important
- e.g. don't save in fixed format, unless you archive for the future...
- just do things that are meaningful...
- many of you just say des sum somewhat mechanically
- or split, then merge; why??

understaning data again...

- again, it is critical that you understand your data...
- so far, you were mostly just doing des and sum
- but there is a ton of other tools that will help you understand your data better
- let's have a look tabstat and table
- let's run examples from help files for these
- can use them with macros and loops

before we begin

- we will have lots of code today
- and some more advanced examples
- again, the purpose of running code is that when you look at it you know what is going on
- don't memorize commands !
- just put some comments, and when you need some code look it up, and reuse it
- copy from others (but acknowledge their work!)

outline

misc

[*] .ado

.ado

- ◇ writing adofiles is the most fancy thing you can do in stata
- ◇ yes, we are at the frontier !
- ◇ we will create our own programs
- ◇ it is lots of fun
- · but also hard work and lots of debugging (error fixing)
- ◇ if you use your own programs, you are an advanced Stata user !
- ◇ writing programs pays off if you use Stata a lot
- ◇ we will see some simple adofiles
- ◇ reference: `http://www.stata-press.com/books/isp.html`

.ado is a bonus

- today's class is somewhat more advanced than what we did so far
- you do not have to write ado programs to get A in this course
  - but it helps in terms of extra credit
  - if you missed points on other things
- macros and loops are essential for your applied work
  - should be in pretty much every dofile
- .ado programs help a great deal, but can manage without them
  - they especially help if you use stata a lot, say > 20hrs a week

macro v loop v .ado

◇ macros: use if you repeat a string over again

◇ loops: use if you do a similar thing over again

◇ .ado: use if you have a complicated task to do over again (usually involving complicated macros and loops)

intuition

- ◇ the idea with .ado files is automation
- ◇ oftentimes, we do the same thing over and over again
  - · there is a bunch of if else statements : if this do that etc...
- ◇ why not write a program for that ?
- ◇ if you do a lot of similar operations
  - · it pays off to spend some time and program it
  - · so that next time you can do it all with push of a button
  - · just say your command and specify options that are task-specific
  - · eg variables, output directory, title, etc

how do i write .adofiles

◇ don’t try to wite them at once and for all

- first write your regular dofile and then
- step by step make it into a program by making macros, loops and optimizing it