27 Commands everyone should know

Contents

27.1 41 commands

Putting aside the statistical commands that might particularly interest you, here are 41 commands that everyone should know:

Getting help	**[U] 4 Stata's help and search facilities**
`help`, `net search`, `search`	
Keeping Stata up to date	
`ado`, `net`, `update`	**[U] 28 Using the Internet to keep up to date**
`adoupdate`	[R] **adoupdate**
Operating system interface	
`pwd`, `cd`	[D] **cd**
Using and saving data from disk	
`save`	[D] **save**
`use`	[D] **use**
`append`, `merge`	**[U] 22 Combining datasets**
`compress`	[D] **compress**
Inputting data into Stata	**[U] 21 Entering and importing data**
`import`	[D] **import**
`edit`	[D] **edit**
Basic data reporting	
`describe`	[D] **describe**
`codebook`	[D] **codebook**
`list`	[D] **list**
`browse`	[D] **edit**
`count`	[D] **count**
`inspect`	[D] **inspect**
`table`	[R] **table**
`tabulate`	[R] **tabulate oneway** and [R] **tabulate twoway**
`summarize`	[R] **summarize**

Data manipulation	[U] 13 Functions and expressions
generate, replace	[D] generate
egen	[D] egen
rename	[D] rename, [D] rename group
clear	[D] clear
drop, keep	[D] drop
sort	[D] sort
encode, decode	[D] encode
order	[D] order
by	[U] 11.5 by varlist: construct
reshape	[D] reshape
Keeping track of your work	
log	[U] 15 Saving and printing output—log files
notes	[D] notes
Convenience	
display	[R] display

27.2 The by construct

If you do not understand the by *varlist*: construct, _n, and _N, and their interaction, and if you process data where observations are related, you are missing out on something. See

[U] 13.7 Explicit subscripting
[U] 11.5 by varlist: construct

Say that you have a dataset with multiple observations per person, and you want the average value of each person's blood pressure (bp) for the day. You could

```
. egen avgbp = mean(bp), by(person)
```

but you could also

```
. by person, sort: generate avgbp = sum(bp)/_N
. by person: replace avgbp = avgbp[_N]
```

Yes, typing two commands is more work than typing just one, but understanding the two-command construct is the key to generating more complicated things that no one ever thought about adding to egen.

Say that your dataset also contains time recording when each observation was made. If you want to add the total time the person is under observation (last time minus first time) to each observation, type

```
. by person (time), sort: generate ttl = time[_N]-time[1]
```

Or, suppose you want to add how long it has been since the person was last observed to each observation:

```
. by person (time), sort: generate howlong = time - time[_n-1]
```

If instead you wanted how long it would be until the next observation, type

```
. by person (time), sort: generate whennext = time[_n+1] - time
```

by *varlist*:, _n, and _N are often the solution to difficult calculations.