The Stata Journal (2008)
8, Number 4, pp. 583–585

Stata tip 69: Producing log files based on successful interactive commands

Alan R. Riley
StataCorp
College Station, TX
ariley@stata.com

So, your interactive Stata session went well and you got some good results. Naturally, you made sure you kept a log file by using the `log` command ([R] **log**).

But, almost inevitably, you also made some errors in your commands. And perhaps you also have within your log some digressions, repetitions, or things that turned out to be not so interesting or useful. How do you now produce a log file based only on the successful commands? More importantly, how do you save the sequence of commands you issued so that you can reproduce your results?

Such questions are longstanding, and there are possible software solutions on various levels. You might reach for the Stata Do-file Editor or your favorite text editor or scripting language, or you might write a program using Stata commands such as `file` ([P] **file**) to edit the log file down to the valuable part. See, for example, Cox (1994) or Eng (2007) for some detailed suggestions.

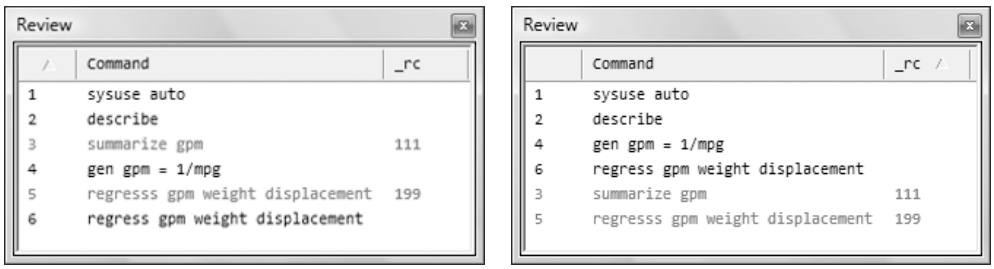
Here I concentrate on two approaches that should help.

The first approach makes a good start by using Stata's Review window, which displays a history of commands submitted to Stata. The following steps will save a do-file consisting of all the interactive commands that did not result in an error.

There are three columns in the Review window: sequence number (the order in which commands were submitted to Stata), command (the command itself), and return code (the return code, or `_rc`, from the command if it exited with an error; this column is empty if the command completed successfully).

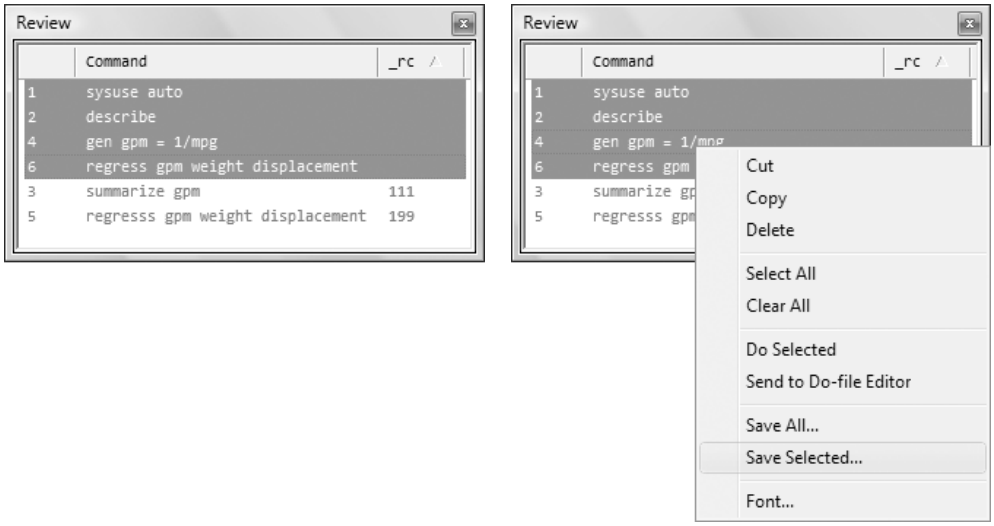
After issuing several commands to Stata interactively, some of which might have resulted in errors, click on the top of the return code (`_rc`) column in the Review window. This will sort the commands in the Review window: now all the commands that did not result in an error are grouped together (and within that group they will be in the order in which they were submitted to Stata). Beneath them will be all the commands resulting in errors, sorted by the return code. The screenshots below show what the Review window might look like before and after doing this.

© 2008 StataCorp LP



The first group is of interest; the second can be ignored. Select all the commands in the first group: click once on the first command in the group to select it, scroll until the last command in the group is visible, and then hold down the *Shift* key while clicking on that last command to select all the commands in the group.

Once all the valid commands have been selected, right-click anywhere in the Review window and choose **Save Selected...** to save those commands to a do-file, or choose **Send to Do-file Editor** to paste those commands into Stata's Do-file Editor.



Naturally, this does not solve the questions of digressions, repetitions, or less worthwhile results. Nor is there an absolute guarantee that rerunning these commands would result in exactly the same state as you ended your session. There is a chance that some of the commands that produced an error changed your data, for example, if one of your commands was to run a do-file that stopped with an error after making certain changes to your data. However, knowing this approach should make your task easier.

The second approach uses `cmdlog`. Make sure that you run both `log` and `cmdlog` simultaneously. Then, with an eye on the log, edit the command log so that it contains only commands that were legal and useful. Then run the command log as a do-file to get the error-free log. In essence, there is less work to do that way.

References

Cox, N. J. 1994. os13: Using awk and fgrep for selective extraction from Stata log files. *Stata Technical Bulletin* 19: 15–17. Reprinted in *Stata Technical Bulletin Reprints*, vol. 4, pp. 78–80. College Station, TX: Stata Press.

Eng, J. 2007. File filtering in Stata: Handling complex data formats and navigating log files efficiently. *Stata Journal* 7: 98–105.