B Advanced Stata usage

Contents

B.1 Executing commands every time Stata is started

Stata looks for the file `profile.do` when it is invoked and, if it finds it, executes the commands in it. Stata looks for `profile.do` first in the directory where Stata is installed, then in the current directory, then along your path (if invoked from the Terminal or a shell script), then in ~/Documents/Stata, and finally along the ado-path (see [P] **sysdir**). We recommend that you put `profile.do` in ~/Documents/Stata.

Say that every time you start Stata, you would like to start a dated log for the session. In ~/Documents/Stata, create the file `profile.do` containing this rather odd-looking command:

```
log using `: display %tCCCYY-NN-DD-HH-MM-SS ///
    Clock("`c(current_date)' `c(current_time)'","DMYhms")', ///
    name(default_log_file)
```

When you invoke Stata, the usual opening appears but with the following additional command, which will be executed:

```
running ~/Documents/Stata/profile.do ...
```

How does the command work? Let's work from the inside out:

- `c(current_date)` and `c(current_time)` are local system macros containing the current date and current time. See [P] **creturn** for more information.
- The left (`) and right (') quotes around the local macros expand them. See [P] **macro** for a full explanation.
- The `Clock()` function uses the resulting date string and the date mask `"DMYhms"` to create a datetime number Stata understands. See [D] **Datetime**.
- The format `%tCCCYY-NN-DD-HH-MM-SS` formats this number in year-month-day-hour-minute-second form because this will make the files sort nicely. See [D] **Datetime display formats** for the details.
- The odd-looking `: display ...` allows the formatted date to be used directly in the command as the file name. This is the advanced concept of an in-line expansion of a macro function. You can see more in [P] **macro**.
- The `log using` command starts a log file, such as shown in [GSM] **16 Saving and printing results by using logs**.
- The `name` option gives the log file the internal name `default_log_file` so that it will not likely conflict with other log files. See [R] **log** for details.
- Finally, the `///` notations are continuation comments so that the three separate lines are interpreted as a single command. See [P] **comments** for more about comments.

There are many advanced Stata programming concepts in this one single command!

profile.do is treated just as any other do-file once it is executed; results are just the same as if you had started Stata and then typed run profile.do. The only special thing about profile.do is that Stata looks for it and runs it automatically.

System administrators might also find sysprofile.do useful. This file is handled in the same way as profile.do, except that Stata first looks for sysprofile.do. If that file is found, Stata will execute any commands it contains. After that, Stata will look for profile.do and, if that file is found, execute the commands in it.

One example of how sysprofile.do might be useful would be when system administrators want to change the path to one of Stata's system directories. Here sysprofile.do could be created to contain the command

```
sysdir set SITE "~/Documents/Stata"
```

See [U] **16 Do-files** for an explanation of do-files. They are nothing more than text files containing sequences of commands for Stata to execute.

B.2 Other ways to launch Stata

You can start Stata by double-clicking on a Stata .dta dataset, a Stata .do do-file, or a Stata .gph graph file. In all cases except for do-files, your current working directory will become the folder containing the file you have double-clicked. Double-clicking a do-file will not change your current working directory unless the do-file is named Stata.do. Putting a Stata.do file into a project's directory allows you to easily launch Stata and set your current working directory to that directory. Do-files opened for editing do not change your working directory.

Stata will behave as you would expect in each case. If you double-click on a dataset, Stata will open the dataset after Stata starts. If you double-click on a graph, the graph will be opened by Stata. If you double-click on a do-file, the do-file will be opened either in the Do-file Editor, or will execute its commands.

If you would rather have Stata execute the commands in a do-file when it is double-clicked, select **Stata > Preferences > General Preferences...**, click on the **Windows** toolbar button, click on the *Do-file Editor* item, click on the **Advanced** tab, and uncheck the *Edit do-files opened from the Finder in Do-file Editor* checkbox.

B.3 Stata batch mode

To run Stata in batch mode, you need to start it in the Terminal. The syntax of the command to start Stata from in the Terminal is

$$\texttt{StataSE} \; \big[\textit{-option} \; \big[\textit{-option} \; \big[\ldots\big]\big]\big] \; \big[\textit{stata\_command}\big]$$

where the options are

Option	Result
-b	set background (batch) mode and log in plain text
-e	set background (batch) mode and log in plain text without prompting when Stata command has completed
-q	suppress logo and initialization messages
-rngstream#	set random-number generator to mt64s (see [R] **set rng**) and set random-number stream to # (see [R] **set rngstream**)
-s	set background (batch) mode and log in SMCL

For you to run Stata from the Terminal, you need to be sure that the shell can find Stata. To do this, you must add the path to the Stata executable in Stata's application bundle to your shell's path. Once that is done, you can invoke Stata from any directory from a shell.

For example, if Stata is installed in /Applications/Stata, then the path to the executable for Stata/SE is /Applications/Stata/StataSE.app/Contents/MacOS. Type StataSE to start Stata/SE.

For Stata/MP, it is /Applications/Stata/StataMP.app/Contents/MacOS. Type StataMP to start Stata/MP.

For Stata/BE, it is /Applications/Stata/Stata.app/Contents/MacOS. Type StataBE to start Stata/BE.

Suppose you had a do-file named bigjob.do. If you want to use Stata in batch mode, typing

```
% StataSE -b do bigjob
```

tells Stata to execute the commands in bigjob.do, suppress all screen output, and route the output to bigjob.log in the same directory. Stata will display a dialog when the commands have finished executing.

Typing

```
% StataSE -e do bigjob
```

tells Stata to execute the commands in bigjob.do, suppress all screen output, and route the output to bigjob.log in the same directory. Stata will simply exit without displaying a dialog when the commands have finished executing.

```
% StataSE -s do bigjob
```

tells Stata to execute the commands in bigjob.do, suppress all screen output, and route the output to bigjob.smcl in the same directory.

Specifying −s −e sets the background (batch) mode and logs in SMCL without prompting when a Stata command has completed.

You can also run the above examples in the background by typing

```
% StataSE -b do bigjob &
% StataSE -e do bigjob &
% StataSE -s do bigjob &
```

Note: Stata runs profile.do before doing bigjob.do, just as it would if you were working interactively.

General notes

While your do-file is executing, the Stata icon will appear on the dock.

The icon has a badge showing a rough percentage of how much of the do-file Stata has executed. (Stata calculates this percentage from the number of characters in the do-file, so the percentage may not accurately reflect the amount of time left for the job to complete.)

If you right-click on the icon on the taskbar, Stata will display a box asking if you want to cancel the batch job.

Once the do-file is complete, Stata will bounce the icon on the dock. You can then click on the icon to close Stata. If you wish for Stata to automatically exit after running the batch do-file, use -e rather than -b.

You do not have to run large do-files in batch mode. Any do-file that you run in batch mode can also be run interactively. Simply start Stata, type log using *filename*, and type do *filename*. You can then watch the do-file run, or you can minimize Stata while the do-file is running.

B.4 Changing Stata's locale

To change the locale of Stata to English, type

```
set locale_ui en
```

To change it back to match the locale set for your operating system, type

```
set locale_ui default
```

For a complete explanation of locales and Stata, see [U] **12.4.2.4 Locales in Unicode**.

B.5 More

If you would like Stata to pause every time the screen fills with results, type `set more on`. This will cause a —more— prompt to appear at the bottom of the Results window whenever there is more information to be displayed than can fit on the screen. This happens, for example, when you are `listing` many observations.

```
. list make mpg

        make                mpg

  1.    Linc. Continental    12
  2.    Linc. Mark V         12
  3.    Cad. Deville         14
  4.    Cad. Eldorado        14
  5.    Linc. Versailles     14

  6.    Merc. Cougar         14
  7.    Merc. XR-7           14
  8.    Peugeot 604          14
  9.    Buick Electra        15
 10.    Merc. Marquis        15

 11.    Buick Riviera        16
 12.    Chev. Impala         16
 13.    Dodge Magnum         16
 14.    Olds Toronado        16
 15.    AMC Pacer            17

 16.    Audi 5000            17
 17.    Dodge St. Regis      17
 18.    Volvo 260            17
 19.    Buick LeSabre        18
 20.    Dodge Diplomat       18

 —more—
```

If you want to see the next screen of text, you have a few options: press any key, such as the Spacebar; click on the **More** button, ⬇ ; or click on the —more— link at the bottom of the Results window. To see just the next line of text, press *Return*. Pressing `q` will interrupt the command. If you click and hold the **More** button, you can also select the **Run to completion** menu item to let the command completely finish.

B.6 Memory size considerations

Memory management in Stata is automatic. For details on efficiency tweaks needed by a very few Stata users, look at [D] **memory**.

For suggested citations, see the FAQ on citing Stata documentation.