

B Advanced Stata usage

Contents

- B.1 Executing commands every time Stata is started
- B.2 Other ways to launch Stata
- B.3 Stata batch mode
- B.4 Memory size considerations

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, and finally along the `ado-path` (see [P] [sysdir](#)). We recommend that you put `profile.do` in `/Users/yourhome/Library/Application Support/Stata`.

Say that every time you start Stata, you would like to start a dated log for the session. In `/Users/yourhome/Library/Application Support/Stata`, create the file `profile.do` containing this rather odd-looking command (the command will be explained at the end of this session):

```
log using `: display %tCCYY-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 ~/Library/Application Support/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 `%tCCYY-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 an extended 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 "/Library/Application Support/Stata"
```

See [U] 16 Do-files for an explanation of do-files. They are nothing more than ASCII 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. 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 **Do-file Editor** toolbar button, 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

```
StataSE [-option [-option [...]]] [stata_command]
```

where the options are

Option	Result
-b	set background (batch) mode and log in ASCII text
-e	set background (batch) mode and log in ASCII text without prompting when Stata command has completed
-q	suppress logo and initialization messages
-s	set background (batch) mode and log in SMCL

The `-q` option starts Stata, but suppresses all the initialization messages, including the Stata logo.

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/IC, it is `/Applications/Stata/Stata.app/Contents/MacOS`. Type `Stata` to start Stata/IC.

For Small Stata, it is `/Applications/Stata/smStata.app/Contents/MacOS`. Type `smStata` to start Small Stata.

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.

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.

B.4 Memory size considerations

Beginning with Stata 12, memory management in Stata became automatic—there is no longer any requirement to manually request memory for Stata when you know you will be using a large dataset. For details on efficiency tweaks needed by a very few Stata users, look at [\[D\] memory](#).