B   Advanced Stata usage

Contents

1.1   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 your bin directory `$HOME/bin`.

Say that every time you start Stata, you would like to start a dated log for the session. In `$HOME/bin`, 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 /home/mydir/bin/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 date-time 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 [GSU] **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 "/opt/stata/ado"
```

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.

1.2 Advanced starting of Stata for Unix

The syntax of the command to start Stata(GUI) is

$$\texttt{xstata} \, \big[\textit{-option} \, \big[\textit{-option} \, [\ldots]\big]\big] \, \big[\textit{stata\_command}\big]$$

The syntax of the command to start Stata(console) is

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

If you have Stata/SE, the commands are xstata-se and stata-se. If you have Stata/MP, the commands are xstata-mp and stata-mp.

The allowable options are

Option	Result
-b	set background (batch) mode and log in plain text (console only)
-h	display usage diagram
-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 (console only)

Typing stata -h does not start Stata, but just shows the syntax diagram for invoking Stata.

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

The -b and -s options specify batch mode; see [GSU] **B.3 Stata batch mode**.

1.3   Stata batch mode

Suppose you had a do-file named `bigjob.do`. If you want to use Stata in batch mode, we suggest using Stata(console). Typing

```
% stata -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 -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

```
% stata -b do bigjob &
% stata -s do bigjob &
```

You may also use redirection, but this is not recommended:

```
% stata < bigjob.do > bigjob.log &
```

Warning: Redirection will not work if your do-file contains either the `#delimit` commands or comment delimiters (`/*` and `*/`, `//`, or `///`). It also cannot create SMCL output. Hence, we recommend using options directly: `stata -s do bigjob &` or `stata -b do bigjob &`.

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

1.4   Using X Windows remotely

Suppose that you are sitting in front of a computer named `local` and that you wish to run Stata on a computer named `neighbor`.

1. Tell X Windows on `local` to allow `neighbor` to use its display by typing `xhost +neighbor`.

2. Be sure that you have set X Windows's `DISPLAY` environment variable on `neighbor` to contain `local:0.0`. X requires this. Important: this variable must be set on `neighbor`.

Having done this, Stata should work:

```
local% xhost +neighbor
local% ssh neighbor
neighbor% setenv DISPLAY local:0.0
neighbor% xstata
```

At this point, either Stata launches, or you see

```
Xlib:  connection to "local:0.0" refused by server
Xlib:  Client is not authorized to connect to Server
```

Here you will have to get help from your network administrator. Proper authorizations have not been given, and these problems have nothing to do with Stata.

To make this process simpler, you may also be able to use the `-X` flag when invoking `ssh`:

```
local% ssh -X neighbor
neighbor% xstata
```

Whether this works depends on your Unix installations. The first series of commands should always work.

1.5   Summary of environment variables

Environment variable	Description
HOME	User's home directory. Default is the directory specified in /etc/passwd.
PATH	Unix executable search path.
SHELL	What to execute when users try to shell out of Stata. Default is /bin/sh.
S_ADO	Sets Stata's ado-path.
STATATERM	Used only if you want to use a different termcap or terminfo entry from what is in your TERM environment variable.
STATATMP	Sets Stata's temporary directory. Default is /tmp.
STATA_PREF_DIR	Location Stata checks for preferences. If not set, Stata will use a subdirectory of $HOME/.stata19/. If you use Stata(console), preferences will be saved in $HOME/.stata19/console/. If you use Stata(GUI), preferences will stored in a subdirectory based on your DISPLAY environment variable, such as $HOME/.stata19/:0.0/.

1.6   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**.

1.7   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 *Enter*. Pressing q will interrupt the command. If you click on the arrow of the **More** button, you can also select the **Run to completion** menu item to let the command completely finish.

1.8   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.