

Basics of Stata

Statistics 8

Last Updated: September 20, 2008 (Stata 10)

1 Description of a Stata Session

1.1 The Toolbar

The “toolbar” refers to the row of icons below the list of menus. You can hold your cursor over an icon to see which it is. Here are some of the buttons on the toolbar:

- The **Open File** button allows you to open a `.dta` data file.
- The **Save File** button lets you to save your data as a `.dta` data file.
- The **Log** button lets you record all your Stata commands and their results (except graphs) as you type them. This file will contain everything, from your mistakes, typos and ... Therefore you are required to edit it before turning it in! The last resort is to cut and paste through the file using a scissor. *IMPORTANT: If you use the Log button to begin your log, be sure to select “.log” as the filetype — this will save it as a text file that will be easy to edit! The alternative is in Stata’s markup and control language (SMCL) which I don’t know anything about!*
- The **Viewer** button allows you to view (but not edit) a file, a help file, or a URL. When the view window opens you can type in a topic for help or type “browse” followed by a website.
- The **Graph** button will bring the graphic window upfront, which will contain the outputs from your graphic commands.
- The **Editor** button will bring a spreadsheet editor containing your data and lets you make changes in it. This is a very nice way to correct a typo in your dataset and also to assign names to variables.
- The **Browse** button will bring the same spreadsheet as under the Editor button but you can now only browse through it and won’t be able to record any changes made in it. This is the safe way to look at your data and not run the risk of destroying it.
- The **Break button** lets you interrupt a Stata command that takes a long time and that you are no longer interested in.

1.2 The Windows

Stata operates with a number of windows. The big window with black background is the **Results window**. All the results of your Stata commands (except graphs which are shown in their own separate windows) appear here. You enter commands in the **Command window**. The **Review window** provides a record of all the Stata commands you enter. You can repeat a previous command by double-clicking the command in the Review window. The **Variables Window** shows a record of all the variables in the data set that you are currently using. You can navigate among the windows with your cursor or by using the Windows menu item (or by using relevant control key as indicated in the Windows menu item). Also, you can close the **Command** and **Variable** windows if you want the **Results** window to be larger. Get them back using the Windows menu item.

2 Structure of a Stata command

You can issue Stata commands in two distinct ways. The Data, Graphics, and Statistics menus provide a menu/command box interface. Choose a command and then fill in the command box. An alternative is to type commands in the **Command** window. The form of all commands is

```
command variables, options
```

or

```
command variable1 variable2 ..., option1 option2 ...
```

where `command` is the name of the command you want to execute, `options` (each option separated by a space) tells Stata how you want to execute the command, and `variables` (each variable name is separated by a space) are the list of variables to use in performing this command. (Note some command names are composed of more than one word.) Note that the menu/command box approach actually will show you the relevant command in the **Review** window. You may find (I have) that it is easier to type in the commands once you know the relevant command.

Sometimes commands have options which you can specify on the same line as the original command, using a comma to separate the command from the options. For example,

```
histogram x1, bin(10)
```

specifies that the histogram should be composed with 10 bins.

End all commands by hitting the “Return” key. If a command goes beyond one line, keep typing — the display will continue into the next line, and the command will not execute until you hit “Return”.

If you need to stop the execution of Stata command hit the the control key and ‘C’ or use the break button on the toolbar.

3 Online Help

To get help within Stata type

```
help
```

at the command line. This will give you a list of topics for which help is available. If you know you need help with a specific command you can also type

```
help commandname
```

These commands will open a viewer window with the relevant information. Often real examples of how to use the command are provided at the end of the help file.

If you do not know the exact name of commands, you can type

```
search commandname
```

This will produce a list of possibly relevant commands from which you may identify the correct command name. Alternatively, you can use the pull down help menu. Use “Search...” if you do not know the name of the command and “Stata Command...” if you do know the name of the command.

4 Log File

Stata can store your commands and their results in a log file. At the start of every Stata session, you should start a log file. You can do this from the Toolbar or using the command:

```
log using filename, text
```

where you choose a name for *filename*. The log file often contains a lot of unwanted output and should not be submitted without editing. *It is important that you use the text option. If you do not, it will be difficult to edit your log.*

You can use `log off` and `log on` to suspend and resume your log. The command `log close` will permanently end the log.

5 Exiting Stata

The smart way to exit Stata is to pull down the “File” menu and select “Exit”. Sometimes when you do this and the data in memory has not yet been saved Stata will not let you exit. In this case, a dialog box will appear giving you the option to “Exit Anyway” or “Don’t Exit.”

To exit Stata from the command line type `exit`. Sometimes Stata won’t let you exit and gives you a message “no;data in memory would be lost”. This message warns you that the data currently in memory has not been saved. To exit you have two choices:

1. Exit without saving the data by typing

```
exit, clear
```

2. Save the data and then exit by typing, for example,

```
save yourdata  
exit
```

6 Getting Data into Stata

6.1 Using Stata-Format Datasets

To use a dataset that has been saved in Stata format use the File menu (select Open), the Open File icon on the Toolbar or the “use” command. Stata-format datasets are usually recognizable by the “.dta” in the filename (e.g. wind.dta). To use a dataset called “wind.dta”, type

```
use wind
```

Note the use command is looking for data files in the Stata directory so it may be easier to just the the Open File icon or the File menu.

If you already have a dataset in memory, Stata will not let you load a new dataset until you type

```
clear
```

to clear out the previous dataset.

Stata-format datasets typically come with the variables already labeled. To see the labels, use the describe command.

6.2 Typing in a New Dataset

Click on the “Editor” button on the toolbar. This will bring up a window that has rows and columns where you can enter your data. Each column represents a variable. You can enter both numeric or character data directly. You can also modify the data (e.g., correct typographical error) using this window. To label variables after entering data, double-click on the box of the spreadsheet immediately above the column containing the data. When you are done, click on “OK” (or on “Cancel” to abandon your changes).

If you want to suffer, you can also do this using the input command to type raw data into Stata. Use the input command followed by a list of names (up to eight characters each) for the variables you want to enter. For example

```
clear (needed only if you are using a Mac and already have data in memory)
input dsun radius rings str7 planet
1.  57.9 2439 0 Mercury
2.  108.2 6050 0 Venus
3.  149.6 6378 0 Earth
4.  end
```

After the input command, Stata prompts with case numbers “1.”, “2.”, and so on. Input columns need not line up, but there must be at least one space between values. Type end to stop data entry and return to Stata’s “.” prompt. Notice that for the non-numeric variable planet it is necessary to put str7 in the input command immediately before planet. The “str7” notifies Stata that the next variable will be a non-numeric (or string) variable and the maximum number of letters in these names is 7 (you can use up to str80). If there are missing values, you may enter a blank; for numeric variables, Stata will record a “.”.

If you made a mistake in entering the data you use the **Editor** window and change the value wrongly entered or the “replace” command. For example, to change the value of radius in the second observation type

```
replace radius = 7050 in 2
```

When using the replace or input commands to enter missing values, you must use a period, ., for numeric variables or a pair of double quotes, “”, for string variables. For example,

```
replace radius = . in 2
replace planet = "" in 3
```

For more sophisticated file operations use the help function to get information on the append and merge commands.

Note, successive input commands will keep adding data to the data already in memory. For example, the command input x followed by a later command input y will result in both x and y being stored in memory. To delete some or all variables from memory use the drop command. See help drop for more information.

6.3 Using Non-Stata-Format Datasets

The only Non-Stata-Format datasets that Stata can read are plain text files. These files are called ASCII files. If you are using a text editing program to type in you dataset, remember to save it in an ASCII format (text format). This may not be the default format. For example if you use Microsoft Word and do not specify the ASCII format (text only format), the saved file will be in a Microsoft Word format which Stata cannot read.

6.3.1 With the dictionary

The dictionary command tells Stata what the variables in the dataset are and can also contain the names and labels of these variables. A dictionary dataset must have the “.dct” extension to be recognized by Stata. If you are not sure whether a file has a dictionary or not, just look at it with any word processor. If the file starts with `dictionary`, then it is a dictionary file, otherwise it isn't. This is what the top of a `.dct` file should look like.

```
dictionary {
    str14  Name
    float  Populatn          "Population"
    float  Income            "Income mean"
    float  Illitrcy          "Illiteracy Rate"
    float  Life.Exp          "Life Expectancy"
    float  Murder            "Murder Rate per 100,000"
}
"Alabama"      3615          3624          2.1          69.05          15.1
"Alaska"       365           6315          1.5          69.31          11.3
.
.
.
```

On each line between the curly brackets, the first column indicates the variable type (`str14`, `float`, ...), the second column is the name of the variable (at most 8 characters), and the final column is a label attached to the variable. The label is very useful since it is used when you plot or graph the variable. Finally, each field has to be separated by at least a space (in the example above, we used a “tab”).

You can input `.dct` files into Stata using the Import option under the File menu or the `infile` command. Using the File menu choose ASCII text with a dictionary and then browse to identify the file you want (the `.dct` file). You can use the `infile` command but need to refer appropriately to the file, e.g., as “H:mydata.dct”. For example

```
infile using `H:mydata.dct'`
```

6.3.2 Without the dictionary

Stata can also read ASCII (or text-only) data files *without dictionaries* using the Import option on the File menu or the `infile` command. Using the Import option will require you to enter a file name and a format. Using the `infile` command you would read a file called “reactor.dat” which has five variables (`site`, `capacity`, `decom`, `start`, `close`, where `site` is a string variable with a maximum of 30 characters and the others are numeric) using the following command:

```
infile str30 site capacity decom start close using reactor.dat
```

Notice that “`str30`” precedes “`site`” to notify Stata that `site` is a string (not-numeric) variable with a maximum of 30 characters. In the ASCII file, missing numeric values should be indicated by a period, `.`, and missing string values by a double quote, “”.

Because this is not a Stata-format dataset, it will not come with attached labels. You have to add labels by following the instructions given in the previous section. (I.e., use the data editor.)

7 Saving Data

To save your data and any attached labels as a Stata-formatted file with the Save option on the File menu or by typing

```
.save filename
```

To use this data later, use the `use` command (that is, use `filename`) or simply double-click on the icon for the saved dataset to reopen it.

To save your data in ASCII format (e.g. in case you want to read it into another statistics package or a text editor) use the Export option from the File menu or the `outfile` command. By default, any file saved using the `outfile` command is given the file extension `.raw`.

8 Saving, Editing and Printing your work

8.1 With a Word Processor Running at the Same Time as Stata

If you have enough RAM memory or you are able to use virtual memory, you should run Stata and a word processor at the same time while you are doing your homework. Doing so will save you time especially when you edit Stata output for your homework.

Having both programs running at the same time allows you to copy output from any Stata window (Select “Copy” from the “Edit” menu in Stata) and paste it in the word processor (Select “Paste” from the “Edit” menu of your word processor.) You can also copy and paste Stata graphs. This will save you from having to print each graph and then cut and paste them into your homework using scissors and tape. Once you have pasted a graph into your homework document, some word processor (like Microsoft Word) will let you edit the graph by double-clicking on them. *An important reminder is that what you want to copy must be in the active window on your desktop.*

Finally, in order to preserve the spacing and the look of Stata tables you have to select the **Courier** font in your word processor. That is, pull down the “Font” menu and select “Courier”. If you keep cutting your work from Stata and pasting it into your word processor, your “report” will almost be done by the time you are done with Stata. To get a hard copy of your report, you can now print it using your word processor.

8.2 When You Cannot Run a Word Processor and Stata at the Same Time

8.2.1 Saving and printing output (except graphs)

You can save a transcript of all text input and output that appears on your screen during your Stata session (**no graphics are saved in the log file**), by using the `log using` command as described earlier. For example, the command

```
log using myfile, text
```

begins a log file called *myfile.log*, that will contain a transcript of all text input and output appearing on your screen until you interrupt the log with `log off`, terminate the log with `log close`, or leave Stata. If you interrupt your log file with `log off` you can turn it back on with `log on`. (You can also begin the log file by clicking on the Log button on the toolbar and selecting “Begin”. In the dialog box that appears, be sure to select a .log file.)

It is suggested that you use your name (first or last) in the title *mylogfile*. The title you give your log file will show up on the output. If you make the title unique, you will have an easier time finding your work when you print it.

Your log file will be saved as an ASCII (or text-only) file. You can edit this file using any word processor. To print the log file, double-click on the log file’s icon. (You may have to select a word processor, e.g., TextEdit or Word.) You can then view, edit, and print the log file using the word processor.

Your log file will be easiest to read if you change the font to **Courier** in your word processor. That is, pull down the “Edit” menu and choose “Select All”. Next go to the “Font” menu and select “Courier”. (Then click the mouse button to un-highlight the text. If you accidentally delete what you see on the screen, you can quit without saving your changes!)

Starting a `log` file is the **first** thing you should do every time you use Stata. You can always cut and paste later (with a word processor, or with scissors and tape) to extract and organize the important results.

The `log` commands will save a record of your Stata commands and their text output. *They will not save graphs and they will not save your data.* To save your data, follow the instructions in the previous section. To save (and print) graphs, follow the instructions in the next two sections.

8.2.2 Saving and printing graphs

You can use the “Save graph” option under the “File” menu to save a graph. For example, first make a graph (here a histogram of the variable called radius),

```
histogram radius
```

this should open a new window with your graph in it, now Select “Save graph” under the “File” menu. You can choose a name and graphic type (postscript or other) for your graph and the location where you would like it to be saved.

Alternatively, you can use the `saving` option as described in the homework.

```
histogram radius, saving(radplot)
```

To print a previously saved graph, double click on the graph file’s icon and select the “File” pull-down menu and select “Print”. You can also print a graph directly, without saving it, by selecting the “Print graph” option under the “File” menu in Stata.

To redisplay a previously saved graph you can just double-click on its icon or, if you are already in Stata, type, for example,

graph use mygraph

9 Command Summary

Here are some commands that you may find useful (this is by no means an exhaustive list of all Stata commands):

anova	general ANOVA, ANCOVA, or regression
by	repeat operation for categories of a variable
ci	confidence intervals for means
clear	clears previous dataset out of memory
correlate	correlation between variables
describe	briefly describes the data (# of obs, variable names, etc.)
diagplots	distribution diagnostic plots
drop	eliminate variables from memory
edit	an alternative to <code>input</code>
exit	leave Stata
generate	creates new variables (e.g. <code>generate years = close - start</code>)
graph	general graphing command (this command has many variations)
graph box	graphs boxplots
graph twoway scatter	graphs a scatter plot
graph twoway connected	graphs a scatter plot, connecting the dots
help	online help
histogram	graphs histograms
if	lets you select a subset of observations (e.g. <code>list if radius >= 3000</code>)
infile	read non-Stata-format dataset (ASCII or text file)
input	type in raw data
list	lists the whole dataset in memory (you can also list only certain variables)
log	save or print Stata output (except graphs)
oneway	oneway analysis of variance
pcorr	partial correlation coefficients
plot	text-mode (crude) scatter plots
predict	calculated predicted values (\hat{y}), residuals (ordinary, standardized and studentized), leverages, Cook's distance, standard error of predicted individual y , standard error of predicted mean y , standard error of residual from regression
qnorm	graphs a normal quantile plot
search	keyword search of commands, often precursor to <code>help</code>
regress	regression
replace	lets you change individual values of a variable
save	saves data and labels in a Stata-format dataset
serrbar	standard error-bar chart
sort	sorts observations from smallest to largest
stem	stem and leaf display
summarize	produces summary statistics (# obs, mean, sd, min, max) (has a <code>detail</code> option)
test	conducts various hypothesis tests (refers back to most recent model fit (e.g. <code>regress</code> or <code>anova</code>) (see <code>help</code> function for info and examples))
ttest	one and two-sample t-tests
use	retrieve previously saved Stata dataset