

## Stata Basics

This handout summarizes some basics regarding file manipulation and command syntax for use in the ‘Stata Command’ window. The easiest way to learn Stata syntax is using the pull-down menus with the mouse. Using these menus generates reusable syntax in your ‘Review’ window.

### Windows in the Stata Console

There are eight windows that can be displayed on the Stata console. Commands for switching between windows are found under the ‘Window’ menu. The windows can be sized and placed in any orientation by clicking-and-dragging on the edge of a window.

For syntax-generating windows (‘Statistics’, ‘Graphics’, etc.), clicking ‘OK’ will execute the command and close the window; clicking ‘Submit’ will execute the command and leave the window open for further manipulation.

### Do-Files

Stata provides a mechanism for scripting analyses through the use of “do-files.” These do-files are useful for replicating analyses, avoiding the storage of huge results files, and assuring that analyses follow an appropriate series of steps.

The do-file editor (the menu icon showing pencil writing on white paper) is used to create and/or edit existing do-files, submit series of commands for immediate evaluation, and saving do-files for later use.

### Loading Datasets from Outside Sources

To load non-Stata data from an external file (*e.g.* Excel spreadsheet, text file, etc.), use the ‘Import’ feature under the ‘File’ menu.

### Loading and Saving Stata Datasets (use, save)

To load a Stata dataset:	<code>use filename</code>
where <i>filename</i> = the complete path of the dataset	
To save a dataset:	<code>save filename</code>
To overwrite the original dataset:	<code>save, replace</code>
Or, you can save it under a different filename:	<code>save newfilename</code>

### Describing and Exploring Data (describe, codebook, summarize, tabstat, table)

The data editor (under the ‘Data’ menu) provides a familiar spreadsheet-like format for viewing and manipulating data by hand.

To list contents of dataset:	<code>describe</code>
To list storage type, display format and labels of particular variables:	<code>describe varname(s)</code>
To list name, labels, frequency distribution, and summary information for all variables:	<code>codebook</code>
To list codebook information for particular variable(s):	<code>codebook varname(s)</code>
To list univariate summary statistics for all variables:	<code>summarize</code>
To list univariate statistics for particular variable(s):	<code>summarize varname(s)</code>
To create tables of summary statistics for particular variable(s):	<code>tabstat varname(s), options</code>

To create contingency tables of statistics for particular variable(s): `table varname(s)`

### Listing Data (list)

To list the data in memory: `list`

To list any number of variables: `list varname(s)`

To list all variables starting with *var*: `list var*`

To list any observation: `list in observation#`

To list observations 1 through 5: `list in 1/5`

To list particular variables in particular observations: `list varA varB in 1/5`

To conditionally list observations: `list varA if varA > 20`

### Saving Output (log)

To save output to a file: `log using filename`

To stop saving output to a file: `log close`

After closing, to continue saving output to original log file: `log using filename, append`

After closing, to continue saving output to a new log file: `log using newfilename`

### Command Syntax

`[by varlist] command [varlist] [=exp] [in range] [weight] [,options]`

'By *varlist*' causes Stata to repeat a command for each subset of the data for which values of the variables in the *varlist* are equal.

If no *varlist* is included, Stata assumes a *varlist* of `_all`, Stata shorthand for including all variables in the dataset.

If *exp* restricts the scope of a command to those observations for which the *exp* is true.

In *range* restricts the scope of a command to a specific observation range.

*Weight* indicates the weight to be attached to each observation.

Many commands have options and '*options*' is the method for invoking these command-specific options.

### Operators

<u>Arithmetic</u>	<u>Logical</u>	<u>Relational (numeric and string)</u>
+ addition	~ not	> greater than
- subtraction	or	< less than
* multiplication	& and	>= > or equal
/ division		<= < or equal
^ power		== equal
+ string concatenation		~= not equal

### Help

To get help on a particular command: `help commandname`

To get help on the help command: `help`

To get a complete list of help topics: `help contents`

To search help and other documents for a particular topic: `search topic`

To obtain all references to a topic, both on-line and in manual:      lookup *topic*  
To learn all about lookup:      help lookup

### Tutorials

To run a Stata tutorial on a particular topic:      tutorial *filename*  
For a complete listing of Stata tutorials:      tutorial contents  
(Examples: intro, tables, regress, logit, survival, ado)

### Entering Commands Quickly and Fixing Mistakes

	<u>Windows/Mac</u>	<u>Unix</u>
To retrieve the previously typed command line:	<i>Page Up</i>	<i>Ctrl-R</i>
To step forward through previous command lines:	<i>Page Down</i>	<i>Ctrl-B</i>
To view a listing of the last 5 command lines:	#review	#review
To see the last <i>n</i> command lines:	#review <i>n</i>	# review <i>n</i>
To review the <i>n</i> <sup>th</sup> previous command line:	<i>n Ctrl-Home</i>	<i>n Ctrl-W</i>

### Editing at the command Line

Use the left and right cursor arrows to move the cursor.

To move the cursor to the beginning of the line:	<i>Home</i>	<i>Ctrl-K</i>
To move the cursor to the end of the line:	<i>End</i>	<i>Ctrl-P</i>
To delete the character at the cursor:	<i>Delete</i>	<i>Ctrl-D</i>
To insert characters before the cursor without overwriting:	<i>Insert</i>	<i>Ctrl-E</i>
To toggle back to overwrite mode:	<i>Insert</i>	<i>Ctrl-E</i>
To delete a line and start over:	<i>Esc</i>	<i>Ctrl-U</i>
To hack off the line at the cursor:	<i>Ctrl-End</i>	<i>Ctrl-X</i>