An introduction to Stata: Part I

Thomas Lumley

October 3, 2003

Stata is installed on computers in the Health Sciences Microcomputer Lab, and is available at a substantial discount to students (and faculty and staff). It will run on any computer you are likely to have.

Until version 8, Stata did not have a well-developed menu system. It was still relatively easy to learn because it has very consistent syntax and unusually good manuals and online help. In version 8 it also has a menu system. As far as we know, Stata is the easiest to use package that can support all the analyses we need in BIOST 517–18–36–37–40. It is widely used in teaching biostatistics, and recently supplanted SPSS as the most-cited statistical package in the *British Medical Journal*. It will still take some effort to learn, especially if it is the first statistical package you have used extensively.

If you don't like my introduction to Stata you will find others linked from the web page, including the one being used in BIOST 511, and a large set of resources at UCLA.

How Stata thinks: At any given time, Stata knows about one data set, in which every *variable* has the same number of *records*. The variables are identified by names, and the records by numbers. Most mathematical or statistical operations work on all the records at once. Stata can temporarily replace this entire data set by smaller pieces, so that it is looking at all the records for men, or all the records for Seattle, or all the records for a particular person.

Stata is designed for interactive use: you type commands into the Command[1] window and they are performed immediately. You can also place a set of commands into a file and run the file. Commands you have typed are stored in the Review window, and a list of the current variables is in the `Variables` window. The PageUp and PageDown keys allow you to recall previous commands, or you can click on them in the Review window

Because Stata runs interactively it is very important to be able to store your commands and output for future perusal and editing. The `log` command (or the File | Log menu item) starts a log that stores everything Stata prints from that point on. In recent versions of Stata it is by default

[1]Notation: This font identifies window, menu and dialog items, so that File | Save means the Save item on the File menu. `This font` identifies things that you type or that Stata prints.

stored in a marked-up format vaguely reminiscent of HTML, and the `translate` command (or File | Translate) will turn it into plain text[2]. To save just the things you type you can click on the upper left corner of the Review window or use the `cmdlog` command. For example

```
log using ozone-sep29
```

will save commands to `ozone-sep29.smcl` in the Stata SMCL format. When you are done, type `log close` to close the log.

Precooked data: Stata has its own format for data sets, which changes with each version. It can read older versions but (obviously) not newer ones. These data sets usually have the extension *.dta*. To load the file `ozone.dta` from the current working directory type

```
use ozone.dta
```

If the file is not in the current working directory you can either change the directory (`cd` or File | Set Working Folder) or specify the full path to the file

```
use C:/TEACHING/517/ozone.dta
```

or use File | Open for a GUI version. You can even specify a web address for the file

```
use http://courses.washington.edu/b517/data/ozone.dta
```

if you have an internet connection. You can create *.dta* files with the `save` command (File | Save As), and this is the usual way to save data sets after you have imported, created or modified them. To save a newly created file[3] do

```
save ozone.dta
```

and to overwrite an existing one

```
save ozone.dta, replace
```

Note that I have not used quotation marks around file names. Quotation marks are permitted, but are needed only when the file name contains spaces.

[2] If you prefer plain text log files type `set logtype text, permanently` to change the default format

[3] If you are using Stata 8 and you want the file to be readable in Stata 7, use `saveold ozone.dta, intercooled`. This isn't very well documented

Raw data: I will always provide data in plain text files (and perhaps additionally in *.dta* files). There are two simple cases for reading such data into Stata. If the first line of the file contains variable names and values are separated by commas (or some other predictable character) then `insheet` will do all the work:

```
insheet using http://courses.washington.edu/b517/data/ozone.csv
```

If the first line does not contain variable names you must give the variable names. The simplest format is one where values are separated by one or more spaces. This can be read with

```
infile ozone solarr wind temp month day using ozone.txt
```

I will use the *.csv* extension for comma-separated data files and *.txt* for space-separated ones.

Labelling Stata 7 and 8 allow variable names to be 31 characters long, but if you need more space you can add a descriptive variable label

```
label variable ozone "Ozone concentration (ppb) from 1pm to 4pm at Roosevelt Island"
```

The label will appear in the Variables window and will sometimes be used by Stata to label output.

Some or all individual values can also be labelled

```
label define monthname 5 "May" 6 "June" 7 "July" 8 "August"  9 "September"
label values month monthname
```

If you are reading in a file that already has category names instead of numbers and you are using `infile`, you can automatically generate these labels

```
infile age:agegp tobacco:tobgp alcohol:alcgp ncases ncontrols using esoph.txt, automatic
```

The name before the column is the name of the variable, the name after the colon is the name of the set of labels (like `monthname`) in the previous example.

Making new variables The `generate` command makes a new variable and `replace` changes values in an existing variable[4].

[4]Most Stata commands can be abbreviated drastically, so you can write `gen` for `generate`. Destructive commands like `replace` typically can't be abbreviated

```
gen high_ozone =  ozone > 80
gen logozone = log(ozone)
gen icky = temp > 90 & wind < 8
gen summer = month == 7 | month == 8 | (month == 9 & day < 23) | (month== 6 & day>21)
```

Note that comparing for equality uses `==`.

If you want to get rid of a variable[5] or some records, use `drop`

```
drop summer high_ozone
drop if ozone == .
```

removes the variables `summer` and `high_ozone` and all the observations for which `ozone` is missing.

To get rid of all the variables, type `clear`.

Missing data: Stata depicts missing data with `.`, but represents it internally as the largest possible number. This often causes problems: the condition `ozone > 80` is true if ozone is missing, and so we actually should have written:

```
gen high_ozone  = ozone >80 & ozone != .
```

To fix our incorrectly created `high_ozone` we could do

```
replace high_ozone = . if ozone == .
```

This introduces another Stata feature: most commands can have an `if` condition added (at the end of the command if the command has no comma, before the comma otherwise) and the command is performed only on the group of records where the condition is true.

Describing your data. `describe` gives structural information about your data: names, labels, types of data. `summarize` gives some basic statistical description: number of non-missing observations, mean, standard deviation, minimum, maximum. Like many Stata commands, `summarize` can be prefixed by `by:` to give summaries for subgroups. This requires the data to be sorted by the subgroup variable

```
sort month
by month: summarize
* Sufficiently recent versions of Stata have bysort, which doesn't need a separate sort
bysort month: summarize
```

[5]For example, if you want to `generate` it again

Note the use of `*` to introduce a comment, which is not interpreted by Stata but simply sent to the output. More summaries can be created with the `detail` option[6] to `summarize`

```
bysort month: summarize, detail
```

We might want to summarize just a few variables

```
bysort month: summarize ozone-temp high_ozone
```

which asks for `ozone`, `temp` and all the variables between them, and `high_ozone`.

Other commands for summaries include `table` and `tabulate`

```
table month, contents(mean ozone mean temp)
tabulate month high_ozone
```

Note that missing values are silently dropped from these tables. You can prevent this with the `missing` option.

Pretty pictures: Before Stata 8 the graphics were fairly limited but still usable.

```
* these are commands for Stata 7 and earlier
* scatterplot
graph temp ozone
* histogram
graph ozone
* box plots
graph ozone, box by(month)
* smoothed scatterplot
lowess temp ozone
```

The easiest way to learn graphics for Stata 8 is with the menu: eg the scatterplot command above is now

```
twoway (scatter ozone temp)
```

The new graphs are prettier, but much slower.

[6] things that go after a comma are called *options*

Doing and redoing: As I mentioned earlier, you can give Stata a whole file of commands to do. Even if you like to work interactively it is a good idea to save up the commands you type and create a single file that can recreate your analysis. If you don't do this you will eventually regret it when new data arrive or when a journal reviewer asks for changes months after your initial analysis. I usually create these `do` files by editing log files.

A file of Stata commands should be a plain text file. If you aren't sure whether a file is plain text under Windows you can look at it in Notepad to check — if you can read it, it's plain text. The file can have any name, but the recommendation is to use a name ending in *.do*. The Stata command `do` (File | Do) will run all the commands in your file. For example, a file `makeozone.do` that creates our New York ozone data can be run with

```
do makeozone
```

Finding help: In addition to me and the TAs, you can try the Stata help system and manuals. There are two commands: `help` asks for help on a specific Stata command, and `search` asks for help on a keyword or topic (Help | Stata command and Help | Search). Some of the help topics refer to commands that are not built in to Stata but can be downloaded, or to entries in the Frequently Asked Questions list from the Stata mailing list. It is worth reading through the entire "Getting Started" guide (on reserve in the library).

People who are planning to use Stata for research and want intensive instruction might consider the NetCourses offered by Stata Corporation (`http://www.stata.com/info/products/netcourse/`)

To be continued: The next installment will describe statistical inference procedures in Stata. If you have ideas for extra information that should be in this guide (either because you know it and think it would be helpful, or because you don't know it and think it would be helpful), tell me.