


Economics 422
Spring Semester 2006
A Basic Guide to STATA
Version of 02/09/06

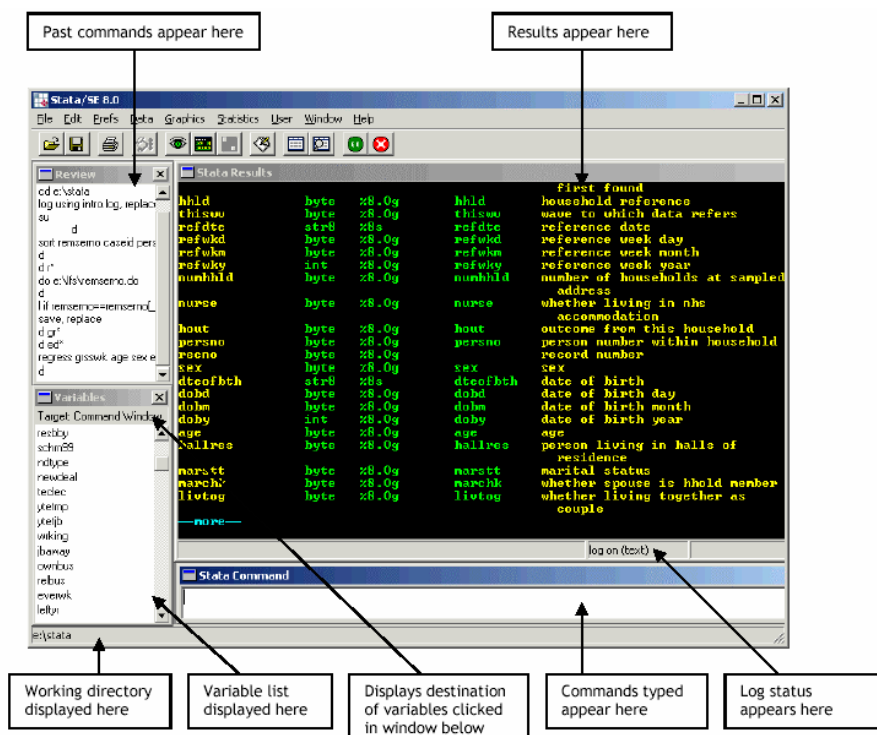
STATA is a computer program which provides very flexible computer environment in which one can conduct various analyses of data. In a few short years, it has become one of the leading programs used by researchers in applied micro economics. Since STATA was written by labor economists, it contains many econometric procedures (fixed-effects, two-stage least squares, sample selection correction, quantile regressions, probit/logit, etc.) used in the analysis of cross-sectional data, which is the focus of ECON422. It is fast and relatively easy to use.

STATA's speed advantage comes from the fact that all data is loaded into RAM. Subsequently, the amount of high memory restricts the size of the problem. Given the size of the data sets we will use in class and the available memory on the lab machines, this should not prove to be much of a constraint.

There is a nice STATA intro book by Lawrence Hamilton which some of you might find useful: *Statistics with STATA* (Updated for Version 9), Duxbury Press.

1. Getting Started

After you logged into the BSOS lab network, STATA can be accessed by clicking the icon , the current version is 9.0. Once in STATA, you will notice 4 windows as:



STATA windows:

Variables: lists variables in the dataset currently loaded in STATA memory

STATA Command: STATA command line, where you enter commands for STATA to execute

STATA Results: shows results of your STATA commands

Review: displays history of your commands (can be re-run by double-clicking)

The typical files handled and produced by STATA can be summarized:

- STATA can readily open and use STATA format data files (**.dta** files). But for most empirical projects, you will receive data in some more general format like ASCII, and you must put the data into a STATA data file. This is accomplished through the use of data dictionaries. Moreover, you can also make use of one of those data converting softwares to transfer your data set into STATA format. An leading example of these softwares is STAT/TRANSFER.
- STATA can take commands either from the Command window, or by running **.do** files that you can create. It is often more convenient to use the do files, since there you can run a batch of commands, whereas the Command window takes one command at a time. Do-files can be typed and saved in the STATA do-file editor (or any other text editor you prefer), they are just plain txt files with the .do extension.
- You can ask STATA to create and record your output in a **.log** file. This is very important if you want to track your works. For example, the line: "log using G:\econ422\evans\CPS85,replace" opens a file called CPS85.log on G:\econ422\evans subdirectory. The "replace" option tells STATA to over write any previous version of the file.

2. Loading Data

The first step is to load your data set into STATA memory so you can work on them. If you are using STATA format data files (**.dta** file), in the STATA command line, type:

```
use G:\econ422\evans\CPS85.dta, clear
```

(The "clear" option clears STATA memory from any previously loaded data; otherwise, the new file will not open). Alternatively, you can click on File -> Open, or the corresponding button on the STATA toolbar, and open the **.dta** file from there.

Unfortunately data does not often come in handy in the STATA format. We can read the raw ASCII data into STATA with data dictionaries. A dictionary defines where the raw data is stored, the variables in the data set, the type of variable (integer, scientific notation, etc.), and a short variable description called label. To use dictionary to load raw ASCII data, type:

```
infile using G:\econ422\evans\CPS85
```

Now that we loaded the data into STATA, we can save it in the STATA format:

```
save G:\econ422\evans\CPS85.dta
```

or

```
save G:\econ422\evans\CPS85.dta, replace
```

if you want to replace your old file.

3. Basic Data Manipulation

Descriptive statistics:

```
summarize [varlist]
```

Typing in this will allow you to summarize the current data, gives basic statistics for all or selected variables, for all, or selected observations. If you want more detailed information on a particular variable (quantiles, medians, skewness, kurtosis, etc.), use the "summarize" command, list the variables, and add option "detail".

```
tabulate [varlist]
```

You can obtain complete distributions for discrete variables by using the "tabulate" command. You can construct two-way contingency tables by listing the two variables in the "tabulate" command. Do not tabulate a continuous random variable.

```
correlate [varlist]
```

This command calculates correlation coefficients. Instead, if you want covariance rather than correlation coefficient, use option "covariance".

Note that you can use the above, as well as most of other STATA commands, on a subset of your dataset by using the "if" statement. Examples:

```
summarize if age >= 30  
tabulate race if educ >= 12
```

Alternatively, you can run commands for subsets of your data using "by" or "bysort" command:

```
bysort race: summarize educ
```

(since "by" command requires sorting first, "bysort" does the sorting itself.)

Working with variables:

To generate a new variable, use "gen". Examples:

```
gen age2 = age^2  
gen hw = wearn/hours  
gen lnhw = ln(hw)  
gen hsch1 = (educ >= 12)
```

After the variables are constructed, you can add a variable label. The syntax for label is illustrated as:

```
label var age2 "age squared"
```

To rename variable:

```
rename [old_name] [new_name]
```

To update values for a variable, use "replace". Example:

```
replace educ = educ - 1
```

Another way to create high school degree dummy variable:

```
gen hschl = 0  
replace hschl = 1 if educ >= 12
```

To drop or keep variable or observations, use "drop" or "keep" commands, examples:

```
drop age (drops age variable)  
keep if age < 50 (keeps observations for which age is less than 50)  
drop in 1/300 (drops observations from 1 to 300)
```

To test, for example, whether weekly earnings for workers belonging to union are the same as for workers not in the union, type:

```
ttest hw, by(union)
```

Regression commands:

To run a simple OLS regression, type:

```
regress depvar [list of independent variables]
```

Example:

The most-often estimated model in labor economics, if not all of economics is the standard human capital earnings function. Log weekly wages has been shown to be roughly linear in education. To estimate this model, run:

```
regress lnhw educ
```

4. Saving Output

You can record output you want in a log file. Open a log file when you want by:

```
log using file_path\log_file_name.log[,replace]
```

Stop and start again recording results by using: "log off " and "log on"

To close log file permanently:

```
log close
```

You can open, view or print the log file from Notepad or Wordpad later on.

5. Getting Help

STATA help feature (Help -> Contents) is often helpful. Yet, there is much more information on commands, usage and examples in the STATA manuals.

6. Exiting STATA

To exit STATA, please do to the command line, type "`clear`" and hit return which clears all variables from memory, then type "`exit`" and hit return.