Introduction to Stata

1 Data Manipulation

1.1 Data Import and Export

To import the data into workspace we simply open the Stata Data Set (any file with extension *.dta*) like we open a file with MS Word, for instance.

Sometimes the data set is to big. Stata allocates only 1Mb to the virtual memory. To extend it we use the command **set memory**. For instance

- **set mem 20m**

will extend the memory to 20Mb, which is frequently enough.

Alternatively to opening the file manually we can use the command **use**., specifying the path to the data set. This looks as:

- **use** *c:\very\long\path\filename*

  Stata assumes that the file *filename* has a .dta-extension

Frequently it is useful to change the working directory in order not to retype the pathname again and again. The default working directory is some internal ZDV directory. To change it use the command **cd**:

- **cd** *newpath*

  For instance **cd** u:\einfuehrung

  all the files will be sought in *u:\einfuehrung*

To see what your current directory is simply type **pwd**.

To save the data on the disk just use **save**. E.g.

- **save** *filename*

  in case the file with the specified *filename* already exists use option **replace** to overwrite it:

- **save** *filename*, **replace**

To import the data that are not Stata-format (extensions other than .dta) two commands can be used:

- **insheet using** *filename*

This command is used when the data are prepared in Excel; important is that when saving Excel spreadsheet one has to give file a .txt extension (tab-separated text file)

Another frequently used spreadsheet extension is ".csv". These files are semicolon-separated and therefore, to import such files one must additionally define the delimiter:

- **insheet using** *filename.csv*, **delimiter(";")**
- **infile** *varnames* **using** *filename*

This command allows importing into Stata an ASCII file; if no extension for file name is specified Stata assumes ".raw" format

To export the data one uses reciprocals of the above commands:

- **outsheet** *varnames* **using** *filename*

This command creates a tab-separated file readable by Excel. If no other format is specified, Stata saves the file with ".out" extension, which can be opened by Excel

- **outfile** *varnames* **using** *filename*

saves the data on the disk as an ASCII (.raw) file

In case the file with the specified name already exists all the export commands also expect **replace** option. E.g.:

outfile *varnames* ***using*** *myfile.raw*, **replace**

overwrites the existing data file myfile.raw.

1.2 Working with Data

Once the data are imported/merged, you might need to do some further manipulations. Here is the list of the most frequently needed commands:

- **describe**

  Returns a description of all variables stored in the workspace. Can be applied to a selected list of variables, e.g.:

  describe *var1 var2*

- **drop** *var1 var2*

  Deletes specified variables from the data. Can be also used to modify the data according to a certain condition. For example:

  drop if var1 == 0

  deletes all observations that correspond to zero observations of variable *var1*.

Here one also sees an example of logical **if**-conditioning. Logical operations are possible almost after every command and encompass such operators as: **==**, **>**, **<**, **>=**, **<=**, **&** ("and") and | ("or"). For instance

drop if var1 == 0 & var2 < 0

deletes all observations that both that correspond to zero observations of variable *var1* and negative observations of variable *var2*;

drop if var1 == 0 | var2 > 0

deletes all observations that correspond to either zero observations of variable *var1* or positive observations of variable *var2*.

Complementary command to **drop** is **keep**

Other useful data manipulation commands deal with generating and replacing variables. All variables are created with the help of the command **generate**. For instance:

- **gen** *var1* = 0

  creates a series of zeros named *var1*;

- **gen** *var1* = *var2* + sqrt(*var3*)

  creates a sum of *var2* and square root of *var3*;

  gen is used with all possible mathematical [+, -, *, /, ^, **exp(***var***)**, **ln(***var***)** ... etc.] and statistical functions (see Stata help for the complete list). Command **gen** also allows **if-**conditioning.

Another very useful command is **replace**. It is the same as **gen** but applies to already existing variables. Syntax is the same:

- **replace** *var1* = *var2*
- **replace** *var1* = *var2* – sqrt(*var3*) **if** *var1* == 0

Apart from these two there also exists an "extended generate" (**egen**) command that allows to perform multi-step changes of the variable and create some useful series (see Stata help). For instance:

- **egen** var1 = fill(1 2)

  creates a sequence 1,2,3, ...

- **egen** has a useful option **by(***var***)**; including it we tell Stata to perform the generation command within the subset of the *var* on which this variable has the same values. To use this command the variable *var* must be sorted first. For example:

  sort *household_id*

  by *household_id*: **egen** *average_age* = mean(*age*)

  finds the average age within every single household

2 Descriptive Statistics Commands

Command **summarize** provides you with the summary statistics of the list of variables (or the whole data set, when no variable is specified). Namely,

- **sum** *var1 var2*

returns number of non-missing observations, mean, standard deviation and sample minimum and maximum. This command also takes an option d which stands for detailed summary. In addition to above quantities

- **sum** *var1 var2*, **d**

will return percentiles of empirical distribution, skewness and kurtosis.

The second main command is **tab** which performs either one-way or two-way tabulation:
tab *var1* – one-way
tab *var1 var2* – cross tabulation

To visualize the data one can call either histogram or kernel density plots. The appropriate command for the histogram is:

- **hist** *var1*

If the variable under study is discrete, use option **, d**:

- **hist** *var1*, **d**

To draw a kernel density type

- **kdensity** *var1*

Furthermore, kernel density command has an option of keeping calculated values of the empirical p.d.f. To run it specify:

- **kdensity** *var1*, **generate(***var_points var_values***)**

As a result Stata will generate a series of the values of empirical p.d.f (*var_values*) and a series of the points at which the kernel estimates were evaluated (*var_points*).

To save any of the above plots use option **saving( )**, e.g.:

- **hist** *var1*, **saving(***myhist***)**
- **kdensity** *var1*, **generate(***var_points var_values***)** saving(*my_density*)

Stata saves graphics in the working directory in the internal .gph format, if no other is specified.

Further useful descriptive statistics commands include:

- **cumul** *var1*, **generate(***var_cdf***)** creates variable *var_cdf* which is an empirical c.d.f. of *var1*; option **generate(***var_cdf***)** is necessary

3 Graphics

To draw plots in Stata there exists a universal command family called **twoway**. Once **twoway** is declared Stata will wait for the type of the graph to drawn. For instance

twoway line – draws a line graph
twoway scatter – a scatter plot, and so on (for other types see Stata Help on *twoway*)

When specifying the plot the variable that determines x-axis must follow the last. E.g.

- **twoway line** *var_y1 var_y2 var_x*

will plot the values of *var_y1* and *var_y2* against the corresponding values of *var_x*

The most important option in the graphical commands is **saving( )**. The expression

- **twoway line** *var_y1 var_y2 var_x*, **saving(***graph1***)**

saves the drawn graph to file *graph1.gph* in the current directory [alternatively, specify the path to another directory in quotas, i.e. **saving(** *"u:\another\folder"***)**]. This option also allows replacing:

- **twoway line** *var_y1 var_y2 var_x*, **saving(***graph1***, replace)**

To call the saved graph execute

- **graph use** *graphname*

To convert current graph into any of the commonly used formats run the command **graph export**. Available formats are .eps, .wmf, .tif (see Help for more), e.g.:

- **graph export** *graphname.eps*, **replace**

4 Simple Regression and Testing

Stata has many different regression models implemented (she is very clever). For the moment let us consider an OLS estimation of the simple linear regression.

For OLS estimation of a linear regression we use the command **regress**. For instance

- **reg** *var_y var_x1 var_x2 var_x3*

will regress the dependent variable (*var_y*) on the set of explanatory variables (*var_x1 var_x2 var_x3*). Dependent variable is always specified first

To test simple linear hypothesis we use the command **test**. E.g.,

- test *var_x1 = var_x2*

will test for equality of estimated coefficients in front of *var_x1* and *var_x2*.

5 .do-Files

Up to now we have been executing all our commands one after another. There exists however a possibility to write them all in one file and let this file run. In this way we will be able to run all our commands at once. This way of working is very convenient, because it allows correcting intermediate mistakes without the need to repeat everything manually from the point the mistake was made onward.

Typical .do-File looks as follows:

****

clear
set mem 20m
cd *u:\einfuehrung*

use *data*
save *work*, **replace**

gen logwage = log(wage)

reg logwage sex age education

save *work*, **replace**

****

In other words this is just a list of commands we want to run. We write them all down, save them with ".do" extension and run. To run a .do-File simply use the File - Do entry in the main menu.