Ninth ARTNeT Capacity Building Workshop for Trade Research
"Trade Flows and Trade Policy Analysis"

June 2013

Bangkok, Thailand

Cosimo Beverelli and Rainer Lanz

(World Trade Organization)

Introduction to STATA

Content

a. Datasets used in Introduction to Stata
b. Resources
c. Importing data into Stata
d. Do files
e. Commands for variable's management and descriptive statistics
f. Macros
g. Loops

a. Datasets used in Introduction to Stata

- To apply some of the Stata commands described in this presentation, we will use two datasets:
  - WDI.dta - a very small subset of the World Development indicators
  - WB_ES.dta – derived from the World Bank Enterprise Surveys
- You can find the datasets in the directory: "Stata_material\data\IntroductionStata\"

b. Resources

- Stata help and Stata manual

- A variety of books covering Stata exist

Web resources:

- Germán Rodríguez's webpage
  - Data management, graphics and programming

- UCLA IDRES' webpage
  - Very comprehensive covering all sorts of topics (data management, analysis,…) with many examples
  - FAQ

- Statalist
  - Typically accessed via a google search

c. Importing data into STATA

- *insheet using filename, clear delimit(";") names*
  - Typically used for text files that are either comma or tab-separated
  - Rarely used alternatives: *infix* (fixed-column format); *infile* (free format)

- Stata12: *import excel using filename , sheet("Ex1") first*
  - Reads excel files directly into Stata
  - Allows to specifiy variables, cellrange and worksheet to import

- StatTransfer
  - Specialised software to transfer data

- Copy paste
  - Sometimes the most efficient way, e.g. when you do not want to write a do file
  - To watch out: the accuracy of copied numbers depends on
    1. how data are formatted in excel, i.e. how many digits are shown, and
    2. your settings in Stata(use *set type double* before copying)

d. Do files

- If you work with STATA, (almost) always use do files
  - E.g. one do file for creating your master dataset and one do file for regressions
  - Do files can also be used to set globals and directories or to run a series of different do files after each other

- Typical commands at beginning of each do file:

```
clear all                          /* removes all data */
set more off, perm                 /* prevents Stata to pause while runnning a do file */
capture log close                  /* closes a log file */

cd "directory"                     /* sets the directory, e.g. "C:\Research\data\" */

log using "filename", replace      /* useful for long do files, allows printing */
capture log close                  /* at the end of a do file that is logged */

use "dataset.dta", replace  /* open dataset; " " are not necessarily needed */
```

e. Commands for variable's management and descriptive statistics

- *generate newvar=exp* *[if]*
  - Creates a new variable
- *replace oldvar=exp* *[if]*
  - Replaces an existing variable
- *rename old_varname new_varname*
  - Renames variable; alternative: *renvars varlist*
- To drop or keep variables you can use
  - *drop varlist* or *keep varlist*
- To drop or keep observations you can use
  - *drop if* or *keep if*

e. Commands for variable's management and descriptive statistics (ct'd)

- *describe*
  - Provides information on dataset (#obs, #vars, size) and on variables (type, labels)
- *sum(marize) varlist*
  - Provides #obs, mean, std. dev., min., max
- *tab(ulate) var1 var 2*
  - Provides one- or two-way tables of frequencies
  - *tab cou sector*
  - Allows the creation of dummy variables with the option generate()
- *table rowvar (colvar), content()*
  - Provides frequencies by default. The option *contents* allows for other statistics
  - *table cou sector, content(mean sales sum d_exp)*
  - *by cou: table sector, content(mean sales sum d_exp)*
- *tabstat varlist, statistics() by()*
  - Another command to calculate summary statistics
  - *tabstat sales if cou=="USA", by(sector)*

e. Commands for variable's management and descriptive statistics (ct'd)

- Commands to identify missings
  - *inspect varlist* e.g. *inspect cou*
  - *codebook varlist* e.g. *codebook cou*

- *duplicates (report/drop/tag/list) varlist*
  - Reports, drops, tags or lists observations that are identical in all variables or identical in the variables specified by *varlist*

- *unique varlist*
  - Reports the number of unique values for varlist

e. Commands for variable's management and descriptive statistics (ct'd)

- *egen newvar=function(varlist or other argument)*
  - Often used command to create new variables, see Stata help

- Often used *egen* functions:
  - *bysort cou sector: egen sales_sec=total(sales), missing*
  - *bysort cou sector: egen sales_sec=mean(sales)*
  - *egen exp_tot=rowtotal(exp_intermediate exp_final)*
  - *egen id_cluster=group(cou sector)*
  - *egen cou_sec=concat(cou sector)*
  - *Further functions include: max, min, count, tag,…*

e. Commands for variable's management and descriptive statistics (ct'd)

- *collapse (mean) varlist (sum) varlist, by(varlist)*
  - Creates an aggregate dataset by e.g. averaging or summing variables across the dimension identified in by()
  - All observations not included in the command are dropped
  - Useful in analysis when moving to a higher level of aggregation, e.g. aggregating trade flows from HS 6-digit to HS 2-digit
  - Useful for calculating descriptive statistics before exporting them to excel using *outsheet* or *export excel*
- *egen* can be used to create aggregates within the disaggregated dataset
  - *bysort cou sector: egen sales_sec=total(sales), missing*
- *duplicates drop* after *egen* gives the same results as *collapse*
  - *keep cou sector sales_sec*
  - *duplicates drop*

e. Commands for variable's management and descriptive statistics (ct'd)

- *destring varlist , replace force*
  - Converts a string variable to a numeric variable
  - Useful when numbers are imported as string into Stata
  - The other way round – numeric to string: *tostring varlist, gen(newvar)*

- *String functions: generate newvar =function()*
  - Allow to manipulate string variables. See Stata help. Some useful functions are:
  - *abbrev ()* – shortens the string the number of indicated characters
  - *length()* – returns the length of the string, i.e. number of characters
  - *subinstr()* – allows to replace or delete particular substrings
  - *substr()* – allows to extract substrings based on its position
  - *upper (lower)* – Changes the entire string to upper-case (lower-case) strings
  - *trim()* – removes leading and trailing blanks of the string

e. Commands for variable's management and descriptive statistics (ct'd)

- *reshape wide (long) 'stub', i() j() options*
  - Reshapes dataset from long to wide format and vice versa
  - Data dimensions such as country, year or sector are normally put in long format
  - 'stub' are variables in *reshape wide* and stubs of variables in reshape long
  - i() are identifying dimensions; j() dimension to change
  - Exercise: Open WDI.dta and reshape it first long and then wide

- To merge datasets use either *merge* or *joinby*
  - *merge (1:1,m:1,1:m,m:m) varlist using filename, update keepusing(varlist)*
  - *joinby varlist using filename, unmatched(both) update*
  - Merge is used to add further variables to observations in the master data
  - Joinby forms all pairwise combination for varlist
  - Exercise: Open WB_ES.dta and merge it with WTI.dta

f. Macros

- See Stata help and Germán Rodríguez's webpage

- Macros are names associated with some text
  - The commands global and local assign strings to global and local macro names

- *global mname [=exp | :extended_fcn | [`]"[string]"[']]*
  - Global macros, once defined, are available anywhere in Stata

- *local lclname [=exp | :extended_fcn | [`]"[string]"[']]*
  - Simplest example: *local c USA JPN*
  - Local macros work only within the do file in which they are defined

- Globals and locals have a variety of uses
  - To define the directories for this class, i.e. directory_definition.do
  - They are used in loops (see next slides)
  - A set of explanatory variables can be grouped under one macro name

g. Loops

- See Stata help and Germán Rodríguez's webpage
- Two main commands: *foreach* and *forvalues*
- Syntax:

```
foreach    lname {in|of listtype} list {
           commands referring to `lname'
           }
```

```
forvalues lname = range {
           commands referring to `lname'
      }
```

g. Loops (ct'd)

- Examples for loops in WB_ES.dta:

```
foreach k in USA JPN {                    /* Loop over any_list */
egen sales_`k'=total(sales) if cou=="`k'"
}

vallist cou, local(c)              /* vallist shows values and creates local */
foreach  k of local c {                   /* Loop over a local  macro */
capture drop sales_`k'
egen sales_`k'=total(sales) if cou=="`k'"
}

forvalues k=1(1)3 {                       /* Loop over sector codes */
egen total_`k'=total(sales) if sector==`k'
}
```

- Foreach can also be used to loop over variables and numbers
  - foreach k of var varlist; foreach k of num numlist