Introduction to Stata Programming

Econometrics I

R. Mora

Department of Economics
Universidad Carlos III de Madrid
Master in Industrial Organization and Markets

Outline

What is Stata?

- Stata is a statistical package and programming language widely used in econometrics
- Stata is available for Windows, Unix, and Mac OS and for 32-bit and 64-bit computers.
- Stata commands usually follow a simple syntax (brackets "[" & "]" mean optional):

```
command [varlist] [if] [in] [weight] [using filename] [,options]
list country lexp gnppc if missing(gnppc)==1 in 1/60, clean noobs
```

- To obtain help on a command (or function) type
  - `help command_name`

```
help help                    // This gives help on the command "help"
```

How you should work with stata

1. Always use a .do file for replicability
2. Keep your results with a .log file
3. Start every .do file with comments on project title, date, what the .do file does, ...

basic_example.do

```
// This is a basic example of how a .do file looks
version 10           // Stata evolves, but your program will always work
clear all                                      // start with a blank sheet
capture log close     // useful if something went wrong in a previous go
log using basic_example.log, text replace            // open the log file
    sysuse lifeexp
    describe
    summarize lexp gnppc
    list country lexp safewater gnppc if missing(gnppc)==1 in 1/60,
clean noobs
log close                                             // close the log file
```

Variable types

- Variable Types (`help datatype` & `help compress`):
  - Numeric variables contain real numbers: $1, 4.564, 0.1, \mathit{float}(.1), ...$
  - There are 5 different types of numeric variables
  - String variables contain (up to 244) ASCII characters : "High Education", " High Education "
- You may convert between numeric and string variables (also `help real()`)
  - string to numeric: `encode`
  - numeric to string: `decode`

Reading and Viewing Data

- Reading data (you can copy data directly from `Excel`,...):
  - `use`: use Stata dataset
  - `input`: you can type data directly (also with the data editor)
  - `infile`: ASCII in free format (`see help infile1`, infile2, & `help infix`)
  - `insheet`: ASCII, csv, one observation per line
- Viewing data:
  - `describe`: gives details of a dataset, such as name of variables, label of variable, etc.
  - `summarize`: summary descriptive statistics of a dataset/variable
  - `list`: shows content of variables
  - `tabulate`: lists each distinct value of a discrete variable and the number of times it occurs

Creating & Modifying Variables

- generate: `generate age_sq = age^2`
- egen: (an addition to generate with many options):
  `egen avewght= mean(weight) if weight<.`
- replace: `replace lwage = 0 if lwage >= .`
- recode: `recode age (0/30=1) (31/50=2) (51/100=3), gen(age_major)`
- tabulate with option generate:
  `tabulate age_major, generate(Dage)` `//creates 3 dummies`

Operators & Functions

- Operators (see `help operators`):
  - Arithmetic: `+`, `-`, `*`, `/`, `^`,
  - Logical and relational: `&`, `|`, `!`, `>`, `<`, `>=`, `<=`, `==`, `!=`
- Functions (see `help functions`):
  - Mathematical: `abs(x)`, `exp(x)`, `log(x)`, `sqrt(x)`...
  - Statistical: `normal(z)`, `invnormal(p)`, `ttail(n,t)`...
  - random-number functions, string functions, matrix functions, ...

Data Management

- To load a file: `use` *`file1`*`, clear // `*`file1`* now becomes data in Stata memory
- To order observations: `sort year month //sort data by year and within year, by month`
- keep & drop: keep/eliminate variables/observations
  `keep age lnweight // keep only these two variables in data`
  `drop in 1/100 // drop 1st 100 obs of all variables`
- collapse: make dataset of summary statistics
  `collapse (mean) weight, by(foreign)`
- To add **observations:** `append using` *`file2`* `// `*`file2`* `may not share variables with file1`
- To add **variables:** `merge 1:1 id_var using file2 // 1 to 1 merge`
- To save data on disk: `save filename, replace`

Macros

- Macros are symbols associated with characters or values
- Global macros remain in memory in the session

```
global mydir
"/media/MEI/Session_05_Introduction_to_Stata_Programming"
global yourdir "C:/Desktop"
```

- Local macros are operational only in the context they are is born

```
local variables = "age agesq education income"
list `variables'
```

Loops & Branches

- Loops are used in repetitive tasks (see also `foreach` and `while`)

```
forvalues mynumber = 1(1)5 {
    display "loop number:  `mynumber'"
    }
```

- Branches: `if`

```
local i = 1991
while `i'<=2010 {
    use lfs`i'.dta, clear
    generate year = `i'
    if `i' > 1991 {
        append using lfs.dta
        }
    save lfs.dta, replace
    local i = `i' + 1
    }
```

The Model

- In general, we will consider as many controls as we feel necessary:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + u$$

- Controls are not correlated with u: $E(x_j u) = 0$, for $j = 1, ..., k$
- We want to
  - get estimates for $(\beta_0, \beta_1, ..., \beta_k)$
  - test hypothesis
  - predict from regression results
- We use WAGE1.DTA to show the use of regress

The wage1.dta

- The file contains data on wages for workers in the USA in 1976
- The number of observations equals 526 (workers)
- Original source: The 1976 Current Population Survey
- Used in textbook: pp. 7, 38, 76-77, 93, 123-124, 180, 190-192, 214, 222-223, 226-228, 232, 235, 254, 260-261, 311, 648 in Wooldridge

The `regress` command

regress *depvar* [*indepvars*] [if] [in] [weight] [, options]

- *depvar*: name of dependent variable
- *indepvars*: list of controls (in addition to a constant)
- typical options:
  - `noconstant`: suppresses constant term
  - `robust`: obtains robust estimates of the variance-covariance matrix (VCE) of the parameter estimates

The standard output from regress

regress *lwage nonwhite female married*

```
      Source |       SS       df       MS              Number of obs =     526
-------------+------------------------------           F(  3,   522) =   39.56
       Model |  27.4762326     3  9.15874421            Prob > F      =  0.0000
    Residual |  120.853519   522  .231520151            R-squared     =  0.1852
-------------+------------------------------           Adj R-squared =  0.1806
       Total |  148.329751   525   .28253286           Root MSE      =  .48117

------------------------------------------------------------------------------
       lwage |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    nonwhite |  -.0513523    .069273    -0.74   0.459    -.1874405    .0847358
      female |  -.3600183   .0425981    -8.45   0.000    -.4437031   -.2763336
     married |   .2312673   .0436792     5.29   0.000     .1454587     .317076
       _cons |   1.660326   .0427254    38.86   0.000     1.576391     1.74426
------------------------------------------------------------------------------
```

Statistical Significance

To check if a regressor is significant, look at its p-value

$$H_0 : \beta_{married} = 0 \qquad H_1 : \beta_{married} \neq 0$$

```
      Source |       SS       df       MS              Number of obs =     526
-------------+------------------------------           F(  3,   522) =   39.56
       Model |  27.4762326     3  9.15874421            Prob > F      =  0.0000
    Residual |  120.853519   522  .231520151           R-squared     =  0.1852
-------------+------------------------------           Adj R-squared =  0.1806
       Total |  148.329751   525   .28253286           Root MSE      =  .48117

------------------------------------------------------------------------------
       lwage |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    nonwhite |  -.0513523    .069273    -0.74   0.459    -.1874405    .0847358
      female |  -.3600183   .0425981    -8.45   0.000    -.4437031   -.2763336
     married |   .2312673   .0436792     5.29   0.000     .1454587    .317076
       _cons |   1.660326   .0427254    38.86   0.000     1.576391    1.74426
------------------------------------------------------------------------------
```

What about *nonwhite*?

Robust Standard Errors: robust

regress *lwage nonwhite female married*, robust

```
Linear regression                               Number of obs =     526
                                                F(  3,   522) =   36.72
                                                Prob > F      =  0.0000
                                                R-squared     =  0.1852
                                                Root MSE      =  .48117

------------------------------------------------------------------------------
             |               Robust
       lwage |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    nonwhite |  -.0513523   .0664889    -0.77   0.440    -.1819711    .0792664
      female |  -.3600183   .0415128    -8.67   0.000     -.441571   -.2784657
     married |   .2312673   .0439628     5.26   0.000     .1449017     .317633
       _cons |   1.660326   .0441697    37.59   0.000     1.573554    1.747098
------------------------------------------------------------------------------
```

- robust gives the same beta estimates (OLS). Estimates for the standard errors, however, use a formula which is generally better than the one used by default

Hypothesis testing: the test command

- test tests linear hypothesis after estimation

$H_o : \beta_{female}$=-.4

```
. test female = -.4

 ( 1)  female = -.4

       F(  1,   522) =    0.93
            Prob > F =    0.3359
```

p-value is larger than 0.10 $\rightarrow$ we cannot reject the null at 10% significance level

The test command (II)

$H_o : \beta_{nonwhite} = \beta_{female}$

```
. test nonwhite =  female

 ( 1)  nonwhite - female = 0

       F(  1,   522) =   15.38
            Prob > F =    0.0001
```

p-value is smaller than 0.01$\rightarrow$ we reject the null at 1% significance level

The test command (III)

$H_o : \beta_{female} = -0.40; \beta_{married} = 0.25$

```
. test (female=-.4) (married=.25)

 ( 1)  female = -.4
 ( 2)  married = .25

       F(  2,   522) =    0.58
            Prob > F =    0.5630
```

we cannot reject the null at 10% significance level

- you can "accumulate" tests:

```
. test female = -.4
. test married = 0.25, accumulate
```

Other results: `ereturn list`

- `regress` is an e-class command: it produces estimates
- the results of an estimation command are automatically saved in macros, scalars, functions, and matrices with names e()
- `ereturn list`: lists all results stored after any estimation command
- results in e( ) are replaced when a subsequent e-class command is executed
- other commands are r-class commands: they get results which are not estimates. They are also store with names e(), but to see them, you have to type `return list`.

Using estimation results after regress

```
. matrix define b=e(b)

. matrix define beta=e(b)

. matrix list beta

beta[1,4]
      nonwhite      female     married       _cons
y1  -.05135233  -.36001835   .23126735   1.6603257
```

we can access and manipulate OLS estimates

Some of the saved results

Matrices

- Coefficients: e(b)
- Variance-covariance: e(V)

Scalars

- No. of observations in regression: e(N)
- No. of parameters: (df_m)
- Degrees of freedom: e(df_r)
- R^2: e(r2)
- Residuals Sum of Squares: e(rss)

Tables from several regressions: estimates

After several regressions, we may want results in one table

1. estimates store *regname#* (as many times as regressions)
2. estimates table *regname1 regname2*

```
quietly regress lwage nonwhite                //quietly suppresses the output
estimates store reg1
quietly regress lwage nonwhite female
estimates store reg2
quietly regress lwage nonwhite female married
estimates store reg3
estimates table reg1 reg2 reg3, b(%7.4f) se(%7.4f) ///
        stats(N r2_a) title("All results")
```

```
est tab reg1 reg2 reg3, b(%7.4f) se(%7.4f) stats(N r2_a) title("All results")
```

All results

Variable	reg1	reg2	reg3
nonwhite	-0.0680	-0.0752	-0.0514
	0.0764	0.0709	0.0693
female		-0.3977	-0.3600
		0.0431	0.0426
married			0.2313
			0.0437
_cons	1.6303	1.8215	1.6603
	0.0245	0.0307	0.0427
N	526	526	526
r2_a	-0.0004	0.1382	0.1806

legend: b/se

Prediction: predict

Obtain predictions, residuals, etc., after estimation

```
regress lwage female married if nonwhite==0                    // white obs
predict ehat , res                                  // we can generate residuals
predict yhat , xb if e(sample)                    // pred.  in estimation sample
predict yhat_w , xb if nonwhite                     // nonwhite wages if whites
```

List of Basic Commands

- `append`
- `capture`
- `clear`
- `collapse`
- `decode`
- `describe`
- `display`
- `drop`
- `egen`
- `encode`
- `ereturn list`
- `estimates store`
- `estimates table`
- `forvalues`
- `generate`
- `global`
- `help`
- `if`
- `infile`
- `input`
- `insheet`
- `keep`
- `list`
- `local`
- `log`
- `matrix define`
- `matrix list`
- `merge`
- `predict`
- `quietly`
- `recode`
- `regress`
- `replace`
- `return list`
- `save`
- `sort`
- `summarize`
- `sysuse`
- `tabulate`
- `test`
- `use`
- `version`
- `while`

Summary

- Stata is a powerful statistical package
- It has many commands to easily manipulate data sets
- It is also a programming language
- OLS is easy to implement using Stata
- Linear hypothesis can be tested using a single command
- Results can be recovered from memory
- In and out-of-sample predictions are available