Introduction to Stata – Session 3

Tarjei Havnes

[1]ESOP and Department of Economics
University of Oslo

[2]Research department
Statistics Norway

ECON 3150/4150, UiO, 2015

Before we start

1. In your folder statacourse: caschool.dta (California test score data)
2. Start STATA from the Start menu
   - 2.1 Or: Go to kiosk.uio.no (Internet Explorer!) and log on using your UIO user name; Navigate to Analyse (english: Analysis); Open Stata

Outline

1. Stata workflow
2. Working with do-files
   - text files with Stata code
   - formatting: make your file readable!
   - comments: include description of the code
3. Regression, prediction, testing
4. Basic graphs
   - scatter plots
   - line plots
   - overlaying graphs
   - basic formatting
   - exporting

Stata workflow

Personal hygiene

In practice you should always try to strictly separate changing & analysing data:

1. first prepare your data for analysis
   - copy data from disk to memory
   - change data (prepare for analysis)
   - save data to disk under new name
2. then analyze these data
   - copy analysis data into memory
   - start logging results to file
   - perform analysis
   - close log file

Advice: one directory per project & start session in project dir

Do files

Until now we have mostly used the command line:

- great to develop but not to reproduce your analysis
- ALWAYS organize your work in Stata scripts

Stata scripts are called do-files after their extension (.do)

Use do-files (with informative names) to organize your work:

- create dataset
  crincome.do makes data file income.dta
- analysis
  andescr.do calculates my descriptive statistics
  anreg.do performs my regression analysis
- making graphs
  grwageplot.do makes the graph wageplot.eps

Note: do-files can call do-files.

- You can create a master do-file which calls the do-files which reproduce your complete preparation and analysis trail

Make a do-file

Use California test score data from course homepage, then make a do-file that does the following

1. Read the data into Stata
2. Keep only read_scr, math_scr, enrl_tot, teachers and el_pct
3. Make new var score equal to mean of reading and math score
4. Make new var str equal the student-to-teacher ratio
5. Label the variables:

enrl_tot	Enrollment
teachers	Teachers
el_pct	Percent english-learners
score	Mean test score
str	Student/Teachers

Style

Space around operators

- `gen x = y + z`

Space after comma

- `gen fx = normalden(x, 0, 1)`

Indent (1 tab) after '{' and close at the level of the opening command

- ```
  if (_rc == 0) {
      di "Warning"
      exit
  }
  ```
  ```

Documenting - Comments

Use comments in your do-files when the code needs explaining or is better readable with a comment

- Single line comments:
  ```
  // comment here
  ```
- Multi line comments:
  ```
  /*
  [commented out]
  */
  ```
- Break lines:
  ```
  list pop19??  /// the rest of the line is
  commented out
      if country=="NOR"
  ```

Make a do-file – cont.

1. Summarize all variables
2. Make a new variable zscore as standardized score, i.e.

$$zscore = \frac{score - mean(score)}{SD(score)}$$

3. Draw a scatter of zscore against str
4. Regress zscore on str
5. Make new var zscorehat as the prediction from the regression
   - *Hint:* use -predict-
6. Draw a scatter of zscore against str, including the predicted regression line
   - *Hint:* use -twoway (scatter y x) (line z x)-

Making tables from regression results

Estimation commands such as -regress- store results like coefficients and covariance matrices

- These can be used to make tables using Stata's -estimates-
  - -help estimates-

To store estimates in memory: -estimates store-
To activate previously stored estimates: -estimates restore-
To table estimates: -estimates table [estnames]-

```
reg zscore str, robust
est store str
reg zscore el_pct, robust
est store elpct
reg zscore str el_pct, robust
est store strelpct
est table str el_pct strelpct
```

The estout-package

`estout` is a user contributed add-on with many options

- you should install such add-ons in a dedicated directory (named e.g. ado or stata)
- this is a little cumbersome when you are working from the server
  - see course web page for how to install programs (add-ons) that you find online or using Stata's -findit- or -net search-
  - this is very useful in practice

Now type `findit estout` , scroll down and click through to install

```
. esttab *, se
```

Make a do-file – cont.

1. Make a new var elhigh equal to 1 if el_pct > mean
2. Table means of score and zscore for the two groups
3. Regress zscore on str controlling for elhigh
4. Table results from this and the previous regression together
5. Make new var zscorehat_elhigh as the prediction
6. Draw a scatter of zscore against str, including the predicted regression line, where both scatter and line are separate for the two groups

Saving your results (logging)

You can save your results to file using -log-

- `log using anauto`

Stata will throw an error when

1. the log file exists
   solution: `log using anauto, replace`
2. the log file is already open
   solution: `close log`
3. when there is no open log
   final solution: `capture close log`

Plain text log file:

- `log using anauto, replace text`

Advice: Always use the same name as the do file

A typical do file (anreg.do)

```
clear
cd "M://My Documents/statacourse"
capture log close
log using anreg, replace
set more off

// do analysis here

// sometimes:
    quietly log close
     // do something that you don't need to log here
    quietly log using anreg, append
     // do further analysis here

log close
// always leave one empty line at the end
```

Make a do-file – cont.

1. Make new var strelhigh as the interaction of str and elhigh
2. Repeat the previous regression, incl. strelhigh
3. Form the prediction zscorehat_elhighint
4. Table results from all the regressions together
5. Draw a scatter of zscore against str, including the predicted regression line, where both scatter and line are separate for the two groups

Hypothesis testing

To do hypothesis testing, use -test-

- test one variable $\beta_1 = 0$:

```
test var1
```

- test several variables, $\beta_1 = 0$, $\beta_2 = 1$:

```
test var1=0 var2=0
```

- test combined hypotheses, $\beta_1 = \beta_2$:

```
test var1 = var2
```

Hypothesis testing

```
. quietly regress zscore str strelhigh elhigh, robust

. test strelhigh

 ( 1)  strelhigh = 0

       F(  1,   416) =    0.32
            Prob > F =    0.5717

. test strelhigh elhigh

 ( 1)  strelhigh = 0
 ( 2)  elhigh = 0

       F(  2,   416) =  107.95
            Prob > F =    0.0000

. gen strellow = str * (1-elhigh)

. quietly regress zscore strellow strelhigh elhigh, robust

. test strellow = strelhigh

 ( 1)  strellow - strelhigh = 0

       F(  1,   416) =    0.32
            Prob > F =    0.5717
```

Hypothesis testing

You can also do tests of e.g. means

```
. ttest zscore, by(elhigh)

Two-sample t test with equal variances
------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
el_pct < |     276    .4122849    .0507242    .8426944    .3124278    .5121421
el_pct > |     144   -.7902127    .0649548    .7794575   -.9186084   -.6618171
---------+--------------------------------------------------------------------
combined |     420   -1.62e-09     .048795           1   -.0959135    .0959135
---------+--------------------------------------------------------------------
    diff |            1.202498    .0844605                1.036477    1.368518
------------------------------------------------------------------------------
    diff = mean(el_pct <) - mean(el_pct >)                        t =  14.2374
Ho: diff = 0                                     degrees of freedom =      418

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 1.0000         Pr(|T| > |t|) = 0.0000          Pr(T > t) = 0.0000
```

Saving your graph

You can save your graph to disk using

```
graph export filename
```

The extension determines the format, e.g.

```
graph export zscore-str.eps
```

if the file exists, use option -replace-

Note:

- Best quality: Vector based formats
  - ps, eps, pdf, wmf/emf (Win only)
- Most portable: Pixel-based formats
  - png

What you should have learned

1. Stata workflow
2. Working with do-files
   - text files with Stata code
   - formatting: make your file readable!
   - comments: include description of the code
3. Regression, prediction, testing
4. Basic graphs
   - scatter plots
   - line plots
   - overlaying graphs
   - basic formatting
   - exporting