

**EDUCATIONAL
ATTAINMENT
AND
EARNINGS
FUNCTIONS**

STATA COMMANDS

INTRODUCTION TO STATA

Stata is an immensely powerful, resource-rich application and its manuals collectively run to several thousand pages. This document provides an alternative to the manuals that will enable you to undertake the *EAEF* regression exercises with the minimal investment of time.

Files

As with all statistical applications, you will be working with two types of file: a data file and output files.

Data files :

As you do the exercises, the file will increase in size because you will occasionally add new variables to it, but you will never need to change its name. When you have added a new variable and wish to save the expanded file, do so with the existing name and overwrite the previous version. You cannot print this file directly because it is in a special format that no printer can recognize. If you wish to print out some of the data, you have to do this from within Stata.

Log files : Output files are described in Stata as log files. You will create many of them, one for each Stata session. Assuming that you wish to preserve the output for each session, you should give each file a different name. To open a log file, click on the button with a scroll on the Stata menu bar. This is the fourth from the left. You are given a choice of type of log file at the time of opening it. The default is a “formatted log file” with extension `.smcl` . Avoid this, and choose instead a plain log file with extension `.log` . Plain `.log` files are ASCII (text) files and can be imported with no fuss into any word-processing package. Incidentally, having specified the name of your `.log` file and the directory (folder) in which it is to be located, it is natural to look for a tab with “open” or “OK” to click. There isn’t one. You have to click on the button marked “save” to open the file. To close the file at the end of your session, click on the scroll icon again and choose the close option.

Stata windows

When you launch Stata, you will see four Windows: a command window, a results window, a variables window, and a review window.

Command window : This one-line window is where you type in your instructions. You can save on typing in two ways. Instead of typing the name of a variable, you can click on its name in the variables window. Second, when you need to give a command that is similar to a previous one, you can do this by editing the previous command rather than starting from scratch. You do this by

not be able to issue any more commands until you have done this. If the bottom of the results file is not visible, you will not be able to see the --more-- instruction and you will think that Stata has hung up on you.

Variables window : This contains a list of the variables in the data set.

Review window : This lists your most recent commands.

Common commands

Here are a few commands that will be useful in the exercises:

- `reg` followed by a list of variable names. The first variable is regressed on the rest.
- `sum` followed by a list of variable names. This produces a table giving the mean, standard deviation, maximum and minimum for each variable listed.
- `tab` followed by one variable name. This produces a frequency distribution for the variable
- `tab` followed by two variable names. This produces a cross-tabulation with the first-named variable providing the rows and the second-named one providing the columns
- `gen` followed by an equation. This creates a new variable defined as the dependent variable of the equation.

Adding an `if` expression at the end of a command, for example `if y>10`, makes it selective as indicated. Most `if` conditions are straightforward, but there is one that is not: a condition which uses an `=` sign, like `if y==10`, must repeat the `=` sign as shown. This is to distinguish between the use of `=` in equations defining variables and its use in tests for equality.

EXERCISES

Exercise 1 Simple regression analysis

You must download a data set in order to do this exercise. Instructions are given at the beginning of the *EAEF* manual. You should download only once.

Open a log file (see the description of the log window above) and name your log file EX1 or something similar. Stata will automatically add the extension `.log` if you have chosen the plain log file in the save as type box..

Exercise 2 Simple regression analysis

From now on the instructions to open the data file and to open a log file will be omitted. If you have just done Exercise 1 and the data file and a log file are still open, there is no need to close either. The output from Exercise 2 will be added to your log file.

```
reg EARNINGS S
```

Exercise 3 *t* tests of regression coefficients and confidence intervals

Use the regression results from Exercises 1 and 2.

Exercise 4 *F* test of the goodness of fit of a regression equation

Use the regression result from Exercise 1.

Exercise 5 Multiple regression analysis

```
reg S ASVABC SM
reg S ASVABC SF
reg S ASVABC SM SF
```

Tip : After running the first regression, press the Page Up key, change SM to SF, and press the Enter key. For the third regression, press Page Up again and add SM to the list of variables. This will save some typing.

Exercise 6 Multiple regression analysis

```
reg EARNINGS S EXP
```

Exercise 7 Multiple regression analysis

```
reg S ASVABC SF
predict ES, resid
reg SM ASVABC SF
predict ESM, resid
scatter ES ESM
reg ES ESM
```

Note : The graph produced by the `scatter` command does not go to the log file and has to be printed separately. If you want a graph for the log file, use `plot` instead of `scatter`. The

Exercise 9 Correlated explanatory variables

```
reg S SM SF ASVAB2 ASVAB3 ASVAB4
reg S SM SF ASVABC
cor ASVAB2 ASVAB3 ASVAB4
```

Exercise 10 Correlated explanatory variables: use of a restriction

```
reg SIBLINGS SM SF
cor SM SF
gen SP = SM + SF
reg SIBLINGS SP
```

Exercise 11 *F* tests of the goodness of fit in a multiple regression model

Use the regression results for Exercises 5 and 6.

Exercise 12 Nonlinear regression analysis

```
gen LGEARN = ln(EARNINGS)
reg LGEARN S EXP
```

Save the data set, with the same name as before, after defining *LGEARN* . If you do not, you will have to define *LGEARN* again the next time you wish to use it.

Exercise 13 Nonlinear regression analysis

```
gen LGWT85 = ln(WEIGHT85)
gen LGHEIGHT = ln(HEIGHT)
reg LGWT85 LGHEIGHT
```

Exercise 14 Box–Cox test

```
sum LGEARN
```

In the next instruction, replace *Z* with the mean of *LGEARN* from the sum command:

```
gen EARNSTAR = EARNINGS/exp(Z)
gen LGEARNST = ln(EARNSTAR)
reg EARNSTAR S EXP
reg LGEARNST S EXP
```

Exercise 17 Regression analysis with multiple categories of dummy variable

```
reg S ASVABC MALE SM SF ETHBLACK ETHHISP
```

Exercise 18 Regression analysis with multiple categories of dummy variable

```
reg LGEARN S EXP MALE ETHBLACK ETHHISP
```

Exercise 19 Ftest of the explanatory power of a group of dummy variables

Use the regression results from Exercises 15–18.

Exercise 20 Evaluation of the effect of changing the omitted category in a regression with dummy variables

```
reg S ASVABC MALE SM SF ETHWHITE ETHHISP
```

Exercise 21 Evaluation of the effect of changing the omitted category in a regression with dummy variables

```
reg LGEARN S EXP MALE ETHWHITE ETHHISP
```

Exercise 22 Dummy variable trap

```
reg LGEARN S EXP MALE FEMALE
```

Exercise 23 Slope dummy variable

```
gen MALEASVC = MALE*ASVABC
```

```
reg S ASVABC SM SF ETHBLACK ETHHISP MALE MALEASVC
```

Exercise 24 Slope dummy variable

```
gen MALES = MALE*S
```

```
reg LGEARN S EXP ETHBLACK ETHHISP MALE MALES
```

Exercise 25 Interactive dummy variable

```
gen MALEBLAC = MALE*ETHBLACK
```

```
gen MALEHISP = MALE*ETHHISP
```

```
reg S ASVABC SM SF MALE ETHBLACK ETHHISP MALEBLAC MALEHISP
```

Exercise 27 Chow test

```
reg LGEARN S EXP ETHBLACK ETHHISP
reg LGEARN S EXP ETHBLACK ETHHISP if MALE==1
reg LGEARN S EXP ETHBLACK ETHHISP if MALE==0
```

Exercise 28 Comparison of a Chow test with an *F*test using a full set of dummy variables

```
gen MALESM = MALE*SM
gen MALESF = MALE*SF
reg S ASVABC ETHBLACK ETHHISP SM SF MALE MALEASVC MALEBLAC
MALEHISP MALESM MALESF
reg S ETHBLACK ETHHISP ASVABC SM SF
cor MALE MALEASVC MALEBLAC MALEHISP MALESM MALESF
```

Exercise 29 Comparison of a Chow test with an *F*test using a full set of dummy variables

```
gen MALEEXP = MALE*EXP
reg LGEARN S EXP ETHBLACK ETHHISP MALE MALES MALEEXP
MALEBLAC MALEHISP
reg LGEARN S EXP ETHBLACK ETHHISP
cor MALE MALES MALEEXP MALEBLAC MALEHISP
```

Exercise 30 Oaxaca decomposition of the difference in male and female earnings

```
reg LGEARN S EXP ETHBLACK ETHHISP if MALE==1
sum LGEARN S EXP ETHBLACK ETHHISP if MALE==1
reg LGEARN S EXP ETHBLACK ETHHISP if MALE==0
sum LGEARN S EXP ETHBLACK ETHHISP if MALE==0
```

Exercise 31 Omitted variable bias

```
reg S ASVABC SM
reg S ASVABC
reg S SM
```

Exercise 32 Omitted variable bias

```
reg LGEARN S EXP
reg LGEARN S
```

Exercise 34 Redundant explanatory variable

```
reg LGEARN S ASVABC EXP MALE ETHBLACK ETHHISP
reg LGEARN S ASVABC EXP MALE ETHBLACK ETHHISP SIBLINGS
cor SIBLINGS S ASVABC EXP MALE ETHBLACK ETHHISP
```

Exercise 35 Proxy variables

```
gen PWE = AGE - S - 5
reg LGEARN S ASVABC MALE ETHBLACK ETHHISP
reg LGEARN S ASVABC MALE ETHBLACK ETHHISP PWE
reg LGEARN S ASVABC MALE ETHBLACK ETHHISP EXP
```

Variation :

```
reg LGEARN S ASVABC MALE ETHBLACK ETHHISP if MALE==1
reg LGEARN S ASVABC MALE ETHBLACK ETHHISP PWE if MALE==1
reg LGEARN S ASVABC MALE ETHBLACK ETHHISP EXP if MALE==1
reg LGEARN S ASVABC MALE ETHBLACK ETHHISP if MALE==0
reg LGEARN S ASVABC MALE ETHBLACK ETHHISP PWE if MALE==0
reg LGEARN S ASVABC MALE ETHBLACK ETHHISP EXP if MALE==0
```

Exercise 36 Test of a restriction

```
gen EXPBEF = EXP - TENURE
reg LGEARN EXP S ASVABC MALE ETHBLACK ETHHISP
reg LGEARN EXPBEF TENURE S ASVABC MALE ETHBLACK ETHHISP
reg LGEARN EXP S ASVABC MALE ETHBLACK ETHHISP if MALE==1
reg LGEARN EXPBEF TENURE S ASVABC MALE ETHBLACK ETHHISP
if MALE==1
reg LGEARN EXP S ASVABC MALE ETHBLACK ETHHISP if MALE==0
reg LGEARN EXPBEF TENURE S ASVABC MALE ETHBLACK ETHHISP
if MALE==0
```

Exercise 37 *t* test of a restriction

```
reg LGEARN EXP S ASVABC MALE ETHBLACK ETHHISP TENURE
reg LGEARN EXP S ASVABC MALE ETHBLACK ETHHISP TENURE
if MALE==1
reg LGEARN EXP S ASVABC MALE ETHBLACK ETHHISP TENURE
```

Exercise 39 Measurement error, instrumental variables estimation and Hausman test

The instrumental variables estimation command in Stata is `ivreg`. It should be followed by the dependent variable, then the explanatory variables not requiring instrumentation, then, in parentheses, the variables requiring instrumentation, followed by an = sign and a list of the instrument(s). Thus the command for an IV regression of `LGEARN` on `S`, `ASVABC`, `MALE`, `ETHBLACK`, and `ETHHISP`, with `SM` instrumenting for `ASVABC`, is

```
ivreg LGEARN S EXP MALE ETHBLACK ETHHISP (ASVABC=SM)
```

To perform the Hausman test, continue with the following commands:

```
estimates store      name1
reg LGEARN S ASVABC EXP MALE ETHBLACK ETHHISP
estimates store      name2
hausman  name1 name2 , constant
```

`name1` and `name2` are names to be supplied by you to identify the IV and OLS regressions in question.

To use `SM`, `SF`, `SIBLINGS`, and `LIBRARY` as joint instruments for `ASVABC`, the first command should be

```
ivreg LGEARN S EXP MALE ETHBLACK ETHHISP (ASVABC=SM SF
SIBLINGS LIBRARY)
```

To perform the Hausman test, continue with the next three commands as before, but use different names for `name1` and `name2`.

Exercise 40 Simultaneous equations estimation

```
reg ASVABC S
ivreg ASVABC (S=SM)
```

Exercise 41 Binary choice models

```
gen COLL=0
replace COLL=1 if S>12
```

Exercise 42 Sample selection bias

```
g COLLYEAR = 0
replace COLLYEAR = S-12 if S>12
g LGARNCL = LGARN if COLLYEAR>0
heckman LGARNCL COLLYEAR ASVABC EXP MALE ETHBLACK ETHHISP,
select(ASVABC MALE ETHBLACK ETHHISP SM SF SIBLINGS)
```