

## PS 233 – Intermediate Statistical Analysis

### Stata Notes

This handout covers a few commands and tricks in Stata.

Chris Zorn of Emory University has prepared an excellent note on several key Stata commands. It can be downloaded from his web-page at:

<http://www.emory.edu/POLS/zorn/Classes/ICPSR2001/Stata4Dummies.pdf>

### USING DO-FILES

A basic structure of a do-file:

```
#delimit ;  
clear;  
set mem 15m;  
set more off;  
  
use "C:\myfile.dta";
```

```
{ your commands, estimation, data transformation, and so on}
```

```
save "C:\mynewfile.dta", replace;
```

*#delimit ;* This is a command that sets the character that marks the end of a command line. The character chosen is ; This command is helpful when writing do-files because otherwise commands should be written in a single line.

*clear*; This command clears the memory from any previous data you had. This means that you are about to start anew. Please, notice that I use ; at the end of the command line.

*set mem 15m*; This command sets the memory allocated to the data. Some memory is needed, but the greater amount of memory allocated to the data, the slower Stata becomes. So, try *15m*, and if that is not enough, increase it. (This is probably not a major concern with fast computers and big RAMs).

*set more off*; If you have used Stata for awhile, you have come across the message – *more* – on the Stata result window. That means that Stata is waiting for you to press a key before continuing the analysis. If you *set more off*, this does not happen and Stata will not pause. If you write do-files that run for some time, you might set more off, have a break, and let Stata work. (Sometimes you want to have the *more* message activated. Try to *set more off* and then type *help regress*.)

*use "C:\myfile.dta"*; This command opens your data file.

Once you have open your data set, you can work on it.

*save "C:\mynewfile.dta", replace*; This command saves your data file. Please, notice that I have chosen a new name for the data set. By saving the data file with a

different name, you can always go back to your original data file, and undo or modify what you did. This is probably one of the most important advantages of do-files: they store a record of all you did on a data file, and allow you to change what you did. The sub-command `, replace;` tells Stata to overwrite an existing data set. The rationale for this sub-command is that it keeps only the latest version of the “C:\mynewfile.dta” data file.

## **OPENING FILES WITH EXCEL**

`insheet using “C:\myexcelfile.csv”, comma;` If you have an Excel file, you need to save it as a CSV file (a comma delimited file). Then, the `insheet` command will read the file into Stata. On some occasions, you also need to include a variable list. So, the command would read `insheet var1 var2 var3 ... using “C:\myexcelfile.csv”, comma;`

## **OPENING A LOG-FILE**

`log using “C:\mylogfile.log”;` This opens a log file. To close it, just type `log close;`

## **REGRESSION, PREDICTED VALUES AND RESIDUALS**

`regress y x1 x2 ... ;` This is the basic command for regression.

To create predicted values  $E(Y_i) = \beta_0 + \beta_1 \times x_{1i}$ , there are two options.

*predict yhat*; This creates the fitted values and stores them in a variable called *yhat*.

Alternatively,

*gen yhat1=\_b[\_cons] + \_b[x1]\*x1*; This command creates the fitted values.

What is important here is that *\_b[\_cons]* is how Stata identifies  $\hat{\beta}_0$ , the constant in the regression model; and *\_b[x1]* is how Stata identifies  $\hat{\beta}_1$  the slope coefficient for the variable *x1*. What you write between brackets [ ] is the name of the variable. (So, if you have a model with several explanatory variables *x1*, *x2*, *x3*, ... then the coefficients are identified this way: *\_b[x1]*, *\_b[x2]*, *\_b[x3]*, ...)

To predict residuals,

*predict res, residuals*; This command computes the residuals and stores them in a variable called *res*.

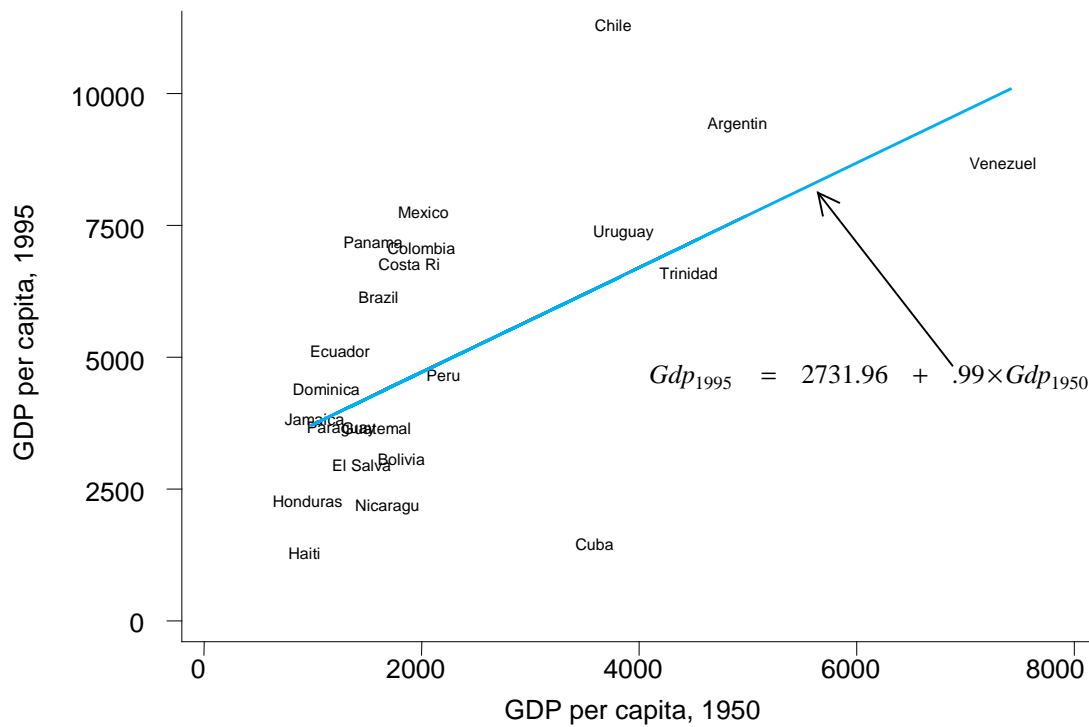
Check *help regress* to find what else *regress* and *predict* can do.

## GRAPHS

If you want to import Stata graphs into Word, you need to save them as Windows Metafiles (.wmf). (Click on File, select Save Graph, and then select the option window metafile). To import the graph into word: launch Word, click on insert, select picture, and

the sub-option “from file.” At this point find your file, and click insert. If you double-click on the figure, you can edit it.)

**Figure 1. Regression of Gdp in 1995 on Gdp in 1950**



**CREATING LAGGED VARIABLES (AND SORTING FILES)**

Let’s suppose that you have a data set with time-series data. Let’s suppose you are interested in the relationship between GDP at year t-1 and Democracy at year t (your

Year	Democracy <sub>t</sub>	GDP <sub>t</sub>	GDP <sub>t-1</sub>
1990	10	20	---
1991	8	22	20
1992	8	18	22

theory claims that there are delayed effects...) So, you need to create a variable that measures GDP at year t-1. Here are the steps you need to follow.

*sort year;*

*gen laggedp = gdp[\_n-1];*

*sort year;* This command sorts your data using the *year* variable.

*gen gdplag = gdp[\_n-1];* This command creates a variable called *gdplag* which is equal to the variable *gdp* the previous year. What you have between brackets *[\_n-1]* is what tells Stata to shift the column of GDP data down one cell. If you need to lag the variable two years: *gen gdplag2 = gdp[\_n-2];* three more years: *gen gdplag3 = gdp[\_n-3];* ...

A complication emerges if you have a pooled time-series cross-section data set,

Cname	cnumber	year	gdp	gdplag	<b>Problem</b>
Brazil	1	1980	10	---	---
Brazil	1	1981	12	10	<b>10</b>
Brazil	1	1982	14	12	<b>12</b>
<b>Brazil</b>	1	1983	13	14	<b>14</b>
<b>Argentina</b>	2	1980	9	---	<b>13</b>
Argentina	2	1981	8	9	<b>9</b>
Argentina	2	1982	11	8	<b>8</b>
Argentina	2	1983	10	11	<b>11</b>

that is, a data set that records data over time for several countries, a typical data set in comparative politics and IR, and elsewhere.

The problem is that when you type: *gen laggedp = gdp[\_n-1]*; Stata literally shifts a column of data down one step. So, the risk is that of using the value of Gdp in Brazil in 1983 to code the  $Gdp_{t-1}$  variable for Argentina in 1980. The table should clarify the problem.

How to solve this issue. Follow these steps.

*sort cnumber year;*

*by cnumber: gen gdplag=gdp[\_n-1];*

*sort cnumber year;* This command sorts the data by country and by year. Pooled Time-series Cross-Section data sets have a variable that identifies countries using some conventional number.

*by cnumber: gen gdlag=gdp[\_n-1];* This command tells Stata to create a 1-year lag variable for the variable gdp country by country.