

Intro to STATA Lecture Notes
Stuart Soroka
Department of Political Science, McGill University
January 2010 *

Introduction to STATA data management, graphing, and bivariate regression commands, using the Lijphart *Patterns of Democracy* dataset.

The Basics

The four (or five, or six) windows:

The **review window** logs all commands (from the command window) as they are entered. Click on an old command, and it will appear again in the command window.

The **variables window** lists all variables in the working file. Click on a variable, and it will appear in the command window.

The **command window** is where all your commands are typed.

The **results window** displays results. It can display only a limited number of lines at a time. If your results are going to be very long, use a **log file**.

The **do-file editor** is a workspace where you can write, edit, and save Stata commands. Rather than entering these commands in the command window, you can run them from the do-file editor. The advantage is that you can easily edit and save all your commands.

The **data editor** allows you to enter, view, or edit your data file. It looks like a spreadsheet. Typically, variables are listed across the top, and cases are listed down the side. This window must be closed in order to run commands in Stata.

Before the Analysis

Datasets can be opened and saved using the icons at the top of the screen. If you have a dataset that is not in Stata format, you can use a separate program called *StatTransfer* to translate the dataset from its current format into Stata format. The advantage of using this program is that it will retain any variable or value labels in the original file.

Stata loads the working dataset into memory. Depending on the version of Stata, the program will allocate a certain amount of memory to storing the data. If the dataset

* These notes are distributed freely, and change regularly based on course requirements. They also borrow liberally from Christopher Zorn's *Stata For Dummies* and various STATA manuals and texts. I apologize for any errors.

you are going to use is very large (ours is this week), you'll need to expand the amount of memory Stata uses. This amount should be at least as large as the dataset you're using, but cannot be larger than the amount of memory left on your computer.

This week's dataset is pretty small, but just for practice we'll allocate 20m of memory to data using the **set memory** command:

```
.set memory 20m
```

You also can check the amount of memory Stata is using by typing **.memory**. If you're running very large models, you might also need to change the maximum number of variables Stata permits in a model. The initial value for STATA8 SE is 400. If you'll need more, use the **matsize** command. The following example changes the maximum number of variables allowed to 500:

```
.set matsize 500
```

It is best ('though not crucial) if you do this before you open your dataset.

Also, if you want to keep track of everything you do in your session, use the STATA log. Go to File > Log > Begin to start a log file, or click on the log button (a scroll). Select a location to save the file, and let it run while you do your work. Everything that scrolls by on your Results window – commands and results – will be recorded in this log file.

Learning Stata Commands

One of the strengths of the Stata program is that it uses relatively simple language syntax. Almost all commands follow this structure:

```
.[by varlist:] command [varlist] [=exp] [if exp] [in range] [weight] [, options]
```

where *varlist* denotes a list of variable names, *command* denotes a Stata command, *exp* denotes an algebraic expression, *range* denotes an observation range, *weight* denotes a weighting expression, and *options* denotes a list of options. Not all commands will use every element. In fact, anything in square brackets is optional, so some command lines may simply be the command itself. The most common elements are probably just:

```
. command [varlist] [, options]
```

That said, knowing this structure will help you understand new commands. First, these new commands will invariably be based on this structure. Secondly, the description of the new command in the **help** file will begin by showing the syntax for that command, using the same elements as those listed above.

The **online help** and **search** facilities in Stata, thankfully, mean that you never really have to remember each specific command. The easiest way to use the help command is by using the drop-down Help menu at the top of the screen.

Setting up the Data

Once you have opened (or entered) your dataset, you'll want to take a preliminary look at the variables. There are several commands that are particularly useful:

.list [*varlist*] [*if exp*] [*in range*] [, **[no]display nolabel noobs doublespace**] – lists the values of variables.

.codebook [*varlist*] [, **all header notes mv tabulate (#)**] – produces a codebook describing the dataset

.inspect [*varlist*] [*if exp*] [*in range*] – displays a summary of a variable, including a small histogram

.describe [*varlist*] [, **short, detail fullnames numbers**] – described contents of data in memory.

.summarize [*varlist*] [*weight*] [*if exp*] [*in range*] [, **[detail | meanonly | format**] – provides summary statistics, such as means and standard deviations.

Recoding and Transforming the Data

Before you begin your analysis, it is likely that you'll want to recode or transform some variables. These are the four most valuable commands:

.generate [*type*] *newvar[:lblname]* =*exp* [*if exp*] [*in range*] – creates a new variable.

.replace *oldvar* =*exp* [*if exp*] [*in range*] [, **nopromote**] – changes the contents of an existing variable.

.recode *varname rule [rule...] [*=el]* [*if exp*] [*in range*] – recodes a categorical variable.

.drop *varlist* or **.drop if** *exp* or **.drop in range** [*if exp*] – eliminates cases or variables

.keep *varlist* or **.keep if** *exp* or **.keep in range** [*if exp*] – eliminates cases or variables (same as **.drop**, but more useful if you are dropping more than you're keeping).

For example, in the Lijphart dataset, the variable *elecsys* gives the type of electoral system in each country. To look at the this variable, use the command

.list countryt elecsys

The second column includes values for *elecsys*. What we are seeing are the value labels, rather than the values of the variable itself. To see the numeric value for each of the variable labels, type

.label list elecsys

Say we want to generate a variable that separates the Pluralist system countries from

all others – a ‘dummy’ variable equal to 1 for Pluralist systems and equal to 0 otherwise. We could do so using the command

```
.generate plsys = 0
.recode plsys 0 = 1 if elecsys == 2
```

Generate can also perform mathematical calculations. You can replicate the variable *numiss*, for instance, as follows:

```
.generate numiss2= iss_eco+ iss_rel+ iss_cul+ iss_urb+ iss_reg+ iss_for+ iss_post
```

Better yet, we can create a new variable we’ll use later. First, drop the variable you’ve just created,

```
.drop numiss2
```

Now, let’s look at the distribution of the *numiss* variable,

```
.summarize numiss, detail
```

Now let’s create a new *numiss2* variable dividing the sample in two as follows:

```
.generate numiss2 = 0
.recode numiss2 0=1 if numiss>=2.5
```

Tables of Frequencies

The first step in searching for uni- or multi-variate trends is often to create a table of frequencies or summary statistics. For categorical variables, tables of frequencies are more useful. These can be generated in several ways in Stata. The most valuable multi-purpose command is **.tabulate**.

The simple one-variable version of the *tabulate* command is as follows:

```
.tabulate variable
```

Or, for instance,

```
.tabulate numiss
```

The options for this command are complicated enough that they won’t be repeated here. We’ll use our knowledge of the Stata command language and the online help facility to use the *.tabulate* command below.

In the meantime, there are two particularly common uses of the *tabulate* command. The first is using it to create a two-way table:

```
.tabulate varname1 varname2 [weight] [if exp] [in range] [, column row nofreq]
```

Or, for instance,

```
.tabulate numiss numiss2
```

which is useful to check if our previous coding has worked. For something more interesting, try

```
.tabulate elecsys numiss2
```

The options listed above are a small minority of those available, but they're probably the most useful. **column** displays column percentages; **row** displays row percentages; **nofreq** suppresses printing the frequencies. If you want to run a two-way table that displays column percentages only, then, you could write:

```
.tabulate varname1 varname2, column nofreq
```

Or, for instance,

```
.tabulate elecsys numiss2, col nofreq
```

Better yet, you could do this and test the hypothesis that there is a relationship between these two nominal variables, using a chi square test. To do so, though, instead of typing in the entire command do one of the following:

- click on the last command in the Review window. It will appear in the Command window, and you can just add **chi** at the end, or
- press [Page Up] on your computer keyboard. The last command will appear in the Command window, and you can just add **chi** at the end, to make,

```
.tabulate elecsys numiss2, col nofreq chi
```

The secondly particularly valuable use of the `.tabulate` command is to combine it with the `.generate` command:

```
.tabulate variable, generate(variable)
```

This command creates indicator variables from a categorical variable. If we want dummies for each electoral system, we can type:

```
.tabulate elecsys, generate(elecsys)
```

Stata automatically gives names to the new variables. (This can also be done with continuous variables – see the Stata online help facility.) Let's have a look at the new variables, and how they relate to the original:

```
.list countryt elecsys elecsys1 elecsys2 elecsys3 elecsys4 elecsys5
```

Variable and Value Labels

We've created new variables manually, and using the `tabulate/generate` command above. The latter set of variables is poorly named, and all the new variables lack labels and value labels – in short, slightly longer descriptions of each variable, and descriptions of the values for categorical variables. The most useful commands for creating names and labels are listed below

To change the name of an existing variable:

```
.rename old_ varname new_ varname
```

To add a variable label to an existing variable:

```
.label variable varname ["label"]
```

To add variable labels, you must define the labels (first line), and then attach those labels to the variable:

```
.label define lblname # "label" [# "label"... ]
```

```
.label values varname [lblname]
```

So just to be thorough, let's add variable and value labels to numiss2:

```
.label variable numiss2 "Issue Dimensions dummy"
```

```
.label define num 0 "less than 2.5" 1 "2.5 or more"
```

```
.label values numiss2 num
```

And to see how they work, generate the table again:

```
.tabulate elecys numiss2, col nofreq chi
```

A Conceptual Map of Democracy – Examining the Lijphart Data

One of Lijphart's main theses is as follows:

"Ten differences with regard to the most important democratic institutions and rules can be deduced from the majoritarian and consensus principles. Because the majoritarian characteristics are derived from the same principle and hence are logically connected, one could also expect them to occur together in the real world; the same applies to the consensus characteristics. All ten variables could therefore be expected to be closely related. Previous research has largely confirmed these expectations – with one major exception: the variables cluster in two clearly separate dimensions... The first dimension groups five characteristics of the arrangement of executive power, the party and electoral systems, and interest groups. For brevity's sake, I shall refer to this first dimension as the *executive-parties dimension*. Since most of the five differences on the second dimension are commonly associated with the contrast between federalism and unitary government... I shall call this second dimension the *federal-unitary dimension*." (pg2-3)

The centrepiece of this discussion comes in Chapter 14, where Lijphart presents a scatterplot of democracies across the two dimensions. We can replicate this diagram using the following commands:

In our dataset, all variables measure degrees of consensus democracy, but Lijphart's diagram uses degrees of majoritarian democracy. The `.generate` commands, then, create two new variables, and reverse the signs (positive to negative, negative to positive) so our diagram will look like Lijphart's.

```
.generate first = firstdim*-1
```

```
.generate second = secdim45*-1
```

```
.label variable first "Executive Parties Dimension"
```

```
.label variable second "Federal-Unitary Dimension"
```

Let's just get a sense for the distribution of these variables by generating histograms as follows:

```
.graph twoway histogram first
```

We could use more columns, so increase the number using the bin command:

```
.graph twoway histogram first, bin(20)
```

Try also

```
.graph twoway histogram second
```

```
.graph twoway histogram second, bin(20)
```

We'll now using the graphing commands to creates the graph as Lijphart shows it in Chapter 14. Start with

```
.graph twoway scatter second first
```

Add lines

```
.graph twoway scatter second first, xline(0) yline(0)
```

And now add labels and a title

```
.graph twoway scatter second first, xline(0) yline(0) mlabel(countryt) title ("Lijphart's Conceptual Map of Democracy")
```

Is there a strong relationship between the two dimensions? Check by drawing a regression line. What we're going to do is put parentheses around the commands for the graph we're already drawing, beginning at scatter, and then add another graph type in separate parentheses at the end of the line:

```
. graph twoway (scatter second first, xline(0) yline(0) mlabel(countryt) title ("Lijphart's Conceptual Map of Democracy")) (lfit second first)
```

Bivariate Regression

Regression use the same basic syntax as the commands we've learned thus far. To review past commands, and start with regression, we'll review Lijphart's discussion in Chapter 8, on electoral systems.

Lijphart discusses the relationship between electoral system and the number of political parties; more precisely he uses 'electoral disproportionality' as a predictor of the number of political parties. He presents the following graph:

```
.graph twoway (scatter effparty disprop4, mlabel(countryt) title ("Disproportionality and Number of Parties"))
```

The x axis ends a little early, so we can be more specific about labeling

```
.graph twoway (scatter effparty disprop4, xlab(0 5 10 15 20 25) mlabel
```

(countryt) title("Disproportionality and Number of Parties"))

And we can draw the regression line

```
.graph twoway (scatter effparty disprop4, xlab(0 5 10 15 20 25) mlabel  
(countryt) title("Disproportionality and Number of Parties")) (lfit effparty  
disprop4)
```

Now let's generate the actual regression:

```
.regress effparty disprop4
```

This regression is a little strange, admittedly, since it suggests a causal relationship running from `effparty` to `disprop4`, and the two have more of a reciprocal relationship. The model does give us the equation for the line drawn in the diagram, however. The lack of a causal relationship here is likely why Lijphart presents the graph, along with the (non-directional) correlation coefficient:

```
.correlate effparty disprop4
```

Note that the R-squared from the regression is of course equal to the Pearson's r squared.

For better examples of the kind of causal relationship we require for regression models, we should examine results from Chapter 16, on 'kinder, gentler' consensual democracies. Table 16.1 presents results from bivariate regressions that we can replicate here. Try, for instance, the regression we did by hand last week:

```
.regress womenpar firstdi1
```

To finish, we'll try to replicate the regressions in Chapter 16. As you do, consider whether you agree with Lijphart's hypothesis. Chapter 16 tables are on the next page...

Table 16.1 Bivariate regression analyses of the effect of consensus democracy (executives-parties dimension) on seventeen indicators of the quality of democracy

	Estimated regression coefficient	Standardized regression coefficient	Absolute t-value	Countries (N)
Dahl rating (1969)	1.57***	0.58	3.44	26
Vanhanen rating (1980–88)	4.89***	0.54	3.75	36
Women's parliamentary representation (1971–95)	3.33***	0.46	3.06	36
Women's cabinet representation (1993–95)	3.36**	0.33	2.06	36
Family policy (1976–82)	1.10*	0.33	1.41	18
Rich-poor ratio (1981–93)	-1.41**	-0.47	2.50	24
Decile ratio (c. 1986)	-0.38**	-0.49	2.20	17
Index of power resources (c. 1990)	3.78*	0.26	1.57	36
Voter turnout (1971–96)	3.07*	0.24	1.46	36
Voter turnout (1960–78)	3.31*	0.30	1.49	24
Satisfaction with democracy (1995–96)	8.42*	0.36	1.55	18
Differential satisfaction (1990)	-8.11***	-0.83	4.51	11
Government distance (1978–85)	-0.34**	-0.62	2.51	12
Voter distance (1978–85)	-5.25**	-0.64	2.63	12
Corruption index (1997)	-0.32	-0.14	0.71	27
Popular cabinet support (1945–96)	1.90*	0.22	1.32	35
J. S. Mill criterion (1945–96)	2.51	0.07	0.42	35

*Statistically significant at the 10 percent level (one-tailed test)

**Statistically significant at the 5 percent level (one-tailed test)

***Statistically significant at the 1 percent level (one-tailed test)

Table 16.2 Bivariate regression analyses of the effect of consensus democracy (executives-parties dimension) on ten indicators of welfare statism, environmental performance, criminal justice, and foreign aid

	Estimated regression coefficient	Standardized regression coefficient	Absolute t-value	Countries (N)
Welfare state index (1980)	4.90***	0.68	3.70	18
Adjusted welfare index (1980)	4.29**	0.58	2.60	15
Social expenditure (1992)	2.66**	0.44	1.94	18
Palmer index (c. 1990)	4.99*	0.30	1.67	31
Energy efficiency (1990–94)	0.93***	0.51	3.50	36
Incarceration rate (1992–95)	-32.12*	-0.30	1.39	22
Death penalty (1996)	-0.35***	-0.44	2.86	36
Foreign aid (1982–85)	0.09*	0.30	1.38	21
Foreign aid (1992–95)	0.10**	0.39	1.86	21
Aid versus defense (1992–95)	5.94***	0.51	2.58	21

*Statistically significant at the 10 percent level (one-tailed test)

**Statistically significant at the 5 percent level (one-tailed test)

***Statistically significant at the 1 percent level (one-tailed test)