

PC Labs for Summer School: Intro

Brendan Halpin, Department of Sociology, University of Limerick

May 29 2023

Contents

1 Stata introductory session 1	1
1.1 Stata	1
1.2 Explore existing data	1
1.3 Generating new and recoding variables	2
1.4 Bivariate analysis	2
1.5 Missing values	3
2 Stata introductory session 2	3
2.1 Using the Do-File Editor	3
2.2 Logging	3
2.3 Statistical tests	3
2.4 Data Entry	4

1 Stata introductory session 1

1.1 Stata

Log on, and start Stata: Hit the Windows key, and type "Stata" in the **search** box.

The Stata window has four panels: the big one is for output, and the wide one at the bottom is for entering commands.

You can also use the mouse and menus, but we will focus on the command language.

1.2 Explore existing data

Load the [lab1.dta](#) file by entering the following command in the command window:

```
use http://teaching.sociology.ul.ie/ssrm/unitb0/lab1.dta
```

Use `describe`, `list`, `tab`, `summarize` to get an idea of what it contains.

In particular, `tab` and `summarize` let you look at how variables are distributed:

```
tab empstat
su grsearn
```

Generate bar charts of `empstat` by using the following commands:

```
graph hbar, over(empstat)
```

What does this command do?

```
graph hbar (mean) grsearn, over(lastexam)
```

Generate histograms of grsearn:

```
histogram grsearn  
histogram grsearn if grsearn > 0
```

1.3 Generating new and recoding variables

Creating new variables is done using generate, usually shortened to gen:

```
gen deduct = grsearn - netearn  
label variable deduct "Deductions from gross pay"  
su deduct  
scatter deduct grsearn
```

We can also recode variables. Note for instance that the 5th, 6th and 7th categories of empstat have very small numbers of cases – it is convenient to move them all into a single category. We do this by creating a copy of empstat, and recoding that.

```
recode empstat 5/7=7, gen(emp2)  
label values emp2 empstat
```

Note how we can give the new variable the old variable's labels.

1.4 Bivariate analysis

Crosstabulations are achieved by tab var1 var2, and percentages are entered like tab var1 var2, col or tab var1 var2, row. Explore some pairs of categorical variables in the data file. For example:

```
tab emp2 sex  
tab emp2 sex, col
```

Graphical equivalents of cross-tabulations are clustered and stacked bar charts:

```
graph hbar, over(sex) over(emp2)  
graph hbar, over(sex) over(emp2) asyvars  
graph hbar, over(sex) over(emp2) asyvars stack  
graph hbar, over(sex) over(emp2) asyvars stack percentages  
graph hbar, over(emp2) over(sex) asyvars stack percentages
```

We can compare mean income across qualifications like this:

```
bysort lastexam: su netearn  
graph hbar (mean) netearn, over(lastexam)
```

Try also:

```
graph hbar (mean) netearn, over(sex)  
graph hbar (mean) netearn, over(sex) over(lastexam)  
graph box netearn, over(lastexam)
```

1.5 Missing values

2 Stata introductory session 2

2.1 Using the Do-File Editor

Find the icon for the do-file editor and open it. Try running commands from it (start from scratch with the lab1.dta file, for instance). It is immediately useful when you want to enter a series of commands, e.g.:

```
clear
use http://teaching.sociology.ul.ie/ssrm/unitb0/lab1.dta
recode empstat 5/7=7, gen(emp2)
label values emp2 empstat
graph hbar, over(emp2)
```

It can also be a good way to build up files that achieve complex tasks, like going from loading a data file, through multiple data manipulation, to producing a specific result. To try this, build up commands in the do-file editor which load the lab1.dta file, do the recode and graph the mean income in each empstat group. Run it from the do file editor (you may need to make `clear` the first command), and then once it works, save it. Then run it from Stata: enter the command "`=do = file.do`" at the Stata command line.

2.2 Logging

You can log your activities to a file. First, turn on logging:

```
log using mylogfile.log, replace
```

Continue doing analysis for a while. Then close the log:

```
tab emp2 sex, col
log close
```

The log represents a record of your activities. You can examine it in any text-editor, or in Stata like this:

```
view mylogfile.log
```

2.3 Statistical tests

Generate a confidence interval around mean net earnings

```
ci mean netearn
```

Download [this spreadsheet \(csv\)](#) file and load it into Stata using

```
clear
import delimited using http://teaching.sociology.ul.ie/ssrm/unitb0/ttest.csv
```

This is paired data – carry out a paired-sample ttest like this:

```
ttest before == after
```

Then calculate the difference between the two, and carry out a one-sample ttest thus:

```
gen diff = after - before
ttest diff == 0
```

How do they compare?

To carry out an independent sample t-test, reload [lab1.dta](#). Test whether earnings differ across gender:

```
use http://teaching.sociology.ul.ie/ws/lab1.dta, clear
ttest netearn, by(sex)
```

Finally, test for association between empstat (recoded if you like) and gender:

```
tab empstat sex, chi
```

2.4 Data Entry

There are lots of ways to enter data into Stata. See above, how we used `import delimited` to import a CSV file. You can also import whole Excel files, if they are in the correct structure (optional headings row, variables in columns, no extraneous material). We can also enter directly into the Stata Data Editor.

See the data in [dataentry.html](#). Try entering it (or part of it) in Stata in the following ways:

- Using the Data Editor
- Copying the data to a file and using `infile age sex reg maths dist using datafile.dat`
- Copying the data to a spreadsheet, saving as CSV and doing `insheet age sex reg maths dist using datafile.dat`

You can label it in the data editor, or better in syntax (in a do-file!):

```
label define malefemale 1 "Male" 2 "Female"
label values sex malefemale
```