Two Page Stata

An introduction to Stata in 2 pages. Commands that you actually type into Stata are represented in `courier font`. `xvar` and `yvar` refer to variables in your data. The treatment here is intended to be extremely brief, in order to create a kind of "cheat sheet" that can be presented in 2 pages. More documentation on any command is available in the printed or PDF Stata manuals, or by typing `help` *`command`*.

Save Your Work

`log using` *`filename,`* `replace` will save a log file of your work. `log close` closes the log file.

Get Acquainted With Your Data

`lookfor` allows you to find variables that contain a specified keyword. This is especially useful in large data sets with many variables. Often abbreviated keywords are the most helpful. e.g. to find a poverty variable, type `lookfor pov`

`describe` tells you about the contents of a specific variable. E.g. `describe xvar yvar.`

`codebook xvar yvar` will produce a nicely formatted codebook of your data which is especially useful if you have added variable labels with the `label variable` command. `codebook` by itself will list every variable in your data and generate a lot of [probably too much] output.

Process Your Data

Data with missing values, often represented as negative numbers (e.g. -99, -9, -8) needs to be recoded so that the missing values are represented as a missing value character (".") that Stata knows to exclude from calculations.

`recode _all (-99/-1 = .)` will recode all negative numbers from -99 to -1 to missing for all variables in your data. `recode xvar (7/9 = .)` changes 7 through 9 to be missing for `xvar`. Indeed, `recode` will change specific values in your data to anything you want, not just missing values.

`recode xvar (oldvalue = newvalue), generate(zvar)`will recode a variable into a new variable, often a good idea.

It is often convenient to rename your data so that the variables have more intuitively understandable names e.g. `rename xvar depression`

You can create new variables out of old variables using `generate newvar = expression` e.g. `generate newvar = oldvar1 + oldvar2`

It is sometimes useful to sort your data. `sort xvar` will sort your data by the values of xvar.

Descriptive Statistics

`summarize` gives you basic descriptive statistics for a variable, such as the mean (average). Especially useful for continuous variables. E.g. `summarize xvar yvar` or `summarize xvar yvar, detail.`

`tabulate` gives you a frequency distribution for your variable. Especially useful for categorical variables. e.g. `tabulate xvar.`

Bivariate Statistics

Tabulating two categorical variables together gives you a cross-tabulation of those variables, e.g `tabulate xvar yvar, row col chi2`

`pwcorr xvar yvar, sig` gives you the pairwise correlation of two continuous variables.

`oneway continous_var categorical_var, tabulate` gives you a oneway ANOVA of a continuous variable over a categorical factor.

Multivariate Statistics

`regress yvar xvar` regresses y on x.
`regress yvar xvar zvar` regresses y on x and z. (Other regression commands follow a very similar format: `command yvar xvar zvar` but are beyond the purview of this 2 page guide.)
`regress yvar xvar i.zvar` regresses y on x and z, treating `xvar` as continuous and `zvar` as a <u>set</u> of categorical indicator variables.

Graphing

`histogram xvar` will give you a nice display of one variable. `histogram xvar, by(yvar)` may be useful for comparing the distributions of two variables over the categories of yvar.

`histogram xvar, percent` will scale the y-axis more intuitively in terms of percentages.
`histogram xvar, discrete` gives a nicer display for categorical variables.

`twoway scatter yvar xvar` gives you a twoway scatterplot of your data.

`sunflower yvar xvar` gives you a sunflower plot of your data.

`twoway lfit yvar xvar` will give you a linear fit graph.

The two syntaxes may be combined e.g. `twoway (scatter yvar xvar)(lfit yvar xvar)`

`graph bar xvar, over(yvar)` is useful for creating a bar graph of a continuous or categorical variable graphed across the categories of a categorical variable.

For all graphs, options after a “,” will be helpful in titling your graph
e.g. `twoway lfit yvar xvar, title(“…”) xtitle(“…”) ytitle(“…”)`

by:

In many cases you may want to look at the results of some calculation for xvar, or xvar and yvar over a third variable zvar. In such cases the `by:` syntax will be especially useful.

For example to look at the correlation of xvar and yvar over different values of zvar.

```
sort zvar
by zvar: pwcorr xvar yvar, sig
```

Comments, questions and corrections most welcome and may be sent to: Andrew Grogan-Kaylor (http://www.umich.edu/~agrogan) @ agrogan@umich.edu.

Last updated: November 29, 2010