Experimental Economics

Tutorial for Undergraduate Students

Summer Term 2016

Stata Tutorial

Tutor: Felix Albrecht

✉ f.albrecht@uni-bonn.de

1 General Rules for Data Work

- Never change the primary data directly.
- Avoid doing things "manually" (meaning not using code). *If you later on try to understand what you did, you won't know.*
- Always comment code. *Undocumented code is a PITA.*
- "Group" similar tasks, to keep code readable.
- Create an understandable folder structure. *The number of files is usually quickly increasing. (e.g. "log", "dta", "csv")*
- Stata uses the US number format. (Decimal numbers are written with a dot.)
- Missing data is marked by a •. It represents a very large random value. *Be careful with the > operator.*
- If you have trouble understanding Stata code, try reading it out aloud. *Stata code is frequently very close to human language.*

General Hints:

- If you are part of a larger project, write a comprehensive descriptive header.
- If you work with several version of a script or several co-authors time codes are useful for naming a file.
- If something requires a lot of manual, repetitive labor (e.g. formatting tables, graphs, transforming data) someone usually has written a script / addon / plugin for it. Time spent searching for such plugins is usually time well spent. *(Not too long.)*

Experimental Data Hints:

- Prepare scripts in advance. *Possible simulate a dummy dataset.*
  - *You create the experiment, so you know how the output data structure should look like.*
  - *You find possible variables of interest.*
  - *You can perform power calculations in advance.*

2 Online Ressources

- UCLA - Stata Help Website
- Stata Corp - Stata Graphics
- UCLA - Stata Graphics Help
- UCLA - What's the correct analysis?

3 Mini Stata Command Reference

Most Important:

- `help + cmd` - opens help dialog with infos on the command, usage, code examples.
- `findit + term` - indexed search. Searches for matches in all available help / description documents
- `if` - condition (`==, <, >, >=, <=, !=, =` = "equal, smaller than, larger than, larger of equal to, smaller or equal to, NOT, NOT") Appended to respective command, e.g. `su age if gender == 1`.

Header Commands:

- `set more off` - disables the `more` break for output that doesn't fit on the screen.
- `clear` - deletes the data from memory.
- `clear matrix` - deletes matrix data, not included in the above.
- `version 11` - forces Stata to use the behavior of a specific version. *(Downward compatibility.)*
- `log close` - closes the log. *(Usually end of file. But Stata throws error, if old log file still open.)*
- `log using ''filename.log''` - starts a "log" of all input and output.
- `cap` - `capture`s Stata error messages. *(If Stata throws an error script execution is stopped.)*
- `qui` - quietly. Keeps Stata from producing the screen output for this command.

Data Reading, Saving, Combining:

- `insheet using` - load csv data. *Watch the delimiter.*
- `outsheet using` - save csv data. *Watch the delimiter.*
- `use` - load `*.dta` - file. *(Stata binary data storage file.)*
- `save` - save `*.dta` - file. *(Stata binary data storage file.)*
- `merge` - merge files together. *(horizontal expansion)*
- `append` - appends rows in identical variables of two datasets. *(vertical expansion)*

Data Cleaning:

- `rename` - rename variables
- `label` - ... variables for descriptive purposes and codebook output. *(Also used to label values of a variable.)*
- `drop` - remove variables or observations from data.
- `keep` - opposite to drop.
- `recode` - values of a variable, e.g. invert factor variables.
- `reshape` - to convert data matrix from `wide` to `long`.
- `destring` - turn string into numbers. *Important when reading CSV files or standardizing text-variables to factor variables.*
- `string` - subgroup of functions of replace for working with strings *Frequently necessary to rewrite identifiers or standardize them.*

Working with Data:

- `br` - open data browser.
- `sort` - sort data from smallest to largest value.
- `order` - reorganize the variables of your dataset.
- `replace` - replace values in a variable or overwrite files.
- `gen` - generate values.
- `egen` - generate values, additional functionality (not interchangeable).
- `su` - `summary` summary of data (e.g. "mean","median",...) *fills* the `r`-scalar. *(See help `su` .)*
- `tab` - (cross)tabulate data. *(Get overview over values of one or two variables.)*
- `collapse` - makes dataset of summary statistics, e.g. `mean contribution by group`.

Experimental Economics Data Analysis:

- `spearman` - Spearman's Rank Correlation Coefficient.
- `ranksum` - Mann Whitney U test.
- `signrank` - Wilcoxon Signed Rank test.
- `tab` - (cross)tabulate data. *(Get overview over values of one or two variables.)*; perform χ^2 and Fisher's exact test.
- `kwallis` - Kruskal Wallis test.
- `reg` - linear regression.
- `logit` - Logit regression for binary dependent variable.
- `probit` - probit regression for binary dependent variable.

- `tobit` - Tobit regression for truncated/censored dependent variable.

Graphics:

- `hist` - draws a histogram.
- `graph twoway scatter` - draws a scatter plot
- `graph twoway line` - draws a line plot
  - `graph twoway (line var1 x-var) (line var2 x-var)` - draw two lines for two variables in one graph
- `graph bar` - draw barcharts

Intermediate Commands:

- `bysort` - automated function execution by grouping variable.
- `do + filename` - to link do files from with one another.
- `local i = 1` - defines a local variable `'i'`. *useful for loops*
- `global name var1 var2 var3` - defines a global variable. *useful for loops*
- `while 'i' <= 200 {...}` - a `While`-loop; runs as long as `i` is smaller than `200`
- `forvalues` - for-loop.
  - `forvalues j = 1/9` - for values 1 to 9
  - `forvalues j = 1,3,5 to 9` - for values 1, 3, 5 to 9
- `foreach` - for-loop – loops over list of variables.
- `preserve` - "stores" the current data layout
- `restore` - restores to previous "preserved" state.
- `duplicates` - gives you duplicate rows. *Be careful, only compares given columns. There might be differences in other columns.*
- `e` - Scalar that is filled with output statistics during a regression. *(Similar to `r`. See help `reg` or `su`.)*

Modules / Plugins:

- `ssc install module name` - installs modules from online repository.
- `estout` - Regression table formatting
- `outreg2` - Regression table formatting
- `lgraph` - Easy line graphs by grouping variable