Introduction to Stata

21 June 2010

J. James Reade

Plan

- Slides based on very helpful Stata course by Kerry Papps at Oxford.
- Introduce Stata:
  - Guide to display windows and toolbars.
  - Introduction to Stata Commands.
  - Data: Entering/loading, creating, manipulating, graphing.
  - Running regressions.
  - Using Log and Do files.
- Slides available at:

  `http://www.jamesreade.co.uk/.../StataSlides1.pdf`

  - Additional material available (data, do files, etc).

Stata Provision in Birmingham

- Stata is a powerful econometric/statistical package.
  - Massively used software package. Oodles of help available online.
- BBS has 56 student lab licences on a software server system:
  - All to be 56 located in Muirhead Tower.
  - Unless computer lab session taking place elsewhere: Server restricts access.
- Stand-alone student copies of Stata can be purchased from Timberlake:
  - Small Stata: £40.
    - ∗ Handles only small datasets.
  - Stata SE: £275.
    - ∗ Handles very large datasets.
  - Stata MP2: £560.
    - ∗ Handles very large datasets quicker than SE.

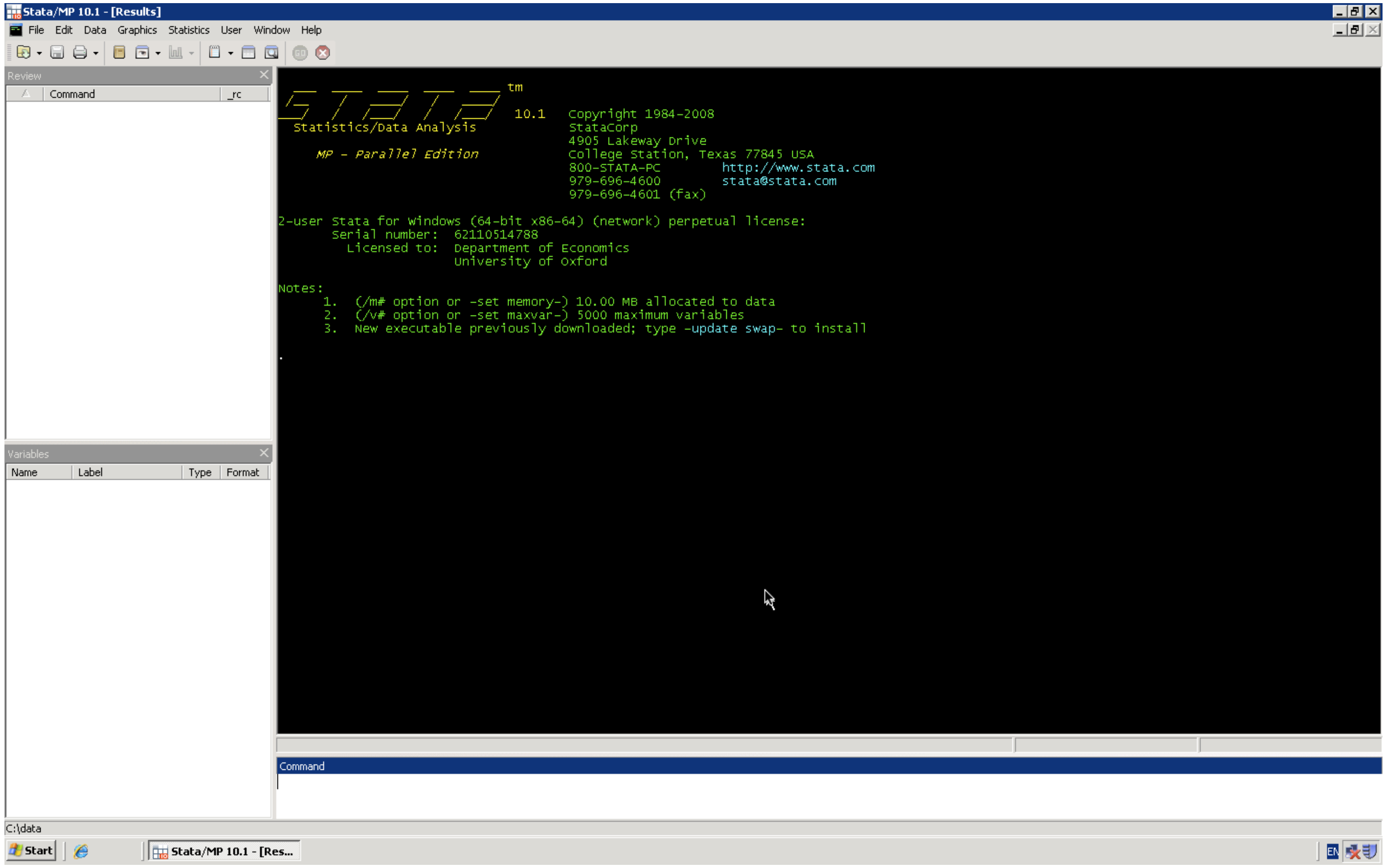
Format of Slides

- Slides, Stata-based and online help has syntax:

  `list` [*varlist*] [`in` *range*]

- `Typewriter style` means type exactly as shown into Stata.
  - Abbreviation of commands often possible.
- *Italicised text* means replace with desired variable name or option.
- `[`Square brackets`]` enclose optional Stata commands. Do not type the brackets in!

Navigating Around Stata



```
                                          tm
  ___  ____  ____  ____  ____
 /__    /   ____/   /   ____/
___/   /   /___/   /   /___/   10.1   Copyright 1984-2008
  Statistics/Data Analysis            StataCorp
                                      4905 Lakeway Drive
     MP - Parallel Edition            College Station, Texas 77845 USA
                                      800-STATA-PC        http://www.stata.com
                                      979-696-4600        stata@stata.com
                                      979-696-4601 (fax)

2-user Stata for Windows (64-bit x86-64) (network) perpetual license:
       Serial number:  62110514788
         Licensed to:  Department of Economics
                       University of Oxford

Notes:
      1.  (/m# option or -set memory-) 10.00 MB allocated to data
      2.  (/v# option or -set maxvar-) 5000 maximum variables
      3.  New executable previously downloaded; type -update swap- to install

.
```

- Results window, Command window, Review window, Variables window.

Get to Know Stata: Open It Up! Tasks

- Locate Stata 11 on your computer and open it.

- Identify the four windows: Results, Command, Review, Variables.

- Open the *Data Editor* either by:
  - **Command Window**: Typing `edit`.
  - **Toolbar**: Hitting *Data Editor* button.
  - **Drop-down Menus**: Data `-->` Data Editor.

- Type some numbers into cells and give the resulting variable a name.

- Exit *Data Editor* and clear memory using `clear` command.

- Investigate the *Help Facility* either by:
  - **Command window**: Typing `help`(*command name*).
  - **Drop-down Menu**: Help.

Ways to Run Stata

- Three ways to run Stata:
  1. Drop-down Menus.
  2. Interactive Mode.
  3. Batch Command Mode.
- **Drop-down Menus** (easiest for beginners):
  - All standard (most often used) commands accessible via drop-down menus.
- **Interactive Mode** (easiest for novices):
  - All commands can be typed into Command window and executed.
- **Batch Command Mode** (recommended):
  - Commands can be collected in a file (Do file) and run as batch.

Stata Commands

- Syntax is **case sensitive**: Must be lower case in Stata.
- Can abbreviate commands when no ambiguity possible.
- Useful initial command:
  - May need to increase standard memory limit from 1MB.

```
set memory #
```

- # must be number followed by `k` (kilobytes), `m` (megabytes) or `g` (gigabytes).
- E.g. `set memory 2g`. Can abbreviate to `set mem 2g`.

Opening Data

- Stata datasets have extension `.dta`.
  - Can import other file types: Later.
- Access data one of three ways:
  - **Drop-down menu**: File `-->` Open.
  - **Toolbar**: `Open (use)` button (furthest left).
  - **Command**: `use` *filename* [, `clear`].
- Only one dataset at a time can be open and in memory in Stata:
  - `clear` allows current dataset in memory to be replaced with `use` command.

Opening Data From the Internet

- *filename* can be file from the internet:
  - Vast archives of online datasets in Stata format.
  - E.g.: `http://www.stata-press.com/data/r11/r.html`
- Do not need to download data and save first:
  - Put weblink in place of *filename*
- Use first dataset from weblink: automiss.dta, so:

```
use ``http://www.stata-press.com/data/r11/automiss.dta'', clear
```

- Can later save dataset onto own computer.

Importing Data

- Data not always in Stata format. Options for non-standard data:
  - Copy and paste into *Data Editor*.
  - Use data transferring software (e.g. Stat-Transfer).
  - Import data into Stata.

- Drop-Down Menus: File --> Import: Importable data formats:
  - Generally ASCII data: `.csv` files probably easiest to use.
  - Can save any Excel file as a `.csv` file in Excel.

- Command: `insheet` [*varlist*] `using` *filename* [, *options*]
  - *options*: `tab`, `comma`, `delimiter(...)`, `clear`.
  - e.g.: `insheet using ``U:\WorldCupSoccer.csv'', comma`

- Other commands: `infile`, `infix`. Use `help` if intrigued.

Inspecting/Understanding Data

- Data stored either in numeric form (real numbers) or alphanumeric (string) form.
  - With `browse` can tell type: Red for alphanumeric, black numeric.
  - Type column in Variables window also gives information.
- `codebook`, `list` allow inspection of dataset:
  - Info on data type, label, missing values, descriptive stats.
  - Both can be restricted to particular variables.
- `tabulate` generates one/two-way tables of frequencies:

`tabulate` *rowvar* [*colvar*]

Immensely Useful Command: `if`

- `if` is conditional statement allowed within most commands.

- E.g. want to tabulate `goals` and `oppgoals` only when England playing.
  - General statement: `tab goals oppgoals`
  - Conditional statement: `tab goals oppgoals if england==1`

- Note double equals (==): For testing equality, single equals (=) for assignment.

- Logical operators can be used with `if` statements:
  - &: Denotes "and", can combine statements. `if england==1 & usa==1`
  - |: Denotes "or". `if england==1 | usa==1`
  - ! or ~: Denote "not", hence "is not equal to": `if england!=1`.

- Cannot use `if` statements on string variables.

Manipulating Data

- Usually want to alter existing data and create new variables.

- Create using `generate` *newvar* = *exp*
  - *exp* can be existing variables in dataset:
    ```
    gen goaldiff = goals-oppgoals
    ```
  - *exp* can involve `if` statements:
    ```
    gen win = 1 if goals>oppgoals
    ```

- Can alter contents of cells in already created variables also:

  `replace` *oldvar* = *exp1* [`if` *exp2*]

- Need `replace` to create dummy variables like `win`:
  - Currently has "." where `goals<=oppgoals`.
  - "." treated as *infinitely high number* by Stata.
  - Need: `replace win = 0 if goals<=oppgoals`.
  - Simpler code: `gen win = (goals>oppgoals)`

More Manipulation of Data

- Can `rename` variables:

  `rename` *oldvarname newvarname*

- Can `label` variables:

  `label variable` *varname “label”*

- Can delete, or `drop` variables:

  `drop` *varlist*

  - Or can `keep` variables:

    `keep` *varlist*
  - Can drop particular observations:

    `drop if` *exp*

Saving

- Having opened data (from internet or wherever) and manipulated it, want to save it.
- Can save as Stata datafile: `save` [*filename*] [, *options*].
  - Main option: `replace`, overwriting existing file with that name.
  - E.g. `save "C:\WorldCupSoccer.dta", replace`
- Can export back to original file type: `outsheet using` *filename* [, *options*]
  - Options are `comma` and other delimiters and `replace`.
  - E.g. `outsheet using "C:\WorldCupSoccer.csv", comma replace`

Data: Tasks

- Open dataset: "WorldCupSoccer.csv" (need to use `insheet`).
- Use `describe` to determine which variables are strings/numeric.
- Rename `result` variable as `outcome`.
- Label `goals` as "goals scored by object team".
- Delete variable `wc`.
- Create variable `worldcup` as sum of all World Cups since 1962.
- Create variable for World Cup host nation.[1]

[1] Hosts: 1962 Chile, 1966 England, 1970 Mexico, 1974 (West) Germany, 1978 Argentina, 1982 Spain, 1986 Mexico, 1990 Italy, 1994 USA, 1998 France, 2002 South Korea and Japan, 2006 Germany.

Data: Tasks

- Drop the variables wc1962, . . . , wc2006.
- Change `result` so that 0.5 denotes a draw and 1 a win.
- Create a variable called `points` denoting how many points team wins:
  - 3 points for a win, 1 for a draw, 0 for a defeat.
- Use `tabular` to describe the average football match outcome.
  - What percentage of matches do England win? Brazil?
- Save your modified dataset (not forgetting to use the `replace` option).
  - Now open the `.dta` version of `WorldCupSoccer` directly from internet.

Advanced Manipulation: Sorting Data

- `sort` puts all observations in dataset into specific order:

  `sort` *varlist*

- Can be useful for observing data, creating variables and merging data.
- E.g. Could sort by date, or by number of goals in game: `sort goals`.
- Can sort by more than one variable: E.g. by team then date: `sort team date`.

Apending, Merging and Collapsing

- Can only keep one dataset in memory but can combine datasets.
- Can append data to open dataset:

  `append using` *filename*

  - Adds extra variables at end.
  - Dataset in memory is **master dataset**.
  - Dataset *filename* is **using dataset**.
- But no matching of observations: Could be important.
  - E.g. Combining two panel datasets with different information on individuals.
- Instead can merge datasets:

  `merge` *varlist* `using` *filename*

Merging

- Stata will merge using common values of observations in *varlist*.
- *varlist* must be present in both datasets.
- Both master and using datasets must be sorted by *varlist*.
- Resulting merged datafile will have extra `_merge` variable:
  - Contains 1 if observation from master, 2 if from using, 3 if both.
  - If doing multiple merges will need to `drop _merge` inbetween.

Merging Exercise[2]

- Consider three datasets containing different information distinguished by `id`:
  1. `http://statistics.ats.ucla.edu/stat/data/stata_faq_multmerge/data1`
  2. `http://statistics.ats.ucla.edu/stat/data/stata_faq_multmerge/data1`
  3. `http://statistics.ats.ucla.edu/stat/data/stata_faq_multmerge/data1`
- Sort each dataset first on `id` and `save`.
- Keep third dataset open and merge other two using `merge`.
- Use `describe` command to consider merged dataset.

[2]From `http://www.ats.ucla.edu/stat/stata/faq/multmerge.htm`

By Group Processing

- May want to execute command separately for different parts of dataset.
  - E.g. Creating variable for individual unit in panel.

- Syntax: `by` *varlist*: *command*

- Need to sort by *varlist* first.
  - Most commands allow `by` prefix.

- **Task**: Create a form variable for football data:
  - Sort by team and date.
  - Use `points` variable to assess form:
    - Define form as **total points in last 4 matches**.
  - Calculate total points (`by team:`) and create form:
    - `by team:  gen form=totpoints`
    - `by team:  rep form=totpoints[_n]-totpoints[_n-4]`

Useful Data Creation Tips

- Data may be categorical: E.g. Responses to questionnaire.
- Effect of information in variable may not be linear.
  - E.g. Strongly agree may be much more likely to buy product.
- Stata allows easy creation of dummies for each 'layer' in a variable.
  - E.g. Create dummy for when variable is 1, when variable is 2.
  - E.g. Create dummy for when variable is "white", "green", "yellow", etc.

  `tab` *varname*, `gen` (*dummyname*)

  - *varname* is variable you want to split up.
  - *dummyname* is prefix to name.
- **Task:** Create dummy for every team using the `team` variable.

Collapsing

- Sometimes want to calculate statistics based on sections of data.
  - E.g. Average profits for company in panel.

`collapse` (*stat*) *varlist1* (*stat*)...[[*weight*]], `by`(*varlist2*)

  - *varlist*s are lists of variables separated by space.
  - *stat* can be `mean`, `sd`, `sum`, `median`,. . .
  - `by`(*varlist2*) specifies groups over which stats to be calculated.

- Be careful: No undo feature once collapsed data.
  - Best to save data before collapsing!

- E.g.: `collapse (mean) age educ (median) income, by(country)`

- **Task**: Create mean goals scored, conceded and points won for each team.

Drawing Graphs

- Final aspect of data manipulation and assessment before running regressions.
- Often want graphical representation of data in write-ups/papers.
- Some graphical commands:
  - `histogram` *varname*, `discrete freq`
  - `scatter` *varlist* [[*weight*]]
- Inserting into write-up documents:
  - Can copy and paste into a Word document. (saving also probably advisable)
  - Save as `.eps` file for importing into LaTeX.[3]
- **Task**: Create scatter plot of goals scored against goals conceded.
  - Save the file and insert it into Word/tex document.

[3] You may also need a converter to change `.eps` files into `.pdf` files for pdfLaTeX for example.

Doing Some Econometrics!

- So far we prepare for empirical work:
  - Load data, manipulate data, draw graphs.
- But now we can start doing actual econometrics!
- Three methods as before:
  - Drop-down menus: `Statistics` menu.
  - Interactive: Regression command followed by *varlist*.
  - Batch code: Do files. See later.

Linear Regression

- Simplest regression model: Ordinary Least Squares (OLS).

- Perform regression of *depvar* on *varlist*:

  `regress` *depvar varlist* [[*weight*]] [`if` *exp*] [, `noconstant, robust`]

  - *depvar*: Dependent variable.
  - *varlist*: Set of independent variables separated by spaces.
  - By default constant included, `noconstant` suppresses it.
  - `robust`: Huber-Weight heteroskedasticity robust standard errors reported.
  - *weights*: If want to run *weighted least squares (GLS)*.

- **Task:**

  - Regress match outcome (`outcome/result`) on `form` and `worldcuphost`.
  - Restrict estimation to just World Cup matches. Do results change?
  - Estimate your regressions using robust standard errors. Are your results affected?

Post-Estimation

- Many commands can be used post-estimation:
  - All refer to most recent model estimated.
- `predict`:
  - `predict` *varname*, `xb` created fitted values.
  - `predict` *varname*, `residuals` created residuals.
- `test`: Tests linear hypotheses (t- or F-tests):
  - `test` *varlist*: All elements of *varlist* jointly equal to zero.
  - `test` *eqlist*: Tests restrictions in *eqlist*, e.g. `test age==experience`.
  - `accumulate` option means hypothesis tested jointly with previous hypothesis.
- `prtab` *varlist*: Predicted values while varying variables in *varlist*.

Post-Estimation

- Stata has remarkable amount of postestimation information:
  - Type `help(regress)` to get some idea.
  - Syntax can be hard: Always try to use the Examples in Help files.
- **Tasks**:
  - Create fitted values and residuals for your variable.
  - Test the joint significance of your variables.

IV Regression

- Instrumental variable regression:
  - When some explanatory variables are endogenous.
    `ivregress` *estimator depvar exogvars* (*endogvars*=*ivvars*) [, *options*]
  - *estimator*: `2sls`, `liml`, `gmm`.
  - *exogvars*: Exogenous variables.
  - *endogvars*: Endogenous variables.
  - *ivvars*: Instrumental variables.

- **Tasks:**
  - Regress `wks_ue` on `tenure` from `nlswork` using OLS.
    - Estimate using IV with `hours` and `c_city` as instruments.
    - Compare your two regressions.
  - Type `help(ivregress)` and run the code under **Examples**.
  - Open `http://www.gseis.ucla.edu/courses/ed231c/notes3/instrumental.html`.
    - Follow the example provided there.

Other Types of Estimators

- Binary dependent variable:
  - Logit: `logit` *depvar indepvars*
  - Probit: `probit` *depvar indepvars*
- Categorical dependent variable:
  - Ordered probit: `oprobit` *depvar indepvars*
  - Ordinal logit: `ologit` *depvar indepvars*
  - Multinomial logit: `mlogit` *depvar indepvars*
- Tobit: `probit` *depvar indepvars*, `ll`(*cutoff*) `ul`(*cutoff*)
- Count data:
  - Poisson: `poisson` *depvar indepvars*
  - Negative binomial: `negbin`*depvar indepvars*

Other Estimators: Tasks

- Use the variable `win` from earlier:
  - Estimate the impact `form` has on probability of winning.
- Use the `result` variable:
  - Estimate an ordered probit and multinomial logit model.
    - Include `form` and `worldcuphost`.
  - Interpret the coefficients.[4]
  - Now include a dummy for each team at the World Cup. What happens?
- Use the `goals` variable:
  - Use a Poisson regression model to estimate the impact form has on goals scored.
  - Estimate a negative binomial regression model and decide which model is appropriate.[5]

[4]These links may be useful here: Ordered probit (http://www.ats.ucla.edu/stat/stata/dae/probit.htm), multinomial logit (http://www.ats.ucla.edu/stat/stata/output/stata_mlogit_output.htm)

[5]See for help interpreting output.

Panel Estimation

- Panel datasets increasingly available for economic analysis.
- Stata long regarded as excellent panel data software package.
- Panel datasets look like any other dataset:
  - But have time index variable and unit (e.g. firm, individual) index.
- Open from internet: `webuse nlswork`.
  - Inspect form of data using `browse`.[6]
- Need to tell Stata how to read the panel dimensions: `tsset` *panelvar timevar*.
  - *panelvar* is unit (e.g. firm), *timevar* is time variable.
  - E.g.: `tsset idcode year` for `nlswork` dataset.

[6] `edit` allows you to edit data cells if you want to.

Panel Estimation and the Time Dimension

- Once `tsset` used, Stata can exploit time dimension of data.
- May want to create lagged variables:

$$\texttt{gen}\ \mathit{varname} = \texttt{L.}\mathit{varname}$$

  - `L2.`*varname* is second lag and so on...

- Stata can be used for time series econometrics but it is not optimal:
  - See OxMetrics tomorrow.
  - Omit *panelvar* from `tsset` to declare data time series.
- **Tasks**:
  - Open nlswork dataset from the internet.
  - Use `tsset` to declare panel dimensions of dataset.
  - Create lagged variable for hours worked.

Panel Estimation

- Generally add `xt` before any regression command for cross section.
- `xtreg` *depvar indepvars* [, `re fe i(`*panelvar*`)`]
  - `i(`*panelvar*`)` can be omitted if `tsset` used.
  - `re` and `fe` are random and fixed effects estimation.
  - Could instead of `fe` enter dummies for time/units.
- **Tasks:**
  - Run cross section regression of `wks_ue` on `tenure`.
  - Estimate with fixed effects and random effects.
  - Include hours and lagged hours in your regression.

Hausman Test: Fixed vs. Random Effects

- Test which estimation method is more appropriate. Procedure:

  1. Run fixed effects regression and save estimates:
     ```
     xtreg depvar indepvars [, fe]
     estimates store fe_name
     ```
  2. Run random effects regression and save estimates:
     ```
     xtreg depvar indepvars [, re]
     estimates store re_name
     ```
  3. Use `hausman` command:[7]
     ```
     hausman fe_name re_name
     ```

- Can also use Hausman test for testing endogeneity.[8]

[7] Order of *fe_name* and *re_name* important in `hausman` command.

[8] See `http://www.gseis.ucla.edu/courses/ed231c/notes3/instrumental.html`.

Panel Tasks

- Return to nlswork dataset: Conduct Hausman test and interpret.
- Create dummy variables for cross-section units.[9]
  - Manually run fixed effect estimate.
  - Are the dummy coefficients significant?

[9] Use the `tab` *varname*, `gen` (*dummyname*) to create dummies.

More Exotic Panel Models

- Random coefficients model: `xtrc` *depvar indepvars* [, *options*]
- Dynamic (I.e. Including lags) panel models:
  - GMM: Various estimators using different moment conditions:
    - `xtabond`, `xtdpdsys`, `xtdpd`.
- If your thesis/dissertation involves panel estimation:
  - Stata online Help very useful indeed.
  - E.g. `http://www.stata.com/stata10/dpd.html`

Log Files

- After a while some output is lost at top of window.
  - Stata only keeps so much in Results window.
- Log file allows recording of *all* output.
  - Useful for reviewing later and for detail when writing up.
- Log files do not save graphics objects.

Outputting Results to LaTeX

- Usually want to write regression results up.
- Stata has excellent facility to output results for write-up:
  - `estout` package.[10]
- After regression use command `eststo` to store results in table.
- `estout` prints simple results table in Results window
- `esttab` produces publication-style results tables:
  - In Results window.
  - Into different file types and formats. *filename* ending in .tex is LaTeX output.

`esttab` [*namelist*] [`using` *filename*] [, *options*]

[10]See `http://repec.org/bocode/e/estout/`

Outputting Results: Tasks

- Store results of Poisson regressions on goals scored and goals conceded using `eststo`.
  - Ensure you use `eststo clear` before you start storing.
- Output results to a `.csv` file.
  - Open the resulting file in Excel: Does it look like you intended?
- Output results to a `.doc` file.
  - Open the resulting file in Word: Does it look like you intended?
- Output results to a `.tex` file.
  - Open the resulting file in WinEdt and compile: Does it look like you intended?

Do Files

- Do files very useful indeed: Collections of individual commands.
  - Generically known as batch files: Can run batch of commands.
- Stata's Do file editor has neat integration with rest of Stata.
- Do files are batch files: Good academic practice to get into.
  - Can recall what you did last time you upened data.
  - Good for writing up papers, making results available, replicable.
- **Task**: Collect all important commands relating to today in one `.do` file.

Random Useful Things to Know

- To operate a command only over certain observations: `in` *start*/*end*.
- `set more off`: Reports output without waiting for user to click.
- Post-estimation: `mfx` provides marginal effects.
- Correlation matrix: `correlate` *varlist*
- Create many dummy variables from string variable:

`tab` *varname*, `gen` (*dummyname*)

Concluding

- Course and slides are introduction to Stata.
- Provided tools to navigate Stata and get going.
- Wealth of online information on Stata: Google search usually helpful.