Introduction to Stata

UW CSSCR Workshop
Zhe Liu
July 2021

Purpose of the workshop

- This workshop introduces the basic usage of Stata for data analysis
- Topics include[1]
  - Stata as a data analysis software package
  - Navigating Stata
  - Data import
  - Exploring data
  - Data visualization
  - Data management
  - Basic statistical analysis

1. The content is modified from UCLA's online source "Introduction Stata for classes".

What is Stata?

- A software package for data management, statistical analysis, and graphics.
- A wide array of statistical tools that include both standard methods and newer, advanced estimation procedures
- Designed for researchers in the fields of econometrics, social science and biostatistics

Tools for data analysis, a comparison

Features	SPSS	SAS	Stata	JMP (SAS)	R	Python (Pandas)
Learning curve	Gradual	Pretty steep	Gradual	Gradual	Pretty steep	Steep
User interface	Point-and-click	Programming	Programming/ point-and-click	Point-and-click	Programming	Programming
Data manipulation	Strong	Very strong	Strong	Strong	Very strong	Strong
Data analysis	Very strong	Very strong	Very strong	Strong	Very strong	Strong
Graphics	Good	Good	Very good	Very good	Excellent	Good
Cost	Expensive (perpetual, cost only with new version). Student disc.	Expensive (yearly renewal) Free student version, 2014	Affordable (perpetual, cost only with new version). Student disc.	Expensive (yearly renewal) Student disc.	Open source (free)	Open source (free)
Released	1968	1972	1985	1989	1995	2008

Why Stata?

Advantages

- Intuitive data management capabilities
- Command driven, but can also be accessed via a point-and-click method
- Syntax is provided and consistent across commands, so easier to learn
- Exceptionally strong support for
  - Specialized statistical analysis
  - Panel data analysis

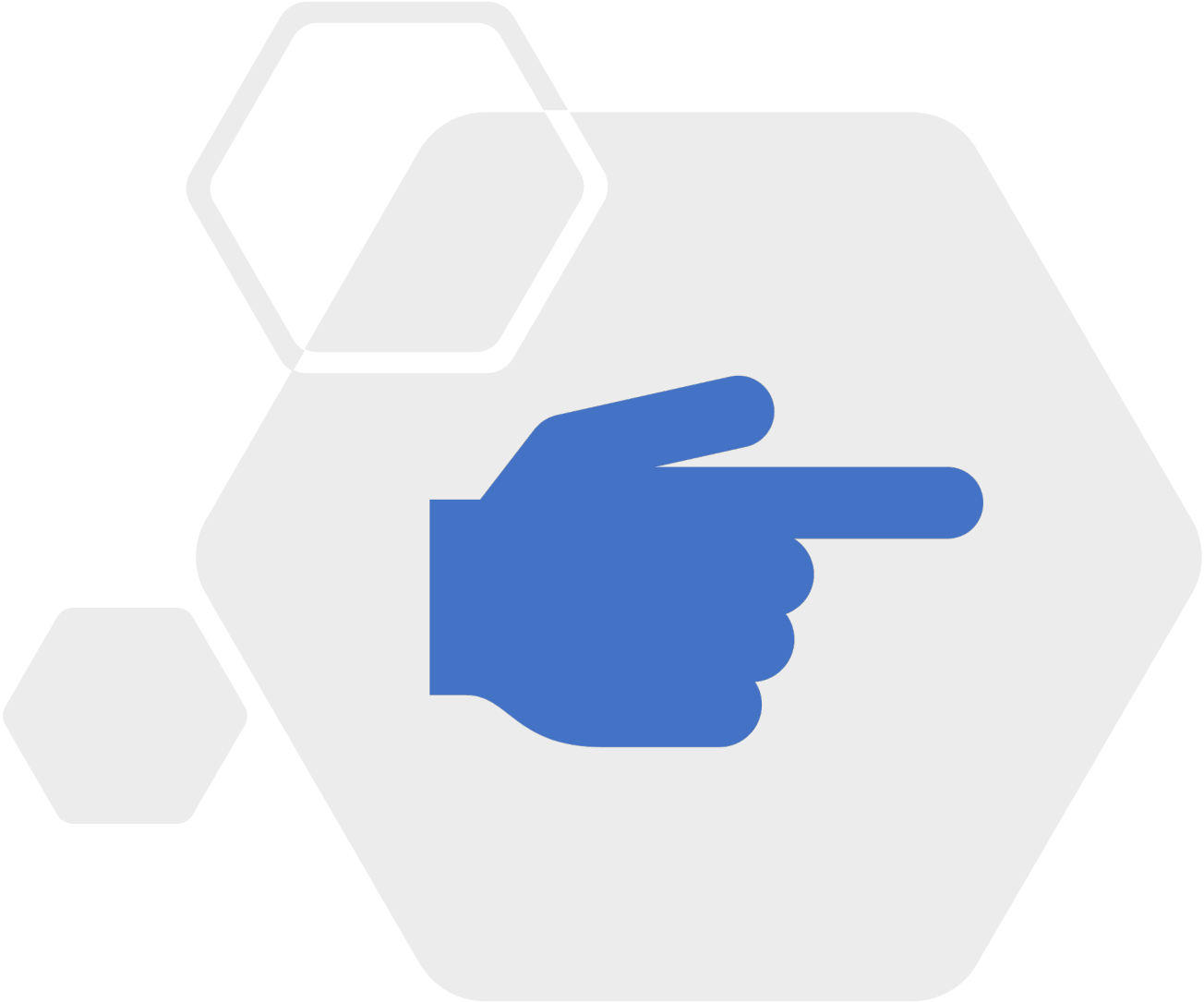
Disadvantages

- Limited to one dataset in memory at a time
- Appearance of output tables and graphics is somewhat dated and primitive
- Community is smaller than R
  - less online help
  - fewer user-written extensions

Acquiring and using Stata at UW

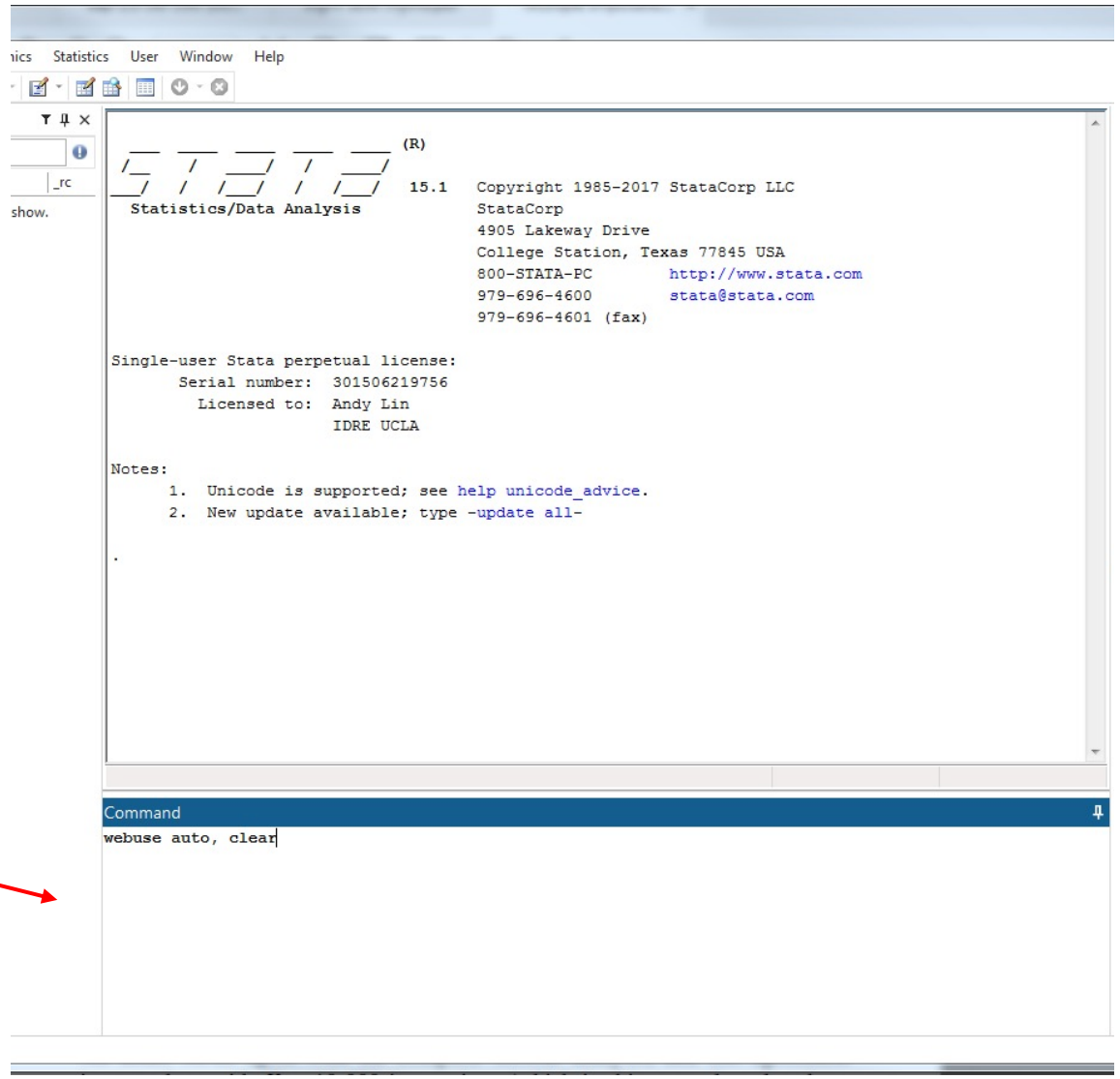
- Order and then download Stata directly from their website, but be sure to use GradPlan pricing, available to UW students
  - IC ≤ SE ≤ MP, regarding size of dataset allowed, number of processors used, and cost
- Stata is also installed in CSSCR labs and on remote desktops
  - https://depts.washington.edu/csscr/rdservices/

Navigating Stata’s interface



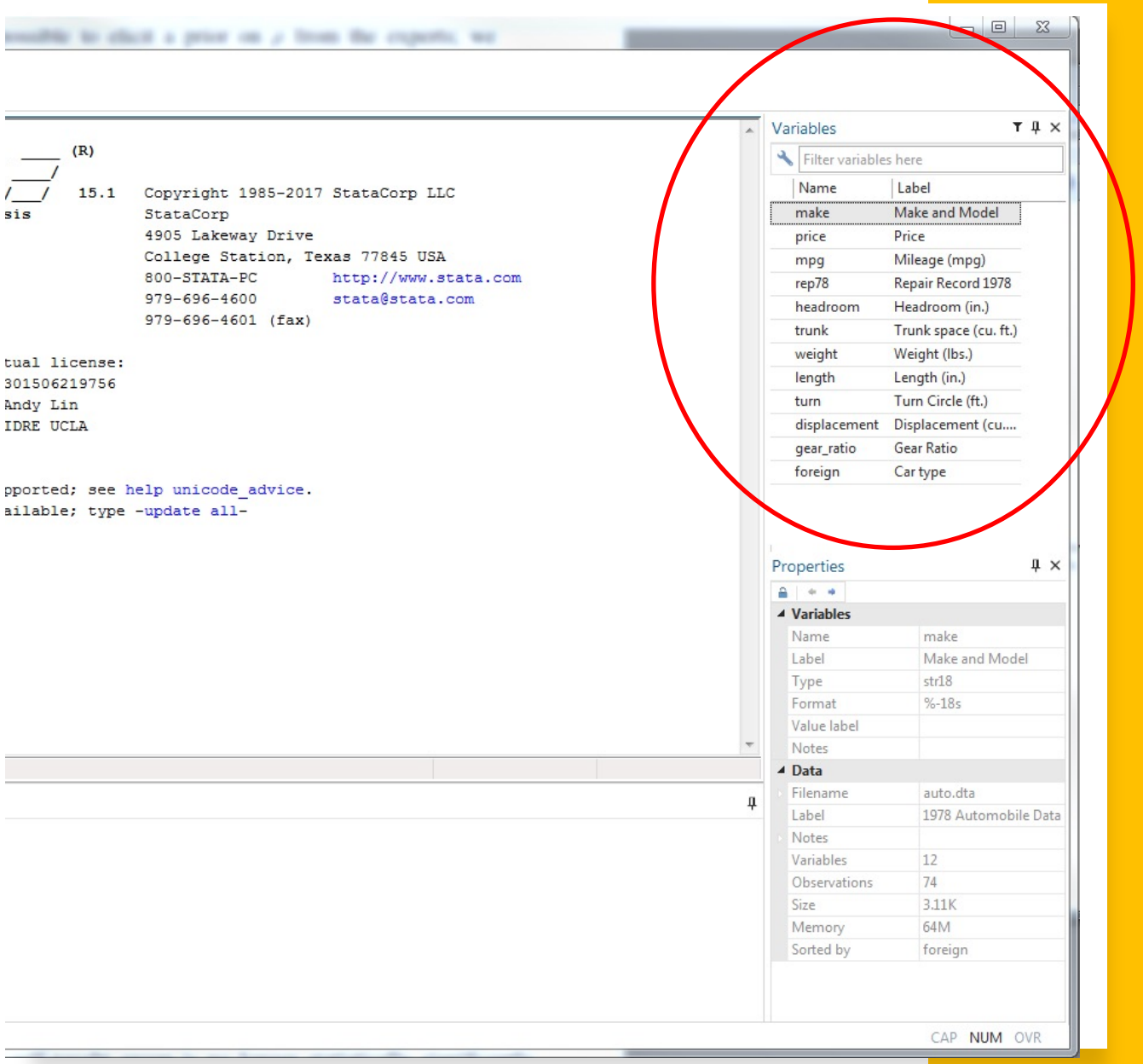
Command window

- You can enter commands directly into the Command window
- This command will load a Stata dataset over the internet



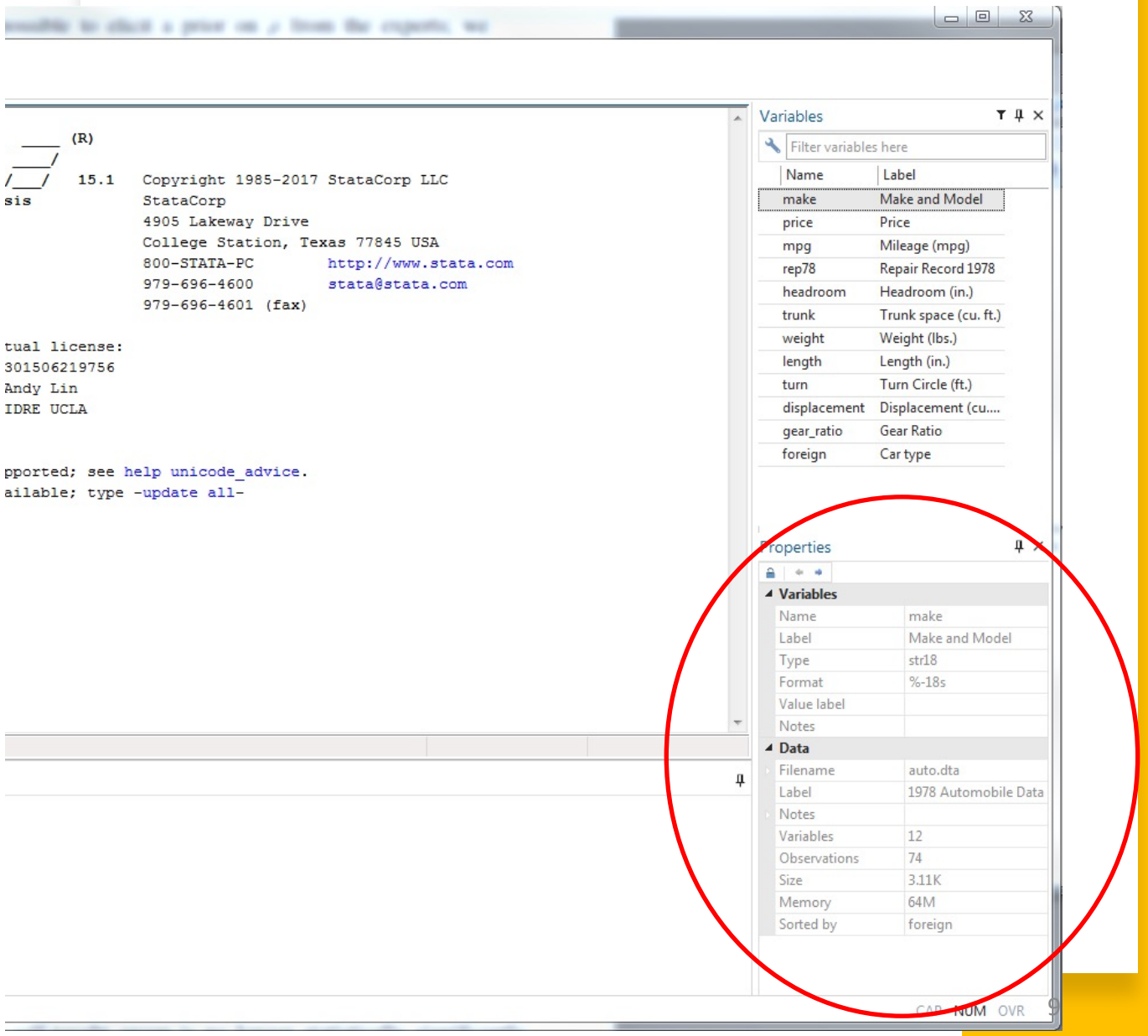
Variables window

- Clicking on a variable name will cause its description to appear in the Properties Window
- Double-clicking on a variable name will cause it to appear in the Command Window

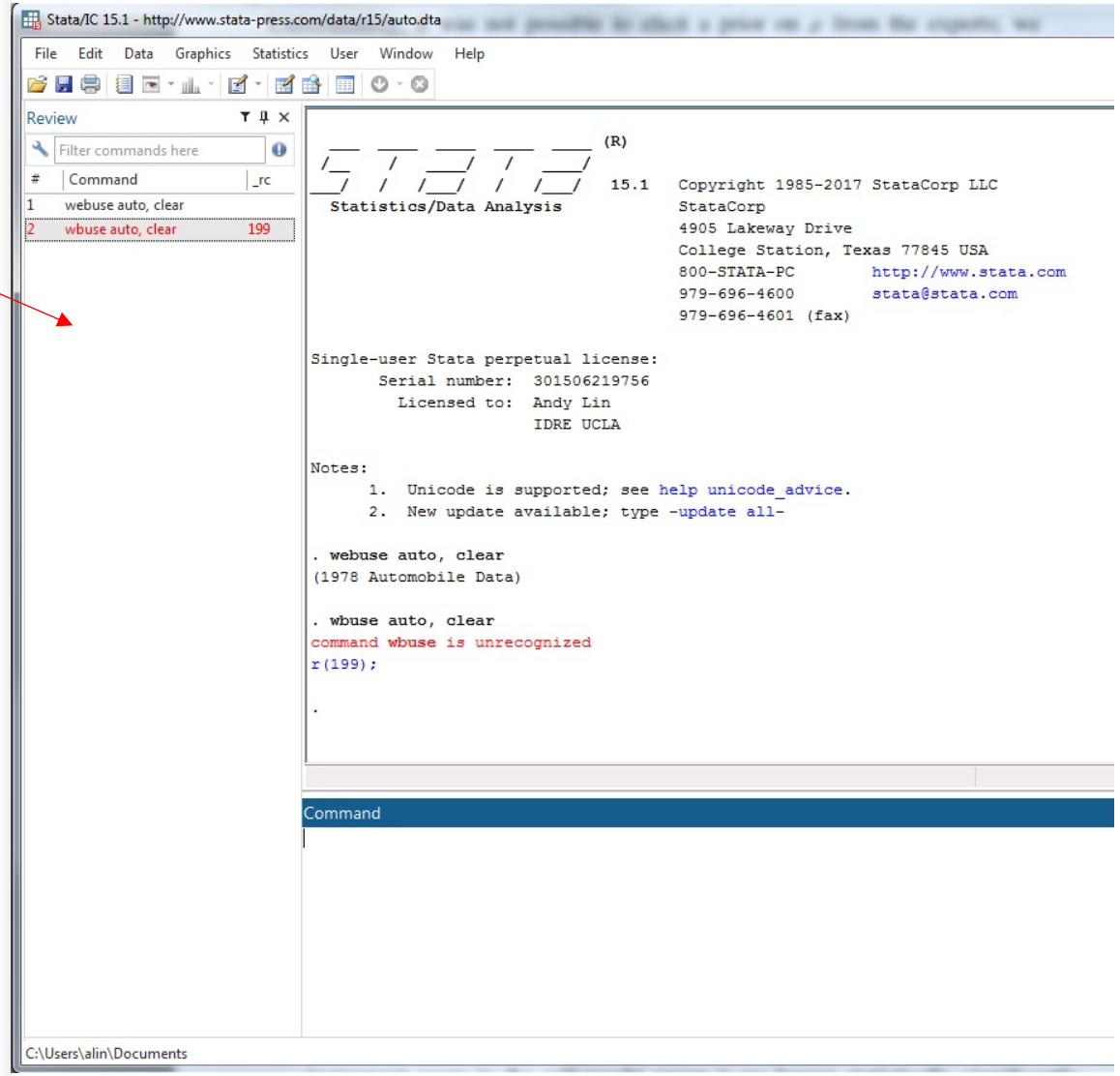


Properties window

- The **Variables** section lists information about selected variable
- The **Data** section lists information about the entire dataset



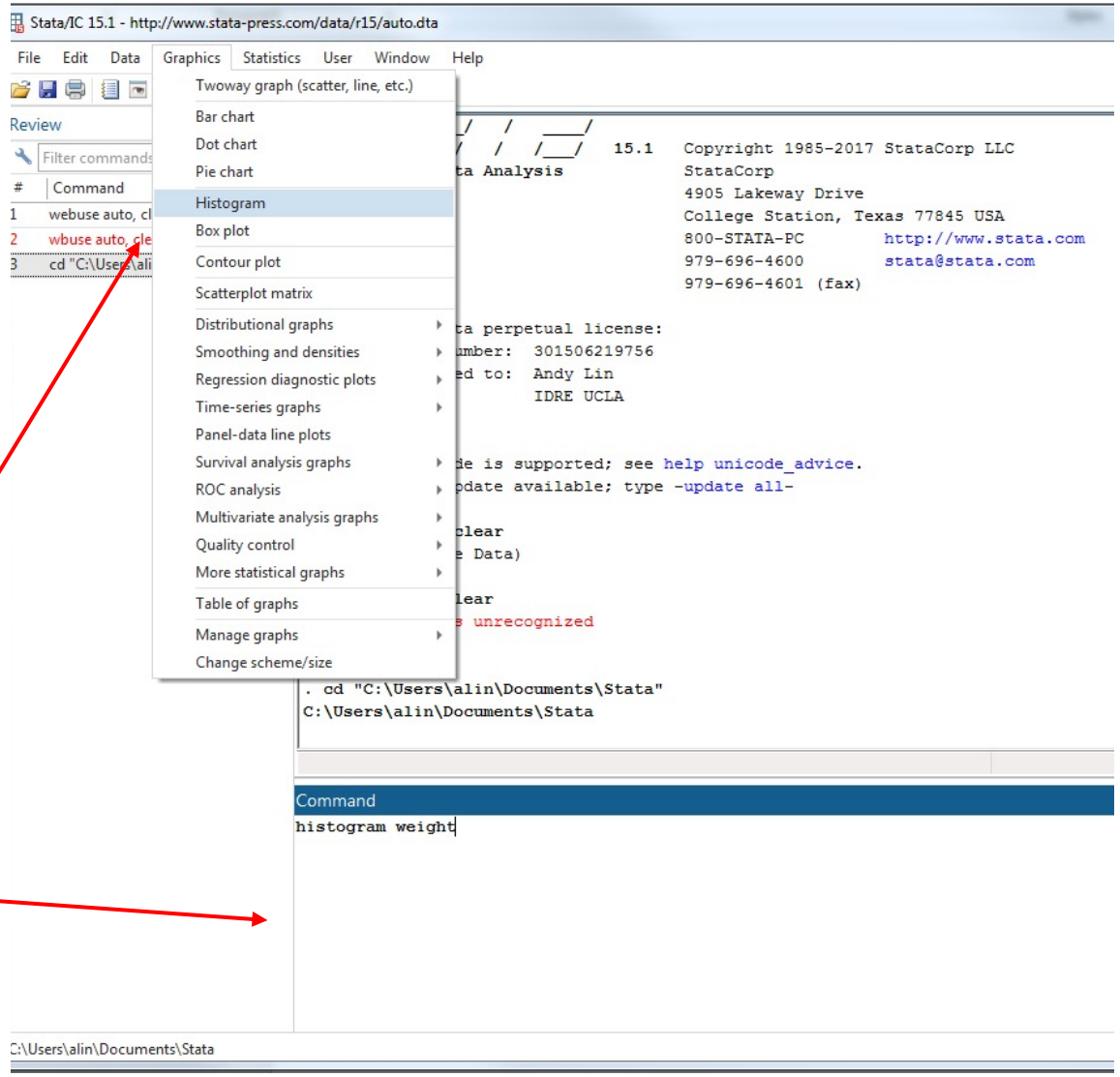
Review Window



- The Review window lists previously issued commands
- Successful commands will appear black
- Unsuccessful commands will appear red
- Double-click a command to run it again

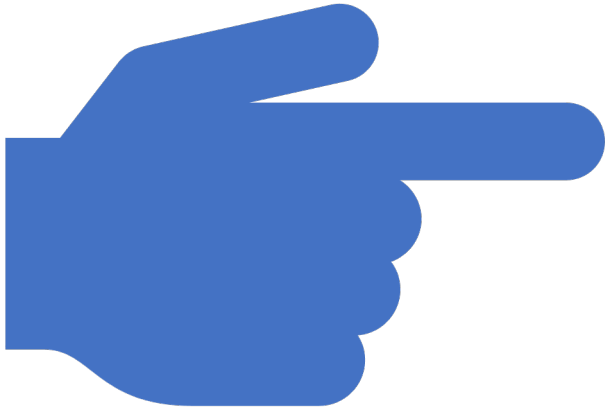
Stata menus

- Almost all Stata users use syntax to run commands rather than point-and-click menus
- Nevertheless, Stata provides menus to run *most* of its data management, graphical, and statistical commands



Help files

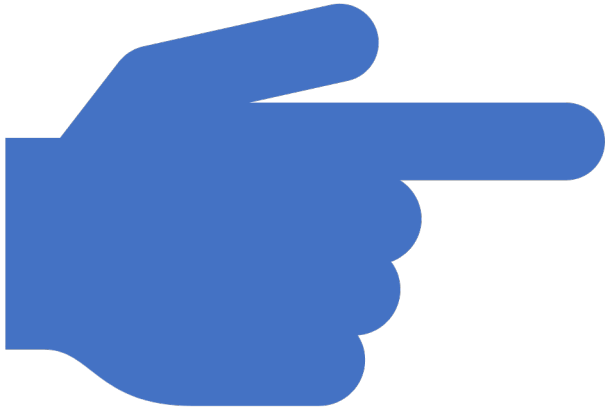
- Precede a command name (and certain topic names) with `help` to access its help file.



Log-files

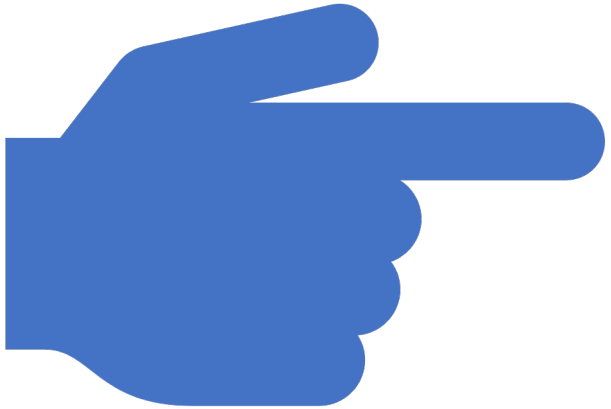
- Stata can create a copy of everything that is sent to the Results window, with the exception of graphs.

- The default in Stata is to save the file with the extension .smcl



Do-files

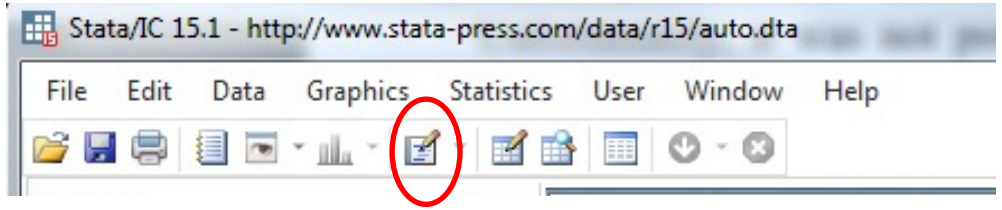
- Stata do-files are text files where users can store and run their commands for reuse
  - Reproducibility
  - Easier debugging and changing commands
- We recommend *always* using a do-file when using Stata
- The file extension .do is used for do-files

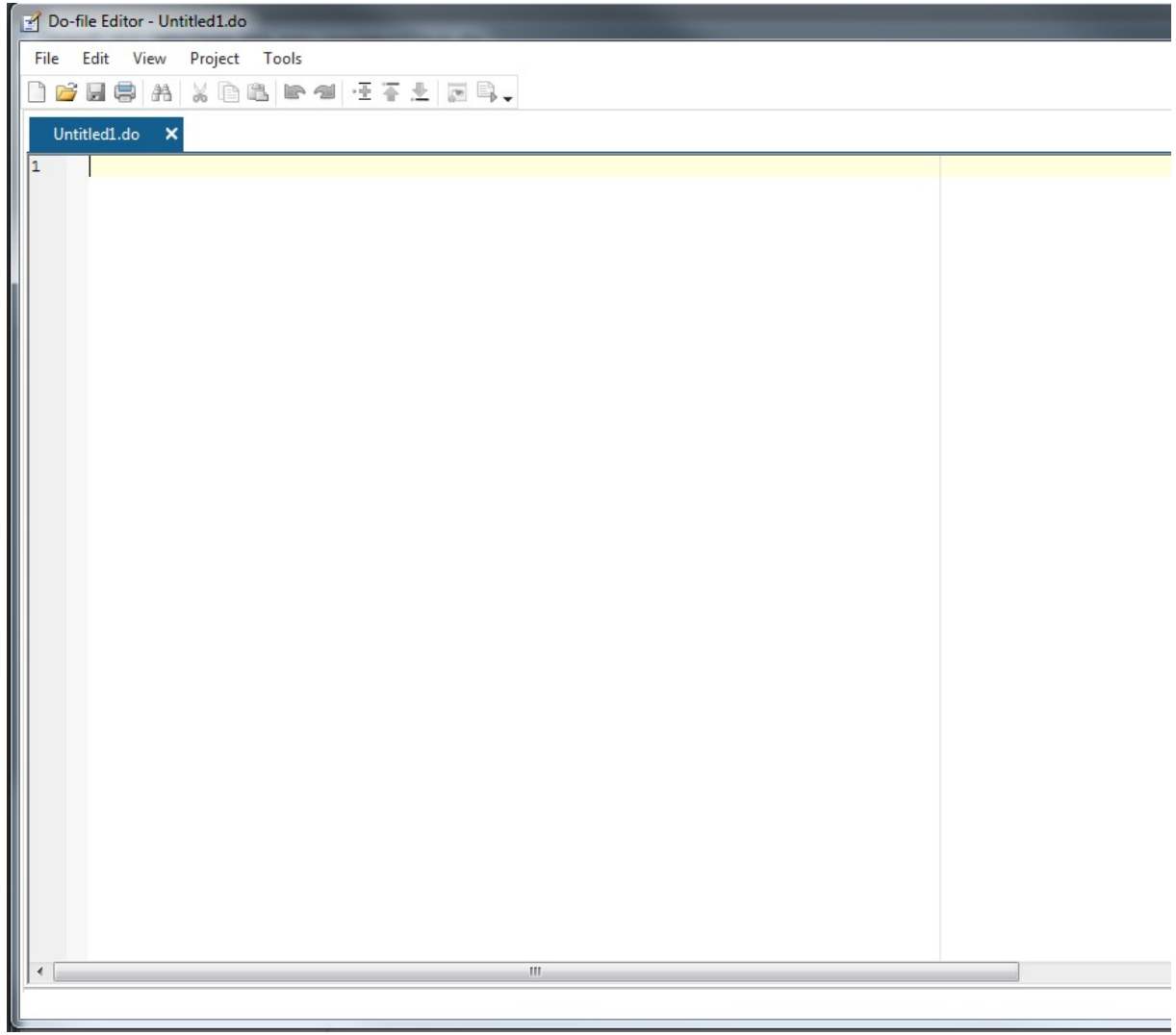


Opening the do-file editor

Use the command `doedit` to open the do-file editor

Or click on the pencil and paper icon on the toolbar





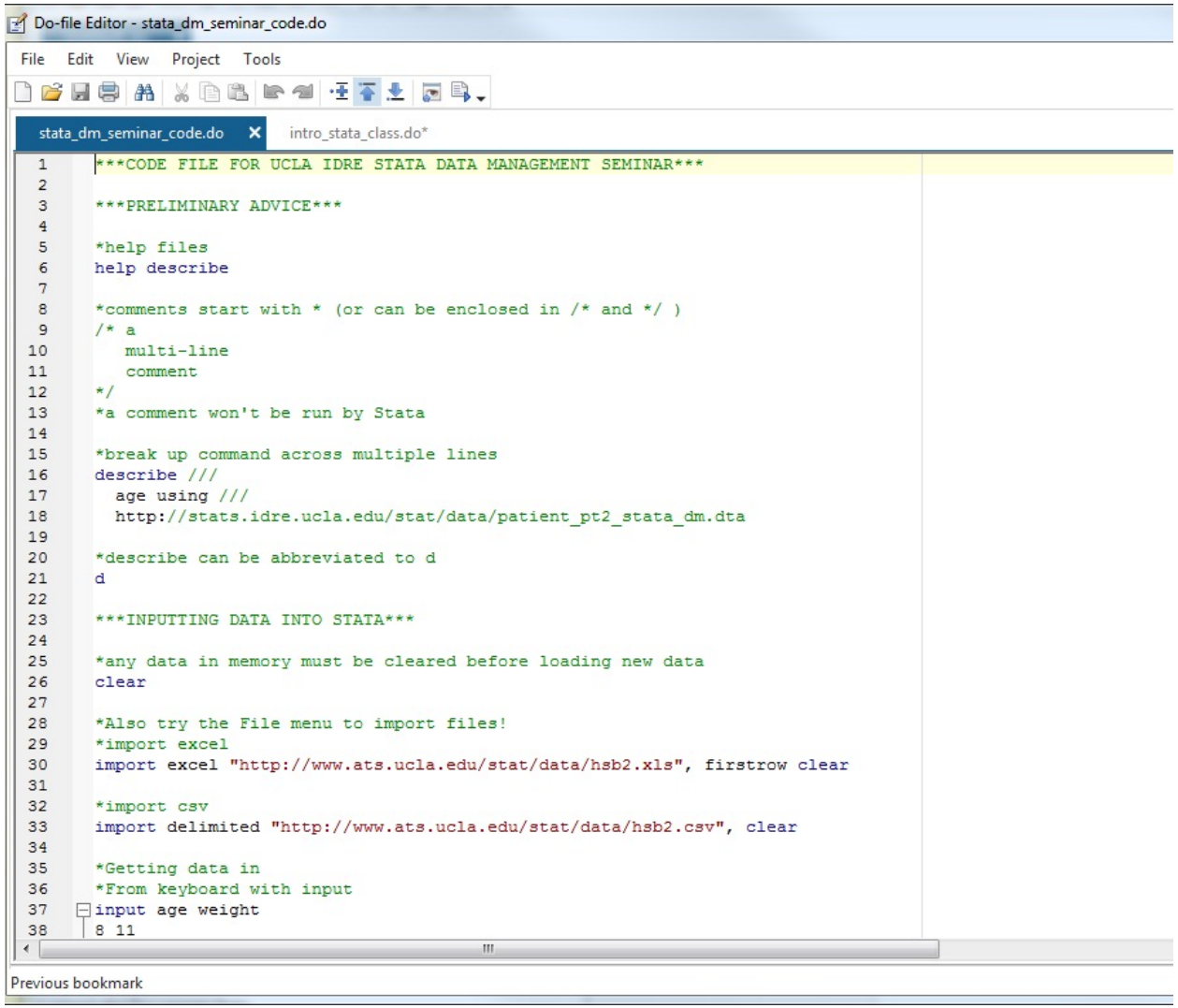
The do-file editor is a text file editor specialized for Stata

Syntax highlighting

The do-file editor colors Stata commands blue

Comments, which are not executed, are usually preceded by * and are colored green

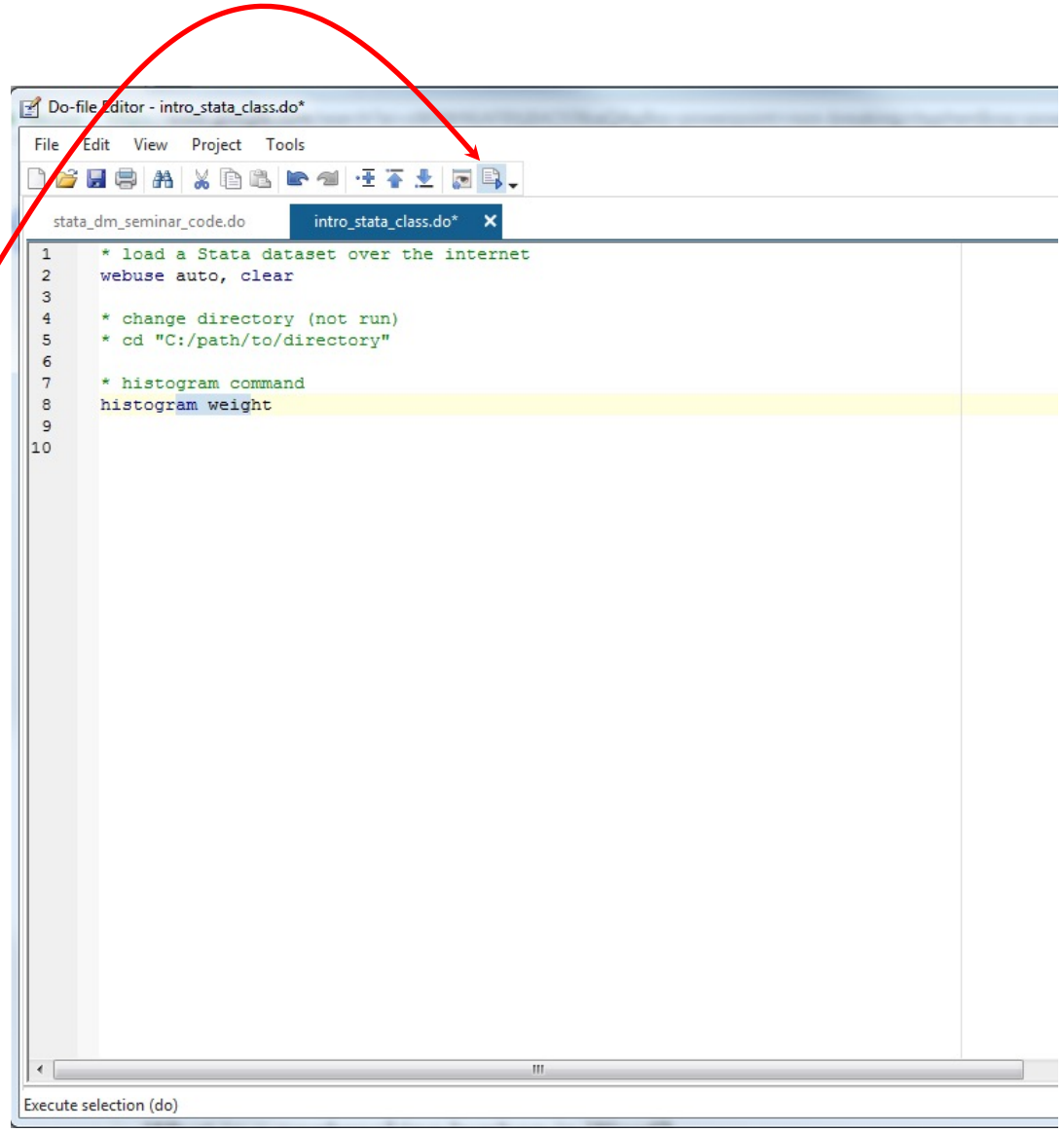
Words in quotes (file names, string values) are colored “red”



Running commands from the do-file

To run a command from the do-file, highlight part or all of the command, and then hit Ctrl-D (*Mac*: Shift+Cmd+D) or the "Execute(do)" icon, the rightmost icon on the do-file editor toolbar

Multiple commands can be selected and executed

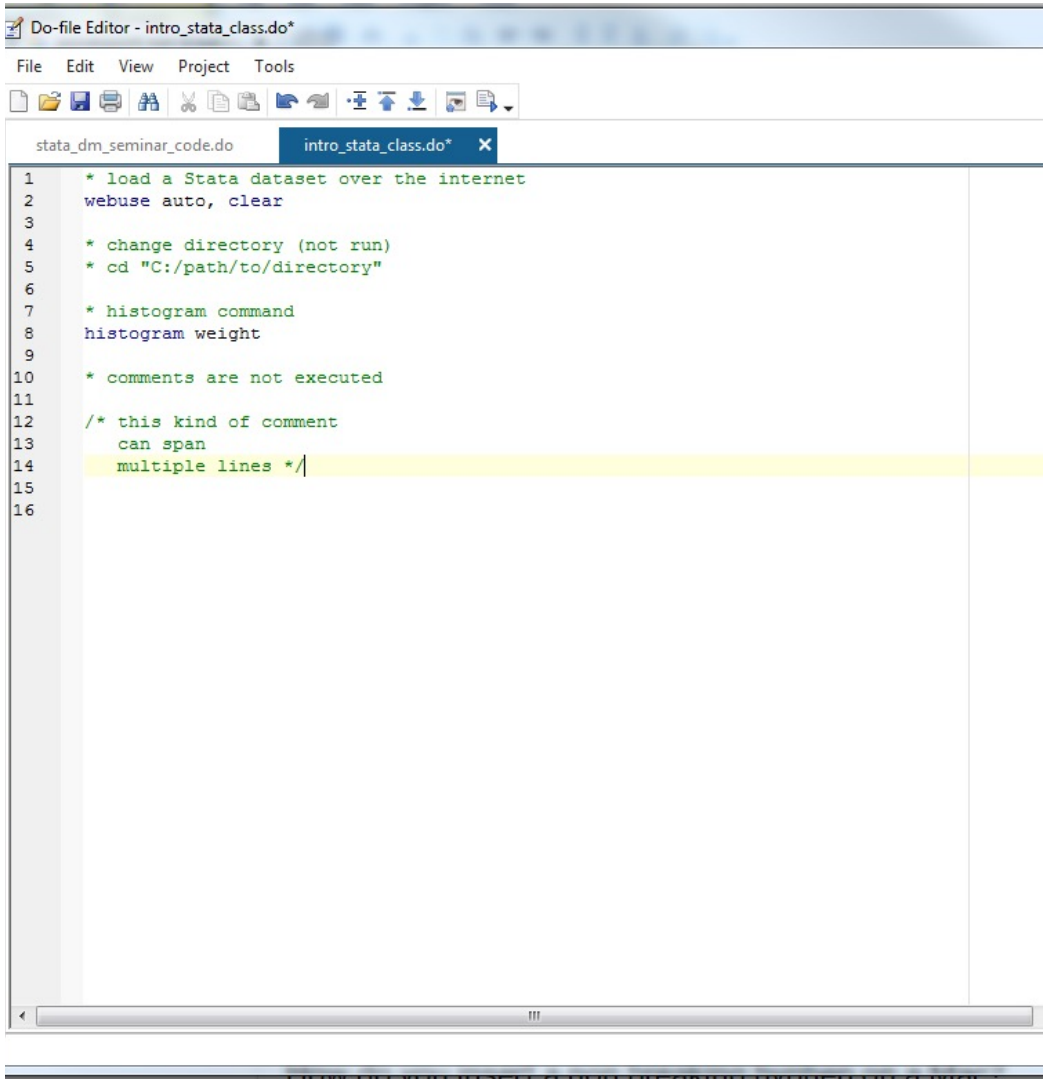


COMMENTS

Comments are not executed, so provide a way to document the do-file

Comments are either preceded by * or surrounded by /* and */

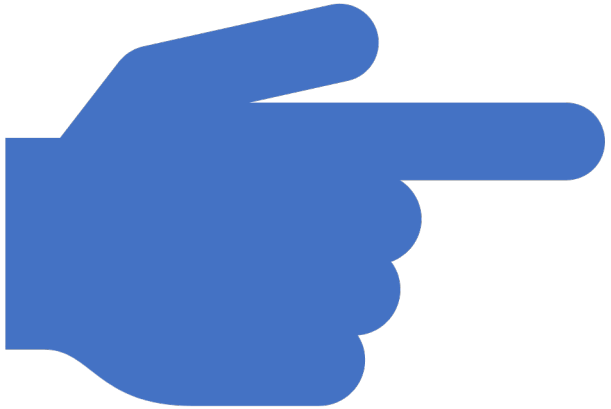
Comments will appear in green in the do-file editor



Log-files

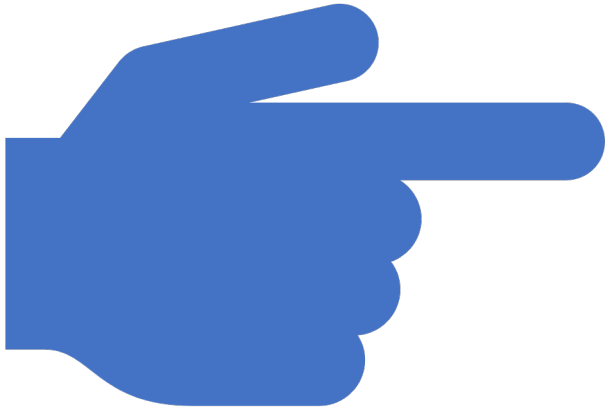
- Stata can create a copy of everything that is sent to the Results window, with the exception of graphs.

- The default in Stata is to save the file with the extension .smcl



Importing data

use	load Stata dataset
save	save Stata dataset
clear	clear dataset from memory
import excel	import Excel dataset
import delimited	import delimited data (csv)



Loading and saving .dta files

- The command `use` loads Stata .dta files
  - Usually these will be stored on a hard drive, but .dta files can also be loaded over the internet (using a web address)
- Use the command `save` to save data in Stata's .dta format
  - The `replace` option will overwrite an existing file with the same name
- The extension .dta can be omitted when using `use` and `save`
- Data import commands like `use` will often have a `clear` option which clears memory before loading the new dataset

```
* read from hard drive; do not execute
use "C:/path/to/myfile.dta“

* load data over internet
use https://stats.idre.ucla.edu/stat/data/hs0

* save data, replace if it exists
save hs0, replace


* load data but clear memory first
use https://stats.idre.ucla.edu/stat/data/hs0, clear
```

Importing excel data sets

- Stata can read in data sets stored in many other formats
- The command `import excel` is used to import Excel data
  - An Excel filename is required (with path, if not located in working directory) after the keyword `using`
- Use the `sheet()` option to open a particular sheet
- Use the `firstrow` option if variable names are on the first row of the Excel sheet

```
* import excel file; change path below before executing
import excel using "C:\path\myfile.xlsx", sheet(“mysheet") firstrow clear
```

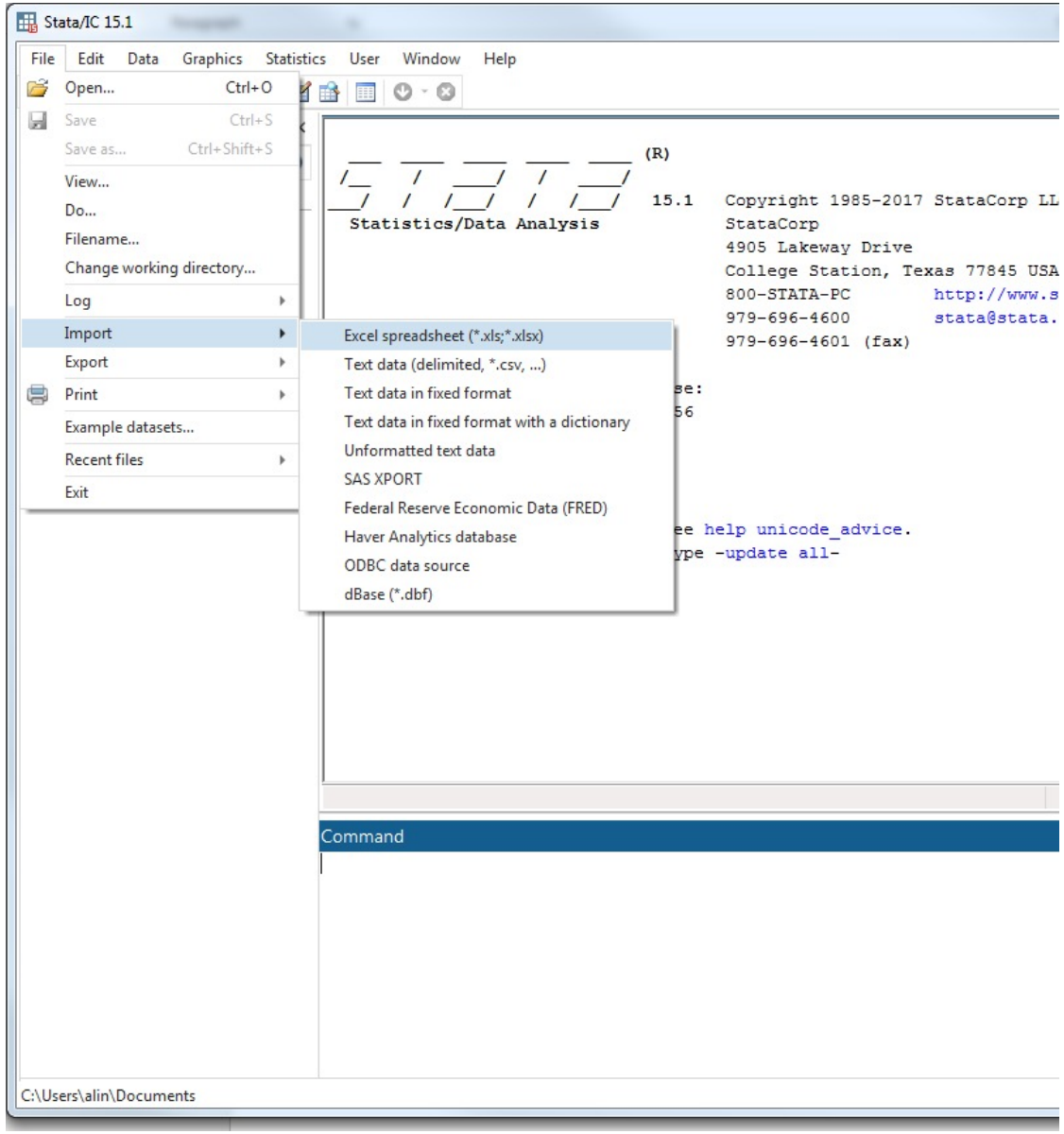
Importing .csv data sets

- Comma-separated values files are also commonly used to store data
- Use `import delimited` to read in .csv files (and files delimited by other characters such as tab or space)
- The syntax and options are very similar to `import excel`
  - But no need for `sheet()` or `firstrow` options (first row is assumed to be variable names in .csv files)

```
* import csv file; change path below before executing
import delimited using "C:\path\myfile.csv", clear
```

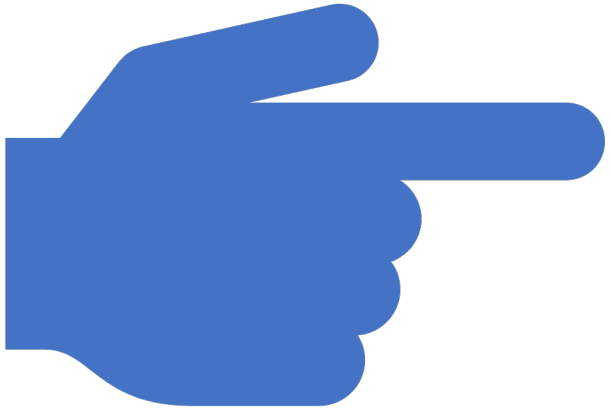
Using the menu to import EXCEL and .csv data

Select File -> Import and then either
“Excel spreadsheet” or
“Text data(delimited,*.csv, …)”



Exploring data

browse	open spreadsheet of data
describe	get variable properties
codebook	inspect variable values
summarize	summarize distribution
tabulate	tabulate frequencies



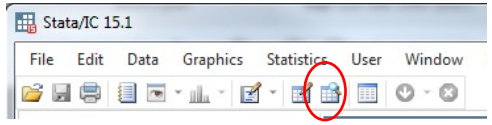
Workshop dataset

- We will use a dataset consisting of 200 observations (rows) and 13 variables (columns)
- Each observation is a student
- Variables
  - Demographics – gender(1=male, 2=female), race, ses(low, middle, high), etc
  - Academic test scores
    - read, write, math, science, socst
- Go ahead and load the dataset!

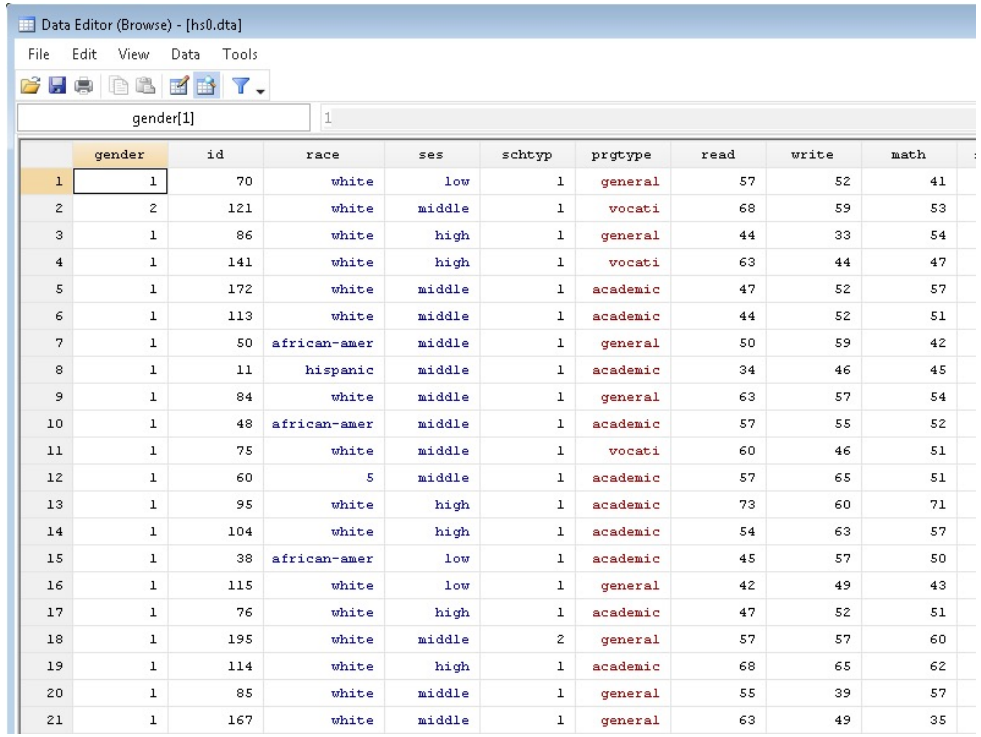
```
* Workshop dataset
use https://stats.idre.ucla.edu/stat/data/hs0, clear
```

Browsing the dataset

- Once the data are loaded, we can view the dataset as a spreadsheet using the command `browse`
- The magnifying glass with spreadsheet icon also browses the dataset



- Black columns are numeric, red columns are strings, and blue columns are numeric with string labels

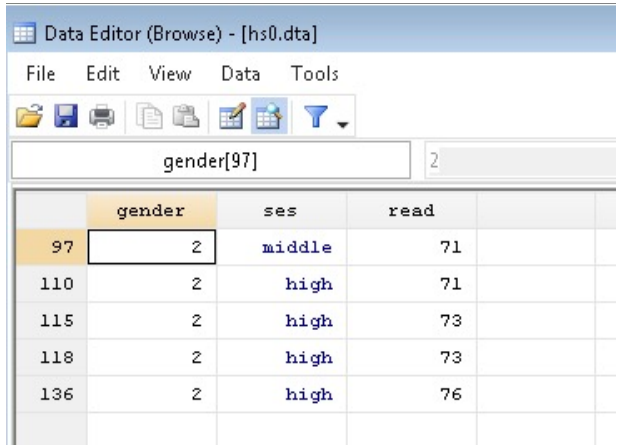


	gender	id	race	ses	schtyp	prgtype	read	write	math
1	1	70	white	low	1	general	57	52	41
2	2	121	white	middle	1	vocati	68	59	53
3	1	86	white	high	1	general	44	33	54
4	1	141	white	high	1	vocati	63	44	47
5	1	172	white	middle	1	academic	47	52	57
6	1	113	white	middle	1	academic	44	52	51
7	1	50	african-amer	middle	1	general	50	59	42
8	1	11	hispanic	middle	1	academic	34	46	45
9	1	84	white	middle	1	general	63	57	54
10	1	48	african-amer	middle	1	academic	57	55	52
11	1	75	white	middle	1	vocati	60	46	51
12	1	60	5	middle	1	academic	57	65	51
13	1	95	white	high	1	academic	73	60	71
14	1	104	white	high	1	academic	54	63	57
15	1	38	african-amer	low	1	academic	45	57	50
16	1	115	white	low	1	general	42	49	43
17	1	76	white	high	1	academic	47	52	51
18	1	195	white	middle	2	general	57	57	60
19	1	114	white	high	1	academic	68	65	62
20	1	85	white	middle	1	general	55	39	57
21	1	167	white	middle	1	general	63	49	35

Stata logical and relational operators

- == equal to
  - double equals used to check for equality
- <, >, <=, >= greater than, greater than or equal to, less than, less than or equal to
- ! not
  - != not equal
- & and
- | or

```
* browse gender, ses, and read
*  for females (gender=2) who have read > 70
browse gender ses read if gender == 2 & read > 70
```



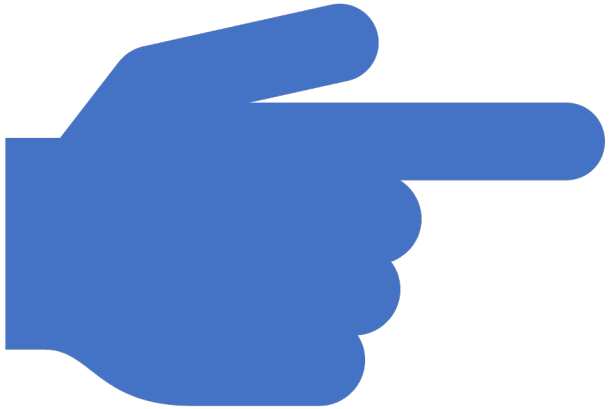
	gender	ses	read
97	2	middle	71
110	2	high	71
115	2	high	73
118	2	high	73
136	2	high	76

Quick exploring

- `describe` provides the following variable properties:
  - storage type (e.g. byte (integer), float (decimal), str8 (character string variable of length 8))
  - name of value label
  - variable label
- `codebook` detailed information about the values of each variable
- `summarize` command calculates a variable's:
  - number of non-missing observations
  - mean
  - standard deviation
  - min and max
- `tabulate` displays counts of each value of a variable
  - useful for variables with a limited number of levels

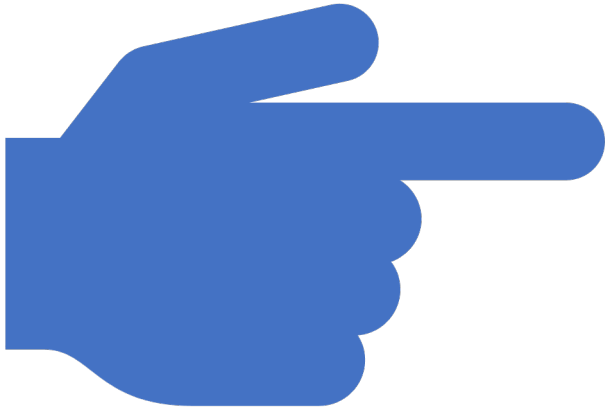
Data visualization

histogram	histogram
scatter	scatter plot



Data management

generate	create variable
replace	replace values of variable
rename	rename variable
label variable	give variable description
egen	extended variable generation
encode	convert string variable to numeric



creating dummy indicators

- It is often necessary to create variables that are 0/1 indicators for belonging to a category of another variable, where 0=FALSE and 1=TRUE
  - often called dummy variables or indicators
  - Remember that Stata often prefers to work with numeric variables

```
* create a variable that equals 1 if prgtype
*  equals academic, 0 otherwise
gen academic = 0
replace academic = 1 if prgtype == "academic"
tab prgtype academic
```

extended generation of variables

- `egen` (extended generate) creates variables using a wide array of functions, which include:
  - statistical functions that accept multiple variables as arguments
    - e.g. means across several variables
  - functions that accept a single variable, but do not involve simple arithmetic operations
    - e.g. standardizing a variable (subtract mean and divide by standard deviation)
- See the help file for `egen` to see a full list of available functions

```
* generate variables with functions rowmean returns mean of all non-missing values

egen meantest = rowmean(read math science socst)

summarize meantest read math science socst
```

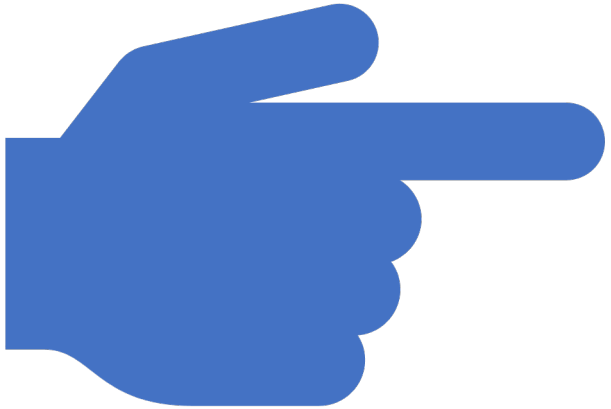
encoding string variables into numeric

- `encode` converts a string variable into a numeric variable
  - remember that some Stata commands require numeric variables
  - `encode` will use alphabetical order to order the numeric codes
  - `encode` will convert the original string values into a set of value labels
  - `encode` will create a new numeric variable, which must be specified in option `gen(`*`varname`*`)`

```
* encoding string prgtype into numeric variable prog
encode prgtype, gen(prog)
```

Basic statistical analysis

ttest	t-tests
correlate	correlation matrices
regress	linear regression
logit	logistic regression



T-tests

- independent samples t-test : whether the mean of a variable is different between 2 groups
- paired samples t-test: assesses whether the means of 2 variables are the same

* independent samples t-test

```
ttest var, by(groupvar)
```

where *var* is the variable whose mean will be tested for differences between levels of *groupvar*

* paired samples t-test

```
ttest  var1 ==  var2
```

correlation

- A correlation coefficient quantifies the linear relationship between two (continuous) variables on a scale between -1 and 1
- Syntax: `correlate` *`varlist`*
- The output will be a correlation matrix that shows the pairwise correlation between each pair of variables

```
* correlation matrix of 5 variables
corr read write math science socst

(obs=195)

             |     read    write     math  science    socst
-------------+---------------------------------------------
        read |   1.0000
       write |   0.5960   1.0000
        math |   0.6492   0.6203   1.0000
     science |   0.6171   0.5671   0.6166   1.0000
       socst |   0.6175   0.5996   0.5299   0.4529   1.0000
```

linear regression

- Linear regression, or ordinary least squares regression, models the effects of one or more predictors, which can be continuous or categorical, on a normally-distributed outcome
- Syntax: `regress` *`depvar`* `varlist`, where *`depvar`* is the name of the dependent variable, and *`varlist`* is a list of predictors.
  - For categorical predictors with the `i.` prefix, Stata will automatically create dummy 0/1 indicator variables and enter all but one (the first, by default) into the regression

LINEAR REGRESSION EXAMPLE

```
* linear regression of write on continuous predictor math and categorical predictor
prog
regress write math i.prog
```

```
      Source |       SS           df       MS      Number of obs   =       200
-------------+----------------------------------   F(3, 196)       =     44.20
       Model |  7214.30058         3  2404.76686   Prob > F        =    0.0000
    Residual |  10664.5744       196   54.411094   R-squared       =    0.4035
-------------+----------------------------------   Adj R-squared   =    0.3944
       Total |   17878.875       199   89.843593   Root MSE        =    7.3764

------------------------------------------------------------------------------
       write |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        math |   .5476883   .0635714     8.62   0.000     .4223166    .6730601
             |
        prog |
    general  |  -1.248212   1.381794    -0.90   0.367    -3.973304     1.47688
     vocati  |   -3.84865   1.426982    -2.70   0.008     -6.66286   -1.034441
             |
       _cons |   25.18496   3.677755     6.85   0.000      17.9319    32.43801
------------------------------------------------------------------------------
```

logistic regression

- Logistic regression is used to estimate the effect of multiple predictors on a binary outcome
- Syntax very similar to regress: `logit` *`depvar varlist`*`,` where *`depvar`* is a binary outcome variable and *`varlist`* is a list of predictors
- Add the `or` option to output the coefficients as odds ratios

Logistic regression example

```
* logistic regression of being in academic program on female and math score
* coefficients as odds ratios
logit academic i.female c.math, or

Logistic regression                               Number of obs     =        200
                                                  LR chi2(2)        =      46.85
                                                  Prob > chi2       =     0.0000
Log likelihood = -114.95535                       Pseudo R2         =     0.1693

------------------------------------------------------------------------------
    academic | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    1.female |   1.144479   .3680227     0.42   0.675     .6093863    2.149429
        math |   1.128431   .0229718     5.94   0.000     1.084293    1.174365
       _cons |   .0018648   .0020288    -5.78   0.000     .0002211    .0157282
------------------------------------------------------------------------------
```

Where to find additional helps?

- Data Manipulation topics: help contents data management
- List of all Estimation Commands: help estimation commands
- Time series: help time
- Survival analysis: help st
- Panel data: help xt
- Survey analysis: help survey
- UCLA STATA resources: https://stats.idre.ucla.edu/stata/