Introduction to STATA

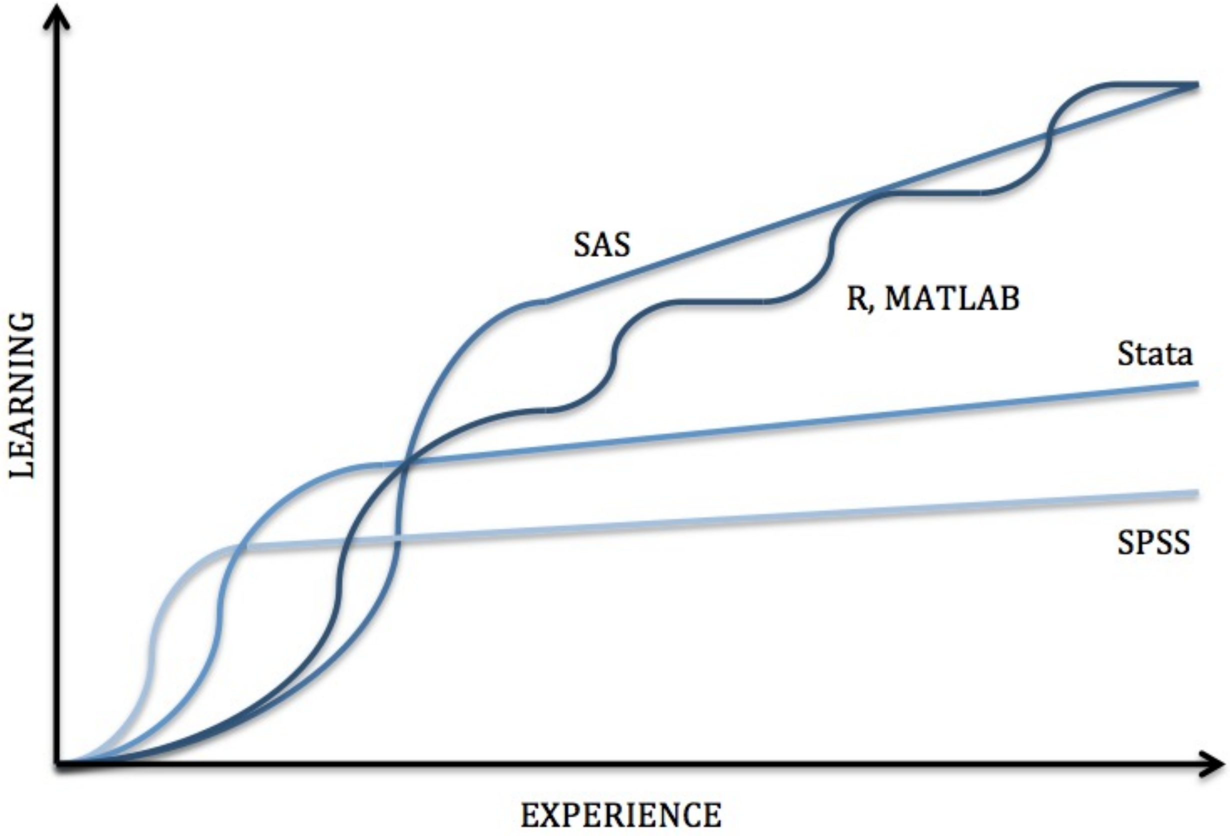
Adrian Rohit Dass
Institute of Health Policy, Management, and Evaluation
Canadian Centre for Health Economics
University of Toronto

September 22, 2023

Outline

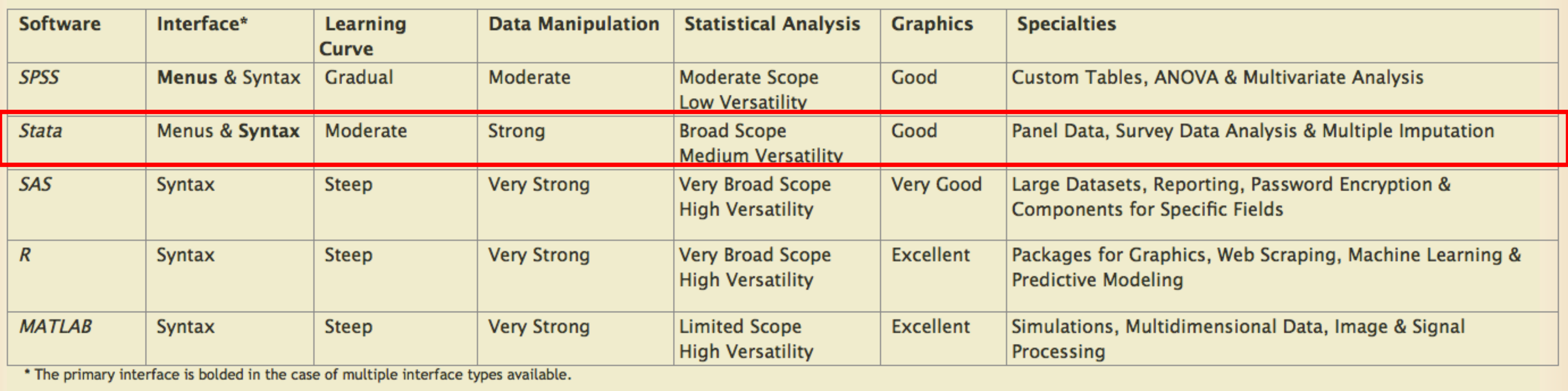
- Why use STATA?
- Reading/Cleaning data
- Regression Analysis
- Post-estimation Diagnostic Checks
- Other Topics in STATA
- Applied Example
- STATA Resources

Learning Curves of Various Software Packages



Source: https://sites.google.com/a/nyu.edu/statistical-software-guide/summary

Summary of Various Statistical Software Packages



Software	Interface*	Learning Curve	Data Manipulation	Statistical Analysis	Graphics	Specialties
SPSS	**Menus** & Syntax	Gradual	Moderate	Moderate Scope Low Versatility	Good	Custom Tables, ANOVA & Multivariate Analysis
Stata	Menus & **Syntax**	Moderate	Strong	Broad Scope Medium Versatility	Good	Panel Data, Survey Data Analysis & Multiple Imputation
SAS	Syntax	Steep	Very Strong	Very Broad Scope High Versatility	Very Good	Large Datasets, Reporting, Password Encryption & Components for Specific Fields
R	Syntax	Steep	Very Strong	Very Broad Scope High Versatility	Excellent	Packages for Graphics, Web Scraping, Machine Learning & Predictive Modeling
MATLAB	Syntax	Steep	Very Strong	Limited Scope High Versatility	Excellent	Simulations, Multidimensional Data, Image & Signal Processing

* The primary interface is bolded in the case of multiple interface types available.

Source: https://sites.google.com/a/nyu.edu/statistical-software-guide/summary

Why STATA?

- Moderate learning curve
- Widely used in economics and other social sciences
- Feature rich for analyzing various types of data (survey data, panel data, etc.)
- Wide array of free, user-written routines to expand the scope of STATA's capabilities
- Support for export of regression results to tables through packages such as "estout" (STATA 16 or older) or Tables feature (STATA 17)

STATA Purchasing options (from UofT)

For more information: https://onesearch.library.utoronto.ca/ic/stata-gradplan-u-t

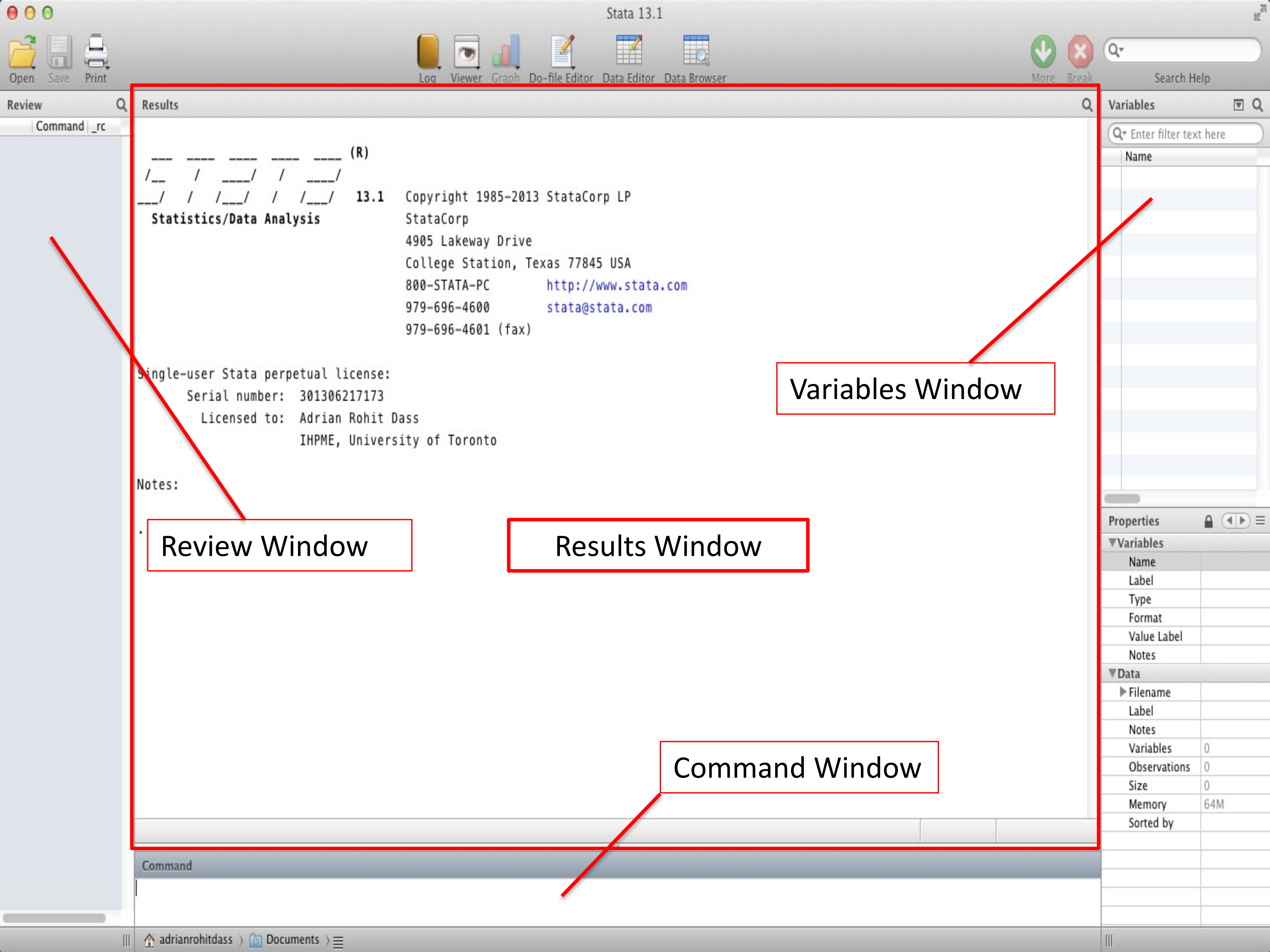
Reading/Cleaning data

STATA Basics

- Contains a menu and syntax based interface
- Prior programming experience is not required, but can be helpful (especially with the syntax based *.do* files)
- Case sensitive, so be careful:

  I.e.

  - regress y x results will result in a successful OLS estimation (if everything else is right)
  - Regress y x results <u>will</u> in an error message



```
  ___  ____  ____  ____  ____ (R)
 /__    /   ____/   /   ____/
___/   /   /___/   /   /___/   13.1   Copyright 1985-2013 StataCorp LP
  Statistics/Data Analysis            StataCorp
                                      4905 Lakeway Drive
                                      College Station, Texas 77845 USA
                                      800-STATA-PC        http://www.stata.com
                                      979-696-4600        stata@stata.com
                                      979-696-4601 (fax)

Single-user Stata perpetual license:
       Serial number:  301306217173
         Licensed to:  Adrian Rohit Dass
                       IHPME, University of Toronto

Notes:

.
```

Variables
Name
Label
Type
Format
Value Label
Notes

Data
Filename
Label
Notes
Variables 0
Observations 0
Size 0
Memory 64M
Sorted by

Variables Window

Review Window

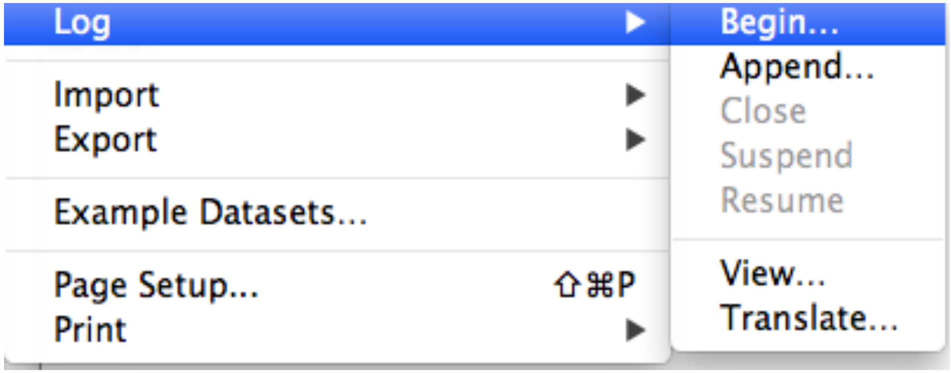
Results Window

Command Window

Starting a Log File

This should generally be your *first* step when using Stata

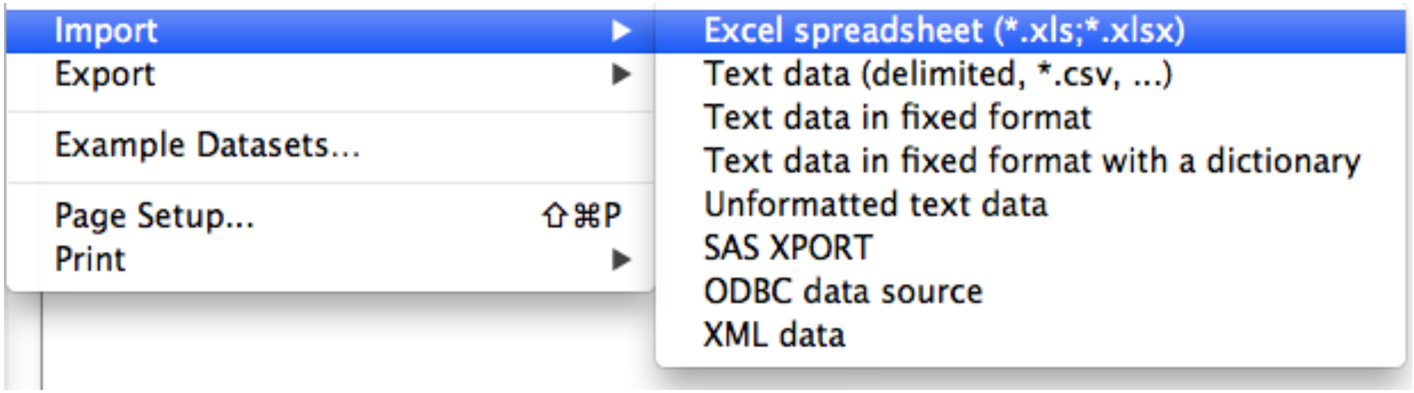
- Menu:
  - File → Log → Begin:



  - Stata will prompt you to name the file. Pick a creative name (E.g: logfile1), then click ok
  - At this point, Stata will record everything you do (importing data, running commands, regression output, etc)
- Syntax:
  - log using filename [, append replace [text|smcl] name(logname)]

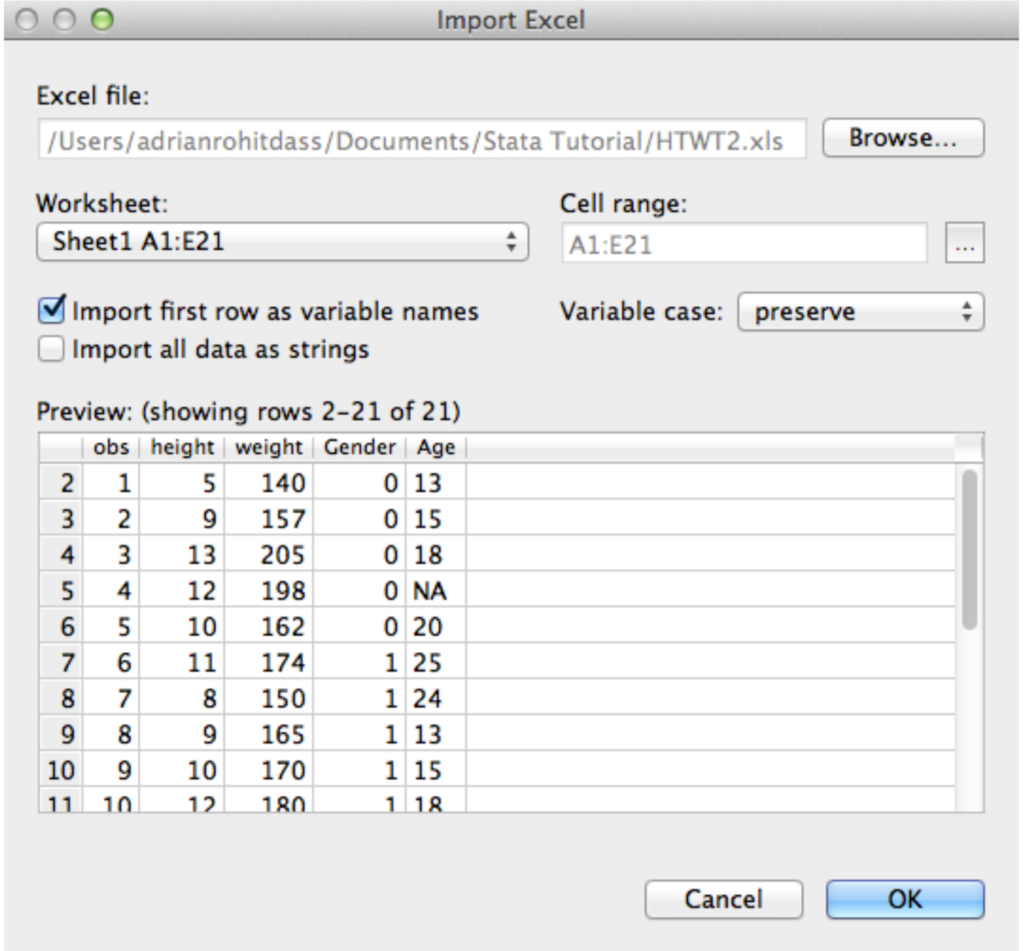
Importing Data into Stata

- Menu
  - File → Import → Choose appropriate option:


Import
Export
Example Datasets...
Page Setup... ⇧⌘P
Print
Excel spreadsheet (*.xls;*.xlsx)
Text data (delimited, *.csv, ...)
Text data in fixed format
Text data in fixed format with a dictionary
Unformatted text data
SAS XPORT
ODBC data source
XML data

  - .csv (Comma Separated) is a common option, but .xls (Microsoft Excel Format) and other formats are compatible too
- Syntax
  - import excel [using] *filename [, import excel options]*
  - For .csv files, command changes to import *delimited*

Importing Data into STATA (Microsoft Excel (.xls))



Excel file: /Users/adrianrohitdass/Documents/Stata Tutorial/HTWT2.xls

Worksheet: Sheet1 A1:E21

Cell range: A1:E21

☑ Import first row as variable names

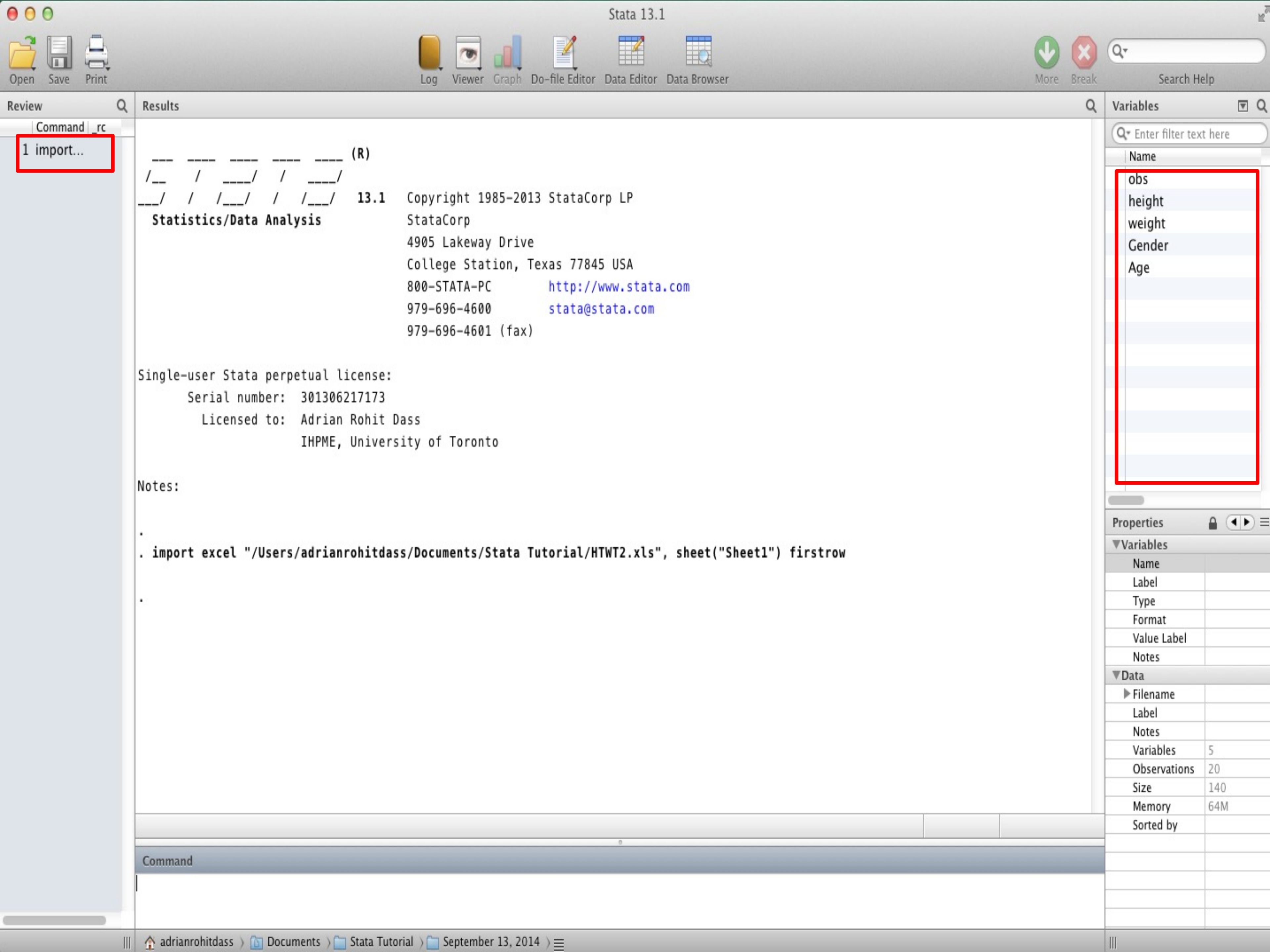
□ Import all data as strings

Variable case: preserve

Preview: (showing rows 2-21 of 21)

	obs	height	weight	Gender	Age
2	1	5	140	0	13
3	2	9	157	0	15
4	3	13	205	0	18
5	4	12	198	0	NA
6	5	10	162	0	20
7	6	11	174	1	25
8	7	8	150	1	24
9	8	9	165	1	13
10	9	10	170	1	15
11	10	12	180	1	18

Once happy with settings, click ok



1 import...

```
 ___  ____  ____  ____  ____ (R)
/__    /   ____/   /   ____/
___/   /   /___/   /   /___/   13.1   Copyright 1985-2013 StataCorp LP
  Statistics/Data Analysis            StataCorp
                                      4905 Lakeway Drive
                                      College Station, Texas 77845 USA
                                      800-STATA-PC        http://www.stata.com
                                      979-696-4600        stata@stata.com
                                      979-696-4601 (fax)

Single-user Stata perpetual license:
       Serial number:  301306217173
         Licensed to:  Adrian Rohit Dass
                       IHPME, University of Toronto

Notes:

.
. import excel "/Users/adrianrohitdass/Documents/Stata Tutorial/HTWT2.xls", sheet("Sheet1") firstrow

.
```

Variables:

- obs
- height
- weight
- Gender
- Age

Properties

▼Variables

Name	
Label	
Type	
Format	
Value Label	
Notes	

▼Data

▶Filename	
Label	
Notes	
Variables	5
Observations	20
Size	140
Memory	64M
Sorted by	

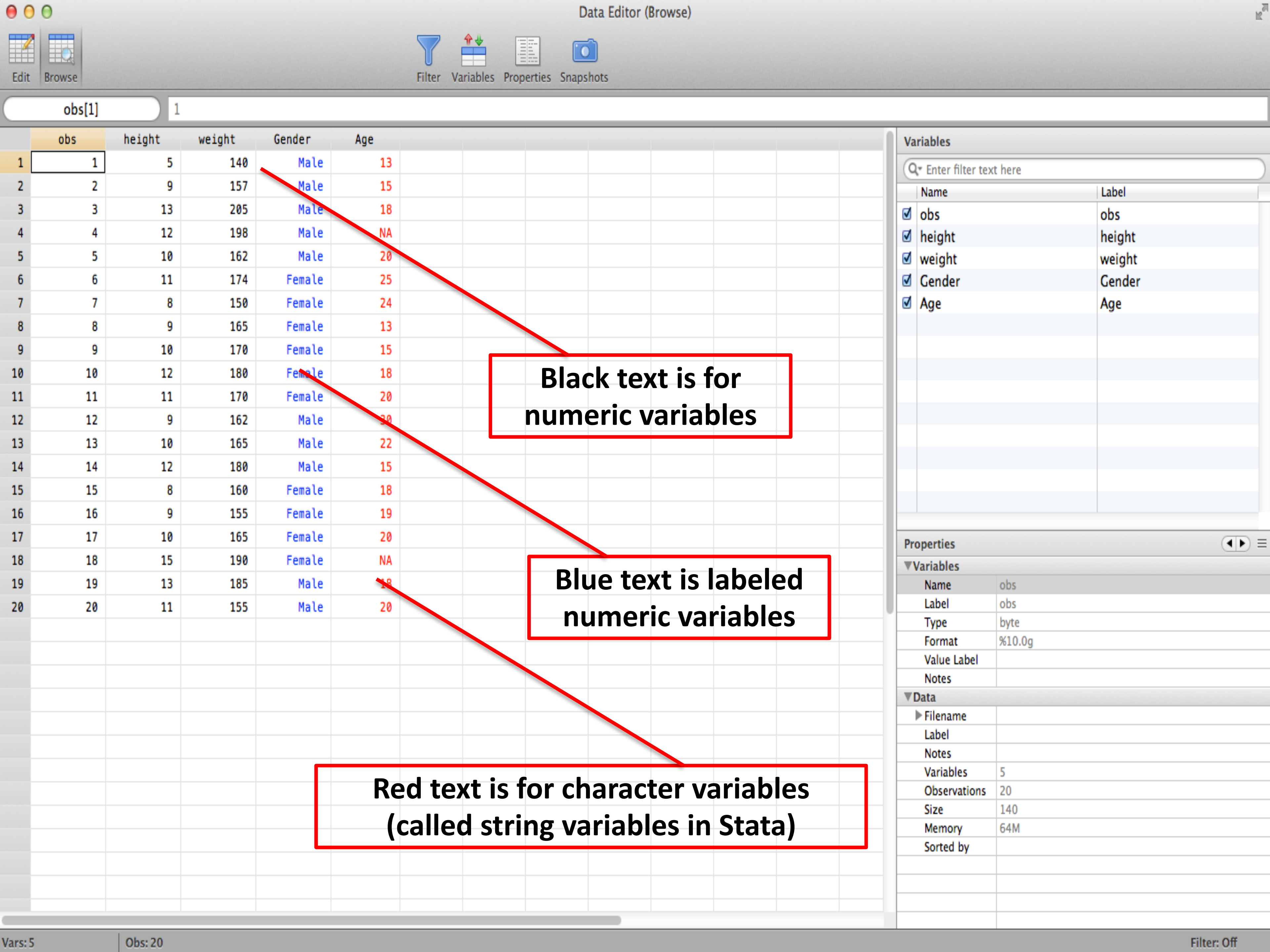
adrianrohitdass 〉 Documents 〉 Stata Tutorial 〉 September 13, 2014 〉

Starting off

Type describe to obtain some useful information about your dataset:

```
Contains data
  obs:            20
 vars:             5
 size:           140
--------------------------------------------------------------------------------------------------------------------------------------
              storage   display     value
variable name   type    format      label      variable label
--------------------------------------------------------------------------------------------------------------------------------------
obs             byte    %10.0g                 obs
height          byte    %10.0g                 height
weight          int     %10.0g                 weight
Gender          byte    %10.0g                 Gender
Age             str2    %9s                    Age
--------------------------------------------------------------------------------------------------------------------------------------
Sorted by:
     Note:  dataset has changed since last saved
```

To look at your data, type browse



obs	height	weight	Gender	Age
1	5	140	Male	13
2	9	157	Male	15
3	13	205	Male	18
4	12	198	Male	NA
5	10	162	Male	20
6	11	174	Female	25
7	8	150	Female	24
8	9	165	Female	13
9	10	170	Female	15
10	12	180	Female	18
11	11	170	Female	20
12	9	162	Male	20
13	10	165	Male	22
14	12	180	Male	15
15	8	160	Female	18
16	9	155	Female	19
17	10	165	Female	20
18	15	190	Female	NA
19	13	185	Male	18
20	11	155	Male	20

Variables

Enter filter text here

	Name	Label
☑	obs	obs
☑	height	height
☑	weight	weight
☑	Gender	Gender
☑	Age	Age

Properties

▼Variables	
Name	obs
Label	obs
Type	byte
Format	%10.0g
Value Label	
Notes	
▼Data	
▶Filename	
Label	
Notes	
Variables	5
Observations	20
Size	140
Memory	64M
Sorted by	

Vars: 5 Obs: 20 Filter: Off

Black text is for numeric variables

Blue text is labeled numeric variables

Red text is for character variables (called string variables in Stata)

Convert Character variable to Numeric

Make use of Stata’s destring command:

```
destring [varlist] , {generate(newvarlist)|replace}
[destring_options]
```

Eg:

```
destring age, replace ignore(NA)
```

Sorting the Observations and Variables

- Sorting changes the order in which the observations appear. We can sort numbers, letters, etc.

- Example (ascending): sort x
  - Note: Use gsort for descending or create a negative version of x and sort

- Ordering changes the order variables in dataset appear.

- Example: order x y z

Changing Existing variables: **rename**

- Command: **rename**

- changes the name of an existing variable

- Example, rename variable ‘ZGMFX10A’ as ‘height’ **rename** ZGMFX10A height

Working with Labels

label give descriptions to variables or data sets

- To label the dataset in memory:
- **label data** “National Population Health Survey”

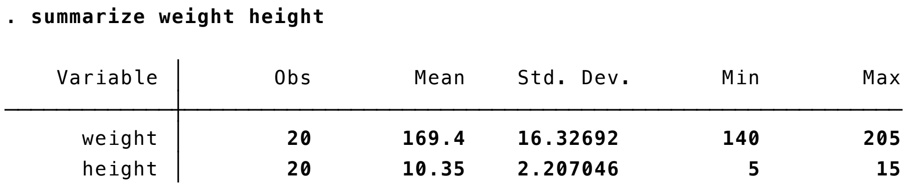
- To label a variable:
- **label var** healthstat “Self-Reported Health Status”

- To label different numeric values the variable may take:
- **label define** vlhealthstat 1 “Excellent” 2 “Very Good” 3 “Good” 4 “Fair” 5 “Poor”
- **label values** healthstat vlhealthstat

Obtaining basic summary statistics

- Summarize command: Use to obtain basic summary statistics of 1 or more variables (mean, standard deviation, min, max, etc.)

summarize [varlist] [if] [in] [weight] [, options]

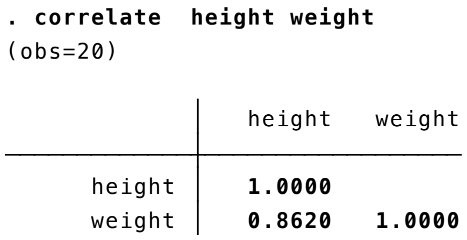


```
. summarize weight height

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
      weight |         20       169.4    16.32692        140        205
      height |         20       10.35    2.207046          5         15
```

- Correlate command: Creates a matrix of correlation or covariance coefficients for 2 or more variables

correlate [varlist] [if] [in] [weight] [, correlate_options]



```
. correlate  height weight
(obs=20)

             |   height   weight
-------------+------------------
      height |   1.0000
      weight |   0.8620   1.0000
```

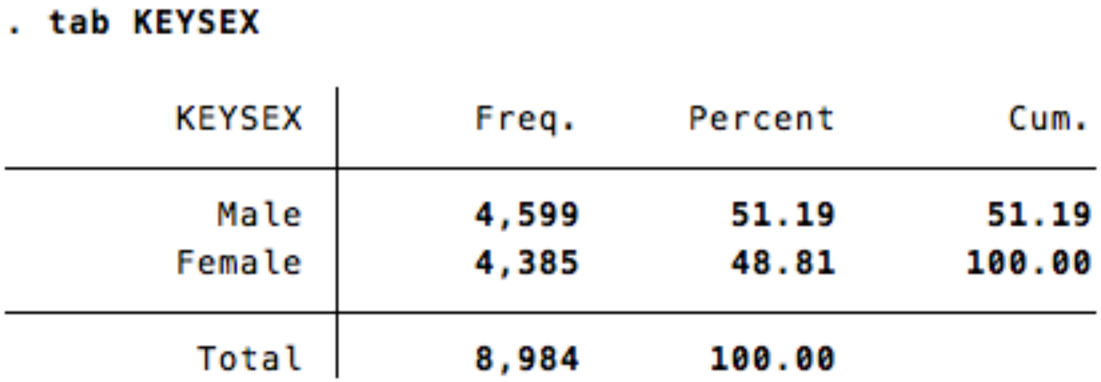
tabulate

- command: **tabulate**

- Calculates and displays frequencies for one or two variables

- Syntax:

- **tabulate** varname [if] [in] [weight] [, options]



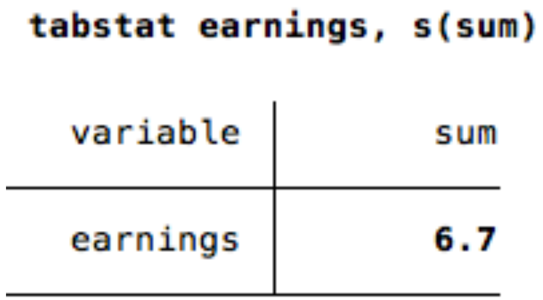
```
. tab KEYSEX

     KEYSEX |      Freq.     Percent        Cum.
------------+-----------------------------------
       Male |      4,599       51.19       51.19
     Female |      4,385       48.81      100.00
------------+-----------------------------------
      Total |      8,984      100.00
```

More detailed descriptives

- Use tabstat command

tabstat varlist [if] [in] [weight] [, options]



```
tabstat earnings, s(sum)

    variable |       sum
-------------+----------
    earnings |       6.7
------------------------
```

- The example above calculates the sum of the variable, but you could specify other statistics as well (median, range, etc.). If you don't specify a particular statistic at the end, then *tabstat* will generate the mean

Creating a new variable: **generate**

- command: **generate**

- Syntax:

- **generate** newvar = exp [if exp] [in range]

- Example:

- **generate** age_sq=age*age

- Notes:

Can type generate or gen for short

Changing Existing variables: **replace**

- Command ‘**replace**’ changes the contents of an existing variable

- Syntax:

  replace oldvar = exp [if exp] [in range]
- **replace** can be using in many circumstances, including
  - Creating binary and categorical variables
  - Fixing values

Ex: Replace responses coded as “no response” (-1 in this case) with missing values

```
replace variable = . if variable == -1
```

Create a Binary Variable

- To create a binary variable (0 / 1):

- Generate a variable equal to 0 for all observations
- Replace it to be 1 for selected observations

- Example, create a binary variable for people with income over $80,000:

```
gen highinc=0
replace highinc=1 if hh_inc>80000
```

Recode Variable

- command: recode
- Syntax:
- recode varlist (rule) [(rule) ...] [, generate(newvar)]
- Example

```
recode sah (1 2 = 1) (3 = 2) (4 5 = 3), gen(sahrecode)
```

Exploring Missing Values

- Missing values are given by "." in STATA
- To count the number of missing values in all variables in dataset, use user-written command **tabmiss**
  - To install, type findit tabmiss in command window
  - To use, type **tabmiss**
- Important Note: you can use "findit" to install other user written commands, as well as help files for commands in STATA
- Can also use **tab** var, m (one variable)

Saving data

If you’ve imported data into STATA from a spreadsheet, text file, etc., you may want to save it as a STATA dataset.

  - This is particularly useful for large datasets, as STATA can generally read its own datasets faster than importing raw data

- Menu: go File → Save (will give you an option to replace the data if it already exists)
- Syntax: **save** [filename] [, save_options]

Graphing/Plotting Data

- Two-way scatter plot

twoway scatter yvar xvar

- Two-way line plot

twoway line yvar xvar

- Two-way scatter plot with linear prediction from regression of y on x

twoway (scatter yvar xvar) (lfit yvar xvar)

- Two-way scatter plot with linear prediction from regression of y on x with 95% CI

twoway (scatter yvar xvar) (lfitci yvar xvar)

Regression Analysis

Fitting a Linear Model To The Data

General notation:

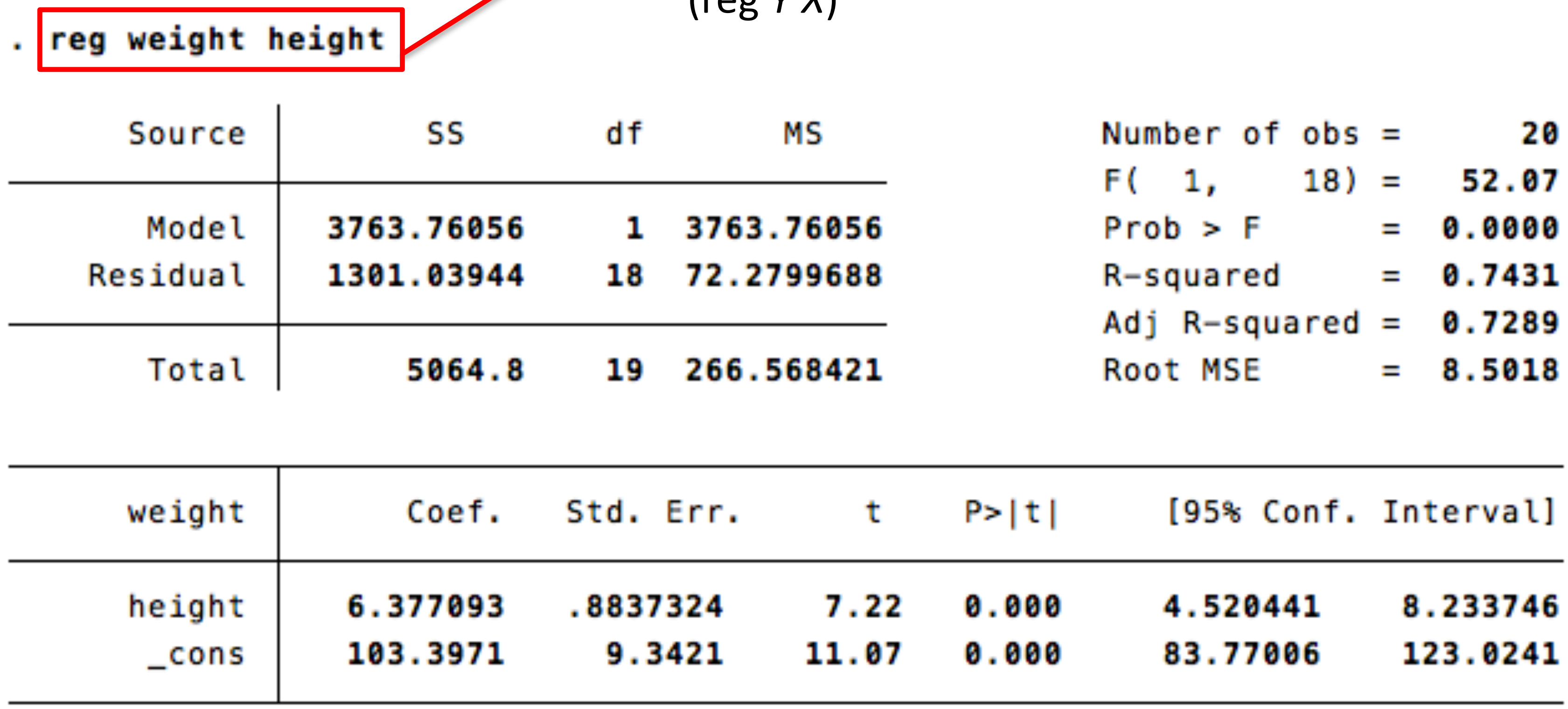
regress depvar [indepvars] [if] [in] [weight] [, options]

Where:

Y is our *dependent* variable
X is our *independent* variable(s)
Note: You may type “reg” instead of “regress”

Fitting a Linear Model To The Data

Stata Output:



```
. reg weight height

      Source |       SS       df       MS              Number of obs =      20
-------------+------------------------------           F(  1,    18) =   52.07
       Model |  3763.76056     1  3763.76056           Prob > F      =  0.0000
    Residual |  1301.03944    18  72.2799688           R-squared     =  0.7431
-------------+------------------------------           Adj R-squared =  0.7289
       Total |      5064.8    19  266.568421           Root MSE      =  8.5018

------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      height |   6.377093   .8837324     7.22   0.000     4.520441    8.233746
       _cons |   103.3971     9.3421    11.07   0.000     83.77006    123.0241
------------------------------------------------------------------------------
```

Follows notation (reg *Y X*)

Post Estimation

Post Estimation

- Obtaining residuals

```
predict residuals, residuals
```

NB: The “residuals” after predict is just the name you want to give to the residuals. You can change this if you want to

- Obtaining fitted values

```
predict fittedvalues, xb
```

Residual Diagnostic and Heteroskedasticity testing

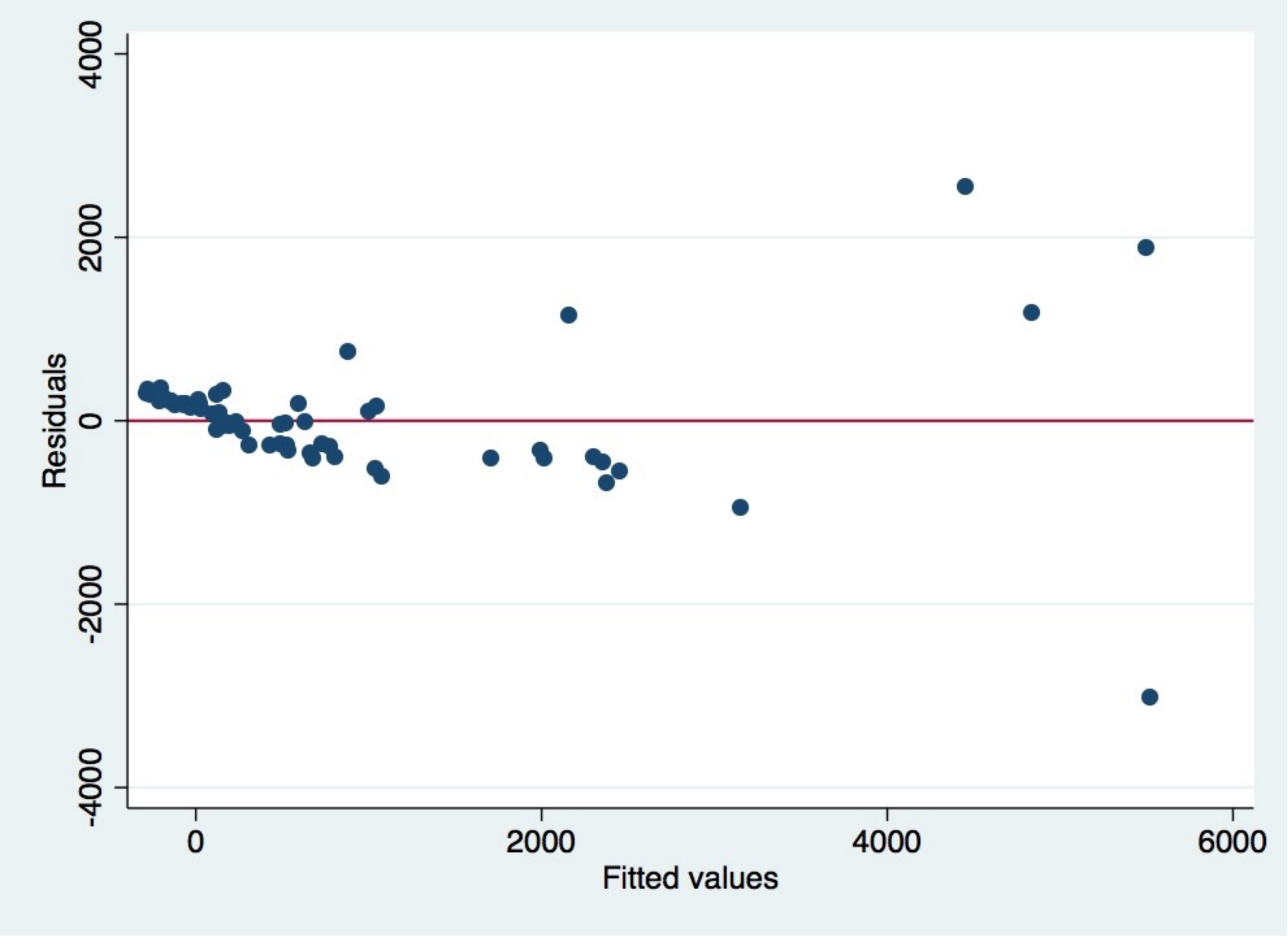
- OLS regression assumes homoskedasticity for valid hypothesis testing. We can test for this after running a regression
- Examine residual pattern from the residual plot

```
rvfplot, yline(0)
```

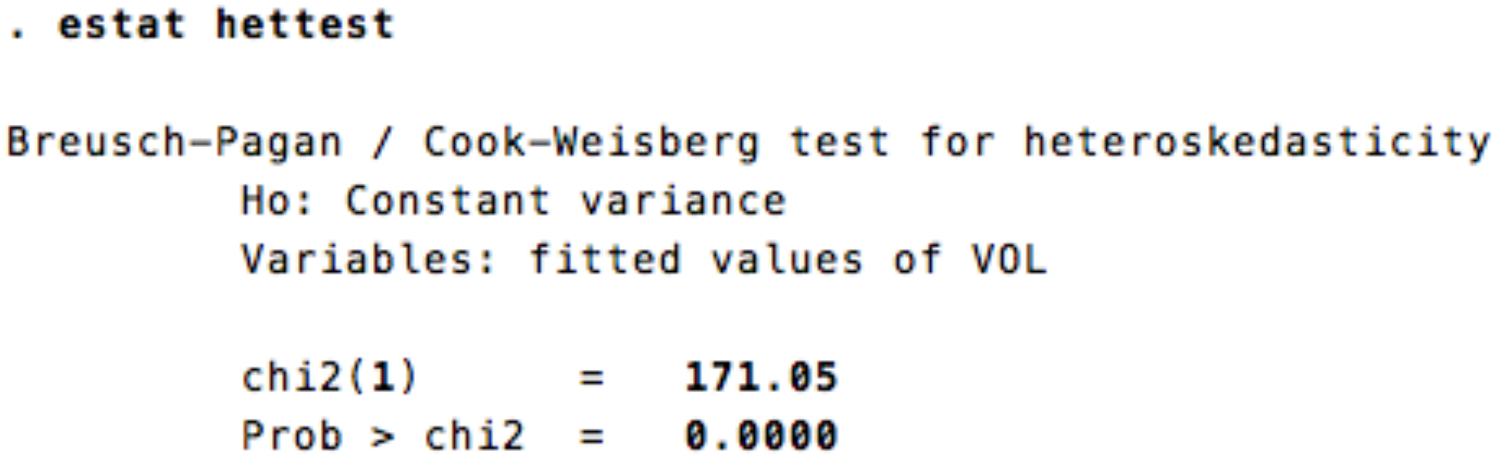
- Heteroskedasticity test

```
estat hettest
```

RVF Plot


Residuals
4000
2000
0
-2000
-4000
0
2000
4000
6000
Fitted values

Test for Heteroskedasticity



```
. estat hettest

Breusch-Pagan / Cook-Weisberg test for heteroskedasticity
         Ho: Constant variance
         Variables: fitted values of VOL

         chi2(1)      =   171.05
         Prob > chi2  =   0.0000
```

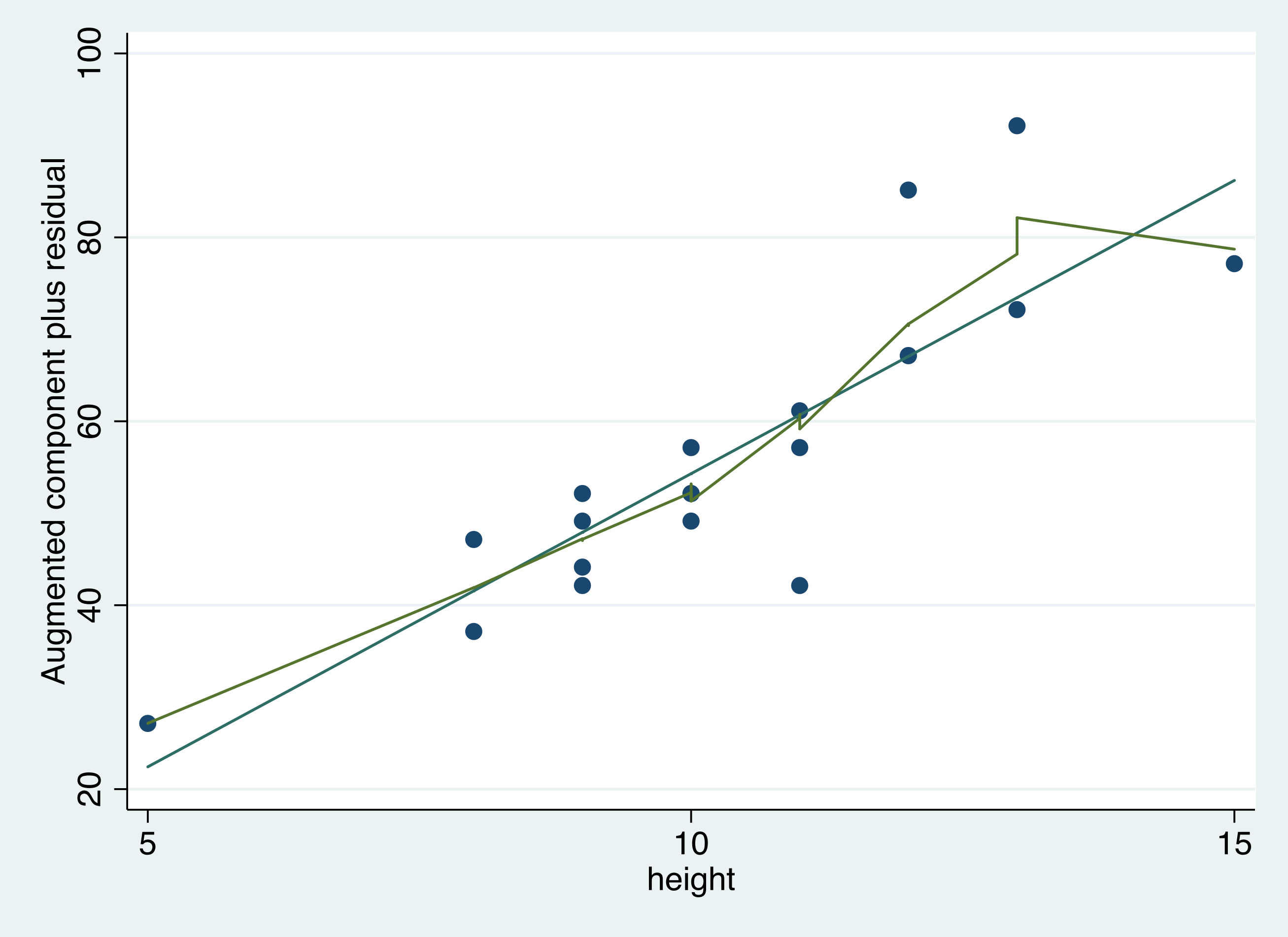
Reject the null (no heteroskedasticity) in favour of the alternative (there is heteroskedasticity of some form).

Linearity testing

- OLS assumes a linear relationship between the Y and X's. We can test for this after a regression:
- Command:

```
acprplot var, lowess
```

ACPRPLOT Stata



Testing for multicollinearity

OLS regression assumption: independent variables are not too strongly *collinear*

Detection:

- Correlation matrix

correlate *varlist* (before regression)

- Variance Inflation Factor

vif (after regression)

Specification testing

- To see if there is omitted variables from the model, or if our model is miss-specified
- Syntax: **estat ovtest**

```
. estat ovtest

Ramsey RESET test using powers of the fitted values of crime
       Ho:  model has no omitted variables
                  F(3, 44) =      6.45
                  Prob > F =      0.0010
```

Standard Errors

- Heteroskedasticity-robust standard errors
  - regress y x_1 x_2…x_n, vce(robust)
- Cluster robust standard errors
  - regress y x_1 x_2…x_n, vce(cluster *clusterid*)
- Bootstrapped standard errors
  - regress y x_1 x_2…x_n, vce(bootstrap)

Storing Estimation Results

- STATA can store the results of your regression via the estimates command:

  estimates store *name*
- This can be very useful in analyzing regression results after running multiple models
- estout package (needs to be installed) can be used to create tables from the regression results that can be exported from STATA. To install, type: ssc install estout, replace

http://repec.org/bocode/e/estout/esttab.html

Other Topics in STATA

Regression commands for other types of outcome variables

- Binary outcomes: probit or logit

(help probit; help probit postestimation)
(help logit; help logit postestimation)

- Ordered discrete outcomes: oprobit or ologit

(help oprobit; help oprobit postestimation)
(help ologit; help ologit postestimation)

- Categorical outcomes: mprobit or mlogit

(help mprobit; help mprobit postestimation)
(help mlogit; help mlogit postestimation)

Panel Data Econometrics

- Pooled Linear Regression

```
regress depvar [indepvars] [if] [in] [weight] [, options]
```

- Random Effects

```
xtreg depvar [indepvars] [if] [in] [, re RE_options]
```

- Fixed Effects

```
xtreg depvar [indepvars] [if] [in] [weight] , fe [FE_options]
```

Working With Do-Files

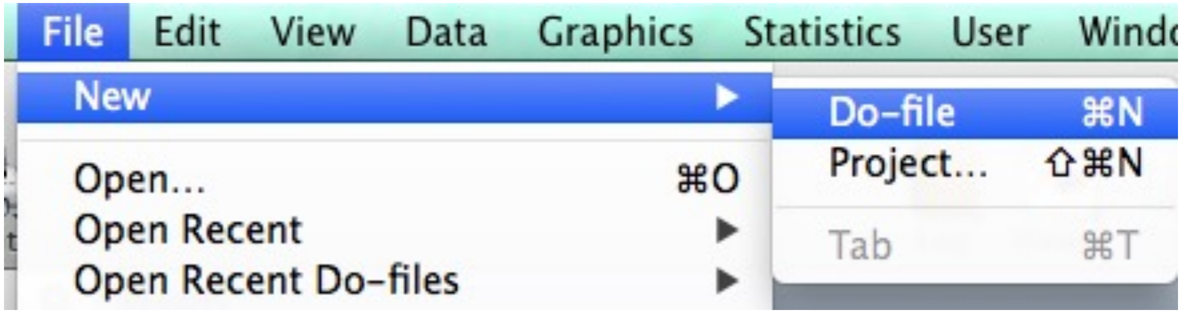
Motivation

Why bother?

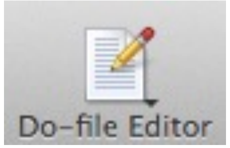
1) We can ovoid tediously running the same set of commands over and over again through the menu/command window
2) Creates a document listing *all* the commands we’ve run
3) Increases our productivity with STATA!

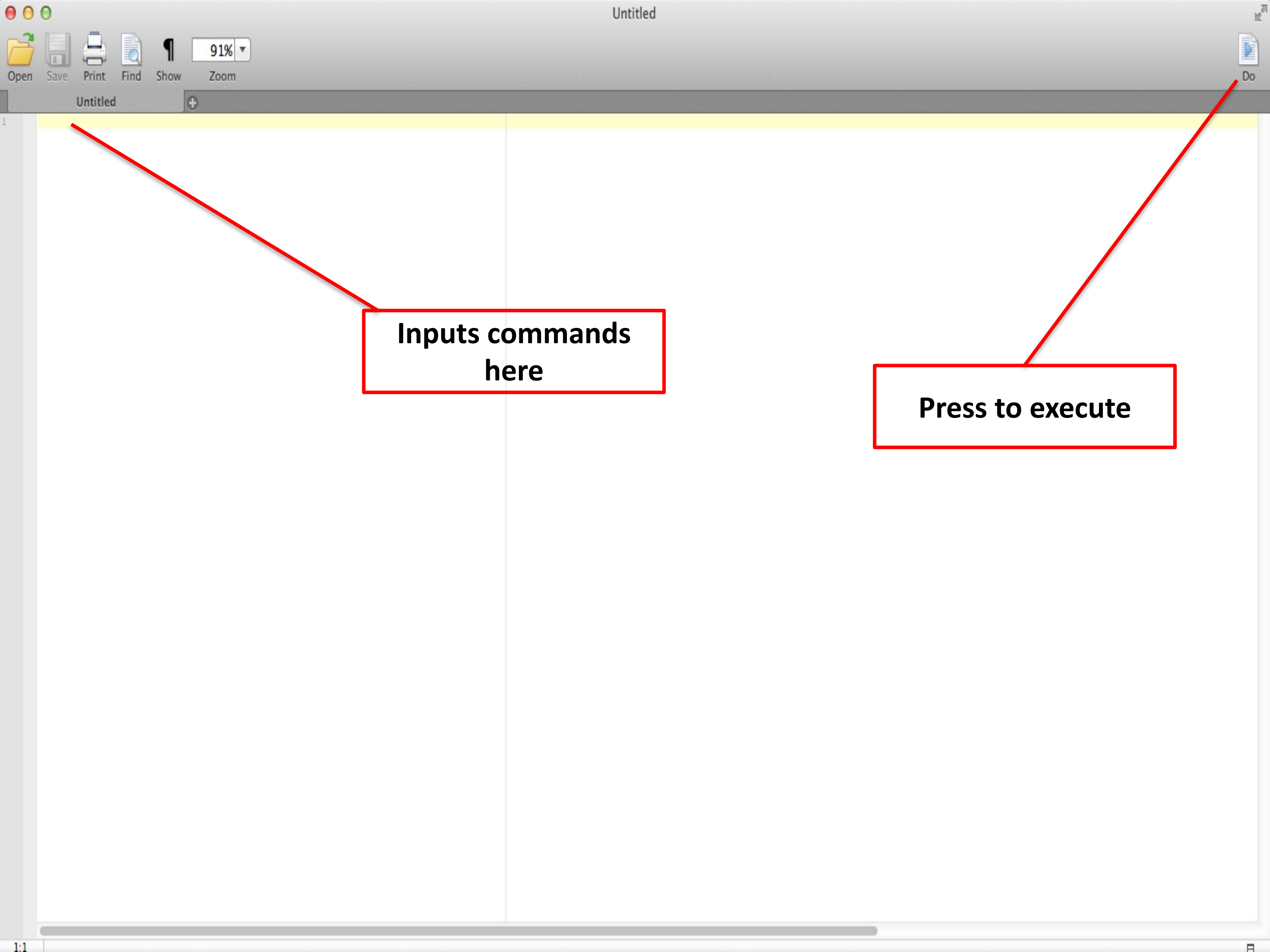
How to get to do file editor:

- File → New → Do-file



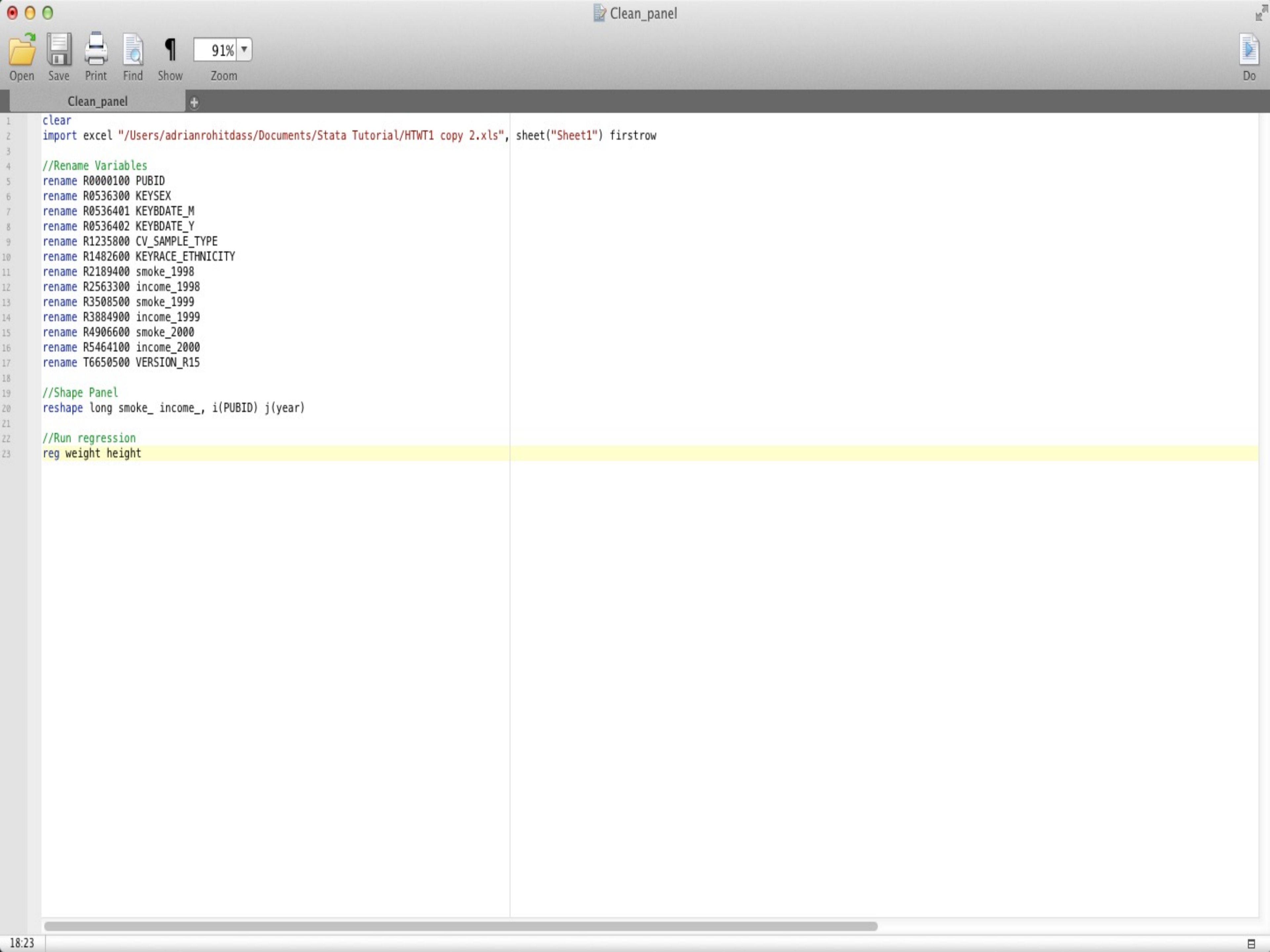
- Or “Do-file Editor” button at top (depending on which version of STATA you have)





Inputs commands here

Press to execute



```
clear
import excel "/Users/adrianrohitdass/Documents/Stata Tutorial/HTWT1 copy 2.xls", sheet("Sheet1") firstrow

//Rename Variables
rename R0000100 PUBID
rename R0536300 KEYSEX
rename R0536401 KEYBDATE_M
rename R0536402 KEYBDATE_Y
rename R1235800 CV_SAMPLE_TYPE
rename R1482600 KEYRACE_ETHNICITY
rename R2189400 smoke_1998
rename R2563300 income_1998
rename R3508500 smoke_1999
rename R3884900 income_1999
rename R4906600 smoke_2000
rename R5464100 income_2000
rename T6650500 VERSION_R15

//Shape Panel
reshape long smoke_ income_, i(PUBID) j(year)

//Run regression
reg weight height
```

Applied Example

- Analysis of Health Expenditure Data in Jones et al. (2013) *Chapter Three*
- The data covers the medical expenditures of US citizens aged 65 years and older who qualify for health care under Medicare.
  - Outcome of interest is total annual health care expenditures (measured in US dollars).
  - Other key variables are age, gender, household income, supplementary insurance status (insurance beyond Medicare), physical and activity limitations and the total number of chronic conditions.
- Data can be downloaded from here (mus03data.dta): https://www.stata-press.com/data/musr.html

Code From Applied Example

```
cd "/Users/adrianr/Desktop/STATA Example" //Change Working Directory

log using "mylogfile.smcl", replace //start log file

clear //remove variables from STATA

use "mus03data.dta" //Load Data

describe //Description of data (output will remain in log file)

table posexp //Frequency table (output will remain in log file)

drop if posexp==0 //Remove individuals with $0 in health expenditures (following example)

*Regression*
reg totexp female income suppins phylim actlim totchr //Regression without age
eststo reg1 //Store results

reg totexp age female income suppins phylim actlim totchr //Regression with age (following example)
eststo reg2 //Store results

rvfplot, yline(0) //RVF Plot
graph export rvfplot.png, replace //Save plot in working directory

estat hettest //Heteroskedasticity test

esttab reg1 reg2 using "myresults.csv", cells(b(fmt(3)star) se(par)) stats (N r2) replace //export results

*Robust regression*
reg totexp female income suppins phylim actlim totchr, robust //Regression without age, HC robust
eststo robust1 //Store results

reg totexp age female income suppins phylim actlim totchr, robust //Regression with age (following example), HC robust
eststo robust2 //Store results

esttab robust1 robust2 using "myresultsrobust.csv", cells(b(fmt(3)star) se(par)) stats (N r2) replace //export results

log close //Close log file
```

STATA Resources

STATA Online Resources

- STATA manuals are freely downloadable from the above site

http://www.stata-press.com/manuals/documentation-set/

- Typing help [topic] in the command window is also useful, but the online manuals generally contain more detail/examples

STATA Online Resources

UCLA Institute for Digital Research and Education

- List of topics and STATA resources can be found here:

http://www.ats.ucla.edu/stat/stata/webbooks/reg/default.htm

Other STATA Resources

- Jones, A.M., Rice, N., d’Uva, T.B., Balia, S. 2013. Applied Health Economics - Second Edition, Routledge Advanced Texts in Economics and Finance. Taylor & Francis
- Cameron, A.C., Trivedi, P.K. 2010. Microeconometrics Using Stata – Revised Edition, Stata Press books.
- Allison, P.D. 2009. Fixed Effects Regression Models, Quantitative Applications in the Social Sciences. SAGE Publications.
- Wooldridge, J. M. (2010). Econometric analysis of cross section and panel data. MIT press
  - Solutions manual (sold separately) contains STATA code and output

Thanks for Listening

Good luck with STATA!