

Econ 294 Lecture 1.pdf

Econ 294 Lecture 10.pdf

Econ 294 Lecture 2.pdf

Econ 294 Lecture 3.pdf

Econ 294 Lecture 4.pdf

Econ 294 Lecture 5.pdf

Econ 294 Lecture 6.pdf

Econ 294 Lecture 7.pdf

Econ 294 Lecture 8.pdf

Econ 294 Lecture 9.pdf

# Applied Economics (STATA) Lab

## ECON 294 A Section 2

### Fall 2016

**Subhra B Saha**

Ph.D in Economics from The Ohio State University, OH – 2008  
Assistant Professor of Economics 2008 - 2015 June at Cleveland  
State University, OH  
Lecturer, UCSC & California State University at Monterey Bay 2015  
– present  
Currently Opening Business in Medical Field

# Contact

- Office: Engineering 2, #453
- Office Hour: Tuesday 11:00 A.M. – 1:00 PM (or by appointment)
- E-Mail: [susaha@ucsc.edu](mailto:susaha@ucsc.edu)

# Today's Agenda

- Syllabus & course policies
- Opening STATA & Datasets in STATA
- Case Sensitivity of Commands
- Logical and Mathematical operators
- Help
- Basic Data Analysis
- Generating Data
- Graphs

# Syllabus

Textbook, Lectures, Assignments,  
Course Policies

# Textbook

- No required textbook
- **Recommended:** Cameron and Trivedi's Microeometrics: Using Stata (Statacorp, College Station, TX).
- Contains step-by-step recipes for conducting different econometric methods.
- Available at S&E Library book reserve

Prof/ta	Saha, Subhra
Course	Economics 294A / 294B ECON 294A / 294B Applied Economics Lab
Cour note	S&E Fall 2016

#### Materials for this course

Title	Author	Call #	Loan Period
Microeconometrics using Stata / A. Colin Cameron, Adrian Colin. Colin Cameron, Pravin K. Trivedi.		Reserves S&E Desk HB139 .C36 2010 NOT CHECKD OUT	2 Hours; No Overnight

# Web Resources

- UCLA's Institute for Digital Research and Education.  
<http://www.ats.ucla.edu/stat/stata/>
- The University of Wisconsin at Madison.  
<https://www.ssc.wisc.edu/sscc/pubs/sfr-intro.htm>
- The Stata List. <http://www.statalist.org>
- Use Google to look for help if you are stuck

# Lab Resources

- Stata IC v14 is installed on all the workstations at all the Windows/PC labs.
- The Windows/PC lab at BE109 (Baskin Engineering 109) and Cowell Apartments are open 24 hours every day throughout the quarter, including holidays occurring during the quarter.
- Many other classes also use the facilities so you may want to check the Lab schedule listed here  
<http://its.ucsc.edu/computer-labs/class-schedules/fall/index.html>
- The PCs run Windows 7 Enterprise 64bit. All of the Windows PCs are configured the same at all the labs and require login with CruzID and Blue password.

# Renting STATA IC (Intercooled Stata) from Stata Corp

6 month License

Perpetual license

Big discount as student of UCSC

# Course Outline

- 10 meetings scheduled (including this one).
- First 5 - data management and programming in Stata.
- Last 5 - will cover econometric tools available in Stata.
- Light reference to econometric techniques

# Topics in the data management

- Commands to
  - generate format and label variables
  - logical operators
  - wildcards
  - encoding and decoding variables
  - handling missing variables
  - installing packages
  - reshaping and collapsing datasets
  - merging and appending datasets

# Topics in the programming

- using macros (local and global),
- scalars and matrices
- control structures, a.k.a. loops (for each, for values, while, for varlist, etc.)
- trace; restore/preserve
- esttab/outreg.

# Econometric Tools

- Non linear analysis
- Non Parametric Methods
- Longitudinal data OR Panel Data
- Matching
- Difference in Difference
- Regression discontinuity
- Instrumental Variables
- The schedule has been set to coordinate well with material from Econ 216.
- Focus on the different tools to perform analysis (command syntax, options, etc.)

# Lectures, Assessments and Final Exam

- Use the Lectures as your guide
- Posted on E Commons before the class each week (by 3 PM)
- Hard copy due in class next week
- Bring a soft copy with you to correct your answers as I go over the answers in class
- Final is Take Home – submit it to me – will give you instructions later!

# Evaluation

Grade Assignment Distribution

Assessed Area	Percent of Grade
Assessments	60%
Final	40%
<b>TOTAL</b>	<b>100%</b>

# Accessing Stata in Lab

1. Login to workstation with your CruzID and Blue password.
2. Open the "Class Folders" folder on the desktop.
3. Open "Economics" folder.
4. Open "Stata" folder.
5. Open "Stata 14".

# What do you expect to see

- Command window – type in your commands/codes
- Output window – see results (including errors)
- Review window – stores the codes you just used
- Variables window – shows variables in use

# What we want to practice

- Do File: Executable file with all your codes in one file
- Avoids mess; stores commands forever
- Add comments to tell you and your colleagues what are you trying to do
- Can become really big
- We will see how to make do files later
- Command Window – really used to test your code

# Saving Files

- Temporary user data files can be saved to the Desktop
- Permanent copies will need to be saved elsewhere.
- User's can save or copy files to their UCSC Home Directory which is mapped as "X:" on the PCs,
- Or use a USB Flash Drive, or email.

# Opening data files from STATA and Saving them for future use

- Files pre loaded in stata: **sysuse dir**
- **sysuse auto.dta, clear**
- **clear** : is an option, which clears any other dataset in the memory and allows stata to load the dataset that you want.
- Viewing data – click on data editor – works for small number of data
- If you modify this file then you need to save it with a different file name.
- save X:\mod\_auto.dta, replace (for PC)
- save “Users/cmto/mod\_auto.dta, replace (for MAC)
- replace: is an option, which allows you to make changes to the code and store a modified dataset
- Good practice: save the original as “raw” or “original” & make changes to it & then save it as modified

# Types of files in stata

- Data: extension **.dta**
- Do Files: Collection of commands **.do**
- Results: **.smcl**
- **ALWAYS SAVE RAW/ORIGINAL FILE**
- When making changes in the original file, save that file separately

# Opening data files NOT stored in STATA and saving them for future use

- Files that you want to use: telling stata where the file is: specify pathname
- **use “C:\Econ294A\original dataset.dta”, clear [for PC]**  
(For not preloaded)
- For opening pre-loaded data set: **sysuse**
- **save “C:\Econ294A\modified dataset.dta”, replace [for PC]**
- **use “Users/cmto/Econ294A/original dataset.dta”, clear [for MAC]**
- **save “Users/cmto/Econ294A/modified dataset.dta”, replace [for MAC]**

# Interpreting code

- C: (This is where Stata is pulling from)
- Econ294 (This is the File name)
- Original dataset.dta (This is the file name)(.dta shows that it is a stata file)

# Commands and their case sensitivity

- Stata commands are all written in lower case letters
- These commands are case sensitive
- **summarize** (summary stats), **count** (# Observations), **describe** (variable description), **tabulate** (frequency by), **table** (make a table of), **tabstat** (table of summary stats)
- Example:
- **summarize** price
- **SUMMARIZE** price

# Mathematical Operators

- You can use stata as your personal calculator
- Type: **2+3**
- Then type: **display 2+3**
- **display 2\*5**
- **display 204/3**
- **display 3-9\*4+2^3**
- **display (3-9)\*4+2^3**

# Logical Operators

- if
- and (&)
- or (|)
- not (! OR ~)
- equal (==)
- greater than(>)
- less than (<)
- greater than equal to (>=)
- less/smaller than equal to (<=)

# Examples

- Type: **count if foreign=1**
- Single equal sign assigns value
- **count if foreign==1**
- **count if foreign==0**
- **count if foreign!=1**
- **count if foreign^=1**
- **count if foreign**
- **count if !foreign**
- If you want two or more conditions met then use **&**
- **count if mpg<=20 & price<6000**
- **count if mpg<=20 & price<3000**
- If you at least one condition met use **|**
- **count if mpg<=20 | price<3000**
- **count if foreign==1 | price>6000**
- **count if (foreign==1 & price<3000) | (foreign==0 & price>5000)**

# Foreign is a dummy variable

- Foreign = 1 means the car is foreign made
- Foreign = 0 means the car is not foreign made

# Treatment of Missing variables

- If you omit == sign after if statement STATA will evaluate whether the variable is greater than 0
- Missing values (.) are interpreted as **infinity**
- **Example**
- **count if rep78**
- **count if rep78 & rep78!=.**
- **count if rep78 & !missing(rep78)**
- **count if rep78>0 & rep78<.**

# Help

- help is a function in stata: you can associate with any command and you will see the syntax of that command
- Type: **table foreign, contents(mean price)**
- **Example:**
- **table foreign, contents(mean price)**

# Data Analysis: describe

- Gives you details about variables
- Example
- **describe**
- **describe, simple**
- **String variable is a non-numeric variable**

# Data Analysis: summarize

- Gives you # observations, mean, median, min, max about variables
- You can do this for all variables or for specific variables
- Example
- **summarize mpg**
- **summarize mpg, detail**
- **summarize**
- Look at variable “make” – it is a string variable (non-numeric)

# Data Analysis: count

- Without additional options or parameters specified, the count command simply counts all observations in the dataset
- More useful with conditions of interest
- Example
- **count**
- **count if mpg<=20**
- **count if mpg>3000**

# Data Analysis: tabulate

- This command lists variables
- Abbreviated as **tab**
- **One or two variables may be used as arguments**
- **Example**
- **tab headroom**
- **tab headroom foreign**
- **tab headroom, summarize(price)**
- **tab foreign, summarize(turn)**

# Data Analysis: table

- This is a great command to learn about dataset
- Example
- **table headroom foreign**
- **table headroom foreign, contents(mean price)**

# Data Analysis: tabstat

- This variable is effective at producing summary stats
- Example
- **tabstat price headroom weight,  
statistics(mean median p95)**
- **tabstat price headroom weight,  
statistics(mean median p95)  
columns(statistics)**

# Generating Variables: `gen`

- We are going to use the generate command (written as `gen`)
- Examples
- **`gen cheap=1 if price<4500`**
- **`sum cheap`**
- STATA does not know which value to assign if price is greater than or equal to 4500 : hence create missing values

# Creating a dummy/indicator variable called cheap

- **Type: gen cheap=0 if price>=4500**
- Because cheap is already defined this gives an error!
- **replace cheap=0 if price>=4500**
- **sum cheap**

# Dropping variables: stata drops variables from memory

drop cheap

Still want to generate a  
dummy/indicator variable for pricey  
cars?

Go back to gen and create “cheap”

# Creating De-Meanned Variables

- This is useful in panel data and time series
- Example:
- **sum trunk**
- **gen dmtrunk = trunk-13.75676**
- **Sum dmtrunk**
- In the programming part of this class, we will learn how to store these results in the program memory & use them in automated fashion

# Graphs: Using Drop Down Menu

- Utilize drop down menu to make graphs
- Benefits: the code used to make the graph will show itself in output window
- You can copy it and paste it in your do file: in case you did not know the proper syntax
- Use drop down menu to create: Two Way Plot

# Step by Step use of Drop Down Menu to make scatter graph

- Step 1: Click on Graphics
- Step 2: Click on Twoway Graph
- Step 3: Click Create
- Step 4: Click Basic Plots & Scatter
- Step 5: Y variable: price X Variable: mpg
- Step 6: Click Accept & OK
- **twoway (scatter price mpg)**

# Overlaying Graphs

- **twoway (scatter price mpg)**
- **twoway (lfit price mpg)**
- **twoway (lfit price mpg) || (scatter price mpg)**

# How to copy graphs?

- Simply left click on the graph
- right click & left click on save
- Paste on word file
- Do not forget to save that word document you are pasting things on.
- You can do this to any stata results

# Assignment 1 posted on eCommons

- Please bring hard copy of your answers to class for submission
- Please bring soft copy to check answers that I will provide
- Lose 10% each day you are late

# Lec 10

- **Instrumental Variables:**
  - **Why bother?**
  - **How to execute IV in STATA?**
  - **Tests of important assumptions using STATA**
- **Regression Discontinuity**
  - **Why Bother?**

**How to execute in STATA**

Random Allocation into Treatment  
and Control Groups is the  
cornerstone of good experiments

**Exogenous Variation leads to  
identification**

**Few Other Issues with RCT:**  
What happens when the individuals in the treatment  
have an option to refuse treatment  
or  
people in control want the treatment?  
We will talk about these complicated cases in Spring

- **Cross Sectional Data**

$$y = \log(\text{wage}_i)$$

- $x = \text{Years of Schooling}$

- Objective: Explain the variation in  $y$  with variation in  $x$

Unit of analysis (i): *individuals*

$$y_i = \alpha + \beta x_i + \delta_1 Control^1_i + \dots + \delta_k Control^k_i + u_i \dots \quad (1)$$

$$\text{cov}(x_i, u_i) \neq 0$$

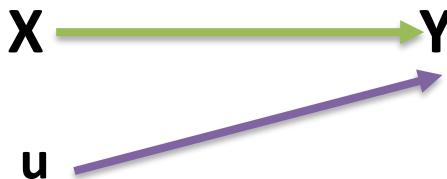
$\hat{\beta}$  is an inconsistent estimate of  $\beta$  with OLS

# Visual Demonstration of the Problem

**Randomized Control Trial:**



**Good Regressions:**



**Bad Regressions:**

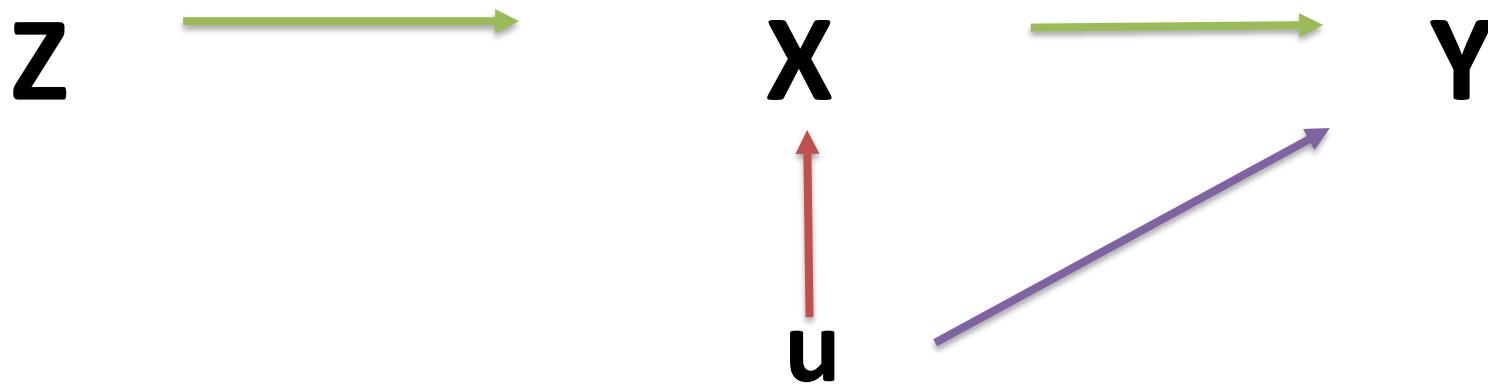


- More able persons tend to get more education i.e. have higher years of schooling.
- More able persons also tend to earn more

If variables in error are specific to “groups” [family, racial makeup, gender makeup] then we can use **fixed effects** utilizing the “within group variation”

But we DO NOT have panel data in this case!

# Visual Demonstration of the Solution: IV/Natural Experiment (DiD)/RD/:



*IV : Strong / Weak*

$$\text{cov}(z_i, x_i) \neq 0$$

*IV : Valid / Invalid*

$$\text{cov}(z_i, u_i) = 0$$

$$y_i = \alpha + \beta x_i + \delta_1 Control^1_i + \dots + \delta_k Control^k_i + u_i \dots \dots (1)$$

*First Stage*

$$x_i = \omega_0 + \theta_1 z^1_i + \dots + \theta_k z^k_i + \eta_1 Control^1_i + \dots + \eta_k Control^k_i + v_i \dots \dots (2)$$

*Use First Stage Estimates to find :  $\hat{x}_i$*

*Second Stage*

$$y_i = \pi_0 + \lambda \hat{x}_i + \gamma_1 Control^1_i + \dots + \gamma_k Control^k_i + \xi_i \dots \dots (3)$$

Want you to understand that the IV variables go in the first stage of the regression but **NOT** in the second stage

This is called exclusion restriction

# Possible IVs for returns to Schooling

- Mother's Education
- Distance from school

$IV=Z$	$cov(Z,X)$	$Cov (Z,U)$
Mother's Education	Strong	Possibly invalid
Distance From School	Possibly weak	Valid

# ivreg2: download package

## 2SLS/GMM

ivreg2 depvar [varlist1] (varlist2=instlist) [if] [in] ,  
options

Options: first, ffirrst, robust,  
cluster

Matrices Stored: F stat & other  
diagnostics – easy to input on the  
results table using outreg2

- The variable that you are instrumenting is called the endogenous variable
- In the returns to education model that we are talking about we are instrumenting: schooling – endogenous variable
- The instruments we are going to use will be: mom\_school and dist

<b>idcode</b>	<b>wage</b>	<b>schooling</b>	<b>mom_school</b>	<b>dist</b>	<b>exp</b>
1	20	10	10	2	2
2	22	12	12	5	3
3	18	18	12	8	4
4	22	15	15	3	3
5	21	17	14	2	4
6	20	20	18	9	7
7	43	25	20	4	1
8	38	28	21	8	1
9	27	22	16	8	2

# IVREG2 Codes, Interpretations and Diagnostics

- use **class10data1**
- sum
- reg wage schooling exp
- estimates store ols
- ivreg2 wage (schooling=mom\_school) exp, first ffirst
- estimates store iv1
- ivreg2 wage (schooling=dist) exp, first ffirst
- estimates store iv2
- ivreg2 wage (schooling=mom\_school dist) exp, first ffirst
- estimates store iv3
- estimates table ols iv1 iv2 iv3, star stats(N)

- Sargan Test Stat:  $H_0$  is the hypothesis that the instrument is valid. So you don't want to reject the null.
- P-value larger than .05 for instruments to be valid

# With Robust Option

- \* With Robust Option
- **reg wage schooling exp, robust**
- **estimates store ols**
- **ivreg2 wage (schooling=mom\_school) exp, first ffirst robust**
- **estimates store iv1**
- **ivreg2 wage (schooling=dist) exp, first ffirst robust**
- **estimates store iv2**
- **ivreg2 wage (schooling=mom\_school dist) exp, first ffirst robust**
- **estimates store iv3**
- **estimates table ols iv1 iv2 iv3, star stats(N)**

Can we do IV with panel data?

Easy Way: Do FE by introducing the panel variable as a dummy variable

Slightly more involved way: Yes with  
**xtivreg2 – need to download it**

<b>idcode</b>	<b>wage</b>	<b>scho oling</b>	<b>mom _sch ool</b>	<b>dist</b>	<b>exp</b>	<b>state</b>
1	20	10	10	2	2	1
2	22	12	12	5	3	1
3	18	18	12	8	4	1
4	22	15	15	3	3	2
5	21	17	14	2	4	2
6	20	20	18	9	7	2
7	43	25	20	4	1	3
8	38	28	21	8	1	3
9	27	22	16	8	2	3

# IVreg2 doesn't take dummies

- Use tab, (var) gen(new var)

# With Robust and State Fixed Effects

- **use class10data2**
- **tab state, gen(st)**
- **\* With Robust and State Fixed Effects**
- **reg wage schooling exp st1 st2, robust**
- **estimates store ols**
- **ivreg2 wage (schooling=mom\_school) exp st1 st2, first ffirst robust**
- **estimates store iv1**
- **ivreg2 wage (schooling=dist) exp st1 st2, first ffirst robust**
- **estimates store iv2**
- **ivreg2 wage (schooling=mom\_school dist) exp st1 st2, first ffirst robust**
- **estimates store iv3**
- **estimates table ols iv1 iv2 iv3, star stats(N)**

# When IV sensitive to FE

- The beta coefficients blow up and change signs and also have small p-values
- Even though Hansen statistic says we are all good

# Regression Discontinuity Design

# Thistlethwaite and Campbell

- They studied the impact of merit awards on future academic outcomes
- Awards allocated based on test scores
- If a person had a score greater than  $c$ , the cutoff point, then she received the award
- Simple way of analyzing: compare those who received the award to those who didn't.

Why is this wrong? - Confounding:  
factors that influence the test score  
are also related to future academic  
outcomes (income, parents'  
education, motivation)

**Thistlethwaite and Campbell noted  
that they could compare individuals  
just above and just below the cutoff  
point**

# Validity

- Simple idea: assignment mechanism is completely known
- We know that the **probability of treatment (i.e. getting merit awards)** jumps to 1 if **test score > c**
- Assumption is that individuals cannot manipulate with precision their assignment variable (think about the SAT)

# Comparable individuals: near cutoff point

- If treated and untreated individuals are similar near the cutoff point then data can be analyzed as if it were a (conditionally) randomized experiment
- If this is true, then background characteristics should be similar near  $c$  (can be checked empirically)
- The estimated treatment effect applies to those near the cutoff point (external validity)

# Aside

- Validity doesn't depend on assignment rule being "arbitrary"
- Hinges on assignment mechanism being known and **free of manipulation with precision**
- Manipulation example 1: Test with few questions and plenty of time
- Manipulation example 2: DMV test to get a driving license
- Again: some manipulation is fine (you can always study harder, for example).
- Precision is the key

- Simplest case is linear relationship between  $Y$  and  $X$

$$Y_i = \beta_0 + \beta_1 T_i + \beta_3 X_i + \epsilon_i$$

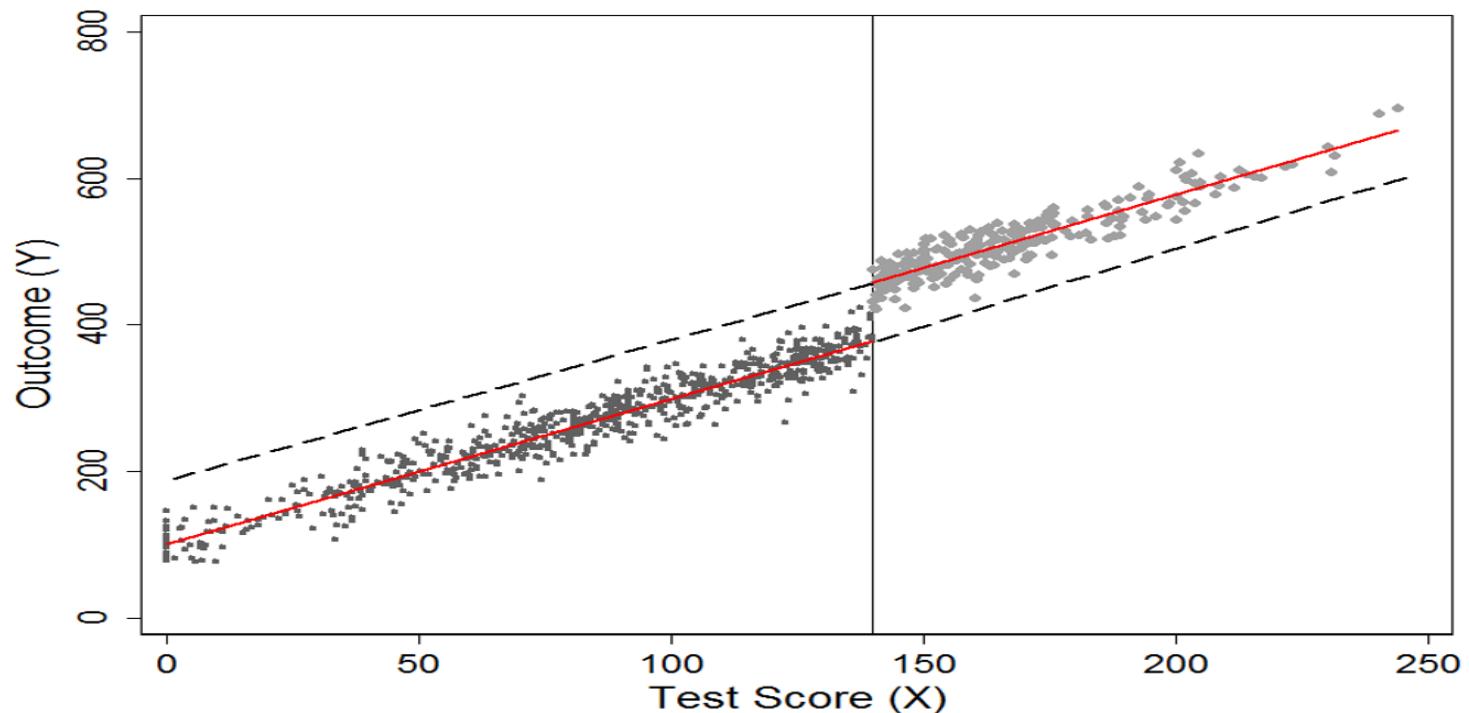
- $T_i = 1$  if subject  $i$  received treatment and  $T_i = 0$  otherwise. You can also write this as  $T_i = \mathbf{1}(X_i > c)$  or  $T_i = \mathbb{1}_{[X_i > c]}$
- $X$  is the assignment variable (sometimes called “forcing” or “running” variable)
- Usually centered at cutoff point
- $Y_i = \beta_0 + \beta_1 T_i + \beta_3(X_i - c) + \epsilon_i$ . Treatment effect is given by  $\beta_1$ .
- $E[Y|T = 1, X = c] = \beta_0 + \beta_1$  and  $E[Y|T = 0, X = c] = \beta_0$ .
- $E[Y|T = 1, X = c] - E[Y|T = 0, X = c] = \beta_1$ .

# Counterfactual = Extrapolation

In RDD the counterfactuals are conditional on X as in a conditionally randomized trial

- Note that the estimation of treatment effect in RDD depends on extrapolation
- To the left of cutoff point only non-treated observations
- To the right of cutoff point only treated observations
- What is the treatment effect at  $X = 130$ ? Just plug in:
- $E[Y|T, X = 130] = \beta_0 + \beta_1 T + \beta_3(130 - 140)$

Dashed lines are extrapolations



## \* Regression Discontinuity

```
clear all  
set obs 1000  
set seed 1234567
```

### \* Generate forcing variable

```
gen x = rnormal(100,  
50)  
replace x=0 if x < 0  
drop if x > 280  
sum x, det
```

### \* Treated if X > 140

```
gen T = 0  
replace T = 1 if x >  
140  
sum T
```

```
* Linear example with treatment effect  
capture drop y  
gen y = 100 + 80*T + 2*x +  
rnormal(0, 20)  
  
scatter y x if T==0,  
msize(vsmall) || scatter y  
x if T==1, msize(vsmall)  
///  
    legend(off) xline(140,  
lstyle(foreground)) || ///  
    lfit y x if T ==0,  
color(red) || lfit y x if T  
==1, color(red)  
ytitle("Outcome (Y)") ///  
    xtitle("Test Score  
(X)")  
graph export  
linear_ex.png, replace
```

### \* Linear example with NO treatment effect

```
capture drop y1  
gen y1 = 100 + 0*T + 2*x +  
rnormal(0, 20)
```

```
scatter y1 x if T==0,  
msize(vsmall) || scatter y1  
x if T==1, msize(vsmall) ///  
    legend(off) xline(140,  
lstyle(foreground)) || ///  
    lfit y1 x if T ==0,  
color(red) || lfit y1 x if T  
==1, color(red)  
ytitle("Outcome (Y)") ///  
    xtitle("Test Score (X)")
```

```
graph export  
linear_ex_noe.png,  
replace
```

# Running a regression to capture the Treatment Effect

- \* Following model recaptures simulated data
- **reg y T x x2**
- \* Could center
- **gen x\_c = x - 140**
- **gen x2\_c = x2-140**
- **gen x3\_c = x3-140**
- **reg y T x\_c x2\_c**
- **You can run the same regression with rdrobust**

- \* Install RDD-related packages

**ssc install rd, replace**

**ssc install cmogram, replace**

- \* Install: rdrobust

**<https://sites.google.com/site/rdpackages/>**

# Housekeeping

- If you are **Section 2** student or a student who comes to this class on **Thursday** then please submit.
- Assignment 1 + Assignment 2 on next Thursday **Oct 6** **in class – hard copy (print out of a word file with codes and results)**
- This applies if you have **switched** from Tuesday to Thursday class.
- If you are **Section 1** student or a student who comes to this class on **Tuesday** then please submit **Assignment 1** on **Tuesday Oct 4** **in class – hard copy (print out of a word file with codes and results)**

# Today

- Few more data cleaning commands
- Writing do–files instead of simply using the command window
- review and replicate
- clear to our colleagues
- how to set up your working directory
- install packages from the Internet
- mmerge
- append
- outreg2

# More Data Cleaning Commands

- Arranging variables: Descending and Ascending Order
  - **sort varname, stable**
- Renaming variables
- **rename**
  - **rename [oldname] [newname]**
  - Creating Dummy Variables using **tabulate** or **tab**  
**sysuse auto, clear**
  - **tab foreign**
  - **tab foreign, gen (f)**

# Do File create/save/execute/edit them

- do–file is a script that contains a series of commands for Stata to execute
- clicking on its icon (a page of paper with a pencil)
- Type: doedit
- Edit do file
- Save file specifying file path with option replace
- We can execute all of the commands listed within a do–file by clicking on the Execute (do) icon – Not the same as do quietly
- Highlighting part of the do–file before clicking the execute icon will only implement the selected commands.

# Comments: Method 1

- Beginning of line: Asterix symbol: \*
- Example:

**\*This is a comment. Stata will not execute this comment**

**sysuse auto, clear**

# Comments: Method 2

- If we want to write comments on the same line as a command we can do so by inserting two slashes “//” after the Stata command.
- Example:

**sysuse auto, clear // I am another form of comment**

# Comments: Method 3

- In order to write longer comments that run over several lines of a do–file we will begin our comment with “/\*” and end it with “\*/”.
- Example:

```
/* This is a long comment intended to show  
that a new Asterisk (*) symbol is not needed at  
the beginning of each new line of the comment  
*/
```

# Line Breaks

- Some of the commands that we will need to write will be rather long. When we encounter such a long command, it is in our best interest to split it up for improved clarity. Two methods for splitting lines are presented below.

# Line Break: Method 1

- Placing “/\*” will break a line at the point where it is inserted. When this is done, the next line of code should begin with “\*/”.
- Example:

```
twoway (scatter price mpg) (lfit price mpg if  
foreign==1) /*  
*/(lfit price mpg if foreign==0)
```

# Line Break: Method 2

- Three consecutive slashes, “///”, can also be used at the desired break point

```
twoway (scatter price mpg) /*  
*/ (lfit price mpg if foreign==1) /*  
*/(lfit price mpg if foreign==0) ///
```

# Combination of both - Uncommon

**twoway (scatter price mpg) /\* makes a scatter plot of price and mpg**

**\*/ (lfit price mpg if foreign==1) /\* overlays a best fit line for foreign cars**

**\*/(lfit price mpg if foreign==0) /// overlays a best fit line for domestic cars**

## Too Much Commentary

The screenshot shows a Windows application window titled "Do-file Editor - Lecture 2\*". The menu bar includes File, Edit, View, Project, and Tools. Below the menu is a toolbar with various icons. The main area contains a code editor with the following content:

```
32
33     ***** The Do-file Begins Here *****
34
35     sysuse auto.dta /* I am opening the preloaded auto.dta data */
36
37     /* Next we will generate a variable that indicates a vehicle is large if
38     it weights more than 4000 pounds and has a length greater than 200 inches.
39     If both conditions are met, the variable, which we will name "large_ind",
40     will take a value of 1. Otherwise the indicator will equal zero. */
41
42     /* After recent discussions with our colleagues,
43     the threshold has changed to 3500 pounds */
44
45     generate large_ind = 1 if weight>=3500 & length>=200
46     replace large_ind = 0 if large_ind!=1
47
48     summ large_ind
49
50     ***** Here ends the do-file *****
51 |
```

The status bar at the bottom shows "Ready" on the left and "Line: 51, Col: 0 CAP NUM OVR" on the right.

## An Acceptable Amount of Commentary

```
Do-file Editor - Lecture 2*
File Edit View Project Tools
Lecture 2*
52
53 *** Begin Do-File ***
54
55 sysuse auto.dta, clear /* I am opening the preloaded auto.dta data */
56
57 /* weight threshold has changed from 4000 to 3500 pounds */
58 generate large_ind = 1 if weight>=3500 & length>=200
59 replace large_ind = 0 if large_ind!=1
60
61 summ large_ind
62
63 *** End Do-File ***
64
```

Ready Line: 63, Col: 7 CAP NUM OVR

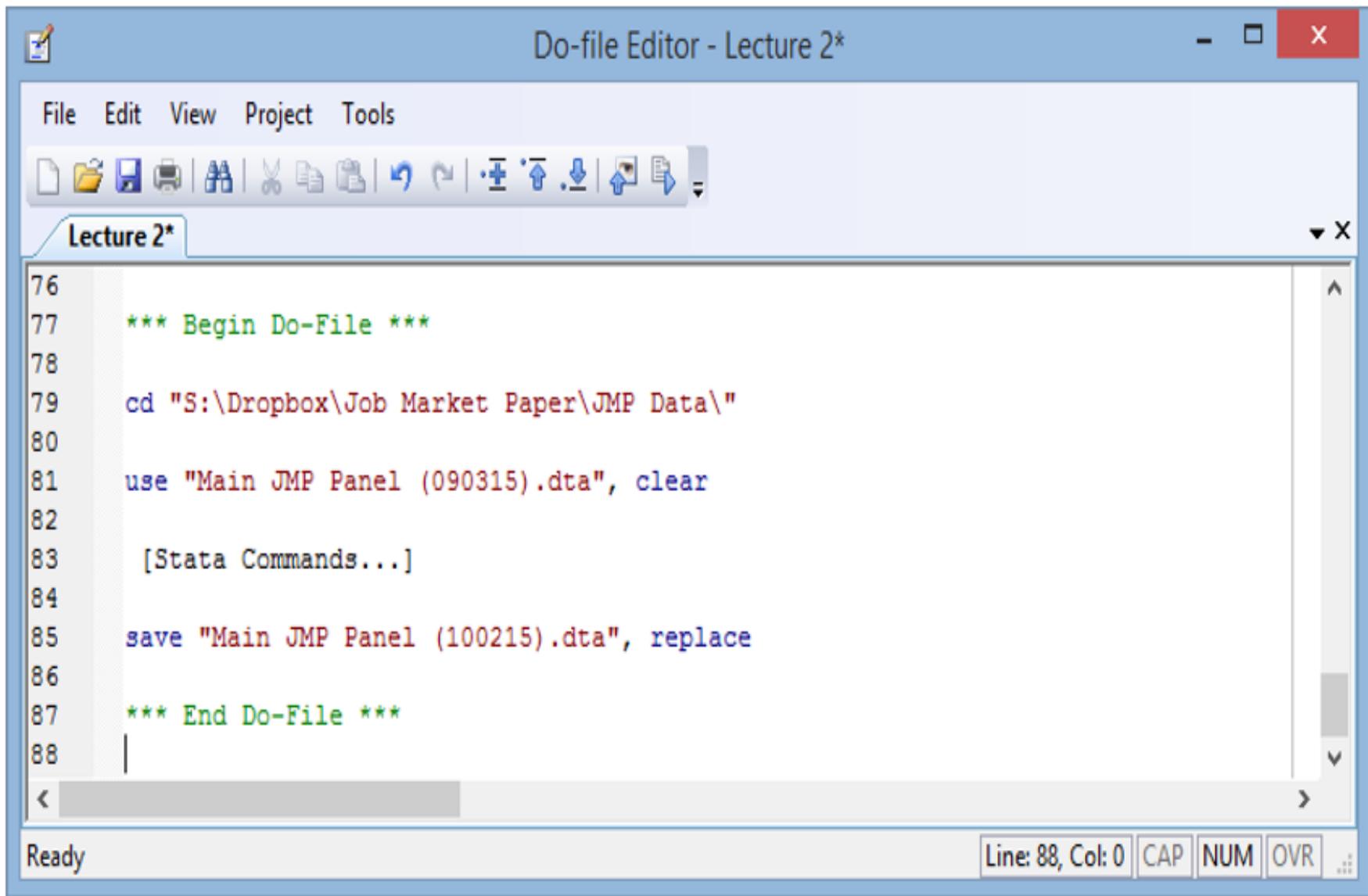
# Starting and Saving a do file

The screenshot shows a Windows application window titled "Do-file Editor - Lecture 2\*". The menu bar includes File, Edit, View, Project, and Tools. The toolbar contains icons for file operations like Open, Save, Print, and Undo/Redo. The main text area displays a Stata do-file script:

```
64  
65 *** Begin Do-File ***  
66  
67 use "S:\Dropbox\Job Market Paper\JMP Data\Main JMP Panel (090315).dta", clear  
68  
69 [Stata Commands...]  
70  
71 save "S:\Dropbox\Job Market Paper\JMP Data\Main JMP Panel (100215).dta", replace  
72  
73 *** End Do-File ***|
```

A yellow callout box with a black border and rounded corners is positioned over the line "Please say that again". The status bar at the bottom left says "Ready" and the bottom right shows "Line: 73, Col: 19" and keyboard indicator lights for CAP, NUM, and OVR.

# Using Change Directory command *cd*



The screenshot shows a 'Do-file Editor - Lecture 2\*' window with the following Stata commands:

```
76
77     *** Begin Do-File ***
78
79     cd "S:\Dropbox\Job Market Paper\JMP Data\""
80
81     use "Main JMP Panel (090315).dta", clear
82
83     [Stata Commands...]
84
85     save "Main JMP Panel (100215).dta", replace
86
87     *** End Do-File ***
88 |
```

The window includes a menu bar (File, Edit, View, Project, Tools), a toolbar with various icons, and status bars at the bottom indicating 'Ready' and 'Line: 88, Col: 0'.

**mmerge:** is used to add *columns* (*new variables*) to an existing dataset using another dataset

But It may not be pre loaded in stata

How would you know? Type

**help mmerge**

What does STATA tell you?

# Installing Packages using “*findit*”

Many commands come preloaded with Stata, but occasionally we will want to use a command that is not already installed

User Written Packages

**Example: *findit mmerge***

Search of official help files, FAQs, Examples, SJs, and STBsWeb resources from Stata and other users(contacting <http://www.stata.com>)

6 packages found (Stata Journal and STB listed first)

---

dm75 from <http://www.stata.com/stb/stb53>

STB-53 dm75. Safe and easy matched merging / STB insert by Jeroen Weesie,  
Utrecht University, Netherlands / Support: [j.weesie@fss.uu.nl](mailto:j.weesie@fss.uu.nl) / After  
installation, see help mmmerge

---

mmmerge from <http://fmwww.bc.edu/RePEc/bocode/m>

'MMERGE': module: Safer and easier to use variant of merge. / mmmerge is an  
extension of merge that automatically sorts the / master and slave data  
sets, allows selection of variables, and / provides more readable output  
describing the result of a merge. / This version (2.5.0) is an update of

---

spagg from <http://fmwww.bc.edu/RePEc/bocode/s>

'SPAGG': module to create aggregate source or target contagion spatial  
effect variable for directed dyadic data / spagg generates an aggregate  
source or target contagion spatial / effect variable for analysis of  
spatial dependence in directed / dyad data. It can create spatial effect

---

spdир from <http://fmwww.bc.edu/RePEc/bocode/s>

'SPDIR': module to create directed dyad contagion spatial effect variable  
/ spdир generates a directed dyad contagion spatial effect / variable for  
analysis of spatial dependence in directed dyad / data. It can create  
spatial effect variables for spatial lag, / spatial-x and spatial error

---

spspc from <http://fmwww.bc.edu/RePEc/bocode/s>

'SPSPC': module to create specific source or target contagion spatial  
effect variable for directed dyadic data / spspc generates a specific  
source or target contagion spatial / effect variable for analysis of  
spatial dependence in directed / dyad data. It can create spatial effect

---

spundир from <http://fmwww.bc.edu/RePEc/bocode/s>

1/25/ 'SPUNDIR': module to create directed dyad contagion spatial effect  
variable / spundир generates an undirected dyad contagion spatial effect /

**Click on any one of the blue links  
& click on install**

Make sure that the package was delivered into your computer and stata recognizes it: type **help mmerge**

# Other Examples of Packages to be installed and their uses

- Package containing commands:  
**estpost/estout/esttab**
- **findit esttab**
- **estpost sum mpg price**
- **esttab ., cells("mean sd count") noobs**

# Merging Data

- **Master Dataset** – the dataset to which you want to add more variables using another file
- **Using Dataset** – this is the data which will be used to add more data to the master
- You will match the two datasets by some variable that they have in common
- i.e. some variable will be common between the **master dataset** and **using dataset**
- Making the data “wider”

# Example

- Download education.dta and wage.dta from ecommons to your X: drive
- Assume ID variable identifies observations in both data sets.
- Type:
- **cd X:\**
- **use education.dta, clear**
- **describe**
- **summarize**
- **use wage.dta, clear**
- **describe**
- **summarize**

# Process

- **Step 1:** Make sure that the Master Dataset is **open** (i.e. **you did not clear it by accident from STATA memory**). In this case **master dataset** is the wage.dta
- **Step 2:** Make sure they have a variable **in common**, by which you should be conducting the merge. To be on the safe side, the variables that needs to be merged should be sorted in the same way.
- **Step 3:** Type: **mmerge id using education.dta**
- The variable we are merging on is *id* and the **using dataset** is education.dta
- **Step 4:** Look at the resulting dataset
- **Step 5:** Interpreting variable *\_merge*

```
. nmerge id using education
```

merge specs		
matching type	auto	
mv's on match vars	none	
unmatched obs from	both	
master	file	wage.dta
	obs	1350
	vars	2
	match vars	id (key)
using	file	education.dta
	obs	1400
	vars	2
	match vars	id (key)
result	file	wage.dta
	obs	1500
	vars	5 (including _merge)
	_merge	100 obs only in master data (code==1) 150 obs only in using data (code==2) 1250 obs both in master and using data (code==3)

# How did the merge perform?

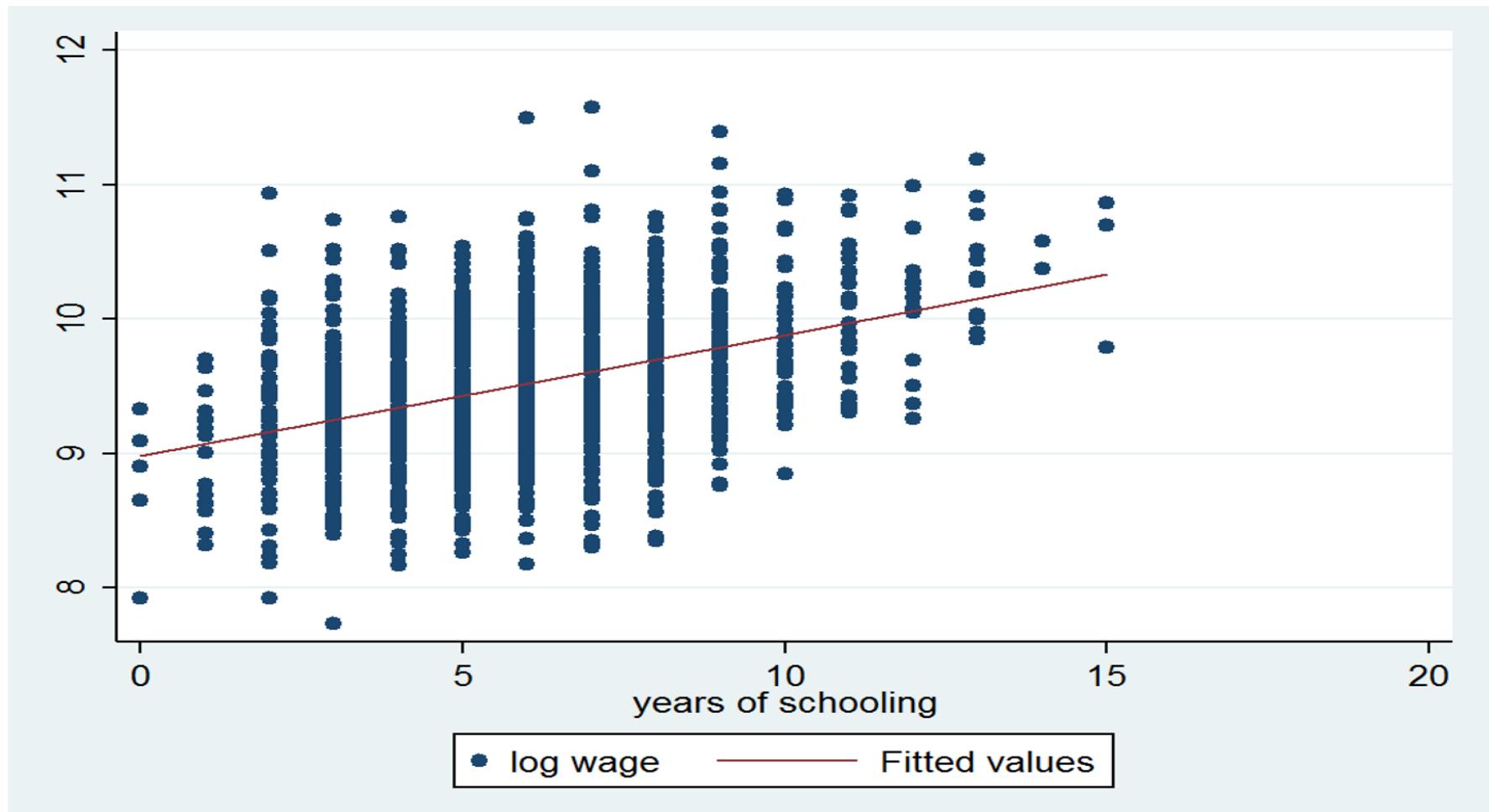
- The resulting data set contains 1500 observations.
- 100 observations only appeared in the master dataset (wage.dta) but not in using (education.dta)
- There are 150 observations for which we only have education data (but no wage data).
- Finally, 1250 observations contain information from both data sets.
- This means that we have full information on education and wages for only 1250 people.
- Note that the mmerge command created a “\_merge” variable which stores the result of the merge.

# How will you use mmerge to add additional files?

female.dta

# Relationship between wage and education

- Type: **twoway (scatter lnwage education) (lfit lnwage education)**



## Easy and safe merging of datasets

### Basic syntax

```
mmmerge match-variable(s) using filename [, {simple | table}  
      umatch(varlist) ukeep(varlist) ]
```

### Full Syntax

```
mmmerge match-variable(s) using filename [,  
      { type(type_value) unmatched(unmatched_value) | simple | table }  
      missing(m_value) nolabel replace update _merge(varname) noshow  
      { ukeep(varlist) | udrop(varlist) } uif(exp) umatch(varlist)  
      { uname(stub) | urename(rename_specs) } xlabel(stub) ]
```

where

type\_value = { auto | 1:1 | 1:n | n:1 | n:n | spread }

unmatched\_value = { both | none | master | using }

missing\_value = { none | value | nomatch }

rename\_specs = oldname newname [\ oldname newname \ ...]

## Description

mmmerge is an extension of merge that makes matched merging **safe**. It requires users to specify the type of match to be performed; mmmerge verifies that the requirements hold. It also makes merging **easy**, though that may not be obvious at

Options for manipulating the using data ("u"-options)

`ukeep(varlist) udrop(varlist)` specifies a varlist in the using data that has to be kept (dropped) before being merged into the master data. It is not valid to specify both `ukeep` and `udrop`. If neither is specified, all variables of the using data are used. The match variable(s) need not be specified in `ukeep`; they are automatically included in `ukeep` (excluded from `udrop`).

`uif(exp)` specifies that only the observations in the using data that meet expression `exp` are to be used. Properness of the key in the using data is determined after `uif` is processed.

`umatch(varlist)` specifies the names of the match variables in the using data. The `umatch` variables are associated with the match variables in the specified order. Clearly, the number of match variables in `umatch` should be the same as the number of matching variables in the master.

`mmerge` renames the `umatch` variables to the master match variable names after `ukeep/udrop` have been processed, but before `urename` is processed. An error occurs if there are naming conflicts.

# Appending Datasets

- The append command combines observations from the using data set with those of the Master data set.
- The append command will make our data set longer.
- If two data sets containing the same variable use a different name for that variable, we will not be able to conduct the append properly.
- Be sure that all variables are named identically across the two data sets. Recall that Stata is case-sensitive.

# Example

- Master data (Western University) and Using Data (Eastern University) – download them to X: from ecommons
- Looking at Summary Stats
- Type:
- **cd X:\**
- **use EasternUniv.dta, clear**
- **describe**
- **summarize**
- **use WesternUniv.dta, clear**
- **describe**
- **summarize**

# Process

- A look at the summary statistics and description of each data set reveals that each contains the same variables, but with different names for the said variable.
- The Eastern University data set contains variables with lowercase names like "gap" and "sat".
- The Western University data set has variable names like "SAT" and "GPA".
- We can use the rename command to change the name of variables in either data set.
- If you have trouble, please type: **help append**

# Codes

- Type
- use "WesternUniv.dta", clear
- ren SAT sat
- ren GPA gpa
- append using EasternUniv
- summarize

# Did the append work? What then? -

## Regression

- Summarize should tell you if you have the desired number of observations
- relationship between GPA and SAT scores  
Graphical Way (twoway scatter)
- But let's try running regressions with Stata using the **regress** command.
- Type
- **regress gpa sat**

# How will you append additional files?

```
. regress gpa sat
```

Source	SS	df	MS	Number of obs	=	1400
Model	1175.91068	1	1175.91068	F( 1, 1398)	=	24954.74
Residual	65.876178	1398	.04712173	Prob > F	=	0.0000
Total	1241.78685	1399	.887624628	R-squared	=	0.9470

	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
gpa						
sat	.0158673	.0001004	157.97	0.000	.0156703	.0160644
_cons	-9.060573	.0704652	-128.58	0.000	-9.198802	-8.922344

I expect that you will learn about the contents of this output in your econometrics coursework. Instead, I will focus our discussion on a method for quickly producing output that can be used in Microsoft Word and Excel.

# Sending Output to word or excel file

- **Outreg2 (my favorite)** and esttab
- The outreg2 command is rather powerful and contains numerous options that are detailed in the lengthy help file provided by Stata. We will cover some of the most important options. A basic application of the outreg2 command is to run it after running a regression. See the output below.

```
. regress gpa sat
```

Source	SS	df	MS	Number of obs	=	1400
Model	1175.91068	1	1175.91068	F( 1, 1398)	=	24954.74
Residual	65.876178	1398	.04712173	Prob > F	=	0.0000
Total	1241.78685	1399	.887624628	R-squared	=	0.9470

	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
sat	.0158673	.0001004	157.97	0.000	.0156703	.0160644
_cons	-9.060573	.0704652	-128.58	0.000	-9.198802	-8.922344

```
. outreg2 using lab2.doc, replace  
c:\ado\plus\o\outreg2.ado  
lab2.doc  
dir : seeout
```

# What do we do now?

- We can click on the “lab2.doc” text that shows up in blue in order to open the .doc file that we just saved. This may not work on a Mac.
- Clicking on the “dir” text will open the directory in which the table was saved.
- Clicking on the “seeout” text will open the table in the Stata data browser.
- Your output should look something like the table below.

VARIABLES	(1)
	gpa
sat	0.0159*** (0.000100)
Constant	-9.061*** (0.0705)
Observations	1,400
R-squared	0.947

Standard errors in parentheses

\* \* \* $p < 0.01$ , \*\* $p < 0.05$ , \* $p < 0.1$

# Adding additional specifications from regressions to the same table

- We may wish to combine the results from several regressions into a single table.
- You can do this by using the “append” option within outreg2.
- Do not confuse the “append” option with the ‘append command that we covered earlier.
- Within outreg2, the “append” option will cause Stata to make tables wider with an additional column for each additional regression.



## Do-file Editor - Lecture 2\*



File Edit View Project Tools



Lecture 2\*

```
111  
112     *** Begin Do-File ***  
113  
114     regress gpa sat  
115     outreg2 using lab2.doc, replace  
116  
117     regress gpa sat if university==1  
118     outreg2 using lab2.doc, append  
119  
120     regress gpa sat if university==2  
121     outreg2 using lab2.doc, append  
122  
123     *** End Do-File ***
```

&lt; &gt;

Line: 124, Col: 0 CAP NUM OVR

VARIABLES	(1) gpa	(2) gpa	(3) gpa
sat	0.0159*** (0.000100)	0.0171*** (0.000115)	0.0145*** (0.000118)
Constant	-9.061*** (0.0705)	-9.826*** (0.0808)	-8.197*** (0.0824)
Observations	1,400	700	700
R-squared	0.947	0.969	0.956

Standard errors in parentheses

\* \* \* $p < 0.01$ , \*\* $p < 0.05$ , \* $p < 0.1$

# Notes

- Using the outreg2 command allows us to swiftly make tables without excessive time spent on formatting.
- Furthermore, the significance levels are clearly presented in the output from outreg2 and the standard errors are accurately displayed below the coefficient estimates.
- Simply copying and pasting the table from the Stata output window into Excel presents a specific formatting challenge.
- Excel reads parentheses as an indication that a number is negative.
- Cells have to be preformatted as “text” in Excel for this problem not to occur.

Important outreg2 options were included in the last table.

- 1. The **label** option replaces variable names with variable labels in the table.
- 2. The **bdec** option determines the number of decimals for the coefficient (b) estimates.
- 3. The **sdec** option determines the number of decimals for the standard errors (s).
- 4. The **ctitle** option specifies the column title.
- 5. The **title** option specifies the main title for the table.

# Example: Improving on the table above

The screenshot shows the Stata Do-file Editor window titled "Do-file Editor - Lecture 2\*". The menu bar includes File, Edit, View, Project, and Tools. The toolbar contains various icons for file operations like Open, Save, Print, and Undo/Redo. The main editor area displays a Stata do-file script:

```
126  
127     *** Begin Do-File ***  
128  
129     regress gpa sat  
130  
131     outreg2 using lab2-table2.doc, ///  
132     word replace label bdec(4) sdec(4) ctitle(Whole Sample) ///  
133     title("GPA and SAT scores, OLS regression")  
134  
135     regress gpa sat if university==1  
136  
137     outreg2 using lab2-table2.doc, ///  
138     word append label bdec(4) sdec(4) ctitle(Eastern University)  
139  
140     regress gpa sat if university==2  
141  
142     outreg2 using lab2-table2.doc, ///  
143     word append label bdec(4) sdec(4) ctitle(Western University)  
144  
145     *** End Do-File ***
```

The status bar at the bottom indicates "Ready", "Line: 146, Col: 0", and keys CAP, NUM, OVR.

GPA and SAT scores, OLS regression			
VARIABLES	(1)	(2)	(3)
	Whole Sample	Eastern University	Western University
SAT scores	0.0159*** (0.0001)	0.0171*** (0.0001)	0.0145*** (0.0001)
Constant	-9.0606*** (0.0705)	-9.8256*** (0.0808)	-8.1974*** (0.0824)
Observations	1,400	700	700
R-squared	0.947	0.969	0.956

Standard errors in parentheses

\* \* \* $p < 0.01$ , \*\* $p < 0.05$ , \* $p < 0.1$

## Saving datasets in Stata 14 so that they can be read by Stata 11, Stata 12, or Stata 13

The `version(#)` option is new to Stata 14, and it specifies which previous `.dta` file format is to be used. `#` may be 13, 12, or 11. The default is `version(13)`, meaning Stata 13 format.

To save a dataset in Stata 14 so that it can be used in Stata 13, use the `saveold` command.

```
. saveold autoold, version(13)  
file autoold.dta saved
```

or

```
. saveold autoold  
file autoold.dta saved
```

To save a dataset in Stata 14 so that it can be used in Stata 12, use the `saveold` command with the `version` option.

```
. saveold autoold, version(12)  
file autoold.dta saved
```

To save a dataset in Stata 14 so that it can be used in Stata 11, use the `saveold` command with the `version` option.

```
. saveold autoold, version(11)  
file autoold.dta saved
```

# Housekeeping

- Section 2 students are turning in Assignment 1 and Assignment 2 today
- **Section 2 students will turn in Assignment 3 on next Thursday Oct 13**
- **Section 1 students will turn in Assignment 2 on Tuesday Oct 11**
- We have learned: How to write .do files, comment, line breaks, save do files, save results using .smcl file; download packages; use outreg2 to transfer results of regressions to word.
- Ugly tables ok with this assignment

# Today

- Best Practices to make your work replicable
- Data Management: **label** – gives info about a variable
- ascribe labels to the individual values in a dummy variable you just created
- Wildcards (to reduce typing) ?, \*
- **recode**: Change numerical variable with uncomfortable values to binary dummy
- **encode**: Change a string variable (non numerical) to numerical/ dummy
- **egen**: extension of “gen” to generate complicated variables e.g. row sum
- Commands by groups of Observations (**by** and **sort**)
- **reshape**
- Submit Assignment 2 and discuss assignment 2 answers

# Replicability

- We want to strive to make our do-files entirely replicable.
- In order for your do-files to be entirely replicable they will need to contain all of the commands that you used to arrive at your results and whatever comments are necessary to communicate what you have done and why.
- This doesn't mean that you need to provide lengthy explanations for why a given regression was run, but an outside, intelligent observer ought to be able to pick up your do-file and arrive at the same results given the same set of inputs.
- These inputs not only include the do-file or do-files that you have written, but also the data sets that were used by your script(s).

# Replicability of Do file

- You can test to see if your do-file is replicable by taking the following steps.
- First close Stata entirely.
- Next reopen Stata and try to run your do-file.
- If the result of these actions is anything other than the expected results that you previously obtained, your do-file is not replicable.
- If you want, you could share do-files with your classmates to see if they work on each other's computer. There are two basic concerns with this second practice.
  - First of all, the file paths are different when using a Macintosh versus Windows PC.
  - You would also need to specify the exact same file paths for your working directory, and ensure that all input data sets are present on both computers.

# More on Replicability

- The second set of concerns revolves around academic integrity.
- While you are encouraged work together and ask questions of your colleagues, any work turned in for credit should be exclusively your own.
- Furthermore, learning to code using any statistical program is best done through doing.
- That is, there is no substitute for the experience gained through attempting and succeeding at writing your own code.

# Best Practices

- Start with “clear all”
- Specify the directory the files are in: use cd
- Do not forget to do: log using filename.smcl & log close

# Do-file Editor - Lab 3\*

File Edit View Project Tools



Lab 3\*

```
14  
15     label variable inc "inc"  
16  
17     label variable inc "income"  
18  
19     label variable inc "household income"  
20  
21     label variable inc "household income, in thousands of dollars per year"  
22  
23     label variable inc "annual household income, |thousands of USD"  
24
```

File Edit View Project Tools



Lab 3\*

```
26  
27     label variable inc "annual household income earned in informal sector \\\\"  
28     and recorded by the Census Bureau as part of the \\\\"  
29     National Informal Income Study, thousands of USD"  
30
```

Ready

Line: 29, Col: 50 CAP NUM OVR

# Label: Information about variables

- \* Label - uses
- **tab foreign**
- **tab foreign, nolabel**
- **gen domestic=1 if foreign==0**
- **replace domestic=0 if foreign==1**
- **tab domestic**
- \* Adding Value to a dummy variable Created
- **gen domestic=1 if foreign==0**
- **replace domestic=0 if foreign==1**
- **tab domestic**
- **label define dom 0 "foreign" 1 "domestic"**
- **label values domestic dom**
- **tab domestic**

# Wildcards – saves typing time

- We can use the question mark “?” To replace exactly one character
- **sum weigh?**
- **gen rep88=rep78 if foreign==1**
- **replace rep88=0 if foreign==0**
- **gen rep79=rep78 if foreign==0**
- **replace rep79=0 if foreign==1**
- **sum rep?8**
- First we generate variables called rep88 and rep79.
- The command “sum rep?8” informs the Stata to look for all variables that start with “rep”, then contain one of any character, and end in 8.
- There are two variables now in the data set that meet these conditions rep 78 and rep 88.
- Make sure you are clear on why the other variable that we generated, “rep79”, fails to meet this criteria.

# Wildcard Asterisk

- We may find ourselves wishing to reference multiple variables that differ by more than one character.
- We can do this using the Asterix wildcard.
- The Asterix can replace any number of characters (including no characters).
- **sum rep\***
- We may be interested in viewing summary statistics for all of the variables in the auto.dta data set that start with the characters “weight”
  
- **sum weight\***
- **sum\*8**
- **sum rep\*78**
- \* What do you expect
- **sum rep\*7?**

# Recode

- **help recode**
- \* Suppose married takes values 1 and 2; you want it to be 0 and 1
- \*recode married 2=0
- \*married = 1, single = 2, widow(er) = 3, unknown = 99
- \*recode married 2=0 3=0 99=.
- \*recode married 2/3=0 99=.

# Encode

- **cd X:\**
- **use household**
- **encode state, gen(scode)**
- **clear**
- **clear all**

# egen

- The egen command can also be used to calculate maximum, minimum, mean absolute deviation, median, mode, rank, standard deviation and many other values. If we wish to perform calculations over the entire row (i.e. the different sources of income earned by each individual) we can use options that begin with “row”. The help file, as always, is very useful for this command.
- **cd X:\**
- **use household**
- **encode state, gen(scode)**
- **egen earnings=rowtotal(income1 income2)**
- **egen earnings\_conservative=rowtotal(income1 income2) if income1!=.& income2!=.**
- **clear**
- **clear all**

# sort, bysort

- We will see, in the next section of this class, that the egen command becomes more useful when we can perform an analysis over subgroups of the dataset. We can do this with the bysort command that we discuss next.
- **cd X:\**
- **use household**
- **gen adult=1 if age>=18 & age!=.**
- **replace adult=0 if age<18**
- **sort hhid**
- **by hhid: egen hhadults=sum(adult)**
- **drop hhadults**
- **bysort hhid: egen hhadults=sum(adult)**

# collapse

- We may have cause to calculate values for groups of observations at a time. As an example, we have data on households. It is reasonable to want to know the number of adults within the household. How can we calculate such a thing?
- **help collapse**
- **collapse (sum) adult (mean) pcincome=earnings, by(hhid)**
- **clear**
- **clear all**

# Exercise 1: mean and Mean Standard Error of earnings by years of schooling

- `cd x:\`
- `use household`
- `egen tot_inc=rsum(income*)`
- `label var tot_inc "total income"`
- `collapse (mean) mean=tot_inc (semean) se=tot_inc,`  
`by(education)`
- `gen low = mean - se`
- `gen high= mean +se`
- `twoway (rcap low high education) (scatter mean`  
`education)`
- `clear`
- `clear all`

# Exercise 2: Mean income (and 95% confidence intervals) by gender

- `cd X:\`
- `use household`
- `egen tot_inc=rsum(income*)`
- `label var tot_inc "total income"`
- `collapse (mean) mean=tot_inc (semean) se=tot_inc, by(female)`
- `gen ci_low = mean - 1.95* se`
- `gen ci_high= mean + 1.95*se`
- `twoway (rcap ci_low ci_high female) (scatter mean female)`

# Reshape : wide to long, long to wide

- help reshape

# Housekeeping

- Section 2 students are turning in Assignment 3 today
- **Section 2 students will turn in Assignment 4 on next Thursday Oct 20**
- **Section 1 students will turn in Assignment 3 on Tuesday Oct 18**
- We have learned: How to write .do files, comment, line breaks, save do files, save results using .smcl file; download packages; use outreg2 to transfer results of regressions to word. At this point, you should be ok using mmerge or merge and outreg2 (even if you are using Mac)

# Today

- Discuss a couple of problems from Assignment 1 (generating dummy variable)
- and Assignment 2 (merging datasets and using commands to create tables with all columns in a table)
- Submit Assignment 3 and discuss assignment 3 answers
- Global Macros
- Local Macros
- Loops: Control Structures – foreach, forvalue and while
- Nested Loops
- Predict
- Exercises

# Global Macro

- Define a global macro
- **sysuse auto.dta, clear**
- **global covariates "mpg headroom"**

\* Stata replaces \$covariates with "mpg headroom"

\* and Stata executes the following line of code: summarize mpg headroom

- Call a global macro
- **summarize \$covariates**
- **regress price \$covariates**

\* Stata replaces \$covariates with "mpg headroom"

\* and Stata executes the following line of code: regress price mpg headroom

- **regress price covariates : should give an error**

# Define and Call a Global Macro

- Important to use “” to define a macro
- When a macro is defined: it's color changes to red – watch for this in do file
- If the color does not change, you have put an incorrect quotation mark.
- Remember, anything can be defined as a macro, not simply variable names.
- Stata will **not evaluate** the contents of the macro until we execute a command in which the macro is specified.
- When a macro is called: the color of the macro changes to teal/greenish blue – watch for this in the do file
- If the color does not change, you have put an incorrect code to call the macro

# Words of Caution

- If we fail to type “**clear all**”, Stata will keep any macros that we have previously defined.
- This may seem convenient at first as we save on keystrokes, but there is a risk involved.
- We need to be certain about the content of our global macros.
- This doesn’t seem like much of an issue now while we’re working with relatively few small data sets.
- However, it can be easy to overwrite one macro with another one by the same name when working on a large project with multiple do-files.

- **clear all**
  - **sysuse auto.dta, clear**
  - **global covariates "mpg headroom"**
  - **global covariates2 "displacement turn"**
- \* We can define a macro as a list of other macros
- \* It is possible to use more than one macro in a single line of code
- **regress price \$covariates \$covariates2**
  - **global covariates3 "\$covariates \$covariates2"**
  - **regress price \$covariates3**
- \* We can save a whole expression as a global:
- **global regression "reg price \$covariates3"**
  - **\$regression**

# Local Macros

- Local macros are defined in a similar manner to their global counterparts.
- We can define a local macro by typing “local”, followed by the macro’s name, and finally the contents we wish to contain within the macro.
- We reference a local macro by preceding the name with the ` symbol and following it with a single quotation mark’
- The opening symbol is located to the left of the one (1) key on a standard American keyboard.

# Forgetfulness of STATA

- If you wish to run a do-file that uses local macros, the local macros must be defined **within the section you are trying to execute.**
- Stata will not work properly if you highlight and execute the lines of code that contain the local macro definitions and their use in later commands separately.
- For example, highlight and run these 4 lines together and then just highlight and run the last line by itself
- **clear all**
- **sysuse auto.dta, clear**
- **local covariates "mpg headroom"**
- **summarize `covariates'**

# How NOT to run local macro

- When we fail to execute the lines of code that define our local macro,
- Stata replaces ‘covariates’ with an empty space.
- This means that the code summarize ‘covariates’ simply operates like “summarize”, which outputs summary statistics for all variables in the data set.
- So you need to run the whole line of code. Stata wont identify the macro.

- **clear all**
- **sysuse auto.dta, clear**
- **local covariates "mpg headroom"**

\* We can display the contents of a local macro by typing the following:  
display `covariates'

- **display "covariates"**

\* Next, use the contents of the local "covariates" in some commands.

- **summarize `covariates'**
- **reg price `covariates'**

\* Again, multiple local macros can be used in a single command

- **local covariates2 "displacement turn"**
- **display ``covariates2"**

\* Locals can be defined within other locals as in the following line of code

- **display ``covariates``covariates2"**
- **regress price `covariates' `covariates2'**
- **local covariates3 `covariates' `covariates2'**
- **regress price `covariates3'**

\* Entire expression can be saved as locals as well.

- **local regression "reg price `covariates3' if foreign==0"**
- **`regression'**

# Local Macro : store predictor variables & conditions

- A common use of locals is to store covariates when we wish to run many regressions on the same set of covariates.
- Conditions may also be stored within local macros.
- See the code below for examples of ways that macros can be used to produce clean and efficient do-files.

**clear all**

**sysuse auto.dta, clear**

**local covariates "mpg length"**

**local if "if foreign==1"**

**regress price weight**

**regress price weight `covariates'**

**regress price weight `covariates' `if'**

**regress headroom weight**

**regress headroom weight `covariates'**

**regress headroom weight `covariates' `if'**

# Control Structures or Loops

- Control structures or loops are very useful in creating efficient code for Stata.
- It should be increasingly clear, now that we are getting into loops, that many ways of completing the same task are almost always available to you.
- We will discuss three main types of loops: **foreach**, **forvalues** and **while**.
- In many cases these various types of loops will be effectively interchangeable.
- Which type you tend to use the most will be a matter of personal preference and familiarity.

# The general structure of loops in Stata

- [control structure name][local macro][list of values or variables][opening bracket] ... <-- Indent for Human Readability-->The lines of code over which Stata will loop ... [closing bracket]
- The indentation of the code within the loop is intended to make it easier for human eyes to decipher what you've done.

# foreach loop

```
clear all
```

```
sysuse auto.dta, clear
```

```
foreach y in price headroom {
```

```
    regress `y' weight
```

```
}
```

```
foreach x in weight mpg foreign {
```

```
    regress price `x'
```

```
}
```

\* No Datasets

```
foreach year in 2011 2012 2013 {
```

```
    gen yr_`year'=1 if Year=="`year'"
```

```
    replace yr_`year'=0 if Year!="`year'"
```

```
}
```

# forvalues loop

- We can use the forvalues type of loop to execute a command for a range of values that we specify.
- In the example below, we generate several variables with random normal distributions and another variable which is a function of these inputs.
- The regression that follows demonstrates that the randomization process worked properly.

\* generate some variables with a random normal distribution

**clear all**

**set obs 500**

**gen x1=rnormal()**

**gen x2=rnormal()**

**gen epsilon = rnormal()**

**gen y = 0.5 + x1 + x2 +  
epsilon**

**reg y x1 x2**

```
drop x*
forvalues i=1(1)100 {
    generate x`i'= rnormal()
}
```

```
* Lets see what the distribution of our variables really looks like
```

```
codebook x1
kdensity x1
```

# Notes

- The codebook command in the kdensity command both offer interesting insight into the distribution of any variable in the data set.
- The codebook command provides information on the variable type, range, number of unique values, mean, standard deviation, the number of missing values, units and several useful percentiles.
- The kdensity command generates an image of a nonparametric estimate of the probability distribution function of a variable.

# while loop example

- \* No Dataset
- reg y x1
- outreg2 using X:\table1, replace
- local i=2
- while `i' <=20 {
- quietly regress y x`i'
- outreg 2 using X:\table1, append
- local 1=`i'+1
- }

# Notes on what the code in the previous slide did

- If we open the table we've created after first running the `outreg2` command and before using the while loop, we will find a simple table with one column of results.
- If we open the table after the while loop is finished running, we should find 20 columns of regression results.
- The inclusion of “quietly” in front of regression in line 164 prevents Stata from printing the regression results in the output window.
- Notice that the resulting table **contains a new row** for each variable because they are not the same covariate across regressions.
- We would like to have a table where each regression result is presented in the same row in order to facilitate comparison.
- We can get around this problem by creating a fake variable that takes on the value of `x1` in the first run and `x2` in the second run.
- See the example below.

# Fixing 20 rows to one row

- \* No Dataset
- reg y random\_covariate
- outreg2 using X:\table1, replace
- local i=2
- while `i' <=20 {
- replace random\_covariate=x`i'
- quietly regress y random\_covariate
- outreg 2 using X:\table1, append
- local 1=`i'+1
- }

# What is wrong with this file

```
local i = 1
while `i'<= 100 {
    reg y x`i'
}
```

# Nested Loops

- A cool and useful feature of loops is that they can be nested within other loops.
- You may want to iterate a given command over two sets of changing inputs.
- Note the use of indentation in the lines of code the follow.
- Consistent indentation practices become even more useful when several loops are nested within each other.
- Before you execute the code that follows, write down each regression that will be run in the correct order.

# Nested Loops Example

- **clear all**
- **sysuse auto.dta, clear**
- **foreach y in price headroom {**
- **foreach x in weight mpg foreign {**
- **regress `y' `x'**
- **}**
- **}**

# Predict

- Now we're going to use the predict command to generate predicted values of an outcome variable that result from running a regression.
- See the help file for predict if you need to.
- Suppose we're interested in seeing if our model has good out of sample prediction power.
- That is, that it can predict reasonably well the observed values of observations not found in the sample.
- Conduct the following steps:

# Steps for predict

- 1. First estimate the model using every observation except for the first one.
- 2. Predict the outcome variable in the first observation using your model.
- 3. Store the value you have predicted for this first observation.
- 4. Continue with the next observation in the data set.
- Once your loop is finished running, use the commands you've learned so far to compare the predicted values of the data with those that are observed.

# Exercises: sysuse nlsw88.dta, clear

- 1. Regress wage on hours by marital status using a **foreach** loop. Use the outreg2 command to save your results to an excel file.
- 2. Regress wage on hours for each occupation in the sample using a **while** loop. Once again, use the outreg2 command to save your results to an Excel file. Be sure to update the counter at the end of your loop.
- 3. Use a **nested loop** to regress wage on hours for each occupation and by marital status. Save these results to an Excel file as well.

```
1 sysuse auto.dta, clear
2
3 gen id = _n // define an id variable with value equal to the observation number
4 gen yhat = . // generate a variable to store your out-of-sample estimate
5
6 local i = 1 // define the local macro i
7 while `i' <= 74 { // Stata will loop over the commands as long as this is true
8     reg price weight length if id!=`i' // run regression for all but i
9     predict yhat_0 if id==`i' // predict the value of price using reg in line
0     replace yhat = yhat_0 if id==`i' // insert prediction for i only
1     drop yhat_0 // drop the estimate you have generated in this part of loop
2     local i = `i' + 1 // set counter +1 before loop repeats - DO NOT FORGET
3 }
```

# Housekeeping

- Section 2 students are turning in Assignment 4 today
- **Section 2 students will turn in Assignment 5 on next Thursday Oct 27**
- **Section 1 students will turn in Assignment 4 on Tuesday Oct 25**

# Today

- Scalars and matrices in stata: return list, ereturn list
- Hypothesis Testing
- Useful application of outreg2 to show contents of return list and ereturn list on .doc files
- Few more important insights into data handling

# Return List and Ereturn List

- So far, we have programmed Stata through the command window and do-files and simply displayed output in the output window.
- We can interact with the output from various Stata commands in another way.
- Some values are stored in memory as **scalars** and **matrices** *after we run certain commands* (e.g. regress).
- We can use the **return list** command after other non-estimation commands to see these values.
- The command **ereturn list** can be used after estimation commands.

```
sysuse auto.dta  
  
summarize mpg  
  
ret list  
  
display r(Var)  
  
display r(kurtosis)
```

Now lets run the command again with different options. Type “d” after a comma to add the details option to the summarize command. *return list* will now show us additional values that can be referenced.

```
summarize mpg, d  
ret list  
. display r(Var)  
display r(kurtosis)
```

- The values displayed are stored in memory until you write another command.
- For instance, summarize a different variable and display the same set of scalars.
- **summarize weight**
- **ret list**
- **display r(Var)**
- **display r(Kurtosis)**
- We may need to use these values at a later stage in our work, perhaps in a loop.
- We can save these values for later use by storing them either as scalars or macros.
- Once you've stored a value as a scalar it will remain in memory until you drop it (**clear all** would accomplish this task).
- **sum mpg**
- **scalar mean\_mpg =r(mean)**
- **sum weight**
- **scalar mean\_weight =r(mean)**

- The ereturn list command works much in the same manner as return list, only in conjunction with estimation commands.
- We refer to the results of ereturn list using the form “e(name)”.
- **regress mpg weight displacement**
- **ereturn list**
- Try calling a few different scalars and matrices.
- **display e(F)**
- **matrix list e(b)**
- **matrix list e(V)**

# Matrices in Stata

- We will not cover working with matrix algebra in Stata in this course.
- Some of the useful results produced by Stata, however, are stored in matrices.
- You may wish to access these results.
- It is generally easier to work with a copy of the matrix than to work directly with it using the e(b) or e(V) syntax.
- The code that follows shows one method for creating a copy of the post estimation matrix, e( ), and listing its contents.

```
. matrix B = e(b)  
  
. matrix list B  
  
. matrix V = e(V)  
  
. matrix list V
```

You can refer to a specific element in the matrix by including square brackets containing the row and column (in that order) numbers separated with a comma. The brackets are placed immediately following the name of your matrix.

```
. * Display the element in row 2, column 1 of the variance matrix  
  
. display V[2,1]  
  
. * There is no 4th row in matrix V  
  
. display V[4,1]
```

We can use a loop to turn each element of a matrix into a scalar.

```
. forvalues i = 1/3 {  
    . display B[1,`i']  
    . scalar B_`i' = B[1,`i']  
}  
. scalar list
```

# Application: Hypothesis Testing

- You can use the stored results generated by running a regression to conduct hypothesis testing.
- The **lincom** command provides a simple way of testing hypotheses regarding your model coefficients.
- **sysuse nlsw88.dta, clear**
- **reg wage hours tenure ttl\_exp age**
- Suppose we have the null hypothesis that the coefficient on age is statistically equivalent to zero.
- If the P-value associated with this hypothesis is less than 0.05, we will reject this hypothesis at the 95th percentile confidence level.
- **lincom hours**
- **lincom tenure**

# Hypothesis Testing on Combination of Coefficients

- We can also use the **lincom** command to test linear combinations of coefficients.
- If we are interested in testing whether the coefficient associated with hours is equal to the coefficient associated with tenure we can do so with the **lincom** command.
- We can test any linear combination of coefficients, however, the test will always be whether or not the expression is equal to zero.
- Instead of telling Stata to test whether the two coefficients are equal to each other, we will conduct a test of whether the difference between the two coefficients is equal to zero.
- **lincom hours – tenure**
- The coefficient on hours is significantly different than zero and the coefficient on tenure is not significantly different from zero.
- Curiously, the two coefficients are also not statistically different from each other.

# Non Linear Combinations

- We can also test whether the elasticity of wage to schooling is equal to one.
- The only restriction using the income command is that the combinations of coefficients must be linear.
- The **nlcom** command allows testing hypotheses that involve nonlinear combinations of estimators.

```
. clear all

. sysuse nlsw88.dta
(NLSW, 1988 extract)

. g lnschooling = ln(grade)
(4 missing values generated)

. g lnwage = ln( wage )

. reg lnwage lnschooling hours tenure ttl_exp age

. lincom lnschooling - 1
. lincom 2 * hours - (1/3) * tenure
. lincom 2 * hours - (1/3) * tenure^2
not possible with test
r(131);
```

# Adding Scalars to Tables

- The last point of discussion today involves the use of return and ereturn values in regression tables using outreg2.

```
clear all

sysuse nlsw88.dta

reg wage hours tenure ttl_exp age

outreg2 using table1, replace ///
    adds("F-stat", e(F))
```

```
reg wage hours tenure ttl_exp age
    summarize wage if e(sample)
    estadd scalar mn_wage = r(mean)
outreg2 using table1, append ///
    adds("F-stat", e(F), "Mean Wage", e(mn_wage))
```

```
reg wage hours tenure ttl_exp age

    summarize wage if e(sample)
    estadd scalar mn_wage = r(mean)

    lincom hours - tenure

    estadd scalar diff = r(estimate)
    estadd scalar se_diff = r(se)
    estadd scalar t_diff = r(estimate)/r(se)

outreg2 using table1, append ///
    adds("F-stat", e(F), "Mean Wage", e(mn_wage), ///
    "H0: hours = tenure", e(diff), ///
    "Difference SE", e(se_diff), "t-stat", e(t_diff))
```

The resulting table should look like this.

VARIABLES	(1)	(2)	(3)
	wage	wage	wage
hours	0.0546*** (0.0115)	0.0546*** (0.0115)	0.0546*** (0.0115)
tenure	0.0367 (0.0260)	0.0367 (0.0260)	0.0367 (0.0260)
ttl_exp	0.285*** (0.0317)	0.285*** (0.0317)	0.285*** (0.0317)
age	-0.121*** (0.0386)	-0.121*** (0.0386)	-0.121*** (0.0386)
Constant	6.714*** (1.572)	6.714*** (1.572)	6.714*** (1.572)
Observations	2,227	2,227	2,227
R-squared	0.084	0.084	0.084
F-stat	50.94	50.94	50.94
Mean Wage		7.800	7.800
H0: hours = tenure			0.0179
Difference SE			0.0288
t-stat			0.622

Standard errors in parentheses

\*\*\* p<0.01, \*\* p<0.05, \* p<0.1

robust: option after regress  
listcoef: option after regress  
(user written command)

Opening Data in STATA from other  
formats: best .csv

# Housekeeping

- Section 2 students are turning in Assignment 5 today
- **Section 2 students will turn in Assignment 6 on next Thursday Nov 3**
- **Section 1 students will turn in Assignment 5 on Tuesday Nov 1**

# Today

Non Parametric Techniques

Histogram

Kernel Densities

Kernel Densities – with different kernels and bandwidths

Finding patterns in data using local linear regressions: using **lpoly lpolyci**

# Non Parametric Analysis

- We will focus on the most basic tools for non-parametric estimation using Stata.
- Relationship between Two Variables: can be extended to multiple variables (not done in this class)
- Useful discussions of the subject matter can be found in Chapter 9 of A. Colin Cameron and Pravin K. Trivedi's book: Microeconometrics, methods and applications.
- Non-parametric econometrics by Pagan and Ullah is an accessible book on the topic.
- A first year PhD level treatment can be found in the textbook Non-parametric econometrics: theory and practice.

# Benefits of Non Parametric Analysis

- Parametric methods of analysis require us to make assumptions about the **distribution of the error term** (for instance, normality) or about the **shape of the relationship** between variables under analysis (e.g. **linearity**).
- Nonparametric methods of analysis have the advantage of not requiring us to make such assumptions.
- This is especially important, when miss specifications of distributional assumptions about errors can lead to inconsistent parameter estimates.

# Data Generating Process (DGP): y variable

It comes from some probability  
distribution

# Histograms

- Histograms and densities are some of the most basic examples of non-parametric methods that are commonly used to analyze data.
- We can plot bivariate relationships (with continuous random variables) with these methods.
- As the output of these nonparametric methods is often graphical it is important to format our graphs clearly and neatly.
- As a result, we will focus on some of the graphing options available in Stata: choosing line width, colors, pattern, etc.

# How to construct a histogram

- In order to construct a histogram, we will split the data into intervals.
- These intervals are referred to as **bins** and they will cover the full range taken by the variable.
- Next we count the number of observations that occur within each bin.
- Finally, we draw a series of rectangles
  - with **base equal to the width of the bin** and
  - **height determined by the number of observations**
  - within each bin.

- clear all
- set seed 2015
- set obs 200
- **gen x=4\*uniform()+7**
- **gen y=3+ sin(x) + 0.32\*rnormal()**
- \* Change Label of variables
- **la var x "wage"**
- **la var y "hours per day"**
- \* Figure 1: Default Bin Size : help histogram
- **hist y, graphregion(color(white)) xtitle(hours per day)**
- **graph export "hist1.png", replace**
- \* Figure 2: Problem of having SMALL bins
- **hist y, bin(30) graphregion(color(white)) xtitle(hours per day)**
- **graph export "hist2.png", replace**
- \* Figure 3: Problem of having LARGE bins
- **hist y, bin(5) graphregion(color(white)) xtitle(hours per day)**
- **graph export "hist3.png", replace**

# Kernel Density Estimation

- We can estimate the probability density function of a random variable using kernel density estimation.
- Nonparametric density estimates have the advantage of producing smoother density estimates than one would obtain using a histogram.
- Kernel density estimates can be used to make inferences regarding the population distribution of a variable of interest.

# First Three Kernel Graphs

- We can use the `twoway` command to superimpose the Kernel density estimate on the histogram.
- The `normal` option will allow you to plot the normal density alongside your kernel density estimate.
- This will allow you to see how close to normal the distribution of your variable is

# Next Kernel Graph : Choice of Kernel – (weights)

- We can use different kernel functions to generate our kernel density estimate graphs. Your choice of kernel should have little impact on the density. The following graph shows the density estimate that results from using three different kernels: Epanechnikov, rectangle, and Gaussian (normal). Note how close the different kernel estimates are to each other.

# Bandwidth Graph: Choice of Bandwidths –(smoothing functions)

- Unlike your choice of kernel function, your choice of bandwidth is important to the shape of your density estimate.
- The topic of optimal bandwidth choice is advanced and beyond the scope of this class. It is perfectly acceptable to use Stata's default optimal bandwidth.
- This bandwidth is chosen by cross-validation.

# Problems of small or large bandwidth

- We can designate a set bandwidth as an option in the `kdensity` command.
- An under-smoothed kernel will result from a choice of a narrow or less than optimal bandwidth.
- Such a density estimate will have a jagged surface that jumps around.
- Some of these jumps found in your sample will also be found in the population, but some will not.
- It is difficult to know which ones are which.
- A choice of a very wide bandwidth will lead to an over smoothed kernel density estimate which will miss many of facets of the distribution.
- We can minimize these problems by relying on the optimal bandwidth provided by Stata.

- \* KERNEL Density Estimation: help kdensity
- \* Figure 4: Kernel Density
- **kdensity y, k(epan) bw(0.21) lw(thick)**
- **graph export "kernel\_1.png", replace**
- \* Figure 5: Kernel Density Estimate with Histogram
- **twoway (hist y, graphregion(color(white))) || (kdensity y, graphregion(color(white))lwidth(thick))**
- **graph export "kernet\_hist.png", replace**
- \*Figure 6: Kernel Density and Normal Distribution
- **kdensity y, normal graphregion(color(white)) xtitle(hours per day)**
- **graph export "kernel\_normal.png", replace**
- \* Figure 7: Kernel Density With Different Kernels
- **twoway (kdensity y, epan legend(label (1 Epanechnikov)) lcolor(red)lw(thick))|| (kdensity y, gaussian legend(label (3 Gaussian)) lcolor(dknavy)lw(thick))|| (kdensity y, rectangle legend(label (2 Rectangle)) lpattern(dash)lw(thick)), graphregion(color(white))**
- **graph export "kernel\_comparison.png", replace**
- \* Figure 8: Kernel Density With Different Bandwidths
- **twoway (kdensity y, bw(0.21) legend(label (1 "bandwidth=0.21")))|| (kdensity y, bw(0.80) legend(label (2 "bandwidth=0.80")))|| (kdensity y, bw(0.10) legend(label (3 "bandwidth=0.10"))), graphregion(color(white)) xtitle(hourly wage)**
- **graph export "bandwidthcomp.png", replace**

# Non Parametric Regressions

- Suppose we have data on the labor supply in hours per day and wages in US dollars per hour.
- We are interested in examining the relation between two variables.
- A simple regression of Y on X leads to the results shown in column one of table 1.
- The coefficient on X is positive and statistically significant at the 99% level of confidence.
- On the surface, this regression output looks very good.

VARIABLES	(1)	(2)
	y	y
wage	0.220*** (0.0287)	0.763*** (0.0257)
high wage		12.89*** (0.539)
wage × high wage		-1.616*** (0.0624)
Constant	0.567*** (0.217)	-2.821*** (0.167)
Observations	200	200
R-squared	0.230	0.856

Standard errors in parentheses

\*\*\* p<0.01, \*\* p<0.05, \* p<0.1

# Trouble in Paradise

- If we look at the next graph, which contains the linear line of best fit that is generated by OLS and a scatter plot of the data, we can see that our model fits the data poorly.
- We can see that there is an inverse “U” relationship between the variables.
- It appears that the slope changes somewhere around a wage equal to eight dollars an hour.
- We could include an interaction term that allows for a different slope when wage exceeds that level.
- Figure 10 displays the results of such a regression graphically.
- Of course, we don’t need to stop at a single structural break.
- Four structural breaks are implemented in Figure11.

- \* Non Parametric Regressions
- \* Figure 9
- **twoway (scatter y x, mcolor(gray) mszie(small)) || (lfit y x, lcolor(black) lwidth(thick)), legend (off) graphregion(color(white))**
- **graph export "wls.png", replace**
- \* Figure 10
- **twoway (scatter y x, mcolor(gray) mszie(small)) || (lfit y x if x<8, lcolor(black) lwidth(thick))|| (lfit y x if x>8, lcolor(black) lwidth(thick)), legend (off) graphregion(color(white))**
- **graph export "wlshighlow.png", replace**
- \* Figure 11
- **twoway (scatter y x, mcolor(gray) mszie(small)) || (lfit y x if x>=7 & x<8, lcolor(black) lwidth(thick)) || (lfit y x if x>=8 & x<9, lcolor(black) lwidth(thick))|| (lfit y x if x>=9 & x<10, lcolor(black) lwidth(thick)) || (lfit y x if x>=10 & x<11, lcolor(black) lwidth(thick))|| (lfit y x if x>=11 & x<12, lcolor(black) lwidth(thick)) , legend (off) graphregion(color(white))**
- **graph export "wlshighlow.png", replace**
- \*\*\*\*

# What if we do not know the break points?

- The procedures we've looked at so far work well when we're absolutely sure of the location of the break points.
- Often, we aren't going to be sure of these locations.
- Instead, we should use the **local linear regressions**.
- Local linear regressions run linear regressions within a given bandwidth of each x value.
- For instance, to examine estimate the slope at  $x = 7$ , local linear regression takes all the data with  $x$  between 6.5 and 7.5 and estimates the slope at that point.
- Then it moves to 7.1, takes all the points between 6.6 and 7.6 and estimates a new slope.
- As the slope estimates contain many of the same input values the estimates will change slowly, rendering a curve that appears continuous.
- We can use the local polynomials (**Ipoly**) to include polynomials in  $x$  to further improve the estimation.
- Figure 12 shows the results of this command.

- \* Local Linear regressions
- **help lpoly**
- \* Figure 12
- **twoway (scatter y x, mcolor(gray) msize(small))**  
**| | | | (lpoly y x, lcolor(black) lwidth(thick)),**  
**legend (off) graphregion(color(white))**
- **graph export "wlshighlowlpoly.png", replace**
- \*\*\*\*\*
- \*\*\*\*\*

# Alternative to Non Parametric Regression Methods

- The commands `fppfit` and `lowess` also provide easy ways of estimating bivariate relations non-parametrically.
- The options are similar to those of `poly`.
- You should familiarize yourself with the options in case you are asked about them on a homework assignment or the final exam.
- The differences between the non-parametric estimating methods is most pronounced at the tails of the distribution, where the fewest data points are used in the estimation.
- Once again, the choice of bandwidth is more important than your method of estimation.

# Confidence Intervals

- In order to get the full story from a regression table, we need not only the coefficient, but the standard errors as well.
- We can facilitate inference regarding our non-parametric estimates by incorporating confidence intervals into our graphics.
- We can generate confidence band simply using the CI versions of lpoly and fpfit.

- \* Alternative Non Parametric Regression Command
- \* Figure 13
- **twoway (lpoly y x, lcolor(gray) lwidth(thick)  
lpattern(dash)) || (lfit y x if x>=7 & x<8, lcolor(black)  
lwidth(thick)) || (lfit y x if x>=8 & x<9, lcolor(black)  
lwidth(thick)) || (lfit y x if x>=9 & x<10, lcolor(black)  
lwidth(thick)) || (lfit y x if x>=10 & x<11, lcolor(black)  
lwidth(thick)) || (lfit y x if x>=11 & x<12, lcolor(black)  
lwidth(thick)) || (lfit y x, lcolor(red)), legend (off)  
graphregion(color(white))**
- **graph export "altpoly.png", replace**
- \* Figure 14
- **twoway (lpoly y x, legend(label(1 "lpoly")))) || (lowess y x,  
legend(label(1 "lowess")))) || (fpfit y x, legend(label(1  
"fracpoly"))), graphregion(color(white))**
- **graph export "lowesslpfit.png", replace**

The above command made a quick plot with the confidence interval. However, some challenges may arise when we want to change some of the options with this command. For example, when we change the line color to black the graph that results may not look like what we expect.

# One way around

- is to first generate variables that contain the values of Y and X in the graph, together with the standard errors.
- Then, we'll take the critical value of the test statistic and construct the upper and lower bounds of the confidence interval.
- Finally we will graph these additional variables as lines.

- \* Alternative Non Parametric Regression Command
- **twoway (lpolyci y x), graphregion(color(white))**
- **graph export "lpolyci1.png", replace**
- **twoway (lpolyci y x, lcolor(black)), graphregion(color(white))**
- **graph export "lpolyci2.png", replace**
- **lpoly y x, gen(xhat yhat) se(sehat) noscatter**
- **\*upperbound control:**
- **gen ub = yhat +1.96\*sehat**
- **\*lowerbound control:**
- **gen lb = yhat - 1.96\*sehat**
- **twoway (line yhat xhat, lcolor(dknavy) lwidth(thick)) || (line ub xhat, lcolor(dknavy) lpattern(dash)) || (line lb xhat, lcolor(dknavy) lpattern(dash)), ytitle("Y Axis Title") xtitle("X Axis Title") legend(off) graphregion(color(white))**
- **graph export "lpolyublb.png", replace**

robust: option after regress  
listcoef: option after regress  
(user written command)

Opening Data in STATA from other  
formats: best .csv

# Housekeeping

- Section 2 students are turning in Assignment 6 today
- **Section 2 students will turn in Assignment 7 on next Thursday Nov 10**
- **Section 1 students will turn in Assignment 6 on Tuesday Nov 8**

# Today

## Finding Marginal Effects

- The marginal effect of a regressor is the change in the conditional mean of your dependent variable when the regressor changes by one unit.
- Today we will discuss several ways of estimating marginal effects using Stata.
  - using OLS
  - Using Probit
  - Using Logit

# use nlsw88.dta

- Suppose we use OLS to estimate the equation specified in Equation (1).
- $Y = \text{wage}$ ,  $X_1 = \text{grade(educ)}$   $X_2 = \text{tenure(exp)}$

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 \quad (1)$$

The marginal effect of grade on wage is given by Equation (2).

$$\frac{\partial Y}{\partial X_1} = \beta_1 \quad (2)$$

# Code

- \* Marginal Effects in OLS: Use Interaction Term
- `clear all`
- `sysuse nlsw88.dta,clear`
- \* Marginal Effects from Single Equation
- `reg wage grade tenure`

When the regression model is linear,  
the coefficient is directly  
interpretable as the marginal effect.

In the case above, the marginal  
effect is the constant,  $\beta_1$ .

The regression results reported below  
indicate that the marginal effect of an  
additional year of schooling averages to  
\$0.70.

- Suppose we are interested in evaluating whether the effect of schooling on wages is different for unionized and nonunionized workers.
- The simplest approach to this question involves estimating linear regressions for each group separately.
- We accomplished this task by simply including the if condition shown in the state output below.

# Code

- \* Marginal Effects of Two Groups
- `reg wage grade tenure if union==1`
- `reg wage grade tenure if union==0`

- Our regression results show that the marginal effect of an additional year of schooling on hourly wages is \$0.52 for union workers and \$0.71 for nonunion workers.
- These point estimates are different but we don't know if this difference is statistically significant.
- Simply observing that their confidence intervals do not overlap is insufficient and may be misleading if an underlying correlation between both coefficients exists.

One method for accounting for correlation between coefficients involves the estimation of a single regression that includes interactions of each explanatory variable with our union variable.

Equation  
(3) presents the regression specification to be estimated.

$$Y = \beta_0 + \beta_1 \cdot Grade + \beta_2 \cdot Tenure + \beta_3 \cdot Union + \beta_3 \cdot Union \times Grade + \beta_4 \cdot Union \times Tenure \quad (3)$$

We can find the marginal effect of schooling by taking derivatives as in Equation (4).

$$\frac{\partial Wage}{\partial Grade} = \beta_1 + \beta_3 \cdot Union$$

- Our union dummy variable takes only two values: zero and one.
- The marginal effect of Grade=  $\beta_1 + \beta_3 \cdot 0$  for **nonunionized workers**
- The marginal effect of Grade=  $\beta_1 + \beta_3 \cdot 1$  for **unionized workers**.
- We can estimate the marginal effects using Stata by first creating the interactions explicitly or by using the **xi** prefix.

# Code

- \* Marginal Effects of Two Groups using interaction term
- `gen tenure_union=tenure*union`
- `gen grade_union=grade*union`
- `reg wage grade grade_union tenure  
tenure_union union`

# xi: prefix

- We can also use the “xi” prefix to include interactions in our regression specifications.
- Now, suppose that were interested in how wages vary with union affiliation and race.
- As race is a categorical variable we will need to include an indicator variable for each race in addition to the union dummy variable.

# Code

- `xi: reg wage i.race i.union`
- If you have like 50 categories for each dummy variable, instead of having to code a dummy for each category, `i.(varname)` automatically codes it for you
- Consider states; there are 50. So 50 dummies.

We can interact union affiliation with race by placing the “\*” symbol between the two “i.regressor” terms.

Note which categories are designated as reference categories and thus dropped from the results

# Things to notice

- The regression output includes an intercept, indicators for each race other than the reference
- We can also include interactions between categorical and continuous variables.
- In the regression above, union affiliation is interacted with the continuous tenure variable.
- If we use the vertical line or pipe character, “|”, in place of the Asterix, “\*”, we can accomplish the same set of interactions **without the separate union indicator**.
- This means that **union and nonunion** workers are **restricted to the same intercept** which is denoted “cons”.

# Examples of xi: Code

- **xi: reg wage i.race\*i.union**
- **xi: reg wage i.union\*tenure**
- **xi: reg wage i.union | tenure**

- In the regression above, union affiliation is interacted with the continuous tenure variable.
- If we use the vertical line or pipe character, “|”, in place of the Asterix, “\*”, we can accomplish the same set of interactions without the separate union indicator. This means that union and nonunion workers are restricted to the same intercept which is denoted “cons”.

- In the following example, I use the pipe character, “|”, for interactions and add the union dummy in explicitly.
- It is important that we specify the regression model properly. In our simple example, we are interested in how union affiliation impacts the marginal effect of another year of schooling on wages.
- In order to properly specify the model we interact **each explanatory variable with the group variable (union)** as well as **include an indicator for our group variable**.
- The inclusion of the full set of interactions is important as it allows both groups to exhibit different intercepts and slopes.

# Code

- **xi: reg wage union i.union|grade  
i.union|tenure**

- Note that the coefficient on grade is 0.71 and the coefficient on the interaction term is -0.19.
- When we add these two coefficients together we obtain 0.52,
- which is the marginal effect estimated for union workers in our original analysis.
- Our estimates of the marginal effects are numerically identical with either method of analysis.

```
/* THE FOLLOWING CODE WILL PRODUCE THE SAME RESULTS */
```

```
xi: reg wage union i.union|grade i.union|tenure
```

```
xi: reg wage i.union*grade i.union|tenure
```

```
xi: reg wage i.union|grade i.union*tenure
```

```
xi: reg wage i.union*grade i.union*tenure
```

# Discrete Choice Models: Probit/Logit

- Understanding how individuals make choices is a fundamental interest in the field of economics.
- We are generally unable to observe the utility one gains from making various choices such as joining a union.
- Under certain circumstances it may be possible to observe the choices that people make and infer something about their preferences.
- If we observe that someone is part of a union we may be able to infer that for them the utility derived from being in a union **is greater** than the utility derived from not belonging to a union.
- Both estimate a CDF. Probit estimates normal, logit estimates exponential.

- If I choose to purchase a bike, I am revealing that the utility I expect to gain from the bike meets or exceeds the utility of the resources I expend on its acquisition.
- The bicycle company might be interested in determining the marginal effects of their marketing activities on the probability that someone buys a bike
- We can use a **bivariate choice model like probit or logit** to estimate the probability of a given event.
- Our outcome of interest in such a situation is binary.
- The thing we are interested in is the probability of obtaining an outcome equal to one.

- In our example, we will be interested in the probability that a worker belongs to a union.
- That is, the probability that the union indicator variable obtains a value of one.
- Let the probability of the outcome being equal to one be denoted  $p$ .
- We will form a regression model to estimate  $p$  as a function of a vector of regressors,  $x$ , and a  $K \times 1$  parameter vector,  $\beta$ .
- See Cameron and Trivedi (2005) p. 466 for a more rigorous and detailed treatment of the subject.
- For our purposes, it is useful to denote the conditional probability as follows:

$$p_i \equiv Pr[y_i = 1|x] = F(x'_i \beta) \quad (5)$$

Let the function  $F(\cdot)$  be specified as needed. If we specify  $F(\cdot)$  as a cumulative distribution function we ensure that  $0 \leq p_i \leq 1$ . The logit model uses the cdf of the logistic distribution as  $F(\cdot)$ .  $F(\cdot)$  is specified as the standard normal cdf in probit models.

Logit models assume the probability is given by the equation (6).

$$Pr(y = 1) = \frac{e^{x\beta}}{1 + e^{x\beta}} = \Lambda(x\beta) \quad (6)$$

$$Pr(y = 1) = \frac{1}{1 + e^{-x\beta}} \quad (7)$$

where  $x\beta = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \cdots + \beta_N \cdot x_N$ . In this case, the marginal effect of  $x_k$  is given by:

$$\begin{aligned} \frac{\partial Pr(y = 1)}{\partial x_k} &= \frac{e^{x\beta}}{1 + e^{x\beta}} \frac{\partial(x\beta)}{\partial x_k} \\ &= \frac{e^{x\beta}}{(1 + e^{x\beta})^2} \beta_k \\ &= \Lambda(x\beta)(1 - \Lambda(x\beta))\beta_k \\ &= Pr(y = 1) \times Pr(y = 0) \times \beta_k \end{aligned}$$

# Interesting Observation about marginal Effects

- The derivation of the marginal effect is **beyond the scope of this lab**.
- What we should take away from the equations above is that the **marginal effect depends on the value of each of the covariates (x variables)**.
- As an example, **the marginal effect of an additional year of schooling would be different for a union worker with two years of tenure and an un-unionized worker with 10 years of tenure**.
- We can estimate a logit regression using the logit command in Stata.
- The syntax of the logit command is very similar to that of the regress command. See the Stata output below

# Codes

- \* Logit Models
- logit union wage age
- predict phat if e(sample), pr
- gen dydwage=phat\*(1-phat)\*\_b[wage]
- gen dydage=phat\*(1-phat)\*\_b[age]
- sum dydwage dydage
- Gives the average marginal effect. If age increases by one unit, the probability of becoming a union member increases on average by (X)

As mentioned earlier, probit models use the standard normal cdf to specify  $F(\cdot)$ .

$$p = Pr[y = 1|x] = \Phi(x'\beta) = \int_{-\infty}^{x'\beta} \phi(z)dz$$

$$Pr[y = 1|x] = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} dx$$

The marginal effect of a continuous variable  $x_k$  is given by:

$$\frac{\partial p}{\partial x_k} = \phi(x\beta)\beta_k \quad (8)$$

where  $\phi(x\beta)$  is the standard normal density evaluated at  $x\beta$  (so that  $\phi(x\beta) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x\beta^2}{2}}$ ). We can use the *normalden* command in Stata to calculate the value of the normal density at a given value of  $xb$ .

# Codes

- \* Probit Models
- probit union wage age
- predict phat\_probit if e(sample),xb
- // Creates phi of the function
- gen index=normalden(phat\_probit)
- gen dydwage\_probit = index\*\_b[wage]
- gen dydage\_probit = index\*\_b[age]
- sum dydwage\_probit dydage\_probit

- The marginal effects estimates differ between the logit and probit models.
- Generally, there will be little difference between the models.
- If we wish to compare models, we should do so using their marginal effects rather than the estimated coefficients.
- The primary caveat to consider is that we do not have a simple method for evaluating whether these marginal effects are significantly different from zero.
- Methods exist, but are beyond the scope of this Stata lab.

# mfx command

- We can quickly estimate **marginal effects evaluated at the mean values of all covariates** using the **mfx** command.
- You will notice that the marginal effects arrived at using the mfx command are very similar to those we calculated manually earlier.
- They are, however, not the same.
- **The average marginal effect** and **the marginal effect evaluated at the average values of the covariates** are not the same.

# Codes

- \* **mfx**
- **probit union wage age**
- **mfx**
- **mfx, at(5,35)**

- The last column of the output, with the title X, reports the values taken by wage and age. In other words, the marginal effects reported in this table are valid for 39-year-olds with an hourly wage of \$7.57.
- **The marginal effects will differ for people with different values of age and wage.**
- We can calculate the marginal effects for someone with any given age and wage using the at option.
- The order of the values that we list using the at option will correspond with the order of covariates in your regression.

# margins command

- The margins command estimates the marginal effects for each observation and then reports the average of these marginal effects.
- Notice that the marginal effects generated using the margins command are identical to those that we calculated earlier by hand.
- Margins is helpful in that it also provides standard errors in confidence intervals for our marginal effects.
- This facilitates testing whether or not the marginal effects are significantly different from zero.
- **We use the margins command to see if the average marginal effects are statistically significant**

# Codes

- \* **Margins**
- **logit union wage age**
- **margins, dydx(wage age)**
  
- **probit union wage age**
- **margins, dydx(wage age)**

# Housekeeping

- Section 2 students are turning in Assignment 7 today
- **Section 1 students will turn in Assignment 7 on Tuesday Nov 15**
- **Assignment 7 was the last HW**
- **No More Homework Assignments Due**

# Lec 8

- **Panel Data**
  - Fixed Effects
  - Random Effects
  - Repeated Cross Sections

# Lec 9

- **Assignment 7 answers**
- **More Data Management Tricks**
- **Taking Leads and Lags**
- **Difference in Difference**

# Lec 10

- **Instrumental Variables**
- **Regression Discontinuity**
- **Maybe Matching Models**

# Why Panel Data?

- **Unobserved Variation: What is hiding in the error term?  $\text{Cov}(X, \text{error})$  NOT equal to 0**
- Panel data allows you to control for variables you cannot observe or measure like cultural factors or difference in business practices across companies or variables that change over time but not across entities (i.e. national policies, federal regulations, international agreements, etc.).
- This is, it accounts for individual heterogeneity

# Benefits & Problems of Panel Data

- you can include variables at different levels of analysis (i.e. students, schools, districts, states) suitable for multilevel or hierarchical modeling.
- Some drawbacks are data collection issues (i.e. sampling design, coverage), non-response in the case of micro panels or cross-country dependency in the case of macro panels (i.e. correlation between countries)

Use the following dataset by  
copying and pasting it in STATA

Save it in X:\ as class8data

$$wage_{it} = \alpha + \beta_1 edu_{it} + \beta_2 exp_{it} + \beta_3 gender_i + \varepsilon_{it}$$

Idcode	time	edu	exp	gender	wage
1	1	12	5	1	10
2	1	12	6	0	12
3	1	8	12	1	18
1	2	16	8	1	15
2	2	12	12	0	17
3	2	8	14	1	20
1	3	20	17	1	25
2	3	12	15	0	28
3	3	16	21	1	22

Pooled OLS

Treat each observation as i.i.d

Inconsistent Estimates

# Codes for Pooled OLS

- sum
- \* Pooled OLS Regression
- **reg wage edu exp gender**
- **reg wage edu exp gender, vce(cluster idcode)**

$$wage_{it} = \alpha + \beta_1 edu_{it} + \beta_2 exp_{it} + \beta_3 gender_i + \varepsilon_{it}$$

*Each observation NOT i.i.d*

$$\text{cov}(ability_{it}, edu_{it}) \neq 0$$

$$\text{cov}(ability_{it}, wage_{it}) \neq 0$$

$\hat{\beta}_1$  : *Not Identified*

# Simple STATA commands used for any Panel Dataset

- **xtset**
- Before we can issue panel data commands in Stata we need to inform the program that our data set has a panel dimension. We can do this with the xtset command. The syntax that we use is `xtset panelvar timevar`, where `panelvar` and `timevar` are the individual/entity and time-period Indicators respectively. Remember to use `xtset` before running any panel regressions.
- **isid**
- When we first encounter a data set, we need to figure out the identifier variables. We can do this using the `isid` command in Stata followed by a list of variables. If the variables that we have listed costs to unique identifiers within the data set the `isid` command will not produce any output.

# Notes on Fixed Effects Estimation

## Using `xtreg`

- Make Sure Data is in **LONG** format before declaring `xt` format
- If the data is in “**wide**” format, you have to turn it into “**long**” format by using “**reshape**” command
- You want x: **panel variable** (individual variable/entity variable/group variable) and t: **time variable** in separate columns – as in the dataset given above
  - Strongly Balanced Panel
  - Weakly Balanced Panel
  - Unbalanced Panel

# Use specialized panel commands after xtset:

- **xtdescribe**: extent to which panel is unbalanced
- **xtsum**: separate within (over time) and between (over individuals) variation
- **xttab**: tabulations within and between for discrete data e.g. binary
- **xttrans**: transition frequencies for discrete data
- **xtline**: time series plot for each individual on one chart
- **xtdata**: scatterplots for within and between variation.

# Codes for Fixed Effects Estimation

## Using **xtreg**

- **xtset idcode time**
- **xtsum wage edu exp gender time**

For time-invariant variable **gender** the ***within*** variation (i.e. within each individual) is zero.

For individual-invariant variable **time** the ***between*** variation (i.e. between one person to another) is zero

- **xtline wage, overlay**
- \* Fixed Effects Regression using **xtreg**
- **xtreg wage edu exp gender, fe**
- **xtreg wage edu exp gender, fe vce(cluster idcode)**

# Why was gender dropped? - The Mathematics Behind Fixed Effects Estimates

Path to an easier & harder estimation scheme

$$y_{it} = x_{it}\beta + \varepsilon_{it} \dots (0)$$

$$\text{cov}(x_{it}, \varepsilon_{it}) \neq 0$$

$$y_{it} = x_{it}\beta + c_i + u_{it} \dots (1)$$

$$\text{cov}(x_{it}, c_i) \neq 0 \quad \text{cov}(x_{it}, u_{it}) = 0$$

$$y_{it} = x_{it}\beta + c_i + u_{it} \dots (1)$$

$$\frac{1}{T} \sum_{t=1}^T y_{it} = \frac{1}{T} \sum_{t=1}^T x_{it}\beta + \frac{1}{T} \sum_{t=1}^T c_i + \frac{1}{T} \sum_{t=1}^T u_{it}$$

$$\bar{y}_i = \bar{x}_i\beta + c_i + \bar{u}_i \dots (2)$$

$$y_{it} = x_{it}\beta + c_i + u_{it} \dots (1)$$

$$\bar{y}_i = \bar{x}_i\beta + c_i + \bar{u}_i \dots (2)$$

$$(y_{it} - \bar{y}_i) = (x_{it} - \bar{x}_i)\beta + (c_i - c_i) + (u_{it} - \bar{u}_i)$$

$$(y_{it} - \bar{y}_i) = (x_{it} - \bar{x}_i)\beta + (u_{it} - \bar{u}_i) \dots (3)$$

$$\tilde{y}_i = \tilde{x}_i\beta + \tilde{u}_i \dots (4)$$

$$y_{it} = x_{it}\beta + c_i + u_{it} \dots (1)$$

Least Squares Dummy Variable Model: Use OLS on 1 after adding dummy for individuals

$$\tilde{y}_i = \tilde{x}_i\beta + \tilde{u}_i \dots (4)$$

Demeaned Model: Use OLS on 4 where variables are demeaned

# Codes for demeaning Variables

- \* Creating Time Mean of Variables
- **bysort idcode: egen edutimeavg=mean(edu)**
- **bysort idcode: egen gendertimeavg=mean(gender)**
- **bysort idcode: egen exptimeavg=mean(exp)**
- **bysort idcode: egen wagetimeavg=mean(wage)**
- \* Creating Demeaned Variables
- **gen wagedemean=wage - wagetimeavg**
- **gen edudemean=edu - edutimeavg**
- **gen expdemean=exp - exptimeavg**
- **gen genderdemean=gender - gendertimeavg**

**sum wagedemean edudemean expdemean  
genderdemean**

Success of fixed Effects estimation  
depends on variation within each  
individual

If you have too many time invariant  
variables like gender, your fixed  
effects estimation will be in trouble

# Codes for the easy, the medium and the hard

- \* Fixed Effects Regression using xi: reg:  
Least Squares Dummy Variable (LSDV)
- **xi: reg wage edu exp gender i.idcode,  
vce(cluster idcode)**
- **estimates store lsdv**
- \* Fixed Effects Regression using xtreg
- **xtreg wage edu exp gender, fe  
vce(cluster idcode)**
- **estimates store fixed**
- \* Regressions with Demeaned Variables
- **reg wagedemean edudemean  
expdemean genderdemean,  
vce(cluster idcode)**
- **estimates store demean**

# Conceptual Difference: Fixed Effects Versus Random Effects

$$y_{it} = x_{it}\beta + c_i + u_{it} \dots \dots (1)$$

*In Fixed Effects Models:*  $c_i$  fixed between the individuals

*In Random Effects Models:*  $c_i$  comes from some distribution

# Codes for Fixed Versus Random Effects

- \* Fixed Effects Regression using `xtreg`
- `xtreg wage edu exp gender, fe vce(cluster idcode)`
- `estimates store fixed`
- \* Random Effects using `xtreg`
- `xtreg wage edu exp gender, re vce(cluster idcode)`
- `estimates store random`
- `estimates table fixed random, star stats(N r2 r2_a)`

# Fixed Or Random Effects

- **xtreg wage edu exp gender, fe**
- **estimates store fixed**
- **xtreg wage edu exp gender, re**
- **estimates store random**
- **hausman fixed random**

Download: nlswork.dta and  
union.dta from

[http://www.stata-  
press.com/data/r14/xt.html](http://www.stata-press.com/data/r14/xt.html)

<http://fmwww.bc.edu/ec-p/data/mus/>

Download:mus08psidextract.dta  
(PSID wage data 1976-82 from Baltagi  
and Khanti-Akom (1990)) wnload

<http://dss.princeton.edu/training/Panel101.dta>

Paste This link in your browser to get Data. We will work with this dataset today

# Codes

- **clear all**
- **use Panel01.dta**
- **tab country**
- **tab year**
- **xtset country year**

# Codes: Data Exploration

\* Exploring Panel Data

**xtline y**

**xtline y, overlay**

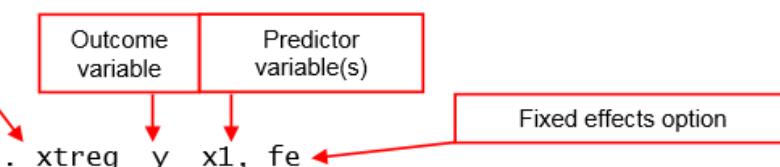
- \* Fixed Effects: Heterogeneity Across Countries
- **bysort country: egen y\_mean=mean(y)**
- **twoway scatter y country, msymbol(circle\_hollow) || connected y\_mean country, msymbol(diamond) || , xlabel(1 "A" 2 "B" 3 "C" 4 "D" 5 "E" 6 "F" 7 "G")**
- \* Fixed Effects: Heterogeneity Across Years
- **bysort year: egen y\_mean1=mean(y)**
- **twoway scatter y year, msymbol(circle\_hollow) || connected y\_mean1 year, msymbol(diamond) || , xlabel(1990(1)1999**

# Codes

- \* OLS Regression
- **regress y x1**
- **twoway scatter y x1, mlabel(country) || lfit y x1, clstyle(p2)**
- \* Fixed Effects Using Least Square Dummy Variables
- **xi: regress y x1 i.country**
- **predict yhat**
- **separate y, by(country)**
- **separate yhat, by(country)**
- **twoway connected yhat1-yhat7 x1, msymbol(none  
diamond\_hollow triangle\_hollow square\_hollow + circle\_hollow  
x) mszie(medium) mcolor(black black black black black black  
black) || lfit y x1, clwidth(thick) clcolor(black)**

$$Y_{it} = \beta_1 X_{it} + \dots + \beta_k X_{kt} + \alpha_i + e_{it} \quad [\text{see eq.1}]$$

**NOTE:** Add the option 'robust' to control for heteroskedasticity



Fixed-effects (within) regression  
Group variable: country

R-sq: within = 0.0747  
between = 0.0763  
overall = 0.0059

corr(u\_i, Xb) = -0.5468

The errors  $u_i$  are correlated with the regressors in the fixed effects model

Total number of cases (rows)	=	70
Number of obs	=	70
Number of groups	=	7
Obs per group: min	=	10
avg	=	10.0
max	=	10
F(1, 62)	=	5.00
Prob > F	=	0.0289

If this number is < 0.05 then your model is ok. This is a test (F) to see whether all the coefficients in the model are different than zero.

Coefficients of the regressors. Indicate how much Y changes when X increases by one unit.	y	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
	x1	2.48e+09	1.11e+09	2.24	0.029	2.63e+08 4.69e+09
	_cons	2.41e+08	7.91e+08	0.30	0.762	-1.34e+09 1.82e+09
29.7% of the variance is due to differences across panels.	sigma_u	1.818e+09				
'rho' is known as the intraclass correlation	sigma_e	2.796e+09				
	rho	.29726926				

(fraction of variance due to  $u_i$ )

Two-tail p-values test the hypothesis that each coefficient is different from 0. To reject this, the p-value has to be lower than 0.05 (95%, you could choose also an alpha of 0.10), if this is the case then you can say that the variable has a significant influence on your dependent variable (y)

$$\rho = \frac{(\sigma_u)^2}{(\sigma_u)^2 + (\sigma_e)^2}$$

$\sigma_u$  = sd of residuals within groups  $u_i$   
 $\sigma_e$  = sd of residuals (overall error term)  $e_i$

t-values test the hypothesis that each coefficient is different from 0. To reject this, the t-value has to be higher than 1.96 (for a 95% confidence). If this is the case then you can say that the variable has a significant influence on your dependent variable (y). The higher the t-value the higher the relevance of the variable.

**NOTE:** Add the option 'robust' to control for heteroskedasticity

Outcome variable	Predictor variable(s)	Random effects option
------------------	-----------------------	-----------------------

. xtreg y x1, re

Differences across units are uncorrelated with the regressors

Random-effects GLS regression  
Group variable: **country**

R-sq: within = 0.0747  
between = 0.0763  
overall = 0.0059

Random effects  $u_i \sim \text{Gaussian}$   
 $\text{corr}(u_i, X) = 0$  (assumed)

Number of obs	=	70
Number of groups	=	7
Obs per group: min	=	10
avg	=	10.0
max	=	10
Wald chi2(1)	=	1.91
Prob > chi2	=	0.1669

If this number is  $< 0.05$  then your model is ok. This is a test (F) to see whether all the coefficients in the model are different than zero.

y	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
x1	1.25e+09	9.02e+08	1.38	0.167	-5.21e+08 3.02e+09
_cons	1.04e+09	7.91e+08	1.31	0.190	-5.13e+08 2.59e+09
sigma_u	1.065e+09				
sigma_e	2.796e+09				
rho	.12664193				(fraction of variance due to $u_i$ )

Interpretation of the coefficients is tricky since they include both the within-entity and between-entity effects. In the case of TSCS data represents the average effect of X over Y when X changes across time and between countries by one unit.

Two-tail p-values test the hypothesis that each coefficient is different from 0. To reject this, the p-value has to be lower than 0.05 (95%), you could choose also an alpha of 0.10), if this is the case then you can say that the variable has a significant influence on your dependent variable (y)

# Fixed Effect or Random Effect:

Hausman Test

# Codes

- \* Random Effects
- **xtreg y x1, re**
- \* Fixed Effects or Random Effects
- **xtreg y x1, fe**
- **estimates store fixed**
- **xtreg y x1, re**
- **estimates store random**
- **hausman fixed random**

	Coefficients			
	(b) fixed	(B) random	(b-B) Difference	sqrt(diag(v_b-v_B)) S.E.
x1	2.48e+09	1.25e+09	1.23e+09	6.41e+08

b = consistent under  $H_0$  and  $H_a$ ; obtained from xtreg  
 B = inconsistent under  $H_a$ , efficient under  $H_0$ ; obtained from xtreg

Test:  $H_0$ : difference in coefficients not systematic

$$\begin{aligned}
 \text{chi2}(1) &= (b-B)'[(v_b-v_B)^{-1}](b-B) \\
 &= 3.67 \\
 \text{Prob>chi2} &= 0.0553
 \end{aligned}$$

If this is < 0.05 (i.e. significant) use fixed effects.

The remaining part of this lecture  
can be ignored – repeat of what we  
already talked about

# Description of a panel: use Panel101.dta

- In this case “country” represents the entities or panels (i) and “year” represents the time variable (t).
- The note “(strongly balanced)” refers to the fact that all countries have data for all years.
- If, for example, one country does not have data for one year then the data is unbalanced. Ideally you would want to have a balanced dataset but this is not always the case, however you can still run the model.

# NOTE:

- If you get the following error after using xtset:
- You need to convert ‘country’ to numeric,  
type:
- encode country, gen(country1)
- Use ‘country1’ instead of ‘country’ in the  
xtset command

# Fixed Effects

(Covariance Model, Within Estimator,  
Individual Dummy Variable Model,  
Least Squares Dummy Variable Model)

# Logic of FE

- Use fixed-effects (FE) whenever you are only interested in analyzing the impact of variables that vary over time.
- FE explore the relationship between predictor and outcome variables within an entity (country, person, company, etc.).

# Assumption1

- When using FE we assume that something within the individual may impact or bias the predictor or outcome variables and we need to control for this.
- This is the rationale behind the assumption of the correlation between entity's error term and predictor variables.
- FE remove the effect of those time-invariant characteristics so we can assess the net effect of the predictors on the outcome variable.

# Assumption 2

- Another important assumption of the FE model is that those time-invariant characteristics are unique to the individual and should not be correlated with other individual characteristics.
- Each entity is different therefore the entity's error term and the constant (which captures individual characteristics) should not be correlated with the others.
- If the error terms are correlated, then FE is no suitable since inferences may not be correct and you need to model that relationship (probably using random-effects), this is the main rationale for the Hausman test (presented later on in this document).

The equation for the fixed effects model becomes:

$$Y_{it} = \beta_1 X_{it} + \alpha_i + u_{it} \quad [\text{eq.1}]$$

Where

- $\alpha_i$  ( $i=1\dots n$ ) is the unknown intercept for each entity ( $n$  entity-specific intercepts).
- $Y_{it}$  is the dependent variable (DV) where  $i$  = entity and  $t$  = time.
- $X_{it}$  represents one independent variable (IV),
- $\beta_1$  is the coefficient for that IV,
- $u_{it}$  is the error term

# Least Square Dummy Variable Model

- The least square dummy variable model (LSDV) provides a good way to understand fixed effects. The effect of  $x_1$  is mediated by the differences across countries.
- By adding the dummy for each country we are estimating the pure effect of  $x_1$  (by controlling for the unobserved heterogeneity). Each dummy is absorbing the effects particular to each country.
- Comparing the fixed effects using dummies with `xtreg` we get the same results.

Another way to see the fixed effects model is by using binary variables. So the equation for the fixed effects model becomes:

$$Y_{it} = \beta_0 + \beta_1 X_{1,it} + \dots + \beta_k X_{k,it} + \gamma_2 E_2 + \dots + \gamma_n E_n + u_{it} \quad [\text{eq.2}]$$

Where

- $Y_{it}$  is the dependent variable (DV) where  $i$  = entity and  $t$  = time.
- $X_{k,it}$  represents independent variables (IV),
- $\beta_k$  is the coefficient for the IVs,
- $u_{it}$  is the error term
- $E_n$  is the entity n. Since they are binary (dummies) you have  $n-1$  entities included in the model.
- $\gamma_2$  is the coefficient for the binary regressors (entities)

Both eq.1 and eq.2 are equivalents:

"the slope coefficient on  $X$  is the same from one [entity] to the next. The [entity]-specific intercepts in [eq.1] and the binary regressors in [eq.2] have the same source: the unobserved variable  $Z_i$  that varies across states but not over time." (Stock and Watson, 2003, p.280)

You could add time effects to the entity effects model to have a *time and entity fixed effects regression model*:

$$Y_{it} = \beta_0 + \beta_1 X_{1,it} + \dots + \beta_k X_{k,it} + \gamma_2 E_2 + \dots + \gamma_n E_n + \delta_2 T_2 + \dots + \delta_t T_t + u_{it} \quad [\text{eq.3}]$$

Where

$-Y_{it}$  is the dependent variable (DV) where  $i$  = entity and  $t$  = time.

$-X_{k,it}$  represents independent variables (IV),

$-\beta_k$  is the coefficient for the IVs,

$-u_{it}$  is the error term

$-E_n$  is the entity  $n$ . Since they are binary (dummies) you have  $n-1$  entities included in the model.

$-\gamma_2$  is the coefficient for the binary regressors (entities).

$-T_t$  is time as binary variable (dummy), so we have  $t-1$  time periods.

$-\delta_t$  is the coefficient for the binary time regressors .

Control for time effects whenever unexpected variation or special events may affect the outcome variable.

# Codes: Ignore – we already know what to expect (from above discussion)

- \* Comparing OLS to LSDV model
- **regress y x1**
- **estimates store ols**
- **xi: regress y x1 i.country**
- **estimates store ols\_dum**
- **estimates table ols ols\_dum, star stats(N)**
- \* Fixed Effects using xtreg
- **xtreg y x1, fe**
- 
- \* Comparing FE to LSDV
- **xi: regress y x1 i.country**
- \* Comparing FE to AREG models
- **areg y x1, absorb(country)**
- **xtreg y x1 x2 x3, fe**
- **estimates store fixed**
- **xi: regress y x1 x2 x3 i.country**
- **estimates store ols**
- **areg y x1 x2 x3, absorb(country)**
- **estimates store areg**
- **estimates table fixed ols areg, star stats(N r2 r2\_a)**

# Big Problem

- What happens if the country variable is itself different across years & the differences affect x variable?
- FE Not appropriate anymore! And fixed effects estimates are not consistent.

# Random Effects

## (Random Intercept, Partial Pooling Model)

# Rationale behind Random Effects

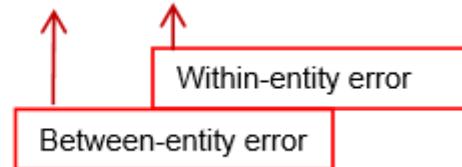
- The rationale behind random effects model is that, unlike the fixed effects model, the variation across entities is assumed to be random and uncorrelated with the predictor or independent variables included in the mode
- If you have reason to believe that differences across entities have some influence on your dependent variable then you should use random effects.

# Benefits & Costs of Random Effects

- An advantage of random effects is that you can include time invariant variables (i.e. gender).
- In the fixed effects model these variables are absorbed by the intercept.
- The marginal effects are not easily calculated

The random effects model is:

$$Y_{it} = \beta X_{it} + \alpha + u_{it} + \varepsilon_{it} \quad [\text{eq.4}]$$



Random effects assume that the entity's error term is not correlated with the predictors which allows for time-invariant variables to play a role as explanatory variables.

In random-effects you need to specify those individual characteristics that may or may not influence the predictor variables. The problem with this is that some variables may not be available therefore leading to omitted variable bias in the model.

RE allows to generalize the inferences beyond the sample used in the model.

**NOTE:** Add the option 'robust' to control for heteroskedasticity

Outcome variable	Predictor variable(s)	Random effects option
------------------	-----------------------	-----------------------

. xtreg y x1, re

Differences across units are uncorrelated with the regressors

Random-effects GLS regression  
Group variable: **country**

R-sq: within = 0.0747  
between = 0.0763  
overall = 0.0059

Random effects  $u_i \sim \text{Gaussian}$   
 $\text{corr}(u_i, X) = 0$  (assumed)

Number of obs	=	70
Number of groups	=	7
Obs per group: min	=	10
avg	=	10.0
max	=	10
Wald chi2(1)	=	1.91
Prob > chi2	=	0.1669

If this number is  $< 0.05$  then your model is ok. This is a test (F) to see whether all the coefficients in the model are different than zero.

y	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
x1	1.25e+09	9.02e+08	1.38	0.167	-5.21e+08 3.02e+09
_cons	1.04e+09	7.91e+08	1.31	0.190	-5.13e+08 2.59e+09
sigma_u	1.065e+09				
sigma_e	2.796e+09				
rho	.12664193				(fraction of variance due to $u_i$ )

Interpretation of the coefficients is tricky since they include both the within-entity and between-entity effects. In the case of TSCS data represents the average effect of X over Y when X changes across time and between countries by one unit.

Two-tail p-values test the hypothesis that each coefficient is different from 0. To reject this, the p-value has to be lower than 0.05 (95%), you could choose also an alpha of 0.10, if this is the case then you can say that the variable has a significant influence on your dependent variable (y)

# Fixed Effect or Random Effect:

Hausman Test

# Codes

- \* Random Effects
- `xtreg y x1, re`
- \* Fixed Effects or Random Effects
- `xtreg y x1, fe`
- `estimates store fixed`
- `xtreg y x1, re`
- `estimates store random`
- `hausman fixed random`

	Coefficients			
	(b) fixed	(B) random	(b-B) Difference	sqrt(diag(v_b-v_B)) S.E.
x1	2.48e+09	1.25e+09	1.23e+09	6.41e+08

b = consistent under  $H_0$  and  $H_a$ ; obtained from xtreg  
 B = inconsistent under  $H_a$ , efficient under  $H_0$ ; obtained from xtreg

Test:  $H_0$ : difference in coefficients not systematic

$$\begin{aligned}
 \text{chi2}(1) &= (b-B)'[(v_b-v_B)^{-1}](b-B) \\
 &= 3.67 \\
 \text{Prob>chi2} &= 0.0553
 \end{aligned}$$

If this is < 0.05 (i.e. significant) use fixed effects.

# Developing a Panel Dataset

# #1

- 1. Whenever you are constructing a new data set, you need to make sure you're comfortable with the various component data sources.
- The more command you have of the institutional background of each data set the easier it will be to work with the data.
- What follows is a list of some basic practices to observe when building a new data set.

## #2, #3

- 2. Figure out which variables identify individuals, households or other units of analysis in your data set.
- It is also important to figure out which if any variables indicate time or date.
- 3. When one unit of analysis is nested within another, it may make sense to generate a new individual identifier

## #4 - #8

- 4. Read the documentation for the data set. If the data comes from a survey, you should read the questionnaires carefully as looking at the data set is insufficient.
- 5. Always check for missing values. Missing values may be coded as 9, 99, 999, -999 or other alphanumeric strings. Remember that we need to recode these as "." In order for Stata to understand them properly.
- 6. When dealing with survey data it is important to know which member of a household or firm filled out the survey.
- 7. Some missing values may be sensible depending on the underlying structure of the data set.
- 8. Finally, use your existing variables to construct the outcome variables that you need.

# Lec 9

- **Assignment 7 answers**
- **More Data Management Tricks**
- **Taking Lags and Differences**
- **Difference in Difference**

# Lec 10

- **Instrumental Variables**
- **Regression Discontinuity**
- **Maybe Matching Models**
- **Final Exam**

# Assignment 7 Answers

Please use the nlsw88.dta dataset for this assignment.

1. (4 points) In the following model, calculate the marginal effect of being affiliated to a union. Show the formula you used to find the marginal effect. Use at least one table and one graph to clearly present your results.

$$wage = \beta_0 + \beta_1 union + \beta_2 tenure + \beta_3 tenure \times union + \beta_4 grade + \beta_5 grade \times union \quad (1)$$

2. (6 points) The following is a logit model to study the determinants of union affiliation

$$\Pr(y = 1) = \frac{e^{x\beta}}{1 + e^{x\beta}} \quad (2)$$

, where  $x\beta = \beta_0 + \beta_1 age + \beta_2 married + \beta_3 age \times married$

- (a) Does the model allow for a different intercept for married and single women, or does it impose the same intercept for both groups?
- (b) Using the subsample of non-married women, estimate the marginal effect of age on union affiliation for this subgroup (in this case  $x\beta = \beta_0 + \beta_1 age$ ).
- (c) Using the whole sample (not only non-married women) estimate the marginal effect of age on union affiliation for non-married women. Hint: look carefully at equation 7 in the lecture notes.
- (d) Using the whole sample (not only non-married women) estimate the marginal effect of age on union affiliation for married women.

# For Obtaining the Marginal Effects & a Table

- **gen ten\_uni=tenure\*union**
- **gen gra\_uni=grade\*union**
- **xi: reg wage tenure grade union ten\_uni gra\_uni**
- **outreg2 using ass7\_table1.doc, replace bdec(4) sdec(4)**
- **reg wage tenure grade if union==1**
- **outreg2 using ass7\_table1.doc, append bdec(4) sdec(4)**
- **reg wage tenure if union==1**
- **outreg2 using ass7\_table1.doc, append bdec(4) sdec(4)**
- **reg wage grade if union==1**
- **outreg2 using ass7\_table1.doc, append bdec(4) sdec(4)**

VARIABLES	(1)	(2)	(3)	(4)
	wage	wage	wage	wage
	Fullsample	Union Subsample	Union Subsample	Union Subsample
tenure	<b>0.1567***</b> (0.0180)	<b>0.1456***</b> (0.0292)	0.1746*** (0.0309)	
grade	<b>0.7118***</b> (0.0395)	<b>0.5210***</b> (0.0641)		0.5637*** (0.0652)
union	3.4794*** (1.0006)			
ten_uni	<b>-0.0111</b> (0.0333)			
gra_uni	<b>-0.1908***</b> (0.0731)			
Constant	-3.0121*** (0.5237)	0.4673 (0.8892)	7.3072*** (0.3079)	1.0245 (0.9035)
Observations	1,866	459	460	460
R-squared	0.246	0.184	0.065	0.140

Standard errors in parentheses

\*\*\* p<0.01, \*\* p<0.05, \* p<0.1

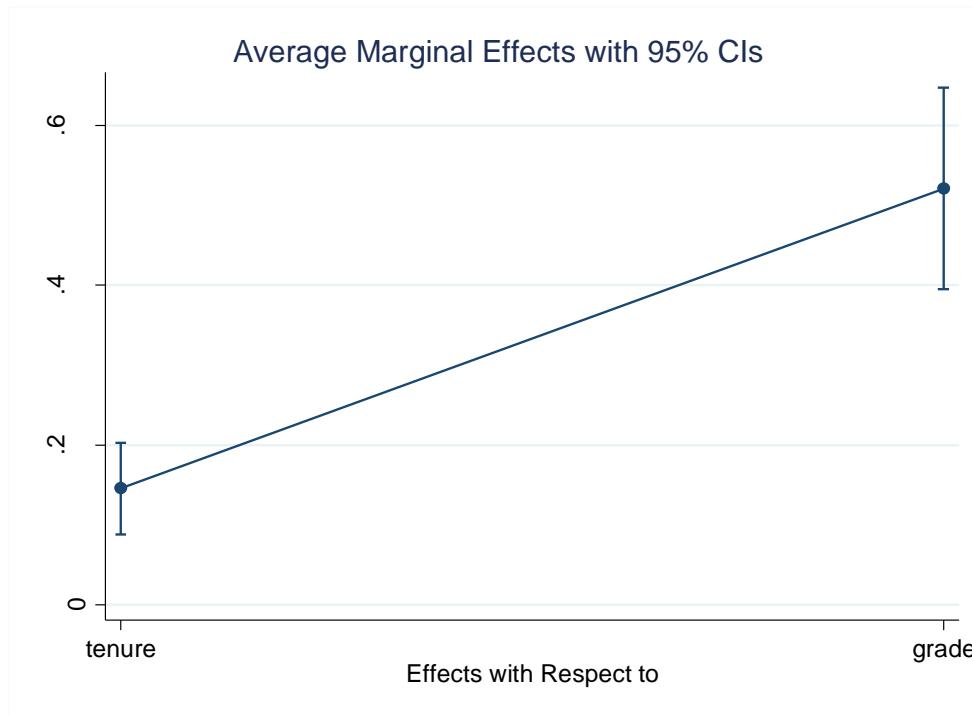
$$\frac{\partial wage}{\partial tenure} = \hat{\beta}_2 + \hat{\beta}_3 \times Union$$

$$\frac{\partial wage}{\partial tenure} = \hat{\beta}_4 + \hat{\beta}_5 \times Union$$

$$\frac{\partial wage}{\partial tenure} = .1456$$

$$\frac{\partial wage}{\partial grade} = .5210$$

# For Obtaining a Graphical Visualization of the Marginal Effects



- **`reg wage tenure  
grade if  
union==1`**
- **`margins,  
dydx(tenure  
grade)`**
- **`marginsplot,  
graphregion(col  
or(white))  
ytitle(marginal  
effect on wage)`**

If you have used lfit: I do not think  
that is appropriate because lfit  
gives the coefficients of yvar and  
xvar ignoring other covariates

Please use the nlsw88.dta dataset for this assignment.

1. (4 points) In the following model, calculate the marginal effect of being affiliated to a union. Show the formula you used to find the marginal effect. Use at least one table and one graph to clearly present your results.

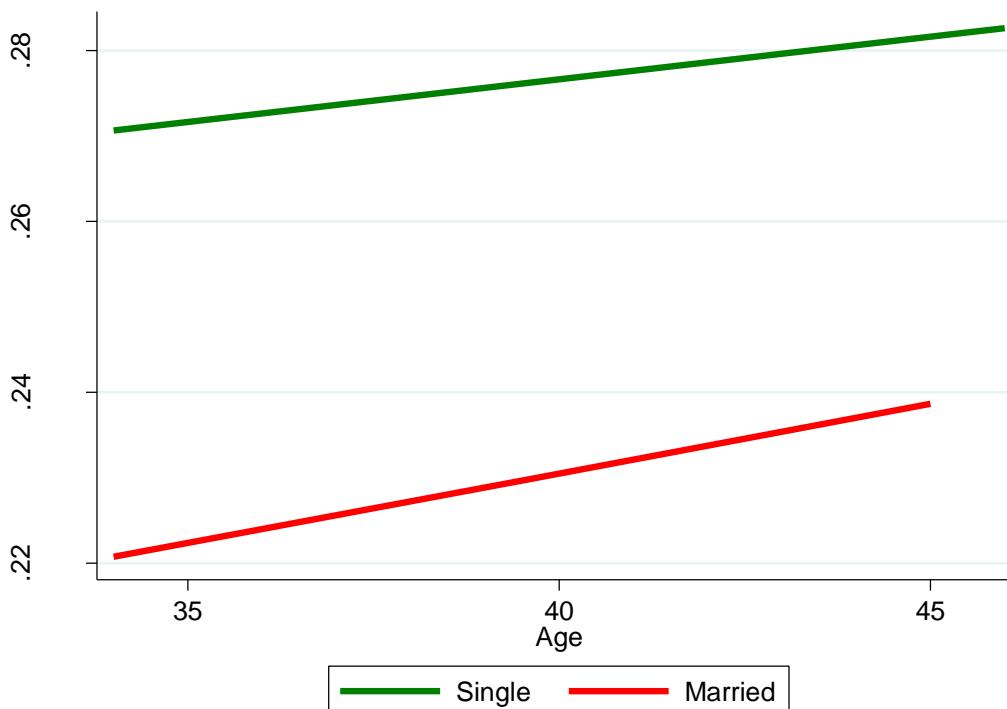
$$wage = \beta_0 + \beta_1 union + \beta_2 tenure + \beta_3 tenure \times union + \beta_4 grade + \beta_5 grade \times union \quad (1)$$

2. (6 points) The following is a logit model to study the determinants of union affiliation

$$\Pr(y = 1) = \frac{e^{x\beta}}{1 + e^{x\beta}} \quad (2)$$

, where  $x\beta = \beta_0 + \beta_1 age + \beta_2 married + \beta_3 age \times married$

- (a) Does the model allow for a different intercept for married and single women, or does it impose the same intercept for both groups?
- (b) Using the subsample of non-married women, estimate the marginal effect of age on union affiliation for this subgroup (in this case  $x\beta = \beta_0 + \beta_1 age$ ).
- (c) Using the whole sample (not only non-married women) estimate the marginal effect of age on union affiliation for non-married women. Hint: look carefully at equation 7 in the lecture notes.
- (d) Using the whole sample (not only non-married women) estimate the marginal effect of age on union affiliation for married women.



### \*Part a

```
gen age_marr= age*marr
logit union age married age_marr
predict phat if e(sample),pr
```

**(Phat is the predicted probability conditional on X,X2,...)**

\* The following gives the graphs for married and non married women

```
twoway (lfit phat age if married==0, lw(thick) lc(green) legend(label(1 "Single")))| |(lfit phat
age if married==1, lw(thick) lc(red) legend(label(2 "Married"))), graphregion (color(white))
xtitle("Age") ytitle("Probability of being in Union")
```

# Problem 2

- \* Part b
- **logit union age married age\_marr if married==0**
- **margins, dydx(age)**
- \* Part c
- **logit union age married age\_marr**
- **mfx, at(married=0)**
- \* Part d
- **logit union age married age\_marr**
- **mfx, at(married=1)**
- **(If you are not restricting a variable to take a specific value, use margins & when you are forcing a variable to take a specific value use mfx)**

# Data Management Tools

Data into Stata

# How to get excel/csv data into stata

## Method 1

Copy and  
Paste

## Method 2

Use File and  
Import  
**WBdata.xls &**  
**Wbcountrydat**  
**a.xls**

## Method 3

Use command (for  
csv data only):  
**“insheet”**

# Codes for reigning in unruly WBdata

- use WBdata
- ren A country
- ren B y1\_1
- ren C y1\_2
- ren D y2\_1
- ren E y2\_2
- ren F y3\_1
- ren G y3\_2
- ren H y4\_1
- ren I y4\_2
- ren J y5\_1
- ren K y5\_2
- ren L y6\_1
- ren M y6\_2
- destring, replace
  
- forvalues j=1/6{
- foreach k in 1 2{
- replace y`j'\_`k'=". " if y`j'\_`k'==".."
- }
- }
- destring, replace ignore(",")
- reshape long y1\_ y2\_ y3\_ y4\_ y5\_ y6\_,  
i(country)j(year)
  
- \* Removing Underscore
- forvalues j = 1/6 {
- rename y`j'\_ y`j'
- }
  
- la var y1 "net energy imports"
- la var y2 "GDP per unit of energy use"
- la var y3 "CO2 emissions: total (thousand metric tons)"
- la var y4 "CO2 emissions: intensity (kg per kg of oil equivalent)"
- la var y5 "CO2 emissions per capita (metric tons)"
- la var y6 "CO2 emissions: kg per \\$ of GDP"
  
- \* xtset country year
  
- encode country, g(cid)
- xtset cid year
  
- save "WBdataXtset.dta", replace

# Time Series Operators: Taking Time Lags and Differences

Important for Time Series Analysis

# Lags & Differences

- The first lag of  $y$  at time  $t$  is the value taken by  $y$  at time  $t-1$ .
- If  $t = 2016$ ;  $t-1$  refers to data from 2015
- The  $k$ -lag of  $y_t$  is equal to the value of  $y$  taken  $k$  periods before  $t$ .
- If  $t = 2016$ ;  $t-k$ ; where  $k=7$  refers to data from 2009
- The difference in  $y$  at time  $t$  is the difference between values of  $y$  at time  $t$  and time  $t-1$ .
- If  $t = 2016$ ; first difference would be  $x_{2016} - x_{2015}$

<b>idcode</b>	<b>time</b>	<b>unemp_per</b>
1	1	10
2	1	12
3	1	18
1	2	15
2	2	17
3	2	20
1	3	25
2	3	28
3	3	22

# Codes

$$\begin{aligned}L1.y_t &= y_{t-1} & L2.y_t &= y_{t-2} & L3.y_t &= y_{t-3} \\D1.y_t &= y_t - y_{t-1} & D2.y_t &= y_t - y_{t-2} & D3.y_t &= y_t - y_{t-3}\end{aligned}$$

```
rename unemp_per y
```

First declare xtset so stata knows its panel data

```
gen y_lag1=L1.y
```

```
gen y_diff1=D1.y
```

## Why Missing Values?

# Difference in Difference

# Introducing Randomized Control Trial (RCT)

X: Medicine

Y: Red Blood Cell Count

Want to Prove that

X  Y

To Eliminate other Factors

- 1) Choose Treatment and Control roughly having similar characteristics
- 2) Randomize Intervention between Treatment and Control
- 3) Calculate Difference between Outcomes of two groups
- 4) Repeat Experiment

	Red Blood Cells in Treatment Group	Red Blood Cells in Control Group
Before the Treatment	$2 = T_1$	$2.3 = C_1$
After the Treatment	$7 = T_2$	$3 = C_2$
Difference = After - Before	$7 - 2 = +5 = T_2 - T_1$	$3 - 2.3 = .7 = C_2 - C_1$

Difference in Difference Estimate  
= Difference for Treatment – Difference for Control  
=  $5 - 0.7 = +4.3 = (T_2 - T_1) - (C_2 - C_1)$

Table 1: Profits by Treatment Status and Time Period

	Pre-Tax Change	Post-Tax Change
Treatment	T1	T2
Control	C1	C2

$$\hat{\beta}_{DD} = (T_2 - T_1) - (C_2 - C_1)$$

# Treatment Control Post Time

## School: 1,2 School: 3 Period: 3

idcode	time	textbook	post	score	tquality
1	1	1	0	20	10
2	1	1	0	22	12
3	1	0	0	18	18
1	2	1	0	22	15
2	2	1	0	21	17
3	2	0	0	20	20
1	3	1	1	43	25
2	3	1	1	38	28
3	3	0	1	27	22

# DOD: Adding a time dimension

$$Score_{it} = \alpha + \beta Textbook_i + \delta Textbook_i \times Post_t + \lambda Post_t + \varepsilon_{it}$$

$$Textbook_i \in [0,1] \quad Post_t \in [0,1]$$

$$i = 1, 2, \dots, S \quad t = 1, 2, \dots, T$$

Table 1: Profits by Treatment Status and Time Period

	Pre-Tax Change	Post-Tax Change
Treatment	T1	T2
Control	C1	C2

$$T1 = \alpha + \beta$$

$$T2 = \alpha + \beta + \delta + \lambda$$

$$C1 = \alpha$$

$$C2 = \alpha + \lambda$$

$$\hat{\beta}_{DD} = (T2 - T1) - (C2 - C1)$$

# Codes

- **gen text\_post= textbook\* post**
- **reg score textbook text\_post post**

$$\hat{\alpha} = 19$$

$$\hat{\beta} = 2.25$$

$$\hat{\delta} = 11.25$$

$$\hat{\lambda} = 8$$

T1	21.25	T2-T1	19.25
T2	40.5	C2-C1	8
C1	19		
C2	27	beta_DD	11.25

$$Score_{it} = \alpha + \beta Textbook_i + \delta Textbook_i \times Post_t + \lambda Post_t + \varepsilon_{it}$$

$$Score_{it-1} = \alpha + \beta Textbook_i + \delta Textbook_i \times Post_{t-1} + \lambda Post_{t-1} + \varepsilon_{it-1}$$

Subtracting the second from the first equation  
and simplifying gives us the following

$$\Delta Score_{it} = \delta Textbook_i \times \Delta Post_t + \lambda \Delta Post_t + \Delta \varepsilon_{it}$$

**reg D.score D.text\_post D.post**

**diff:  
ssc install diff  
help diff**

Parallel Paths Assumption (Without treatment, the average change for the treated would have been equal to the observed average change for the controls),  
covariates

# Panel Data Application

<http://fmwww.bc.edu/ec-p/data/mus/>

Download:mus08psidextract.dta  
(PSID wage data 1976-82 from Baltagi  
and Khanti-Akom (1990)) wnload