Department of Economics
University of California
Berkeley

Spring 2018
Economics 154
Professor Martha Olney

Stata Lesson
Thursday February 15, 2018

[1] Where to find the data sets

http://www.econ.berkeley.edu/~olney/spring18/econ154

There are four versions of the dataset there: txt, excel, stata, and sas transport file. Download all four to your desktop.

[1.1] To transfer an excel data set to stata format

File -> Import -> Excel spreadsheet (.xls, .xlsx), -> Browse to file -> choose option "Import first row as variable names" -> "ok"

If the file is in .csv or .txt format,
File -> Import -> Text data (delimited, *.csv), -> Browse to file ->"ok"

[1.2] To read in data that are in fixed format

First, hope that you don't have to do this! It was common back in the old days (say, 10 years ago) but is far less common today.

Read the manual!
Use infix
Create a "dictionary file" that specifies the locations of the variables and their new names

[1.3] To open a stata data set

start stata
click on the "open file" icon
know what folder your data set is in
click on the data set that has a .dta extension

if it's on a webpage, you may be able to just double-click on the link

[2] To save your work (**important**! You want to be able to remember what you did)

Create a log file: (always do this!)
- File - Log - Begin
  - (or, click on the icon that looks like a spiral notebook)
- Choose either formatted log file (*.scml; only readable within stata)
  - Or text log file (*.log; readable within any word processor) **Better!**
- Give the file a name that includes the date & maybe more
  - "2-15-18 in class.log" for instance

[3] To look at the data

describe Shorthand: des

list (Careful, this can go on and on; use red X to stop)
If it listed all observations at once, "set more on" will give you only one screen at a time.

Data editor/browse (the little icon that has grid lines & a magnifying glass)
Or data > data editor > data editor (browse)

"Tabulate" tells you how many different occurrences there are of each observed value of a variable; useful for variables with just a few possible values; not useful for continuous variables. Including two variables gives you a cross-tab table.

tab *variable-name* (gives a list of all the values and their frequencies)

e.g.

tab occ How many teachers, principals, superintendents?

tab occ sex By gender, how many teachers, principals, superintendents?

"Summarize" tells you a variety of summary statistics.

summarize *variable-name*
or sum *variable-name* (gives brief summary statistics)

sum *variable-name*, detail (gives longer list of summary statistics)

[3.1] To look at the dataset sorted by some variable

bysort *variable-name*: sum *variable name* shorthand: bys

e.g.

bysort sex: sum totear

[3.2] To re-sort the variables

Note: there is no "undo" in stata!! Once you re-order your variables, you can't change your mind and put them back in the previous order.

order *varlist*
will put variables in the order you type, with everything else at the bottom in the existing order; can use *varname1 - varname'n'* to specify the *n* variables between *varname1* and *varname'n'*

e.g.
order id sex male female occ occup1-occup3

To put variable list in alphabetical order,
order _all, alpha — alphabetizes all variables

order *varlist*, alpha — alphabetizes variables in list only

eg
order id-occup3, alpha

[4] To start naming the variables

label variable *variable-name "variable label"* — shorthand: la var

e.g.
Label variable pob "Place of Birth" — Quote marks important

[4.1] Or, use the variable manager

Click on the icon just to the left of the grayed out down arrow

To name a variable, click on the variable name on the left
Then, type the long variable name into the "Label" box on the right, click "apply"

[5] To name the values of variables; two steps

Step one:
label define *labelname value "name" value "name"*

e.g.
Label define sexlabel 1 "male" 2 "female"

Step two:
label values *variablename labelname* — shorthand: la val

e.g.
Label values sex sexlabel

[5.1] Or, use the variable manager

Click on the icon just to the left of the grayed out down arrow

To create a label, click on "manage" to the right of "value label" in the box on the right

Next, click "create label"

Then type in a label name ("sexlabel" or "yesno" or whatever)
e.g., occlabel
Now, type in the values and the labels on the right hand side, clicking "add" after each set
e.g., 1=teacher; 2=principal; 3=superintendent

When done, click "ok" and then "close"

Now, go back to the relevant variable (here, occ), click on the value label box on the right, click on the drop-down menu, and the label "occlabel" should show up.

When you're done, click "apply" before closing out the Variables Manager

[6] To create new variables

generate *newvariable = some function of existing variables* shorthand: g or gen
e.g.
gen num_months_paid = totear / earnmo

[6.1] Creating indicator (dummy) variables

https://www.stata.com/support/faqs/data-management/creating-dummy-variables/ is great

[6.1.1] To create indicator (dummy) variables

generate *indicatorvar = (existing variable == value for indicator to equal 1)*
e.g.
generate male = (sex == 1) Important: double = sign
or
generate grad = (college == 2) Parentheses optional
or
generate young = age<30

Be careful!! Stata treats missing values as positive infinity (not as .), and so when I create "male" the 0's will include not only the females in the data set but also the observations with missing values. Solution

generate young = age < 30 if !missing(age)

(See also 6.3 – we haven't yet defined missing values for this data)

[6.1.2] To create several indicator (dummy) variables at once

tab *existingvar*, gen(*indicatorvar*)

e.g.

tab occ, gen(occup)

Note that the variables are named occup1, occup2, occup3 (not teacher, principal, superintendent). Consider changing the variable names using ren

[6.1.3] To use "factor variables" instead of indicator (dummy) variables

You can substitute "i.occ" in many commands and Stata reads that as "create [temporary, unsaved] dummy variables for each of the values of occ *other than the comparison or base group*"

summarize i.occ — Gives you summary stats for each of the values of occ *other than the base group (by default, the first group)*

regress *outcomevariable* i.*factorvariable*

e.g.

regress totear i.occ

[6.2] To replace values of existing variables

replace *varname* = *newvalue* if *varname* == *oldvalue*

e.g.

```
gen female = sex
replace female = 0 if female == 1
replace female = 1 if female == 2
```

[6.3] To replace missing values of existing variables

replace *varname* = . if *varname* == *missingvalue*

e.g.

Replace age = . if age == -9

If all variables use -9 for the missing value, can instead do this

mvdecode _all, mv(-9)

If some but not all variables use -9 for the missing value, then this works

mvdecode age hours terms earnmo totear colyr board , mv(-9)

Are some -9 values actually "0" rather than "missing"? If so, recode those

replace *varname* = 0 if *varname* == -9

[7] To graph the data

Click “graphics” along the top menu

Choose the type of graphs (start with “twoway graph”)
Click “create”
Choose a particular graph (try “basic plots”, “scatter plot” to start)

Specify x (horizontal) axis variable and y (vertical) axis variables

Specify any “if” restrictions

Click “submit” and the graph will pop up (the more data, the longer it takes)

Edit your specifications as you wish; click “submit” again

when you have it as you like it, **save your graph**
(I find .wmf format to be the most portable format)

[8] To run a linear regression

regress *depvariable independentvariables*

e.g.

Regress totear male

[8.1] Running regressions with restrictions

Want only observations for teachers

Add if statement at the end of the command:

e.g.

regress totear male if occ==1

[9] To run a probit (or dprobit or logit or logistic)

When your dependent variable is 0/1, OLS (here, "a linear probability model") is not the best approach. OLS coefficients are easy to interpret but the method will predict values <0 and >1, which makes no sense. So we use nonlinear regression models designed for 0/1 (binary) dependent variables: probit or logit.

The difference between probit and logit is which probability density function we think is best for describing the dependent variable. Probit assumes standard normal cdf; logit assumes logistic cdf. For our purposes it is unlikely to matter.

Probit *depvariable independentvariables*

e.g.

```
Generate ownhome = (ownhm == 2)
Probit ownhome male totear age
```

Those coefficients are not directly interpretable. They are for the equation

$\Pr(Y = 1 \mid X_1, X_2) = \Phi(\beta_0 + \beta_1 X_1 + \beta_2 X_2)$, where Φ is the cumulative standard normal distribution function (visions of Stat 20 now dance in your head)

To get probit results that are directly interpretable

Dprobit *depvariable independentvariables*

e.g.

```
Dprobit ownhome male totear age
```

If instead you want to use logit, the command is just

logit *depvariable independentvariables*

```
Logit ownhome male totear age
```

Here the equation being estaimted is $\Pr(Y = 1 \mid X_1, X_2) = \dfrac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2)}}$

To get results reported as odds ratios (common in sociology, less common in econ)

```
Logistic ownhome male totear age
```

[10] Creating interaction terms

Suppose you think that gender matters, but not as a constant shift factor. You think instead that the effect of gender is to alter the return to college education. In that case you need to create "an interaction term"

*Gen male_grad = male * grad*

Now include this as a variable in your regression, along with just "grad." The coefficient on "male_grad" tells you if men have a different return to being a college grad than women do.

[10.1] Interpreting interaction terms

The coefficient on the interaction term tells you whether the effect of the variable of interest (here, grad) is different for male than it is for female.

The coefficient on the interaction term does not tell you whether "grad" has a statistically significant effect for men.

Sometimes you want the variable itself (grad) and the interaction term (male_grad) because you want to know if the effect differs, here by gender

reg totear i.occ grad male_grad

Sometimes you want only the interaction terms and not the variable itself, because you want to know if the effect is statistically significant for each gender

*gen female_grad = (1-male)*grad*
to see what we just did, do this
bys sex: sum grad male_grad female_grad
and then this regression
reg totear i.occ female_grad male_grad

[10.2] Interaction terms with factor variables

If I don't want to save my interaction terms, I can use factor variables.

The general format is

i.*variablename#variablename*

For instance

i.occ#sex creates interaction terms between occupation (with the first group as the base) and sex (with the first group as the base)

Reg totear i.occ i.occ#sex

If you want to specify a different base, that specification goes after the i and before the .

Reg totear ib3.occ ib3.occ#sex

or

Reg totear i.occ i.occ#b2.sex

[11] Now, let's play with the data set

Estimate a relationship between total earnings and its determining variables. Think first about:

a) how will you take age into account
b) does gender matter? Does it shift the earnings function, or change some of the returns, or both?
c) do you want separate equations for teachers, superintendents, principals, or do you just want a shift factor for different occupations?

[12] "do" files

When you have a lot of commands, write a "do" file which contains all of the commands and then have stata run the "do" file
Save the "do" file in ascii format, with the suffix .do. For instance "monday.do"
Then type
do Monday
or
do monday, nostop
(Nostop is an option that tells stata to keep going even if it encounters an error)

If you create a do file and immediately execute it, your log file will show each line of the dofile separately. Worthwhile to save the do file (use a logical file name perhaps with a date in it), because most people wind up having to do all the initial data work twice.

In do files, you can have for/next loops and all the usual coding features.
Very very handy.

You can also comment, which is important for replication
/* starts a comment
*/ ends a comment
*/ this is a comment */*

/ this is a comment
that runs over more than one line */*

In a do file, if you have a command that takes >1 line, you indicate "I'm not done yet!!" with three slashes at the end of a line. (See the example under misc. other, 13.4)

[13] Miscellaneous other

[13.1] Multicollinearity.

If you have collinear independent variables, it's a problem if what you care about is one particular coefficient (e.g., coeff on 'male') but not if what you care about is overall best fit for explaining dependent var. www.stat.tamu.edu/~hart/652/collinear.pdf

[13.2] Oaxaca

To do Blinder Oaxaca decomposition, google "stata oaxaca"
Or "help oaxaca" from within stata, but I find that google produces better results

[13.3] Saving output from one particular regression

To save regression results so you can later make a table, immediately after you run the regression type

Estimates store *nameYouWantToGiveIt*

[13.4] Creating tables

To create tables with stored results, I use "esttab"

For instance, I ran 4 regressions and after each typed stored the estimates. Then created table with esttab.

```
        regression blah blah blah
          estimates store row1_249_5yr
        regression blah blah blah
          estimates store row1_239_5yr
        regression blah blah blah
          estimates store row1_218_5yr
        regression blah blah blah
          estimates store row1_208_5yr
esttab row1_249_5yr row1_239_5yr row1_218_5yr row1_208_5yr , ///
        not se(%4.3f) drop(_cons *.year)  sca(F r2_w idp jp) star(* 0.10 ** 0.05 *** 0.01) ///
        title ("Table 3, column 3, Panel C, adding in 5 yr growth trend")
```

[13.5] Identifying subset of observations that are in a particular regression

Immediately after you run a regression type

gen *newvariablename* = e(sample)

which creates a new variable with 0/1 for the observations in the regression (useful for creating summary stats tables)

e.g.

```
reg totear grad male male_grad if occ==1
gen teachersonly=e(sample)
```