Lab 3: Regressions and Visualization

Simon Halliday

ECO311, Fall 2016

Revision Quiz

In your notes for your exercises for this lab, try to answer the following quiz questions based on Lab 2 *before class* and *without checking the lab*! In your exercises, have three categories for each question, “M" for my answer, “TPS” for think-pair-share, and “N" for after notes.

a. What command do you use to make the original code in a variable to be *missing*?

b. What command provides the summary statistics of a variable? What option do you have to specify to get information about the median, mode, etc?

c. What command do you need to use to create a table with summary statistics?

d. Explain in English what the following set of commands do:

```
gen someschool = 1
replace someschool = 0 if w1_r_b7 == 25
replace someschool = . if w1_r_b7 == .
```

e. Explain what you think the following commands do and explain why we might use the variable defined on the third line of the commands, `lnpcy`:

```
egen hhsize = count(pid), by(hhid)
gen pcy = hh_income / hhsize
gen lnpcy = ln(pcy)
```

f. Explain what you would do to complete the following tasks:

   i. Tag one member of each household
   ii. Temporarily save your data in memory
   iii. Get rid of all the household members who haven’t been tagged
   iv. Find the average of the variable `pcy` defined above
   v. Go back to the data in memory before you eliminated the household members.

g. What process would you go through to visualize the relationship between years of education and the variable `pcy` we defined above?

When you get to class, check your answers with your neighbors (Think-Pair-Share). Once you have checked with your neighbors, go back to the lab to check what you didn’t remember from the lab notes.

Doing Stata

By the end of this lab, you should have:

- Use the `reg` (`regress`) command to run a regression in Stata.
- Understand and interpret a simple linear regression with a *continuous independent variable*
- Understand and interpret a simple linear regression with an *independent dummy variable*
- Understand a multiple regression with independent *continuous* and *dummy* variables
- Use the command `predict` to analyze whether the output of a regression is a good fit or not in a graphic representation of the data
- Understand what *dummy variables* do to the regression line for different categories of observations
- Understand what *interaction terms* do to the regression line for different categories of observations
- Export regression output using a variety of different commands `outreg`, `estout` and `putexcel`

Introduction

Before you start this lab session remember to do the following:

- Direct your working directory to the Lab3 folder (or Labs folder if you prefer)
- Make sure that your directory structure is consistent with Lab1 (CommandFiles, log, cmdlog, etc) so you can use *relative file paths*.
- Open a `log` file to track your work:

```
log using ./ProcessingAnalysis/CommandFiles/logs/
    yourlastname_lab3.log
```

  Be sure to specify the file extension `.log`

- Open a cmdlog file

```
cmdlog using ./ProcessingAnalysis/CommandFiles/cmdlogs/
    yourlastname_lab3.do
```

After you have followed these steps, copy the NIDS data into your OriginalData folder for Lab3 and then open the data.

Simple Linear Regression

In the plots above we have seen there seems to be a linear relationship between age and years of complete education for adults. We go ahead and run a regression analysis. The command to run a regression is `regress` which can be abbreviated to `reg`. The `reg` command gets used in the following way:

```
reg y x, options
```

The y-variable is the variable you are trying to understand or the *dependent* variable. The x-variable is an *independent* variable you are trying to use to explain the *dependent* variable.

For example, let's look at a relationship between two variables in the NIDS data: `edyears` and `w1_r_b6`.

```
reg edyears w1_r_b6 if w1_r_b6 >= 18 & w1_r_b6 <= 80
```

To run the regression above, you may need to `generate` the `edyears` variable from Lab 2.

```
recode w1_r_b7 (-9/-3 24 = .) (25 = 0) (13 16 = 10) (14
    17 = 11) (15 = 12) (18 = 13) (19 = 14) (20 = 15) (21
    22 = 16) (23 = 17), gen(edyears)
```

Make sure that you can interpret all coefficients and statistics in the regression output.

HINT if you are rusty – some good standard text book treatments of regression analysis are in Gujarati and Porter's *Basic Econometrics* or Wooldridge's *Introductory Econometrics*. You can also check some of the basics in your ECO220 textbook, Smith's *Essential Statistics, Regression, and Econometrics*. A free statistics book is Open Intro Statistics.

What is the relationship between `w1_r_b6` and the dependent variable `edyears`?

We can use the regression output to predict outcomes for each individual *given* their observable characteristics for the variable `w1_r_b6`:

```
predict p0
```

What does `predict p0` do? It uses the linear regression with one variable, as you would with a basic line $y = a + mx$, but here y is replaced by $p0$, a is the constant from the regression and m is the slope, or the coefficient from the regression, i.e.

$$y_i = \beta_0 + \beta_1 x_i \quad (1)$$

$$p0 = constant + \beta_1 \bar{x} \quad (2)$$

But, the predicted line also needs to be evaluated at the *average* for our graph to make sense, i.e. at $\bar{x}$.

We can now look at a plot of the predicted values (from the regression) versus the data. First, we need to generate mean education for each age group:

```
egen meaned2 = mean(edyears), by(w1_r_b6)
```

Now we can use `graph twoway` (or the drop-down menus) to generate the line graph:

```
twoway (line p0 meaned2 w1_r_b6 if w1_r_b6 >= 18 &
    w1_r_b6 <= 80, sort)
```

Another useful way to visualize the data is with the command `lfitci` which is an abbreviation for the idea of "linear fit with confidence interval":

```
twoway (lfitci edyears w1_r_b6 if w1_r_b6 >= 18 & w1_r_b6
     <= 80, sort) (line meaned2 w1_r_b6 if w1_r_b6 >= 18 &
     w1_r_b6 <= 80)
```

```
twoway (lfitci edyears w1_r_b6 if w1_r_b6 >= 18 & w1_r_b6
     <= 80, sort) (line meaned2 w1_r_b6 if w1_r_b6 >= 18 &
     w1_r_b6 <= 80)
```

The confidence interval around the mans is given by the grey band around the blue line. This shows us something about how *precise* the measure is.

We can add the line for the original data's means back in with the following:

```
twoway (lfitci edyears w1_r_b6 if w1_r_b6 >= 18 & w1_r_b6
     <= 80, sort) (line meaned2 w1_r_b6 if w1_r_b6 >= 18 &
     w1_r_b6 <= 80, sort)
```

Regression with Dummy Variables

Now let's look at the difference in average years of completed education for adults (18 to 80 years old) men and women by regressing years of education on gender:

```
reg edyears male if w1_r_b6 >= 18 & w1_r_b6 <= 80
```

If you need to generate your `male` variable you can use the following code:

```
gen male = w1_r_b4 == 1
```

This code for generating a variable produces the same as the following:

```
gen male = 0
replace male = 1 if w1_r_b4 == 1
```

How do we interpret the coefficient for male?

1. remember that the regression line always passes through the mean values of our data: $Y = a + bX_1$
2. if we consider women only, then each woman has `male == 0`, which implies that predicted mean education for females $= a$.
3. if we consider men only, each man has `male == 1`, thus predicted mean education for males $= a + b$
4. So, the difference in predicted mean education for men and women is b.

Let's check that this is the case.

```
tab male if w1_r_b6 >= 18 & w1_r_b6 <= 80, sum(edyears)
```

From our table above, generate the difference in male/female education (use the `display` command and copy the values from the table). `display` can be abbreviated to `di`.

```
di 8.2970445 - 7.9991006
```

The difference is 0.2979439. How does this number compare to what we saw in the regression?

Multiple Regression

So far we have looked at two *simple* regressions, where a y-variable is predicted by one x-variable. But, it could be the case that a dependent variable is predicted by a many independent variables. For example, it could be that age and gender both play a role in predicting education.

So, let's regress years of education on age and gender.

```
reg edyears w1_r_b6 male if w1_r_b6 >= 18 & w1_r_b6 <= 80
```

Exercise 1

a. What do the coefficient and statistics for `w1_r_b_6` mean?

b. What do the coefficient and statistics for `male` mean?

c. What did the if statement do?

In Lab 2 we observed distinct *racial differences* in the graphs of years of completed education and age for both adults and children. Because this analysis involves many variables, we will use *multiple regression analysis* to try to understand the effects of the different variables. Before we do that, though, we're going to look at graphics interpreting the original regression output along with mean education for each of the race groups. Remember we computed `meaned` in Lab 2. Here are the commands in case you need to re-run them:

```
egen meaned = mean(edyears) , by(w1_r_b6 w1_best_race)
```

Let's revisit our simple regression model from earlier in this lab by looking at a plot of the *fitted* equation and the *mean* level of education at each age for Africans or whites separately:

```
twoway (line meaned w1_r_b6 if w1_best_race == 1 &
    w1_r_b6 >= 18 & w1_r_b6 <= 80 & number == 1, sort) (
    line meaned w1_r_b6 if w1_best_race == 4 &  w1_r_b6 >=
     18 & w1_r_b6 <= 80 & number == 1, sort) (line p0
    w1_r_b6 if w1_r_b6 >= 18 & w1_r_b6 <= 80 & number ==
    1, sort), legend(label(1 "Black African") label(2 "
    White"))
```

Notice that we used the legend option above to label the two different ethnicities to help us understand the fit of the model.

Exercise 2

a. Does the model fit the data well? (simple Yes/No).

b. Why did you say Yes/No above?

Figure 1 shows one fitted regression line and the different means determined by the data.

```
reg edyears w1_r_b6 african coloured indian if w1_r_b6 >=
     18 & w1_r_b6 <= 80
```

If you didn't use your data from last week, then you may need to generate the race dummy variables:

```
tab w1_best_race, gen(race)
rename race1 african
```

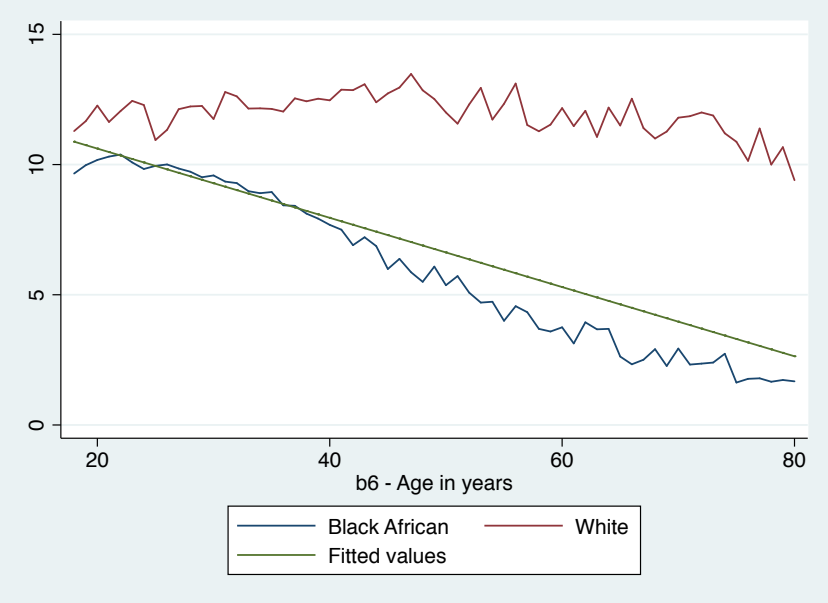

Figure 1: Here we have one fitted regression line, with a line for the mean education for each race group (Black and White). Does the fitted regression line (the predicted line) "fit" the data for White working-age South Africans well?

```
rename race2 coloured
rename race3 indian
rename race4 white
```

Having run the regression, let's look at a scatter plot for our new model. First predict the outcomes from the mode:

```
predict p1
```

Second, construct the graph for the two race groups for the working age population:

```
twoway (line meaned w1_r_b6 if w1_best_race ==1 & w1_r_b6
    >= 18 & w1_r_b6 <= 80 & number ==1, sort) (line
   meaned w1_r_b6 if w1_best_race == 4 &  w1_r_b6 >= 18 &
    w1_r_b6 <= 80 & number == 1, sort) (line p1 w1_r_b6
   if w1_best_race == 1 & w1_r_b6 >= 18 & w1_r_b6 <= 80 &
    number == 1, sort) (line p1 w1_r_b6 if w1_best_race
   == 4 & w1_r_b6 >= 18 & w1_r_b6 <= 80 & number == 1,
   sort)
```

Notice, that plot was not labeled in a way that was easy to make sense of, I've labeled it below:

```
twoway (line meaned w1_r_b6 if w1_best_race  == 1 &
   w1_r_b6 >= 18 & w1_r_b6 <= 80 & number == 1, sort) (
   line meaned w1_r_b6 if w1_best_race == 4 &  w1_r_b6 >=
    18 & w1_r_b6 <= 80 & number == 1, sort) (line p1
   w1_r_b6 if w1_best_race == 1 & w1_r_b6 >= 18 & w1_r_b6
```

```
 <= 80 & number == 1, sort) (line p1 w1_r_b6 if
w1_best_race == 4 & w1_r_b6 >= 18 & w1_r_b6 <= 80 &
number == 1, sort), legend(label(1 "Black African")
label(2 "White") label(3 "Black Fitted") label(4 "
White Fitted"))
```

The labels correspond to different positions, here, 1, 2, 3, and 4. You need to tell each label position what you want it to be.

If you want to export the graph to your current working directory, you can use the following command:

```
graph export race_fitted.pdf, replace
```

The command above exports the graph to .pdf format and saves it in the current working directory. It also *replaces* another file that might have had the same name.

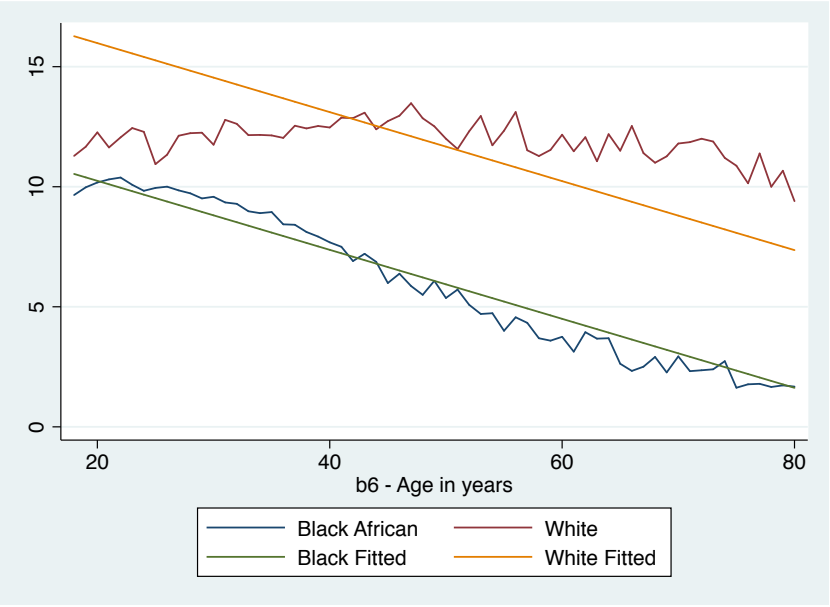

Figure 2: Here we have two fitted regression lines, with each dummy variable changing the *intercept* of the fitted regression line. Notice that the fitted lines are *parallel* because the linear relationship is assumed to be *identical* across the two groups.

Figure 2 shows parallel fitted regression lines (i.e. with the same slope) and different intercepts determined by the *dummy variables*.

Using xi for dummies

To introduce race into our model we needed to create three dummy variables. We can do this using `gen` and `replace` or `tab` with the `gen` option. There is a quicker way to introduce categorical variables into your regression model using the `xi:` command. The `xi:` command works by placing it at the *beginning* of the regression equation and then specifying the categorical variables you

want Stata to expand into its constituent categories by""tagging" them with an "i" in front of each target variable. The command:

```
xi: reg edyears w1_r_b6 i.w1_best_race if w1_r_b6 >= 18 &
    w1_r_b6 <= 80
```

produces the same results as:

```
reg edyears w1_r_b6 african coloured indian if w1_r_b6 >=
    18 & w1_r_b6 <= 80
```

The results are not exactly the same because when using `xi` Stata, by default, selects the *first* category in the specified variable as the *reference* category (or *omitted dummy*. In this instance Africans would be the reference category. If we wanted Whites to be the reference category we could preface the regress command with the char varname[omit] statement. To make category 4 (whites) of the race variable the reference category type:

```
char w1_best_race[omit] 4
xi: reg edyears w1_r_b6 i.w1_best_race if w1_r_b6 >= 18 &
    w1_r_b6 <= 80
```

The results should now be identical to those produced without the `xi` command.

Interaction terms and different slopes

The model above allows for different *intercepts* for each race but from the plot it seems that we should allow for different *slopes* for each race group. How do we allow different categories to have different slopes? We need to introduce *interaction terms* between race and age. An interaction term is a variable that tries to find whether there is a different slope by seeing whether the linear variable (e.g. age) and the category (e.g. race) *interact* independently of the other categories and the existing linear relationship. We could create an interaction for race and age for each race separately by doing the following

```
gen africanage = african*w1_r_b6
gen colouredage = coloured*w1_r_b6
gen indiannage = indian*w1_r_b6
```

and so on but it will be much easier to use the xi command.

```
char w1_best_race[omit] 4
xi: reg edyears i.w1_best_race*w1_r_b6 if w1_r_b6 >= 18 &
    w1_r_b6 <= 80
```

Now we will want to use the regression output to predict what the relationship should be and to graph the predicted relationships:

```
predict p2
```

```
twoway (line meaned w1_r_b6 if w1_best_race == 1 &
    w1_r_b6 >= 18 & w1_r_b6 <= 80 & number == 1, sort) (
    line meaned w1_r_b6 if w1_best_race == 4 & w1_r_b6 >=
    18 & w1_r_b6 <= 80 & number == 1,sort) (line p2
    w1_r_b6 if w1_best_race == 1 & w1_r_b6>=18 & w1_r_b6
    <= 80 & number == 1, sort) (line p2 w1_r_b6 if
    w1_best_race == 4 & w1_r_b6 >= 18 & w1_r_b6 <= 80 &
    number == 1, sort), legend(label(1 "Black African")
    label(2 "White") label(3 "Black Fitted") label(4 "
    White Fitted"))
```

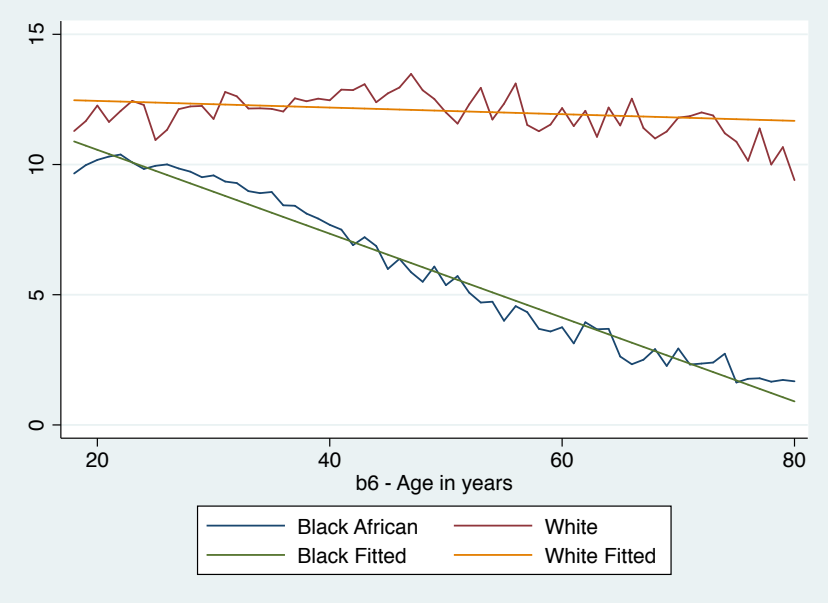

Figure 3: Here we have two fitted regression lines, with each dummy variable changing the *intercept* of the fitted regression line and a fitted *linear* relationship defined by the coefficient of the linear variable and the *interaction term*. Notice that the lines are *no longer parallel*, because the linear relationship is shown to *differ* across the two groups with the statistically significant interaction terms.

Figure 3 shows the fitted regression lines from the regressions showing the *different slopes* of the lines for the two ethnic groups along with the different *intercepts* determined by the dummy variables for each race group.

Regressions with other categorical variables

Let us consider some additional variables for our analysis about the years of education someone accumulates: whether there has been a death in the family in the past 24 months and whether the household has access to electricity.

```
gen death = w1_h_c1 == 1
replace death=. if w1_h_c1==.|w1_h_c1<0
```

The first command, `gen death = w1_h_c1 == 1` has the same effect as the following two commands.

```
gen death = 0
```

```
replace death = 1 if w1_h_c1 == 1
```

We could also have created our death variable using the `recode` command as follows.

```
recode w1_h_c1 (-3 = .) (2 = 0), gen(death)
```

Note that we were careful to exclude the missing observations. It would be incorrect to regard household with a missing answer on the question as *not* having experienced a death.

Exercise 3

a. Generate a dummy variable called `electricity` that is equal to 1 if the household is connected to the electricity supply, and 0 otherwise. Be sure to take missing data into account. (remember you can use `lookfor` to find variables with a word you're interested in).

b. Create a variable called `pcminc` for per capita household monthly income. Use the `w1_hhincome` and `w1_hhsizer` variables.

Now, before we use these variables in regression analysis, let's generate confidence intervals for the electricity, death, education and per capita household income variables.

```
ci electricity death pcminc edyears
```

Note, though, that these variables are *household-level* variables so we will want to generate frequency tables with only *one observation* per household, otherwise we'd be *over-counting* their incidence from large households

Recall in the previous lab how we used the `egen` command to create a dummy variable that was equal to one for only the first observation from each household.

```
egen hh_one = tag(hhid)
```

Exercise 4

a. Generate a confidence interval for the proportion of households that have electricity.

b. Show a frequency table of the number of households that have experienced a death in the past 24 months.

c. What is the mean household income for people who have 12 years of education and people who have 15 or 16 years of education?

Now, let's use per-capita income and electricity in a regression:

```
reg edyears w1_r_best_age pcminc electricity if
    w1_r_best_age>=7 & w1_r_best_age<=17
```

Can you intepret the coefficients and statistical significance of the coefficients on `pcminc` and `electricity`? Add `death` as an additional dummy variable. What does its coefficient and significance suggest?

Exporting Regression Output

We don't merely want to have the output in Stata. Rather, we would like to present the results in a paper using MSWord, GoogleDocs, LaTeXor a similar word processor. That means we want tables that we can easily import into one of these programmes.

Stata has packages that can do this for us: `outreg2` and `estout` produce publication-ready tables. Both of these packages are `ado` files. You can see the various locations Stata uses by typing

ADO FILES ado files are user-written do files that you can install from online repositories. Once installed, these commands are automatically loaded when you start Stata and you can use them like any other built-in Stata command.

```
sysdir
```

From which you should get output like:

```
STATA:    /Applications/Stata/
BASE:     /Applications/Stata/ado/base/
SITE:     /Applications/Stata/ado/site/
PLUS:     ~/Library/Application Support/Stata/ado/plus/
PERSONAL:    ~/Library/Application Support/Stata/ado/
    personal/
OLDPLACE:    ~/ado/
```

The following example shows you the normal way to install the outreg2.ado file. Due to the set up in the lab you may not be able to install ado files this way but it will work on your own computer.

```
net search outreg2
```

This will list several packages that relate to `outreg2`. Click on `outreg2` and follow the instructions to install the ado files. They will be installed in the `PERSONAL` directory.

Now we can use `outreg2` in the same way that we use any built-in Stata command. To see what`outreg2` does type the following:

```
reg edyears w1_r_best_age if w1_r_best_age >= 7 &
    w1_r_best_age <= 17
outreg2 using lab3.xls, bdec(3) excel replace
```

You can now navigate to your current working directory (Lab 3, I hope) and open the file lab3.xls. You will see that `outreg2` has exported the main results of the regression to the Excel file. Additional regressions can be added as columns to the file as we go (as you will have seen in research articles). We can add columns by specifying the option `append`:

```
reg edyears w1_r_best_age pcminc electricity if
    w1_r_best_age >= 7 & w1_r_best_age <= 17
```

```
outreg2 using lab3.xls, bdec(3) excel append
```

The outreg2 command has lots of options that allow you to tailor the regression output to your tastes. To explore these options type `help outreg2`. A really useful option allows you to put the regression output into a word document by specifying the file extension as .doc. For example,

```
reg edyears w1_r_best_age if w1_r_best_age >= 7 &
    w1_r_best_age <= 17
outreg2 using lab3.doc, bdec(3) replace
reg edyears w1_r_best_age pcminc electricity if
    w1_r_best_age >= 7 & w1_r_best_age <= 17
outreg2 using lab3.doc, bdec(3) append
```

Another useful option is `label`. If you have been careful to label each of the variables you have generated, then outreg2 will use these labels rather than only the variable names in Stata, e.g. it will use the label "Years of education" rather than the Stata variable name "edyears." See the same word document as above, but with the option label specified each time in outreg2.

REMINDER You can label variables by using the command `label variable variablename "Label for Variable"`. Remember too that you can abbreviate `label variable` to `lab var`.

```
reg edyears w1_r_best_age if w1_r_best_age >= 7 &
    w1_r_best_age <= 17
outreg2 using lab3.doc, bdec(3) replace label
reg edyears w1_r_best_age pcminc electricity if
    w1_r_best_age >= 7 & w1_r_best_age <= 17
outreg2 using lab3.doc, bdec(3) append label
```

If you are accustomed to use LaTeX, then you can also export to .tex.

Another option: estout

As is often the case with Stata, there is more than one way to easily manage your regression output. An example of another command that creates excel tables from your regression results is shown below. You will need to install the estout ado files by typing

```
net search estout
```

Once you have installed `estout`, run the code below and then use help to explore some of the options.

```
reg edyears w1_r_best_age if w1_r_best_age >= 7 &
    w1_r_best_age <= 17
estimates store m1
reg edyears w1_r_best_age pcminc electricity if
    w1_r_best_age>=7&w1_r_best_age<=17
estimates store m2
estout * using lab3b.xls
```

Each time you use the `estout` command, you first 'store' the results, then you use `estout` to export them to the kind of output you prefer.

Also, you can check a tutorial about estout here: http://www.ats.ucla.edu/stat/stata/faq/estout.htm.

And another: putexcel

Recently, the World Bank and various other institutions have started to use a new command: `putexcel` (which has been incorporated into the core installation of Stata 13.1 and 14). With `putexcel`, the coder has a lot more control over where each item goes. We shan't go through the use of `putexcel` here for the moment, but it is worth investigating if you want to have more control over what you export to MS Excel, e.g. very specific summary statistics or the results from confidence intervals, and what the table itself looks like.

Here are three useful links to learn about `putexcel`:

- Stata.com: http://www.stata.com/stata-news/news29-1/export-tables-to-excel/
- Stata video tutorial: https://www.youtube.com/watch?v=MUQ3E8hIQZE
- A Prezi on using `putexcel` https://prezi.com/helkgmnounx0/putexcel-/

The `putexcel` command is incorporated into Stata itself. You can export to Excel with the drop-down menus:

```
File > Export > Results to Excel spreadsheet (*.xls;*.
    xlsx)
```

Select solutions to coding exercises

1. No solutions for this lab yet.

Commands and options in Lab 3

reg	predict	twoway lfitci
xi: reg	char ... [omit]	ci
sys dir	net search	outreg2 using filename.extension
bdec(3)	excel	replace
append		label
estout	estimates store	putexcel
twoway, legend label		