Week 1: Stata and regression models

Marcelo Coca Perraillon

University of Colorado
Anschutz Medical Campus

Health Services Research Methods I
HSMP 7607
2020

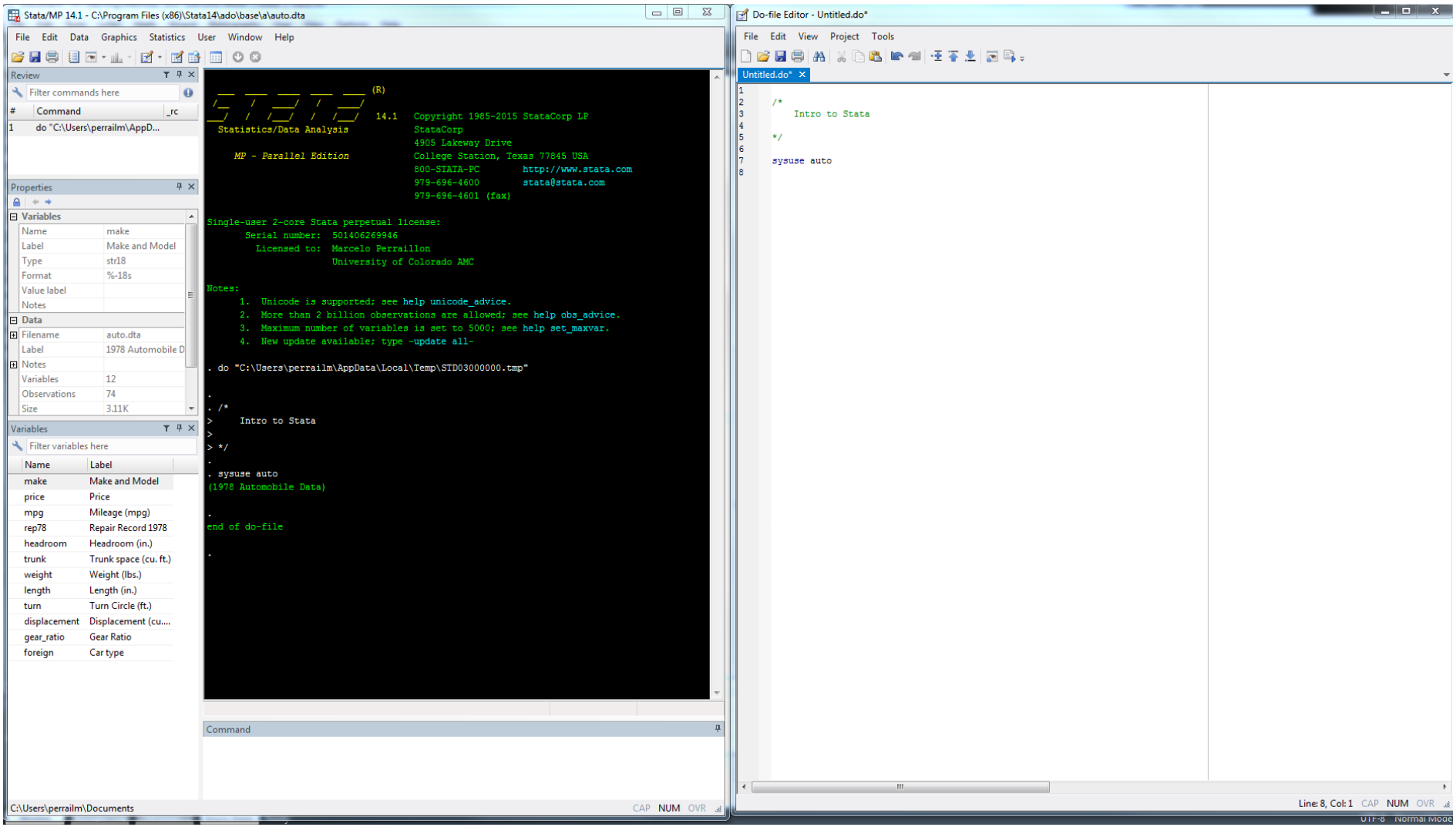
Outline

- Stata basics
- Stata commands
- Exploring a dataset
- Running linear/OLS models
- Review of OLS properties
- Comparing models
- Model effects

The big picture

- Today's class serves two purposes: a basic introduction to Stata and a review of linear/OLS models
- Read chapters 1 and 2 of Cameron and Trivedi for more features
- Review online tutorials on Stata. See cheat sheet on Canvas
- I'll introduce more commands and tricks during the semester as part of the lecture examples and homeworks
- **I'll always answer Stata questions for homeworks so email me when in doubt**

A good way of working with Stata (if you have a large monitor)



Interacting with Stata

- You can enter code in interactive mode or the command window
- The command window is Useful for quick checks and to get help but as a rule, **don't use it**. Use a "do" file
- Do files are text files with a ".do" extension: a collection of Stata code and note
- **Always comment your code**. Comments should be 50% to 70% of your code. You future self will thank you
- **Always** write a "do file" with comments to preserve your work. Select the text in the do-file editor and press **Control+D** to run the code (or use the menu)
- You can use DOS or Unix commands like: pwd, ls, cd, dir, cls
- Today's do file file is called **Week 1 Stata.do**

Importing data and syntax structure

- All homeworks and examples will use data in Stata format (extension .dta).
- Stata has many ways of importing data. Type "help import"
- **Stata documentation is extensive and outstanding**. You can access the PDF documentation by clicking on the blue text (for example, [D] Import)
- Or by using the menu: Help and then click on PDF documentation
- Stata syntax is **consistent**
- In general there is a command name (or two names) followed by selection of variables and then a comma followed by options (sometimes there is a prefix before the command)
- For example: `help tabulate oneway` or `bysort oldid: egen newid = group(oldid)`
- All commands have a shortcut: tabulate same as **tab**, summarize same as **sum** (see underlined text in the help file)

Today's data

- We will use data from the Medical Expenditure Panel Survey (MEPS)
- The MEPS is a survey of families and individuals, their medical providers, and employers across the United States. It covers the non-institutionalized population in the US
- Great source of data on the cost and use of health care and health insurance coverage
- Public files can be downloaded for free. Data available from 1996 (most current year is 2019)
- You can get more data through the Rocky Mountain Data Center `https://www.colorado.edu/rocky-mountain-research-data-center/`
- See `https://www.meps.ahrq.gov/mepsweb/`
- We will use an extract that comes with Deb, Norton, and Manning (2017) and another from Cameron and Trivedi (2010)

Today's "do" file

```
/*
   HSR Method I - Week 1 intro to Stata and regression  review
*/
cd "H:\Teaching\Methods 2020\lectures\Week 1 Overview and Stata\code"
log using "Week 1 Stata", text replace
*** Load data from DNM
use http://www.stata-press.com/data/heus/heus_mepssample

// ---- Explore the data
desc
* Variables we will use today
lookfor ed
codebook exp_tot age female race_*

* Explore variables
sum exp_tot age female race_*
tabstat exp_tot age female race_*, stats(mean sd median min max)
tabstat exp_tot age female race_*, stats(mean sd median min max) columns(statistics
hist exp_tot
```

Organize your work!

- Something you don't learn in graduate school –from a class – is proper organization and documentation of your code and projects
- Should your dissertation be a one big do file? In which folder should you save results versus code? Several? How should you name your variables? Should you use labels? Should you copy-paste results or should you have an automated way to ensure you can reproduce your research? Do you remember in which file is that p-value you mention in page 10? More importantly, are you going to **remember two years later** when a reviewer ask you to do a supplementary analysis?
- Stata has many features to help with that. The editor has projects. You can add notes to datasets, variables (type `help notes`). You can create tables from within Stata so you don't rely on copy-paste (see commands `esttab`, `outreg2`, `putexcel`, `putdoc`)
- I recommend this book (somewhat out-of-date): Long, J. The **workflow** of data analysis using Stata. College Station, TX: Stata Press, 2009.

Explore, explore, explore, explore

- **Don't even think about running a model until you understand the dataset**
  1. List all variables using the command `describe`
  2. Understand more about the storage type by using the command `coodebook`
  3. Use the data editor/browser to actually see the data. Sometimes it helps to change the order: `order foregin` (now variable foreign is the first variable in the dataset)
  4. Check for **missing values** (commands `misstable, tabmiss`). Are missing values coded -99, 99, NA, or .? **In Stata missing values are** $+\infty$ (!!) (In SAS, $-\infty$)
  5. Calculate basic statistics for all the variables you will use: commands `summarize, tabulate, tabstat`
  6. Explore two- or three-way relationships between variables: commands `graph twoway, histogram, lowess, tabulate`
- Be careful about how the data are stored versus how they are displayed. Variables can have labels; values of variables can also have labels

Check data

```
desc exp_tot age female race_*

              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------------------
exp_tot         long    %12.0g                Total medical care expenses
age             byte    %8.0g                 Age
female          byte    %9.0g      lb_female
                                              Female
race_bl         byte    %14.0g     lb_race_bl
                                              Black
race_oth        byte    %14.0g     lb_race_oth
                                             Other race, non-white and non-black

codebook age
-------------------------------------------------------------------------------------------------------------
age
-------------------------------------------------------------------------------------------------------------
                  type:  numeric (byte)
                 range:  [18,85]                      units:  1
         unique values:  68                       missing .:  0/19,386
                  mean:   45.3609
             std. dev:    17.387
           percentiles:        10%       25%       50%       75%       90%
                                23        31        44        57        72
```

Some stats

```
. sum exp_tot age female race_*
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     exp_tot |     19,386     3685.25    9768.475           0     440524
         age |     19,386    45.36088      17.387          18         85
      female |     19,386    .5495719    .4975494           0          1
     race_bl |     19,386    .1382441    .3451649           0          1
    race_oth |     19,386    .0653564    .2471601           0          1

. tabstat exp_tot age female race_*, stats(mean sd median min max)
   stats |   exp_tot       age    female   race_bl  race_oth
---------+------------------------------------------------
    mean |   3685.25  45.36088  .5495719  .1382441  .0653564
      sd |  9768.475    17.387  .4975494  .3451649  .2471601
     p50 |       952        44         1         0         0
     min |         0        18         0         0         0
     max |    440524        85         1         1
```

Stats plus macros

```
* Stata has macros
global tsopts stats(mean median sd min max) columns(statistics)

tabstat exp_tot if race_bl==1, by(female) $tsopts

* same as tabstat exp_tot if race_bl==1, by(female) stats(mean median sd min max) columns(statistics)

Summary for variables: exp_tot
     by categories of: female (Female)
female |      mean       p50        sd       min       max
-------+--------------------------------------------------
  Male |  2741.284       291  8038.086         0    108606
Female |  3821.302      1064  7914.097         0    104467
-------+--------------------------------------------------
 Total |  3401.788       682  7978.381         0    108606
----------------------------------------------------------

tabstat exp_tot if race_bl==0, by(female) $tsopts
Summary for variables: exp_tot
     by categories of: female (Female)
female |      mean       p50        sd       min       max
-------+--------------------------------------------------
  Male |  3148.219       554  10595.89         0    440524
Female |  4227.677      1403  9484.035         0    236749
-------+--------------------------------------------------
 Total |  3730.723       996   10025.4         0    440524
----------------------------------------------------------
```

Linear/OLS regression

- Last class we saw that you can estimate **descriptive**, **causal**, or **predictive** models
- We could replicate the tables above using a regression model. The model is:

  $y_i = \beta_0 + \beta_1 female_i + \beta_2 Black_i + \beta_3 female * Black_i + \epsilon_i$
- That's the standard way of writing the *population* model. But it helps to write the estimated model:
- $E[y_i|\boldsymbol{x}_i] = \hat{\beta}_0 + \hat{\beta}_1 female_i + \hat{\beta}_2 Black_i + \hat{\beta}_3 female * Black_i$
- So the model is a **conditional expectation function**. What is the average health care expenditure in 2004 for Black male?:
  $E[y_i|\boldsymbol{x}_i] = \hat{\beta}_0 + \hat{\beta}_2$
- **Digression**: That's an example of a **saturated model**: The model includes a parameter for each possible value the outcome could take. Note that if we included age, we would need 67 dummies because there are 68 unique ages in the dataset – and we would their interactions (`codebook age`)

Verify (note factor notation)

```
. reg exp_tot i.female i.race_bl i.female#i.race_bl
      Source |       SS           df       MS      Number of obs   =    19,386
-------------+----------------------------------   F(3, 19382)     =     20.42
       Model |  5.8285e+09         3  1.9428e+09   Prob > F        =    0.0000
    Residual |  1.8439e+12    19,382  95137150.3   R-squared       =    0.0032
-------------+----------------------------------   Adj R-squared   =    0.0030
       Total |  1.8498e+12    19,385  95423097.6   Root MSE        =    9753.8

------------------------------------------------------------------------------
           exp_tot |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------------+----------------------------------------------------------------
            female |
          Female  |   1079.458   151.4039     7.13   0.000     782.6931    1376.222
                  |
           race_bl |
       Black race  |  -406.9347   322.1181    -1.26   0.206    -1038.314    224.4446
                  |
    female#race_bl |
Female#Black race  |   .5598868   415.1616     0.00   0.999    -813.1927    814.3125
                  |
             _cons |   3148.219   111.2202    28.31   0.000     2930.218     3366.22
------------------------------------------------------------------------------

. di _b[_cons] +  _b[1.race_bl]
2741.2843
```

Stata saves information after estimation

```
.ereturn list
scalars:
                  e(N) =  19386
               e(df_m) =  3
               e(df_r) =  19382
                  e(F) =  20.42139562138809
< output omitted>
matrices:
                  e(b) :  1 x 9
                  e(V) :  9 x 9
functions:
             e(sample)
. matrix list e(b)
e(b)[1,9]
            0b.          1.         0b.          1.  0b.female#  0b.female#  1o.female#   1.female#
        female      female     race_bl     race_bl  0b.race_bl  1o.race_bl  0b.race_bl   1.race_bl
y1           0   1079.4578           0  -406.93475           0           0           0   .55988681
         _cons
y1   3148.2191
```

- Stata stores results according to the type of command. Use `return list` or `ereturn` (there is also `sreturn list`)
- For more, see

  `https://stats.idre.ucla.edu/stata/faq/how-can-i-access-information-stored-after-i-run-a-command-in-s`

Defining new variables

- You create new variables with `generate`, `replace` or other commands

```
gen      older65 = 0
replace  older65 = 1 if age > 65 & age ~=.
* Don't
generate older65_1 = (age > 65)
* Yes
generate older65_2 = (age > 65) if !missing(age)
sum older65*

   Variable |        Obs        Mean    Std. Dev.       Min        Max
------------+---------------------------------------------------------
     older65 |     19,386    .1512947    .3583452          0          1
   older65_1 |     19,386    .1512947    .3583452          0          1
   older65_2 |     19,386    .1512947    .3583452          0          1

* Quartiles
xtile ageq = age, n(4)
tabstat age, by(ageq) stats(mean sd median min max)

Summary for variables: age
     by categories of: ageq (4 quantiles of age)
    ageq |      mean        sd       p50       min       max
---------+--------------------------------------------------
       1 |  24.64043   4.00162        25        18        31
       2 |  38.02606  3.717267        38        32        44
       3 |  50.63896  3.753583        50        45        57
       4 |  69.71156  8.253133        69        58        85
---------+--------------------------------------------------
   Total |  45.36088    17.387        44        18        85
------------------------------------------------------------
```

Review: Algebraic properties of linear/OLS regression

- Wooldridge covers three "algebraic" properties of linear/OLS models
  - The sum of the residuals is zero: $\sum_{i=1}^{n} \hat{\epsilon_i} = 0$ (**Confusion alert**: ϵ is the error term; $\hat{\epsilon}$ is the residual)
  - The sample covariance between the residuals and the regressor(s) x is zero
  - The means of Y and X are always in the regression line
- All these facts follow from the first-order conditions (see Wooldridge Chapter 2) and `https://perraillon.com/s/week-4-SLR-II.pdf`
- The first one is important to understand "regression towards the mean." It states that on average, we do not make prediction mistakes. OLS chooses β that makes this true
- The second can be confusing if you understand causality using the **conditional independence assumption** (economists in the room). Conditional independence is about the error term ϵ in the **population**; the second algebraic property is about the residual $\hat{\epsilon}$ **in any sample**

Verify

```
. qui reg exp_tot i.female i.race_bl i.female#i.race_bl
* same as reg exp_tot i.female##i.race_bl

. predict yhat if e(sample)
(option xb assumed; fitted values)

. predict resis, res

. sum exp_tot resis yhat

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     exp_tot |     19,386     3685.25    9768.475          0     440524
       resis |     19,386    .0000566    9753.073  -4227.677   437375.8
        yhat |     19,386     3685.25    548.3344   2741.284   4227.677

.
. corr resis female race_bl
(obs=19,386)

             |    resis   female  race_bl
-------------+---------------------------
       resis |   1.0000
      female |   0.0000   1.0000
     race_bl |  -0.0000   0.0499   1.0000
```

- Note that the mean of the outcome and the mean of the the predicted value is the same. But note that the standard deviation of the prediction is a lot smaller. Most predictions tend towards the mean. Type `help factor variables`

Easy to compare models with Stata

```
* Let's add age as a continuous variable
qui reg exp_tot age i.female
est sto m1
* Quadratic
qui reg exp_tot c.age##c.age i.female
est sto m2
. est table m1 m2, star stats(N r2 r2_a bic)

----------------------------------------------
    Variable |      m1              m2
-------------+--------------------------------
         age |  129.22625***   -21.968896
             |
      female |
     Female  |   889.4179***    859.15941***
             |
 c.age#c.age |                  1.5453143***
             |
       _cons | -2665.3656***    562.81848
-------------+--------------------------------
           N |      19386           19386
          r2 |  .05576973       .05839129
        r2_a |   .0556723       .05824555
         bic |  410126.32        410082.3
----------------------------------------------
      legend: * p<0.05; ** p<0.01; *** p<0.001
```

- Note that the mean of the outcome and the mean of the the predicted value is the same. But note that the standard deviation of the prediction is a lot smaller. Most predictions tend towards the mean

Interpretation

- **Always write down your models** (!!)
- Our model (following the table in previous slides) is now (dropping the "hat" on the betas to make it easier to write):
- $E[y_i|\boldsymbol{x}_i] = \beta_0 + \beta_1 age_i + \beta_2 female_i + \beta_3 age_i^2$
- The effect of age is $\frac{\partial E[y_i|\boldsymbol{x}_i]}{\partial age} = \beta_1 + 2 * \beta_3 age$
- **So not a single effect of age**. The effect of age depends on the value of age
- At 30, the effect of age is $\beta_1 + 2 * \beta_3 * 30$
- **Marginal effects** in Stata make this calculation easy as we will see. Essentially, Stata computes a **numerical derivative** rather than an **analytical derivative** as above

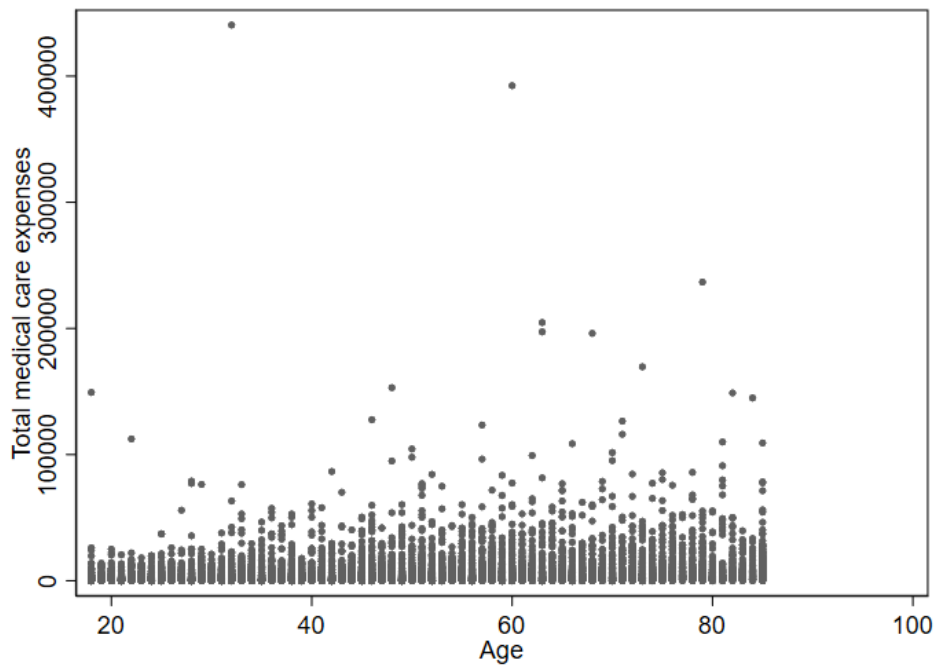
Effects "by hand" and with the **margins** command

```
qui reg exp_tot c.age##c.age i.female
matrix list e(b)
<output omitted>

di  _b[age] + 2*_b[c.age#c.age]*30
70.749963
margins, dydx(age) at(age=(20 30 40 50 60 70 80 90)) vsquish
Average marginal effects                        Number of obs     =     19,386
Model VCE    : OLS

Expression   : Linear prediction, predict()
dy/dx w.r.t. : age
1._at        : age             =          20
<output omitted>
------------------------------------------------------------------------------
             |            Delta-method
             |      dy/dx   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
age          |
         _at |
          1  |   39.84368   12.78323     3.12   0.002     14.78745    64.89991
          2  |   70.74996   8.872776     7.97   0.000     53.35856    88.14137
          3  |   101.6562   5.426229    18.73   0.000     91.02037    112.2921
          4  |   132.5625   3.945165    33.60   0.000     124.8297    140.2954
          5  |   163.4688   6.089931    26.84   0.000      151.532    175.4056
          6  |   194.3751   9.696018    20.05   0.000     175.3701    213.3801
          7  |   225.2814   13.65067    16.50   0.000     198.5249    252.0379
          8  |   256.1877   17.72206    14.46   0.000     221.4509    290.9245
------------------------------------------------------------------------------
```

Graphs

- Stata has great graphical capabilities
- Implemented with the command `graph`. We will often use `graph twoway` to make scatter plots, line plots and more
- You can use shortcuts. Instead of `graph twoway scatter` you can just write `scatter`

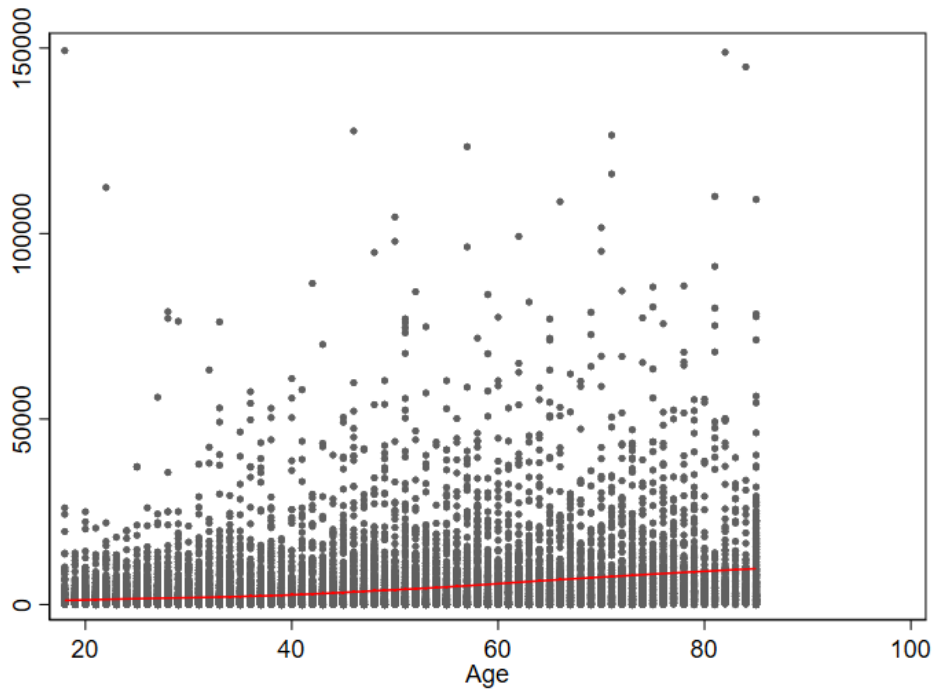
```
scatter exp_tot age, msize(vsmall)
graph export exp_age.png, replace
```



Smoothing

- "Smoothing" techniques are great to see trends. See Cameron and Trivedi section 2.6.6 and Stata help for the commands `lowess`, `lpoly`
- We will use `lowess` to have a sense of the relationship between expenditures and age

```
lowess exp_tot age, gen(y_smooth) nograph
scatter exp_tot age if exp_tot <=150000, msize(vsmall)  ///
   || line y_smooth age if exp_tot <=150000, sort color(red) legend(off)
graph export low.png, replace
```



More on smoothing

- Smoothing techniques are exploratory tools that are handy for exploring relationshipts. They are **exploratory** though
- We will mostly use **kdensity** (kernel density), **lowess** ((Locally Weighted Scatterplot Smoothing), and **lpoly** (local polynomial regression kernel-weighted)
- These type of method are called semiparametric since you do estimate parametric models but use them to obtain non-parametric estimates of interest
- See Cameron and Trivedi Chapter 2

Other helpful features

- You can create and recode variables with `generate, egenerate, replace, recode`
- As with any programming language, you can do loops: `foreach, forvalues`
- Many, many tools to manipulate data: `append, merge, reshape, xpose, collapse`
- Tons of tools to manipulate strings and regular expressions: `strmatch, substr, regexm, trim`
- Great graphical capabilities: command `graph`
- We won't cover some of these features in the semester but it's worth learning about them

Summary

- We will review the issues covered today many times during the semester
- But do go over the readings. Make sure you remember key results from linear and logistic regression. My notes could help: `https://perraillon.com/hsr-methods-old`
- **Review today's do file**. I did more things that because of time I couldn't cover today
- We will use the linear model in situations in which we know that is the wrong model because we know that it's still modeling the unbiased conditional expectation
- In some situations, however, running the correct model could result in **different effects** and conclusions. In the linear model, effects are additive. **In non-linear model, the values of all the covariates in the model affect the effect of one variable on the outcome**
- But sill, the linear/OLS is an incredible useful tool