PLS 802 (2001, Spring)

Stata Learning Module: A Sample Stata Session

This is from the *Getting Started with Stata for Windows* manual.

For this class we will use **auto.dta** shipped with Stata. If you wish to follow along, you must load this data. Launch Stata and choose **Open** from the **File** menu. Select the **auto.dta** file from the directory in which you installed Stata.

use c:\stata\auto, clear
(1978 Automobile Data)

The data that we loaded contains

```
. describe

Contains data from c:\stata\auto.dta
  obs:            74                          1978 Automobile Data
 vars:            12                          11 Sep 1998 10:08
 size:         3,478 (99.6% of memory free)
-------------------------------------------------------------------------------
   1. make         str18  %-18s                Make and Model
   2. price        int    %8.0gc               Price
   3. mpg          int    %8.0g                Mileage (mpg)
   4. rep78        int    %8.0g                Repair Record 1978
   5. hdroom       float  %6.1f                Headroom (in.)
   6. trunk        int    %8.0g                Trunk space (cu. ft.)
   7. weight       int    %8.0gc               Weight (lbs.)
   8. length       int    %8.0g                Length (in.)
   9. turn         int    %8.0g                Turn Circle (ft.)
  10. displ        int    %8.0g                Displacement (cu. in.)
  11. gratio       float  %6.2f                Gear Ratio
  12. foreign      byte   %8.0g       origin   Car type
-------------------------------------------------------------------------------
Sorted by:  foreign
```

The codebook command is a great tool for getting a quick overview of the variables in the data file. It produces a kind of electronic codebook from the data file. Have a look at what it produces below.

. codebook

Another useful command for getting a quick overview of a data file is the inspect command. Here is what the inspect command produces for the auto data file.

. inspect

Listing can be informative

The list command is useful for viewing observations. Here we look at **make mpg** for the first 10 observations.

. list make mpg in 1/10

```
    make                              mpg
 1. AMC Concord                       22
 2. AMC Pacer                         17
 3. AMC Spirit                        22
 4. Buick Century                     20
 5. Buick Electra                     15
 6. Buick LeSabre                     18
 7. Buick Opel                        26
 8. Buick Regal                       20
 9. Buick Riviera                     16
10. Buick Skylark                     19
```

. sort mpg
. list make mpg in 1/5

```
    make                            mpg
 1. Linc. Continental               12
 2. Linc. Mark V                    12
 3. Linc. Versailles                14
 4. Merc. XR-7                      14
 5. Cad. Deville                    14
```

Which 5 cars yield the highest gas mileage?

. list make mpg in -5/-1

```
    make                             mpg
70. Toyota Corolla                   31
71. Plym. Champ                      34
72. Subaru                           35
73. Datsun 210                       35
74. VW Diesel                        41
```

< Descriptive statistics >

Generating Summary Statistics with summarize

For summary statistics, we can use the **summarize** command.

Question: Not being familiar with 1978 prices, what is the average price of a car in this data?

. summarize
. summarize price

```
Variable |     Obs        Mean   Std. Dev.       Min        Max
---------+-----------------------------------------------------
   price |      74    6165.257   2949.496       3291      15906
```

Aside: **summarize** works like **list** without arguments it provides a summary of all of the data:

Question: what is the median MPG?

We can use the **detail** option (of the **summarize** command) to get more detailed summary statistics.

. summarize mpg, detail

```
                              mpg
-------------------------------------------------------------
      Percentiles      Smallest
 1%           12             12
 5%           14             12
10%           14             14       Obs                  74
25%           18             14       Sum of Wgt.          74
50%           20                      Mean            21.2973
                        Largest       Std. Dev.      5.785503
75%           25             34
90%           29             35       Variance       33.47205
95%           34             35       Skewness       .9487176
99%           41             41       Kurtosis       3.975005
```

Question: What is the average price of cars that are below and above the mean MPG?

. summarize price if mpg < 21.3

```
Variable |     Obs          Mean   Std. Dev.        Min         Max
---------+-----------------------------------------------------
   price |      43       7091.86   3425.019        3291       15906
```

. summarize price if mpg >= 21.3

```
Variable |     Obs          Mean   Std. Dev.        Min         Max
---------+-----------------------------------------------------
   price |      31      4879.968   1344.659        3299        9735
```

Aside: **if** can be suffixed to any command. This is one of Stata's more useful features.

Descriptive statistics, making tables

The tabulate command is useful for obtaining frequency tables.

Problem: Obtain counts of the number of domestic and foreign cars.

. tabulate foreign

```
    Car type|      Freq.     Percent        Cum.
------------+-----------------------------------
   Domestic |         52       70.27       70.27
    Foreign |         22       29.73      100.00
------------+-----------------------------------
      Total |         74      100.00
```

The **tab1** command can be used as a shortcut to request tables for a series of variables (instead of typing the **tabulate** command over and over again).

. tab1 rep78 foreign

```
-> tabulation of rep78

      rep78 |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |          2        2.90        2.90
          2 |          8       11.59       14.49
          3 |         30       43.48       57.97
          4 |         18       26.09       84.06
          5 |         11       15.94      100.00
------------+-----------------------------------
      Total |         69      100.00

-> tabulation of foreign

    foreign |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |         52       70.27       70.27
          1 |         22       29.73      100.00
------------+-----------------------------------
      Total |         74      100.00
```

To get mpg value separately for foreign and domestic, we could use the **summarize( )** option as part of the tabulate command.

. tabulate foreign, summarize(mpg)

```
            |          Summary of mpg
    foreign |        Mean    Std. Dev.       Freq.
------------+------------------------------------
          0 |   19.826923    4.7432972          52
          1 |   24.772727    6.6111869          22
------------+------------------------------------
      Total |   21.297297    5.7855032          74
```

Descriptive statistics, correlation matrices

We can use the **correlate** command to get the correlatoins among variables. Let's look at the correlations among **mpg** and **weight**

Question: What is the correlation between MPG and weight of car?

. correlate mpg weight

```
(obs=74)

        |      mpg   weight
--------+------------------
     mpg|   1.0000
  weight|  -0.8072   1.0000
```

Problem: Compare the correlation for domestic and foreign cars.

. correlate mpg weight if foreign==0

```
(obs=52)

        |      mpg   weight
--------+------------------
     mpg|   1.0000
  weight|  -0.8759   1.0000
```

. correlate mpg weight if foreign==1
(obs=22)

```
        |      mpg   weight
--------+------------------
     mpg|   1.0000
  weight|  -0.6829   1.0000
```

Note: We could have obtained this by typing **by foreign: correlate mpg** instead.

Descriptive statistics, correlation matrices, continued

Aside: We can produce correlation matrices containing as many variables as we wish.

```
. correlate mpg weight price weight length displ
(obs=74)

        |      mpg   weight    price   weight   length    displ
--------+------------------------------------------------------
     mpg|   1.0000
  weight|  -0.8072   1.0000
   price|  -0.4686   0.5386   1.0000
  weight|  -0.8072   1.0000   0.5386   1.0000
  length|  -0.7958   0.9460   0.4318   0.9460   1.0000
   displ|  -0.7056   0.8949   0.4949   0.8949   0.8351   1.0000
```

Graphing data

Problem: We know the average MPG of domestic and foreign cars differs. We have learned that domestic and foreign cars differ in other ways as well, such as in frequency-of-repair record. We found a negative correlation of MPG and weight—as we would expect—but the correlation appears stronger for domestic cars. Examine, with an eye toward modeling, the relationship between MPG and weight. Begin with a graph.

. graph mpg weight

Typing **graph** *y x* draws a graph of *y* against *x*. The relationship, we note, is nonlinear.

Note: When you draw a graph, the Graph window appears, probably covering up your Results window. Click on the **Results** button to put your Results windows back on top. Want to see the graph again? Click on the **Graph** button.

Next, we draw separate graphs for foreign and domestic cars.

. sort foreign
. graph mpg weight, by(foreign) total

Syntax note: **by()** is on the right of the command, therefore **graph** did whatever it is that it does with the grouping information. What **graph** did is draw separate graphs for domestic and foreign cars in a single image. We have only two groups, but **graph** will allow any number—the individual graphs just get smaller. The **total** option added an overall graph to the image.

If we had placed the **by** in front, **by foreign: graph mpg weight** we would have obtained separate graphs on separate screens for each value of foreign.

Analysis note: The relationship is not only nonlinear; the domestic-car relationship appears to differ from that of foreign cars.

Model estimation: linear regression

Restatement of problem: We are to model the relationship between MPG and weight.

Plan of attack: Based on the graphs, we judge the relationship nonlinear and will model MPG as a quadratic in weight. Also based on the graphs, we judge the relationship to be different for domestic and foreign cars. We will include an indicator (dummy) variable for foreign and evaluate afterwards whether this adequately describes the difference. Thus, we will estimate the model:

$$mpg = b_0 + b_1*weight + b_2*weight^2 + b_3*foreign + e$$

foreign is already a 0/1 variable, so we only need to create the weight-squared variable:

. gen wtsq = weight^2

. regress mpg weight wtsq foreign

```
  Source |       SS       df       MS                  Number of obs =      74
---------+------------------------------               F(  3,    70) =   52.25
   Model |  1689.15372     3   563.05124               Prob > F      =  0.0000
Residual |   754.30574    70  10.7757963               R-squared     =  0.6913
---------+------------------------------               Adj R-squared =  0.6781
   Total |  2443.45946    73  33.4720474               Root MSE      =  3.2827

------------------------------------------------------------------------------
     mpg |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
  weight |  -.0165729   .0039692     -4.175   0.000      -.0244892   -.0086567
    wtsq |   1.59e-06   6.25e-07      2.546   0.013       3.45e-07    2.84e-06
 foreign |    -2.2035   1.059246     -2.080   0.041        -4.3161   -.0909003
   _cons |   56.53884   6.197383      9.123   0.000       44.17855    68.89913
------------------------------------------------------------------------------
```

Model estimation: linear regression, continued

Aside: Stata can estimate many kinds of models, including logistic regression, Cox proportional hazards, etc. Click on **Help**, choose **Search...**, and enter **estimation** for a complete list or look up estimation in the index of the *Stata Reference Manual*.

We interrupt this quotation to let you try **search estimation** for yourself.

Continuation of attack: We obtain the predicted values:

. predict mpghat

Comment: Be sure to read [U] **23 Estimation and post-estimation commands**. There are a number of features available to you after estimation—one is calculation of predicted values. **predict** just created a new variable called **mpghat** equal to

.0165729**weight** + 1.59*10^-6**wtsq** - 2.2035**foreign** + 56.53884

Model estimation: linear regression, continued

We can now graph the data and the predicted curve.

Continuation of attack: We just created **mpghat** with **predict**. We could graph the fit and data, but we want to evaluate the fit on the foreign and domestic data separately to determine if our shift parameter is adequate. Thus, we will draw the graphs separately:

```
. sort weight
. graph mpg mpghat weight if foreign==0, connect(.l) symbol(Oi)

. graph mpg mpghat weight if foreign==1, connect(.l) symbol(Oi)
```

graph mpg mpghat weight says to graph **mpg** vs. **weight** and **mpghat** vs. **weight**.

connect(.l) says do not connect the **mpg** vs. **weight** points—that is the '**.**'—but do connect (with straight lines) the **mpghat** vs. **weight** points—that is the '**l**' (*el*). It is necessary to sort the data by the *x*-variable—in this case weight—before graphing so that the points are connected in the right order.

symbol(Oi) says use big circles for the **mpg** vs. **weight** points—that is the '**O**' (capital "oh", not a zero)—but use the invisible symbol (no symbol at all) for the **mpghat** vs. **weight** points—that is the '**i**'.

Model estimation: linear regression, continued

Problem: You show your results to an engineer. "No," he says. "It should take twice as much energy to move 2,000 pounds 1 mile compared to moving 1,000 pounds, and therefore twice as much gasoline. Miles per gallon is not a quadratic in weight, gallons per mile is a linear function of weight."

You go back to the computer:

```
. gen gpm = 1/mpg
. label var gpm "Gallons per mile"
. sort foreign
. graph gpm weight, by(foreign) total
```

. regress gpm weight foreign

```
 Source |       SS       df       MS                  Number of obs =      74
---------+------------------------------               F(  2,    71) =  113.97
   Model |  .009117618     2  .004558809               Prob > F      =  0.0000
Residual |   .00284001    71      .00004               R-squared     =  0.7625
---------+------------------------------               Adj R-squared =  0.7558
   Total |  .011957628    73  .000163803               Root MSE      =  .00632

------------------------------------------------------------------------------
     gpm |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
  weight |   .0000163   1.18e-06     13.743   0.000       .0000139    .0000186
 foreign |   .0062205   .0019974      3.114   0.003       .0022379    .0102032
   _cons |  -.0007348   .0040199     -0.183   0.855      -.0087504    .0072807
------------------------------------------------------------------------------
```

You find foreign cars in 1978 less efficient. Foreign cars may have yielded better gas mileage than domestic cars in 1978, but this was only because they were so light.

Summary of other commands

Assign a label to the datafile currently in memory
. label data "1978 auto data"

Assign a label to the variable foreign
. label variable foreign "the origin of the car, foreign or domestic"

Create the value label foreignl and assign it to the variable foreign
. label define foreignl 0 "domestic car" 1 "foreign car"
. label values foreign foreignl

Create a new variable len_ft which is length divided by 12
. generate len_ft = length / 12

Change values of an existing variable named len_ft
. replace len_ft = length / 12

recode mpg into mpg3, having 3 categories, 1 2 3 using **generate** and **replace if**
. generate mpg = .
. replace mpg3 = 1 if (mpg <=18)
. replace mpg3 = 2 if (mpg >=19) & (mpg <=23)
. replace mpg3 = 3 if (mpg >=24) & (mpg <.)

Recode mpg into mpg3a, having 3 categories, 1 2 3 using **generate** and **recode**.
. generate mpg3a = mpg
. recode mpg3a min/18=1 19/23=2 24/max=3

Recode mpg into mpgfd, having 2 categories, but using different cutoffs for foreign and domestic cars
. generate mpgfd = mpg
. recode mpgfd min/18=0 19/max=1 if foreign==0

```
. recode mpgfd min/24=0 25/max=1 if foreign==1
```

With generate and replace

you can use + - for addition and subtraction

you can use * / for multiplication and division

you can use ^ for exponents (e.g. length^2)

you can use ( ) for controling order of operations

< Other operators and functions >

Logical operators used in Stata

~	not
==	equal
~=	not equal
!=	not equal
>	greater than
>=	greater than or equal
<	less than
<=	less than or equal
&	and
|	or

* Egen

egen stands for extended generate and is an extremely powerful command that has many options for creating new variables. Here is a list of some of the other options:

Egen Functions

count	number of non-missing vlaues
diff	compares variables, 1 if different, 0 otherwise
fill	fill with a pattern
group	creates a group id from a list of variables
iqr	interquartile range
ma	moving average
max	maximum value
mean	mean
median	median
min	minimum value
pctile	percentile
rank	rank
rmean	mean across variables
sd	standard deviation
std	standard scores
sum	sums

Some Estimation Procedures in Stata

anova	**analysis of variance and covariance**
arch	**autoregressive conditional heterosce. family of estimators**
arima	**autoregressive integrated moving average models**
bsqreg	**quantile regression with bootstrapped standard errors**
clogit	**conditional logistic regression**
cnreg	**censored-normal regression**
cnsreg	**constrained linear regression**
ereg	**maximum-likelihood exponential distribution models**
glm	**generalized linear models**
glogit	**weighted least squares logit on grouped data**
gprobit	**weighted least squares probit on grouped data**
ivreg	**instrumental variable and two-stage least squares regression**
lnormal	**maximum-likelihood lognormal distribution models**
logistic	**logistic regression**
logit	**maximum-likelihood logit regression**
mlogit	**maximum-likelihood multinomial logit models**
mvreg	**multivariate regression**
nbreg	**maximum-likelihood negative binomial regression**
nl	**nonlinear least squares**
ologit	**maximum-likelihood ordered logit**
oprobit	**maximum-likelihood ordered probit**
poisson	**maximum-likelihood poisson regression**
probit	**maximum-likelihood probit estimation**
qreg	**quantile regression**
reg3	**three-stage least squares regression**
regress	**linear regression**
rreg	**robust regression using IRLS**
sureg	**seemingly unrelated regression**
tobit	**tobit regression**
vwls	**variance-weighted least squares regression**
zinb	**zero-inflated negative binomial model**
zip	**zero-inflated poisson models**

test & **predict** are commands that can be used in conjuction with estimation procedures. There are too many combinations of estimation, **predict** and **test** to get into in this class, other than to say that they provide very powerful tools for researchers and are worth the time spent learning them.

Intro to Graphics

1.0 Stata commands in this unit

. **stem**
. **graph**
    graph types
    histogram
    box
    bar
    oneway
    twoway
    matrix
. **kdensity**
. **pnorm**
. **rvfplot**
. **rvpplot**

2.0 Demonstration and Explanation

2.1 Stem-and-leaf Plots

. **use hsb2, clear**
. **stem math, lines(2)**

The **stem** command produces a stem-and-leaf diagram. The **lines(2)** option sets the output to two lines per digit, which in this case, makes the output a little cleaner.

2.2 The Graph Command

. **graph math, histogram bin(11) normal**
. **kdensity math, normal**

The **graph** command produces many types of graphic plots. The **histogram** option naturally produces histograms. The **bin(11)** option indicates how many categories to break the data into. Eleven was chosen so as to be similar to the **stem** command above. The **kdensity** produces a type of a smoothed histogram. In both **histogram** and **kdensity**, the **normal** option superimposes a normal curve on the graph.

. **sort prog**
. **graph math, box by(prog) total**
. **graph read math socst, box**

The **box** option produces box-and-wisker plots. The **by(prog)** option produces a box plot for each level of the variable **prog**, but only if the data have been sorted on the **prog**. The total option produces a box plot for all the observations, across all level of **prog**. The second box plot example produces separate box plots for each of the variables listed.

. **graph math, bar by(prog) means**
. **graph read math socst, bar means**

The **bar** option produces vertical bar charts. The first bar chart looks at 'math' for each level of 'prog.' It is necessary for the data to be sorted by 'prog' which we did in the previous step. The **means** option produces bar graphs of means.
The second example produces a bar chart of means for the three variables listed after the **graph** command.

. **graph math read science, oneway**

The **oneway** option produces a one-dimensional frequency plot. Notice how easy it is to compare the frequency distributions to two or more variables simultaneously.

. **graph math read, twoway**
. **graph math read, twoway oneway**
. **graph math read, twoway box**

The **twoway** option produces a bivariate scatterplot. Three examples are given: 1) The scatterplot only, 2) the scatterplot along with oneway plots of the marginal distributions, 3) the scatterplot along with box plots of the marginal distributions.

. **graph math read science ses, matrix half**

The **matrix** option produces a bivaarite scatterplot for each of the variables listed. The **half** option suppresses the symmetric upper portion of the output, producing larger individual plots.

2.3 Normal Probalility Plot

. **pnorm math**

The **pnorm** command produces a normal probability plot.

2.4 Some Regression Related Plots

. **regress math read science ses**
. **rvfplot, yline(0)**
. **rvpplot read, yline(0)**
. **rvpplot science, yline(0)**
. **rvpplot ses, yline(0)**

It is easy to create various residual plots using the **rv** commands. The **rvfplot** command produces a plot of the residuals vs the predicted values (fitted). The rvpplot command produces plots of redisuals vs independent variables (predictors). The **yline(0)** option produces a horizontal line at the values of zero on the y-axiz.

3.0 Try the commands on your own

. **use hsb2, clear**
. **stem math, lines(2)**
. **graph math, histogram bin(11) normal**
. **sort prog**
. **graph math, box by(prog) total**
. **graph math read science, oneway**
. **graph math read, twoway box**
. **graph math read science ses, matrix half**
. **pnorm math**
. **regress math read science ses**
. **rvfplot, yline(0)**
. **rvpplot read, yline(0)**
. **rvpplot science, yline(0)**
. **rvpplot ses, yline(0)**

I do, I do

1.0 Stata commands in this unit

. **do**

2.0 Demonstration and Explanation

Sometimes you may want to use the same commands on more than one file but you don't want to have to type them in more than once. Other times its easier to collect all of your Stata commands together in one place and do all at once rather than one at a time. The do-file allows you to place commands in a file and run them all at once. Any command that you can type in on the command line can be placed in a do-file.

2.1 Creating a do-file

Do-files are created with the do-file editor or any other text editor. Any command which can be executed from the command line can be placed in a do-file. Here are some commands that could be placed in a do-file:

set more off
cd A:\stata
use hsb2, clear
generate lang = read + write
label variable lang "language score"
tabulate lang
tabulate lang female
tabulate lang prog
tabulate lang schtyp
summarize lang, detail
table female, contents(n lang mean lang sd lang)
table prog, contents(n lang mean lang sd lang)
table ses, contents(n lang mean lang sd lang)
correlate lang math science socst
regress lang math science female
set more on

Let's look at a do-file that contains these commands that is on our floppy disk.

. cd A:\stata
. type hsbbatch.do

2.2 Running the do-file

. **do hsbbatch**

The **do** command runs the do-file.

Web Notes

The Stata Class Notes and Stata Learning Modules pages are available on the World Wide Web by visiting ...

http://www.ats.ucla.edu/stat/stata/notes/ or http://www.ats.ucla.edu/stat/stata/modules/