Introduction of STATA

News: There is an introductory course on STATA offered by CIS

Description: **Intro to STATA**

- On Tue, Feb 13th from 4:00pm to 5:30pm in CIT 269
  Seats left: 4 Windows, 7 Macintosh
  For undergrads and grad students [register]
- On Wed, Feb 28th from 6:30pm to 8:00pm in CIT 269
  Seats left: 4 Windows, 5 Macintosh
  For undergrads and grad students [register]

STATA is a statistical software package that is widely used by students and researchers in economics. It is used mainly to analyze and model large datasets. You will learn how to create a histogram and frequency distributions. Regression analysis and hypothesis testing will also be covered. Please note that this session will *not* cover the foundational statistics, only how to use STATA to perform them, so some prior knowledge of statistics is extremely helpful.

Go to http://training.brown.edu/index.php?forgroup=undergrads for registration.

Download STATA SE 9.2 installer from CIS

Go to http://www.brown.edu/Facilities/CIS/software.html and download STATA installer which is suitable for your own computer platform. Then install it following the instructions.

Notice: *KeyAccess* must be installed and connected to the Brown campus *KeyServer* to run Stata.

Basics

Start STATA

After installation, go to START menu and open Stata9. Also you can create a shortcut icon put on the desktop and double click the icon to open it. Four Windows will appear. They are Command line, Results window / Review window / Variables box.

Type the commands in the Command line and "Enter", both the commands and the results will show in the Results window. Review window will keep a record of what commands you have entered during your session. When you open a data file in STATA, the variables will be listed, along with their label, in the variables box to the left of the main screen.

Importing data and Creating log files

1. Import a STATA data set

Under the STATA directory, there is a 'auto.dta' file which is a stata data set. Go to the menu bar, click open file, there will be a window for the path search. Choose C:\Program Files\Stata9\, open auto.dta file, now the auto.dta has been imported into our working space. The other way to import stata data set is to type **use "C:\Program Files\Stata9\auto.dta"** in the command line.

2. Create a log file

A log file is to save all the work you **will** do showed in the results window including all the commands and the results. Go to the menu bar "file", scrolling down to "log" and "begin", choose the location where you want to save the results. For example, create a folder called AM169 under your C drive and name the file "auto.smcl". Or in the command line, type **log using "C:\AM169\auto.smcl".** In this way, the results will be stored in this file, and we can edit the results later when we finish**.**

Saving the results, the data and Logging out

1. Saving the results

Type **log close**. Now the log file is finished in the stata output format. Type **translate auto.smcl auto.log.** Once you have a file ends as .log, you can open it in a text editor like Notepad or word**.** Of course if you didn't open a log file, you can still save the results by copy & paste. Open a new word file (or Notepad as you like), highlight the results you want in the Results Window, go to the menu bar "edit", choose "copy", then "paste" them into the new word file.

2. Saving the data

If you didn't change the data set during your past work, you don't need to save the data. If you did make changes of the data, and want to save it, type **save auto, replace** or **save auto2**. The former gives you an updated auto.dta data set. The latter creates a new data set called auto2.

3. Exit STATA

Now, you are ready to exit, you have two choices. You can type **exit** in the Stata Command line or simply click on *File* (on the menu bar at the top of the screen) and then scroll down to *Exit.*

STATA help

Within STATA you can manually choose the help function from the menu bar and type in a search query, or you can type "help *topic*" into the command line.

Data Management

Data Checking and Summarizing:

Here are a few commands that you can use to explore your data. **Stata is case sensitive.**

- **Edit:** brings up a spreadsheet-style data editor for entering new data and editing existing data.
  Browse: is similar to edit, except that it will not allow to change the
- **Describe:** gives you basic information on the number of observations, variable names and labels.
- **List:** displays the values of variables. If no *varlist* is specified, the values of all the variables are displayed.
- **Sort:** arranges the observations of the current data into ascending order based on the values of the variables in *varlist*.
- **Tabulate(**for short, **Tab)**: produces one-, two-way tables of frequency counts.
  Tab1: produces a one-way tabulation for each variable specified in *varlist.*
  Tab2: produces all possible two-way tabulations of the variables specified in *varlist.*
- **Summarize(**for short, **Sum):** calculates and displays a variety of univariate summary statistics.
- **Inspect:** provides a quick summary of a numeric variable that differs from that provided by summarize or tabulate. It reports the number of negative, zero, and positive values; the number of integers and non-integers; the number of unique values; the number of missing; and produces a small histogram. Its purpose is not analytical -- instead it allows you to quickly gain familiarity with unknown data.

Examples: (All the examples will use the dataset auto.dta)

```
. use "D:\Program Files\Stata9\auto.dta"

. describe

Contains data from C:\Program Files\Stata9\auto.dta
  obs:                74                          1978 Automobile Data
  vars:               12                          13 Apr 2005 17:45
  size:           3,478 (99.9% of memory free)    (_dta has notes)
-------------------------------------------------------------------------------
              storage  display    value
variable name   type   format     label      variable label
-------------------------------------------------------------------------------
make            str18  %-18s                 Make and Model
price           int    %8.0gc                Price
mpg             int    %8.0g                 Mileage (mpg)
rep78           int    %8.0g                 Repair Record 1978
headroom        float  %6.1f                 Headroom (in.)
trunk           int    %8.0g                 Trunk space (cu. ft.)
weight          int    %8.0gc                Weight (lbs.)
```

```
  length              int    %8.0g                 Length (in.)
  turn                int    %8.0g                 Turn Circle (ft.)
  displacement        int    %8.0g                 Displacement (cu. in.)
  gear_ratio          float  %6.2f                 Gear Ratio
  foreign             byte   %8.0g       origin    Car type
  ------------------------------------------------------------------------------
Sorted by:  foreign

. tab foreign

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

. sum mpg

    Variable |        Obs        Mean    Std. Dev.       Min        Max
  -------------+--------------------------------------------------------
         mpg |         74     21.2973    5.785503        12         41


. tab foreign, sum(price)

             |        Summary of Price
    Car type |        Mean   Std. Dev.       Freq.
  -----------+------------------------------------
  Domestic |   6,072.423   3,097.104          52
     Foreign |   6,384.682   2,621.915          22
  -----------+------------------------------------
       Total |   6,165.257   2,949.496          74

. tab foreign rep78,sum(price)

                Means, Standard Deviations and Frequencies of Price

             |                    Repair Record 1978
    Car type |         1         2         3         4         5 |     Total
  -----------+--------------------------------------------------------------+----------
  Domestic |     4,564.5  5,967.625  6,607.074  5,881.556     4,204.5 |  6,179.25
             | 522.55191  3,579.357  3,661.267  1,592.019  311.83409 | 3,188.969
             |         2         8        27         9         2 |        48
```

Foreign	.	.	4,828.667	6,261.444	6,292.667	6,070.143
	.	.	1,285.613	1,896.092	2,765.629	2,220.984
	0	0	3	9	9	21
Total	4,564.5	5,967.625	6,429.233	6,071.5	5,913	6,146.043
	522.55191	3,579.357	3,525.14	1,709.608	2,615.763	2,912.44
	2	8	30	18	11	69

Graphics

- **Graph box**: draws vertical box plots. In a vertical box plot, the y axis is numerical, and the x axis is categorical.

  Graph hbox: draws horizontal box plots. In a horizontal box plot, the numerical axis is still called the y axis, and the categorical axis is still called the x axis, but y is presented horizontally, and x vertically.
- **Histogram:** draws histograms of *varname*, which is assumed to be the name of a continuous variable unless the discrete option is specified.
- **Graph twoway scatter/Graph twoway line**: The leading graph is optional. If the first (or only) plot is scatter, you may omit **twoway** as well, and then the syntax is **scatter** for short. The same thing to **line**.

Notice: Being a plottype, **line** may be combined with other plottypes in the *twoway* family, as in

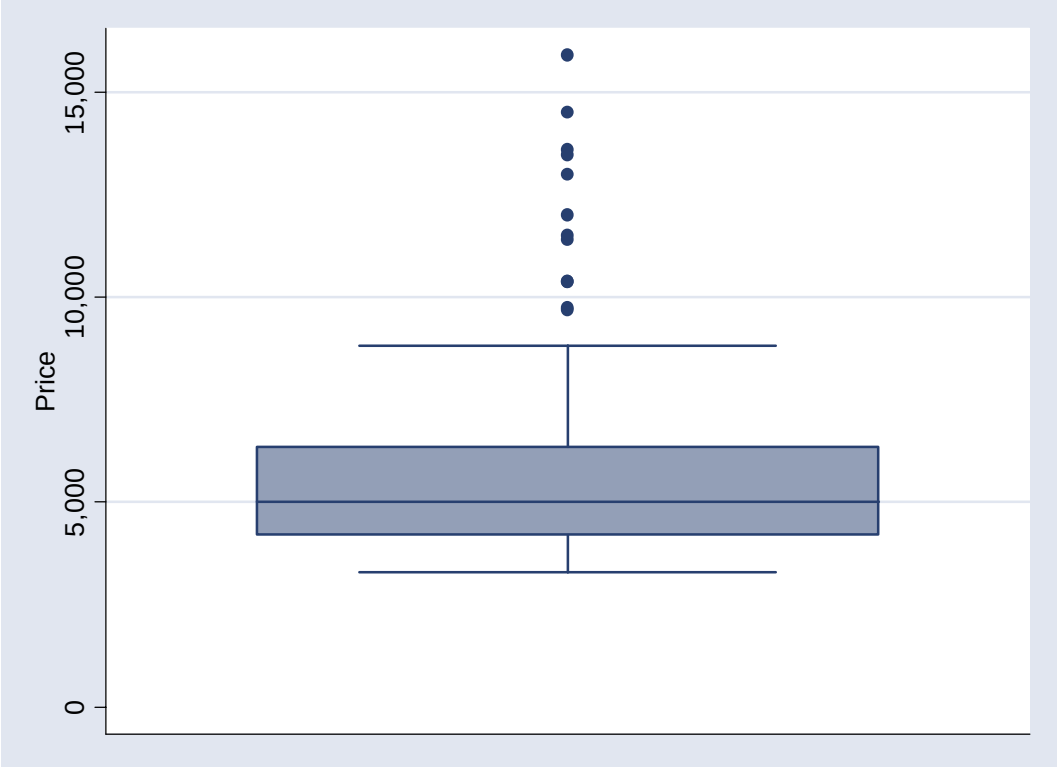
. twoway (line ...) (scatter ...) (lfit ...) ...

which can equivalently be written

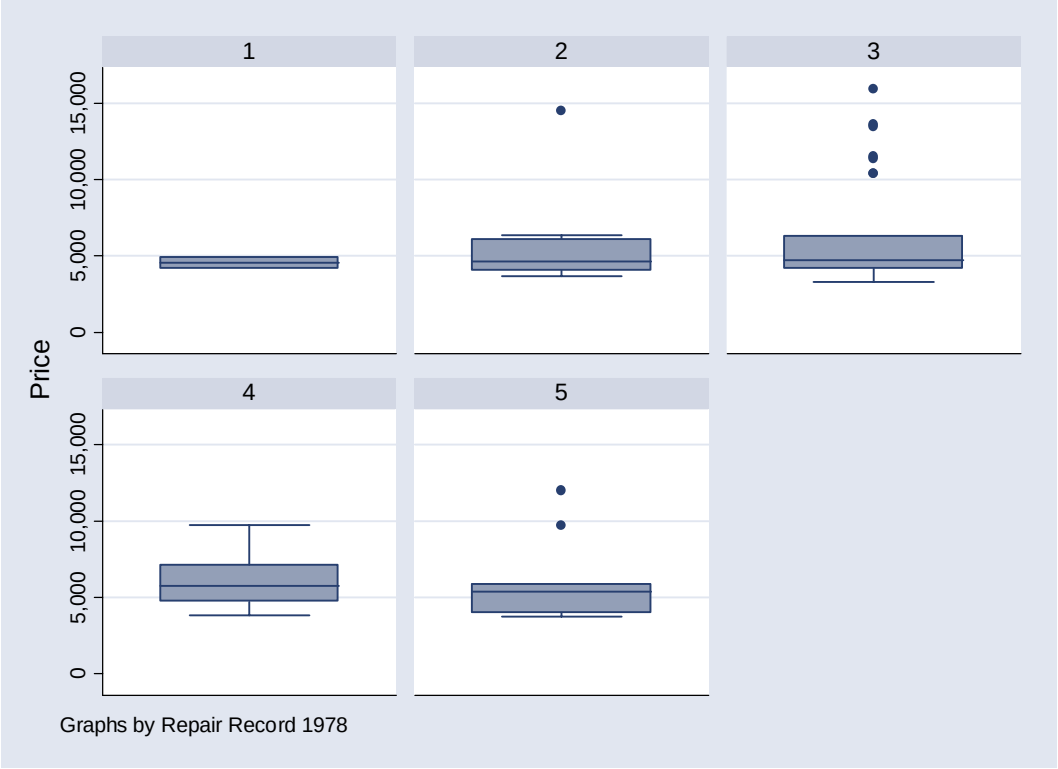
. line ... || scatter ... || lfit ... || ...

Examples:

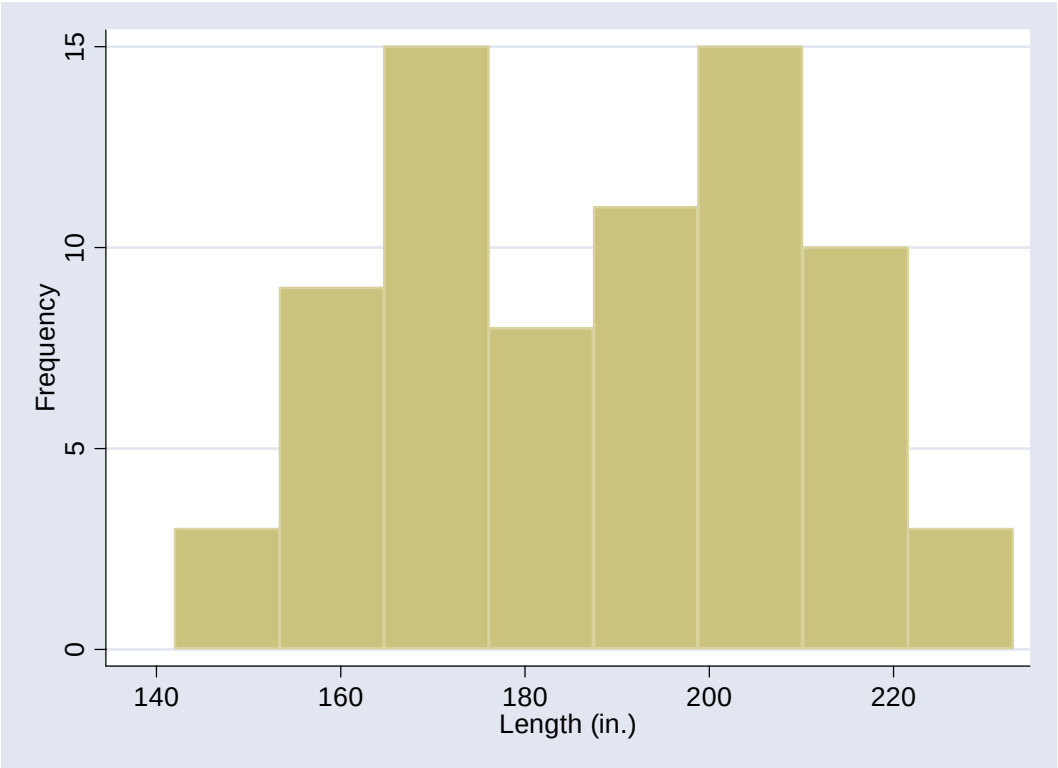
. graph box price



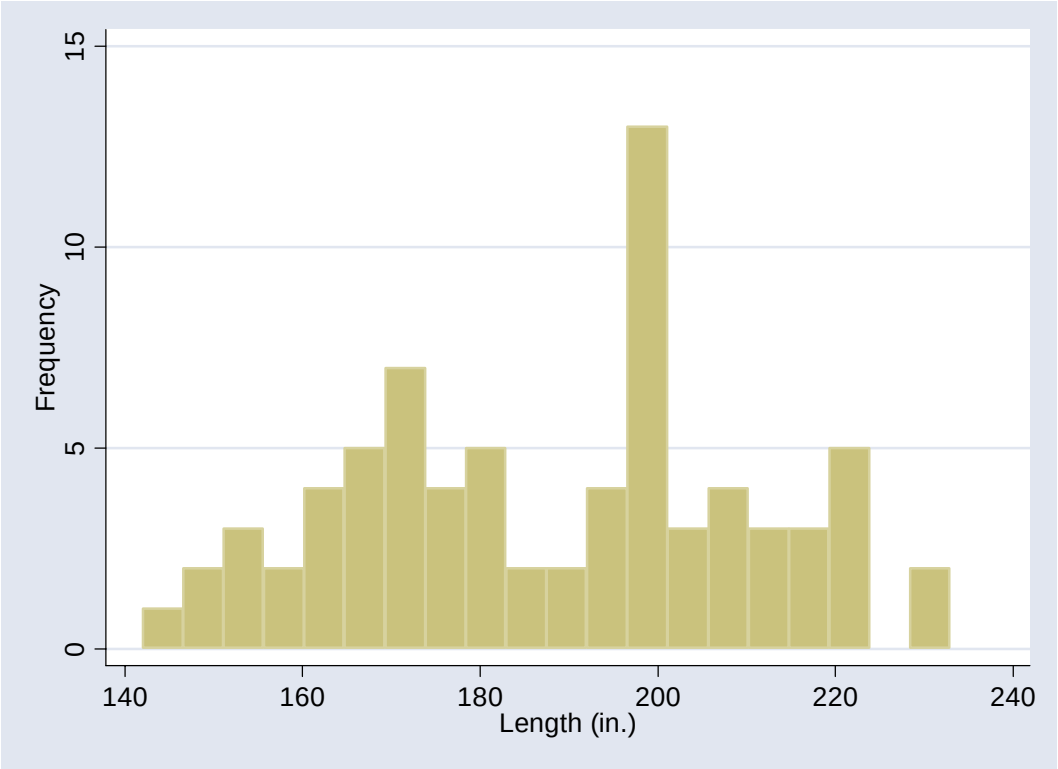
. graph box price, by(rep78)



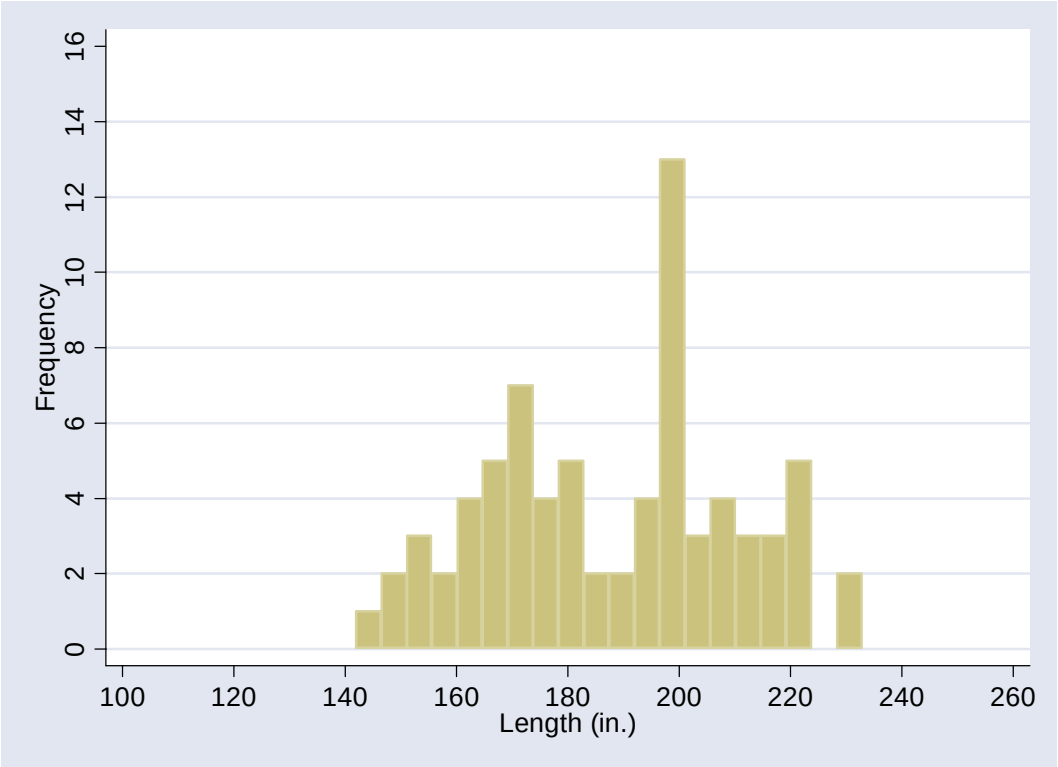
. histogram length,frequency
(bin=8, start=142, width=11.375)



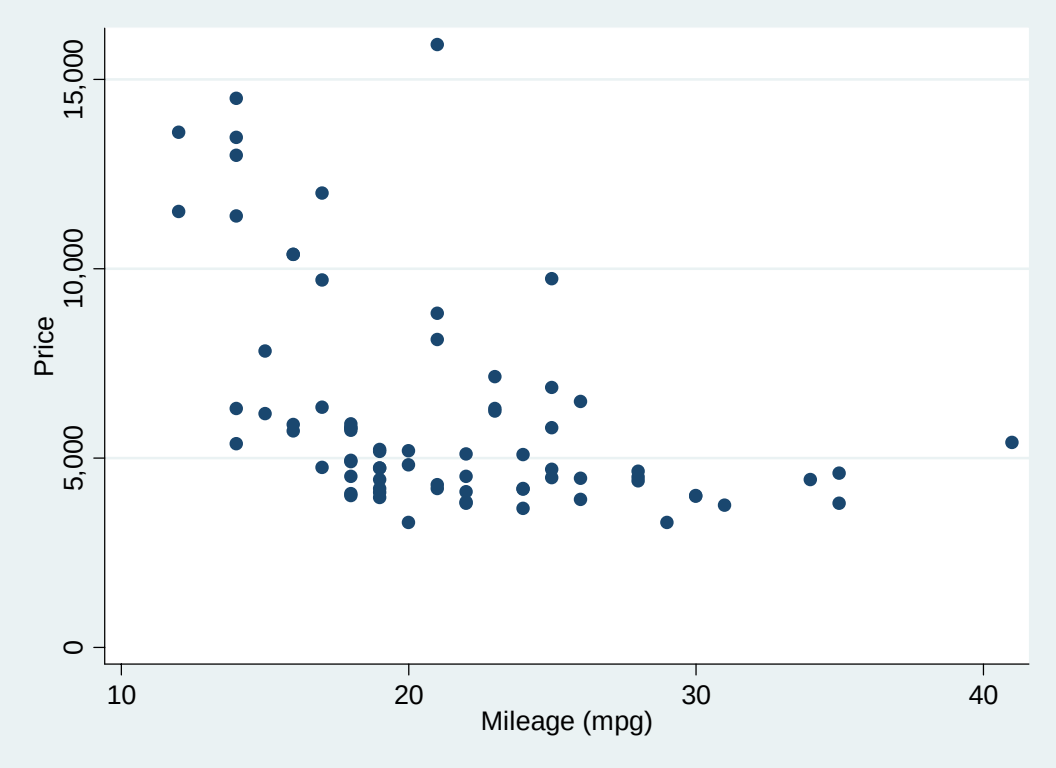
. histogram length,frequency bin(20)
(bin=20, start=142, width=4.55)



. histogram length,frequency xlabel(100(20) 260) ylabel(0(2) 16) bin(20)
(bin=20, start=142, width=4.55)



. scatter price mpg



Probability calculation using STATA

1. Using STATA as a Calculator

- **Display:** displays strings and values of scalar expressions.

Examples:

. display 0.75^4
.31640625

. display exp(3)
20.085537

2. Probability distributions and density functions

- **Binomial(n,k,p)**: returns the probability of k or more successes in n trials when the probability of a success on a single trial is p. Here n > k > 0 must be non-negative integers, and p > 0.
- **F(n1,n2,f):** returns the cumulative F distribution with n1 numerator and n2 denominator degrees of freedom, for n1, n2 > 0; returns 0 unless f > 0.
- **normal(z):** returns the cumulative standard normal distribution.
  normalden(z): returns the standard normal density.
  normalden(z,s): returns the rescaled standard normal density. normalden(z,s)= normalden(z)/s if s>0 and s not missing, otherwise, the result is missing (.).
  normalden(x,m,s): returns the normal density with mean m and standard deviation s. normalden(x,m,s) = normalden((x-m)/s)/s if s>0, and m and s are not missing, otherwise, the result is missing (.).
- **invnormal(p):** returns the inverse cumulative standard normal distribution; normal(z) = p then invnormal(p) = z.

3. Generating random numbers using STATA

- **uniform():** returns uniformly distributed pseudorandom numbers on the interval [0,1). uniform() takes no arguments, but the parentheses must be typed. (See matrix functions for the related matuniform() matrix function.)
  When being used with inverse cumulatives, can draw from other distributions.
  For example:
- **invnormal(uniform())**: returns normally distributed random numbers with mean 0 and standard deviation 1. (uniform()) within invnormal() takes no arguments, but the parentheses must be typed.

Examples:

1. Find the 95% quantile of $N(0,1)$

. display invnormal(0.95)
1.6448536

2. **Generate a sequence of N=50 numbers from $N(0,1)$ and a sequence of N=50 numbers from $N(2,5^2)$.**

. set obs 50
obs was 0, now 50

. gen x=invnormal(uniform()) (sampled from $N(0,1)$)

. gen y0=invnormal(uniform()) (sampled from $N(0,1)$)

. gen y=2+5*y0 (simulated from $N(2,5^2)$)

. browse

Linear Regression

- **Regress**: fits a model of *depvar* on *indepvars* using linear regression.
- **Predict:** calculates predictions, residuals, influence statistics, and the like after estimation. Exactly what **predict** can do is determined by the previous estimation command; command-specific options are documented with each estimation command. Regardless of command-specific options, the actions of predict share certain similarities across estimation commands:

1) **predict** *newvar* creates *newvar* containing "predicted values"--numbers related to the E(y|x). For instance, after linear regression, **predict** *newvar* creates *xb* and, after **probit**, creates the probability F(xb).

2) **predict** *newvar, xb* creates *newvar* containing *xb*. This may be the same result as (1) (e.g., linear regression) or different (e.g., **probit**), but regardless, option *xb* is allowed.

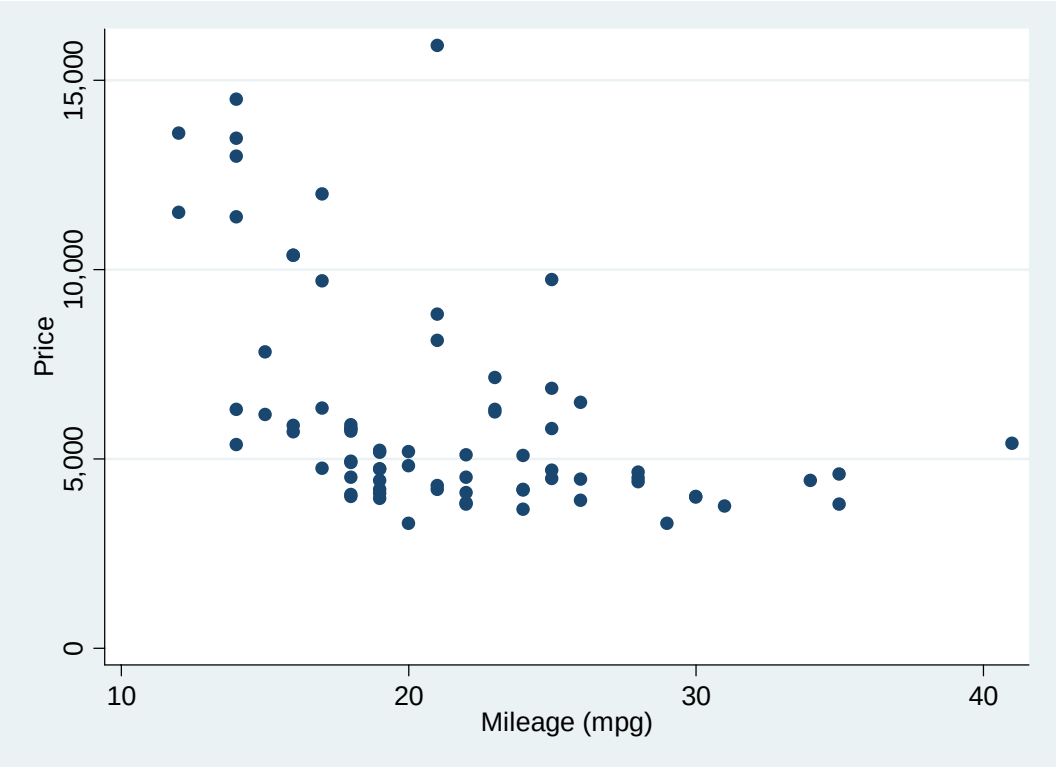
3) **predict** *newvar, stdp* creates *newvar* containing the standard error of the linear prediction *xb*.

4) **predict** *newvar, other_options* may create *newvar* containing other useful quantities; see help for the particular estimation command to find out about other available options.

5) *nooffset* added to any of the above commands requests that the calculation ignore any offset or exposure variable specified by including the offset(varname) or exposure(varname) options when you fitted the model.

Examples:

. scatter price mpg



. regress price mpg

Source	SS	df	MS	Number of obs =	74
				F(1, 72) =	20.26
Model	139449474	1	139449474	Prob > F =	0.0000
Residual	495615923	72	6883554.48	R-squared =	0.2196
				Adj R-squared =	0.2087
Total	635065396	73	8699525.97	Root MSE =	2623.7

price	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
mpg	-238.8943	53.07669	-4.50	0.000	-344.7008	-133.0879
_cons	11253.06	1170.813	9.61	0.000	8919.088	13587.03

. predict yhat
(option xb assumed; fitted values)

. scatter price mpg || line yhat mpg