Data Analysis with Stata 12

Tutorial

November 2012

Table of Contents

Section 1: Introduction

1.1 About this Document

This document is an introduction to using Stata 12 for data analysis. Stata is a software package popular in the social sciences for manipulating and summarizing data and conducting statistical analyses. This is the second of two Stata tutorials, both of which are based on the 12th version of Stata, although most commands discussed can be used in early versions also.

The following sections provide information on running a variety of statistical tests and inference procedures. Readers with at least some basic statistical knowledge are best suited for these tutorials, although we do attempt to explain each process in as much detail as possible. In this tutorial, we also assume that the reader is familiar with the Stata interface, importing and exporting files, and running basic data manipulation commands. If this is not the case, please see our "Getting Started" tutorial before continuing.

1.2 Documentation

Similar to the SAS statistical software package, Stata can be intimidating to first-time users who are not familiar with the syntax language. However, Stata 12 has drop-down menu options for most analytic, graphical, and statistical commands (similar to, but not as extensive as, SPSS). As tempting as the drop-down menus are, we still recommend that you become familiar with the Stata syntax as it is more efficient and leads to fewer errors. However, we do present both options whenever possible.

Among the many reasons why we prefer to use syntax over the drop-down menus is the extent of support material to turn to when you run into problems with your code. First and foremost, we recommend using the "help" feature within Stata itself (described in detail in Section 8 of the "Getting Started" tutorial). Additionally, you can use the following:

1) Stata manuals (some are available at the PCL for check-out)
2) Stata's own website has a modest amount of FAQ's in the support section: http://stata.com/support/faqs/
3) The SSC's website to find more answers to FAQ's: http://ssc.utexas.edu/software/faqs/stata

1.3 Accessing Stata

If you are a faculty, student, or staff member at the University of Texas at Austin, you may access Stata 12 in several ways:

1) License a copy from ITS Software Distribution Services (http://www.utexas.edu/its/sds).
2) Access the program via the Windows Terminal Server for a small yearly fee. To use the terminal server, you need an ITS computer account (either a personal or departmental) and then validate the account for Austin (AMS) services. Details on obtaining an ITS computer account and connecting to the Windows Terminal Services server may be found in the following FAQ: http://ssc.utexas.edu/software/stat-apps-server . If you have difficulties accessing Stata 12 on the Windows Terminal Server, call the ITS Helpdesk at 512-475-9400 or send e-mail to help@its.utexas.edu.
3) Stata is also available at certain labs around campus, and your department may also provide it via a server or in one a lab room. Check with your advisor or chair on the availability of Stata in your department.

1.4 Getting Help

If you have questions about how to use Stata or interpret output, you can e-mail them to stats@ssc.utexas.edu, or visit http://ssc.utexas.edu/consulting/free-consulting to make an appointment via our online scheduler. The SSC Division also offers introductory-level short courses on Stata, as well as on other statistical software packages, each semester. Visit http://ssc.utexas.edu/courses/short for this semester's schedule, registration information, and course descriptions. Also on the SSC website, you'll find more details about our consulting services, as well as frequently asked questions and answers about using Stata and other statistical software.

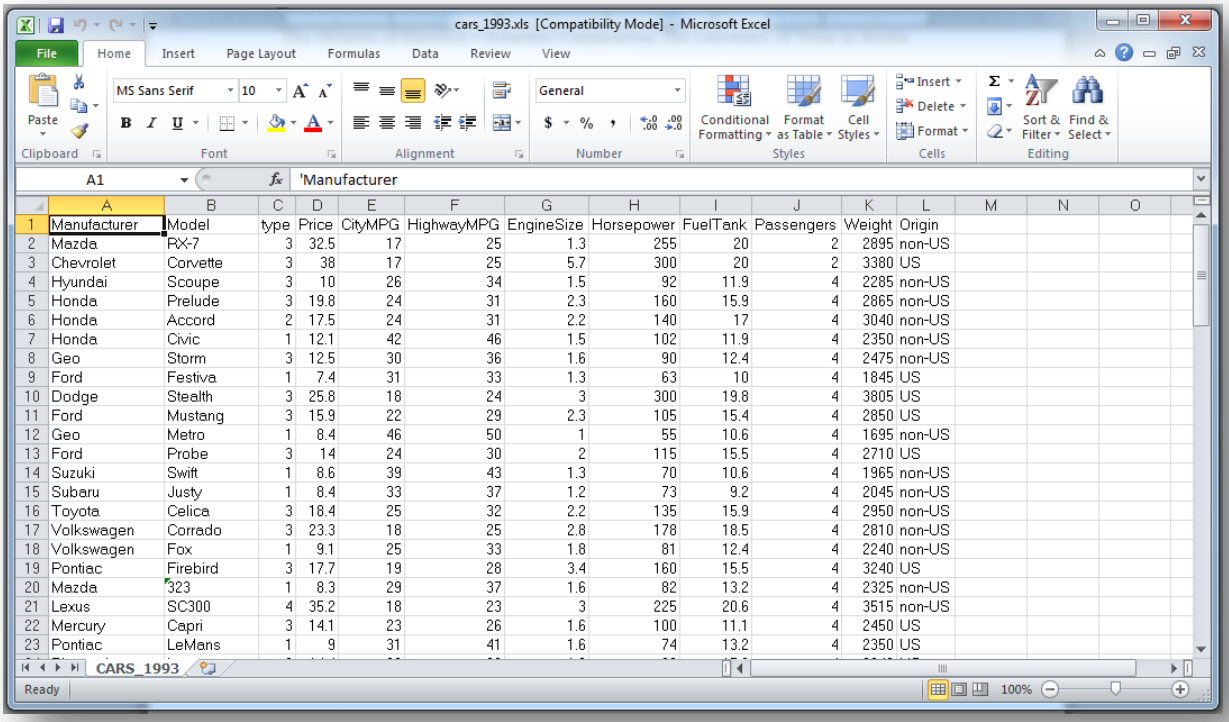
Section 2: The Example Dataset

Throughout this document, we will be using a dataset called *cars_1993.xls*, which was used in the previous tutorial and contains various characteristics, such as price and miles-per-gallon, of 92 cars. In order to follow along with the examples, please download this data by clicking HERE.

Note that this is also the same example dataset we use in the "SAS: Getting Started" tutorial, and the file is actually one of the example datasets from SAS, which provides information about the *cars_1993* file and is represented below:

> **Name:** cars_1993
>
> **Reference:** This represents a subset of the information reported in the 1993 Cars Annual Auto Issue published by Consumer Reports and from Pace New Car and Truck 1993 Buying Guide.
>
> **Description:** A random sample of 92 1993 model cars is contained in this data set. The information for each car includes: manufacturer, model, type (small, compact, sporty, midsize, large, or van), price (in thousands of dollars), city mpg, highway mpg, engine size (liters), horsepower, fuel tank size (gallons), weight (pounds), and origin (US or non-US). The data are excellent for doing descriptive statistics by groups or an ANOVA or regression with price as the response variable. Note that violations of the assumptions are probably present and transformation of the response variable is most likely necessary.

Below is what the file should look like once you download and open it in Excel:



	A	B	C	D	E	F	G	H	I	J	K	L
1	Manufacturer	Model	type	Price	CityMPG	HighwayMPG	EngineSize	Horsepower	FuelTank	Passengers	Weight	Origin
2	Mazda	RX-7	3	32.5	17	25	1.3	255	20	2	2895	non-US
3	Chevrolet	Corvette	3	38	17	25	5.7	300	20	2	3380	US
4	Hyundai	Scoupe	3	10	26	34	1.5	92	11.9	4	2285	non-US
5	Honda	Prelude	3	19.8	24	31	2.3	160	15.9	4	2865	non-US
6	Honda	Accord	2	17.5	24	31	2.2	140	17	4	3040	non-US
7	Honda	Civic	1	12.1	42	46	1.5	102	11.9	4	2350	non-US
8	Geo	Storm	3	12.5	30	36	1.6	90	12.4	4	2475	non-US
9	Ford	Festiva	1	7.4	31	33	1.3	63	10	4	1845	US
10	Dodge	Stealth	3	25.8	18	24	3	300	19.8	4	3805	US
11	Ford	Mustang	3	15.9	22	29	2.3	105	15.4	4	2850	US
12	Geo	Metro	1	8.4	46	50	1	55	10.6	4	1695	non-US
13	Ford	Probe	3	14	24	30	2	115	15.5	4	2710	US
14	Suzuki	Swift	1	8.6	39	43	1.3	70	10.6	4	1965	non-US
15	Subaru	Justy	1	8.4	33	37	1.2	73	9.2	4	2045	non-US
16	Toyota	Celica	3	18.4	25	32	2.2	135	15.9	4	2950	non-US
17	Volkswagen	Corrado	3	23.3	18	25	2.8	178	18.5	4	2810	non-US
18	Volkswagen	Fox	1	9.1	25	33	1.8	81	12.4	4	2240	non-US
19	Pontiac	Firebird	3	17.7	19	28	3.4	160	15.5	4	3240	US
20	Mazda	323	1	8.3	29	37	1.6	82	13.2	4	2325	non-US
21	Lexus	SC300	4	35.2	18	23	3	225	20.6	4	3515	non-US
22	Mercury	Capri	3	14.1	23	26	1.6	100	11.1	4	2450	US
23	Pontiac	LeMans	1	9	31	41	1.6	74	13.2	4	2350	US

Section 3: Descriptive Statistics and Graphs

3.1 Introduction

Almost all analytic procedures begin with running descriptive statistics on the data. Doing this familiarizes you with the properties of your dataset, including mean values, measures of spread, and the frequency of observations for different values of categorical variables. The following section explores the commands in Stata 12 that summarize data, both numerically and graphically, for both quantitative and qualitative variables.

3.2 Univariate Descriptives

As seen in the first tutorial, the **`summary`** command will output the mean, standard deviation, minimum, maximum, and the number of observations for a specified *numeric* variable or set of variables:

```
. sum Price EngineSize FuelTank
```

Variable	Obs	Mean	Std. Dev.	Min	Max
Price	92	19.04891	8.623728	7.4	47.9
EngineSize	92	2.661957	1.041538	1	5.7
FuelTank	92	16.64457	3.29166	9.2	27

You can get more specific details of those variables by adding the **`detail`** option after the list of variables. The output will contain common quartiles and the variance, skewness, and kurtosis statistics (related to the second, third, and fourth moments of the distributions of the variables). Below is the example with the three variables from above. The output continues past the main window, which you can see by hitting **Spacebar** or almost any other key:

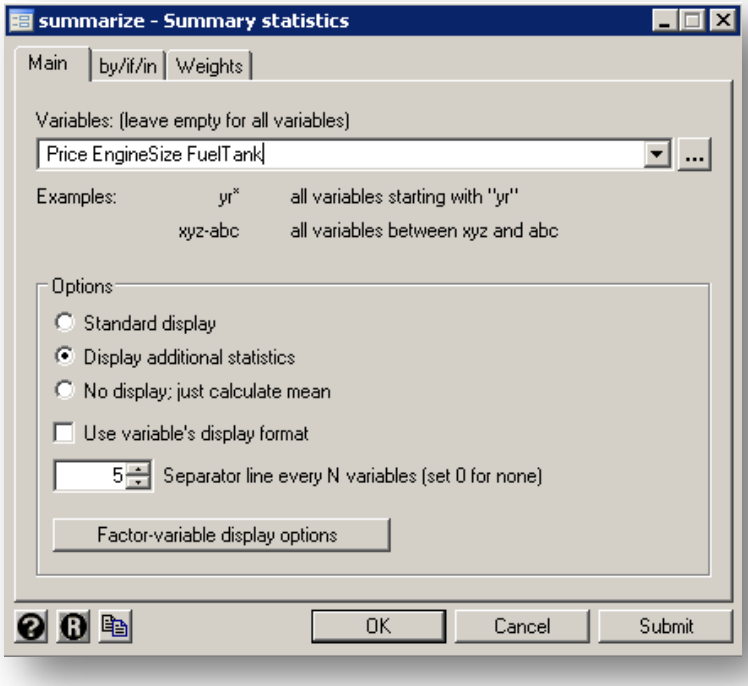
```
. sum Price EngineSize FuelTank, detail

                            Price
-------------------------------------------------------------
      Percentiles      Smallest
 1%           7.4           7.4
 5%           8.4             8
10%           9.8           8.3       Obs                  92
25%         12.15           8.4       Sum of Wgt.          92

50%          17.6                     Mean           19.04891
                        Largest       Std. Dev.      8.623728
75%            23          37.7
90%          32.5            38       Variance       74.36868
95%          36.1          40.1       Skewness       .9919349
99%          47.9          47.9       Kurtosis       3.551329

                         EngineSize
-------------------------------------------------------------
      Percentiles      Smallest
 1%             1             1
--more--
```

These skewness and kurtosis statistics can be hard to interpret. If you are testing for the normality of a variable and need a p-value for these measures, use the **`sktest`** command, shown below for the *Price* variable:

```
. sktest Price

                    Skewness/Kurtosis tests for Normality
                                                      ------- joint ------
    Variable |   Obs   Pr(Skewness)   Pr(Kurtosis)  adj chi2(2)  Prob>chi2
-------------+-------------------------------------------------------------
       Price |    92        0.0003         0.1949        12.12      0.0023
```

From the output, we see that *Price* is significantly skewed (and we can see it is positively skewed from the value of 0.99 in the previous output) but the kurtosis is not significant. Having a significant skewness or kurtosis suggests that a variable is not normally distributed. You may further confirm this by viewing a histogram of the variable (see Section 3.3).

These summary statistics can also be run by going to **Data → Describe Data → Summary Statistics…** To obtain the detailed output, simply click the "Display additional statistics" option:



The **tabstat** command also has the capability to output many of the same statistics. However, you must list out each statistic after the command that you want in the output. If you are using syntax, we recommend **summary, detail** because you do not have to specify each statistic you want.

For categorical variables, the **tabulate** command will output a frequency table of every response (as seen below for the *Origin* variable). You can abbreviate this command with simply **tab**:

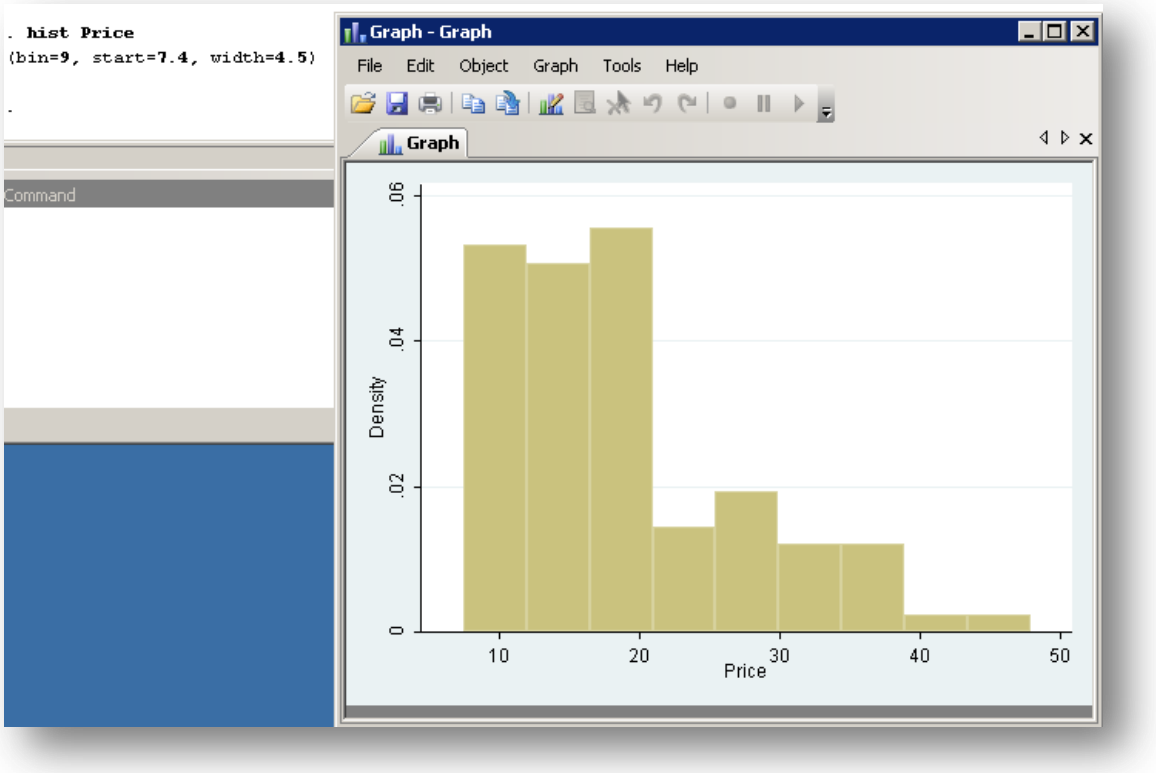
```
. tab Origin

    Origin |      Freq.     Percent        Cum.
-----------+-----------------------------------
        US |         48       52.17       52.17
    non-US |         44       47.83      100.00
-----------+-----------------------------------
     Total |         92      100.00
```

We can see that the dataset is roughly split in half in terms of US-made cars versus foreign-made cars. You can also run the **tabulate** command by going to **Statistics → Summaries, tables, and tests → Tables**.

3.3 Graphical Displays

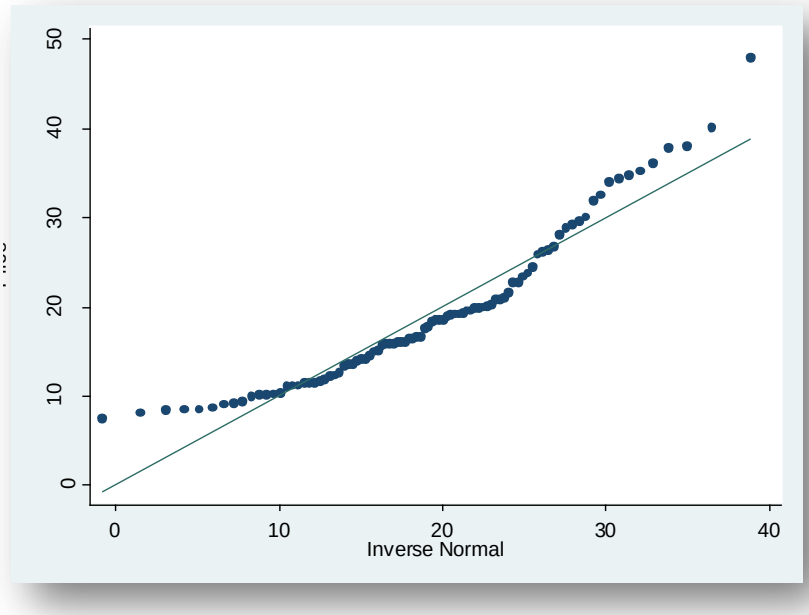
This section presents how to display a single numeric or categorical variable, as well as a pair of two variables. You should select the type of graph you want based on the type of variable or variables you wish to display visually.

For a single numeric variable, you can make a histogram with the **`hist`** command. It will select a default number of bins, which you can also specify if needed. You can enter the syntax shown in the picture below, or go to **Graphics → Histogram**. Without specifying any options, Stata will choose a default bin size, which is displayed in the output window:



After seeing the *Price* histogram, you might want to inspect a normal quantile-quantile plot (QQ-plot), which compares the distribution of the variable to a normal distribution. You can do this with the following command:

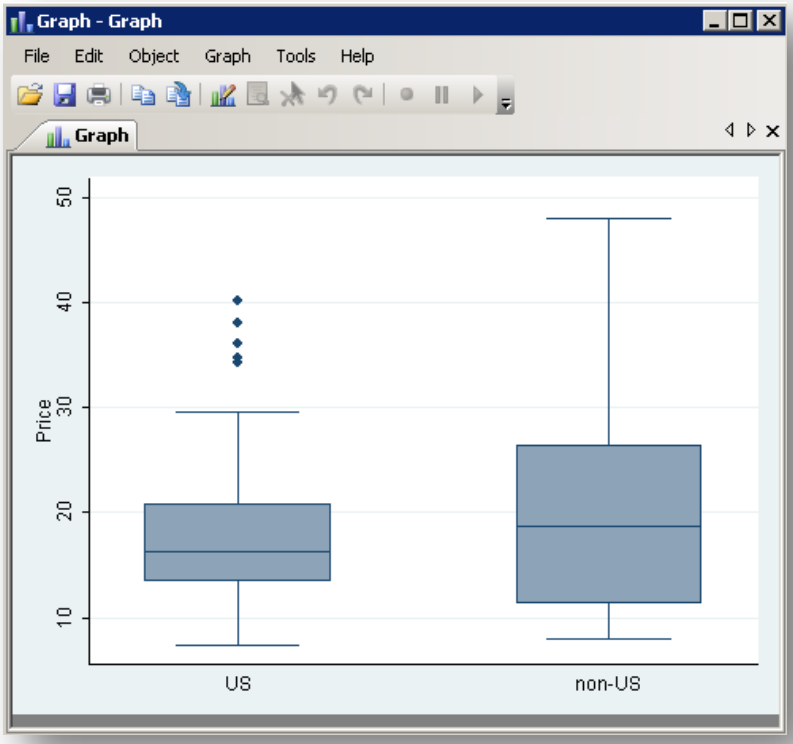
```
qnorm Price
```



The above plot confirms that *Price* is skewed left, and departs from a normal distribution. To numerically present this, you can ask Stata for the skew and kurtosis statistics, including p-values, as we did in Section 3.2.

Another way to display a continuous variable is with a box plot. Often, researchers want to compare the distribution of a continuous variable for two or more different groups (for example, when running an ANOVA procedure). Again, you can produce these with either syntax or by going to **Graphics → Box Plot**. Below, we show the boxplots for vehicle price based on origin (US or non-US):

```
graph box Price, over(Origin)
```

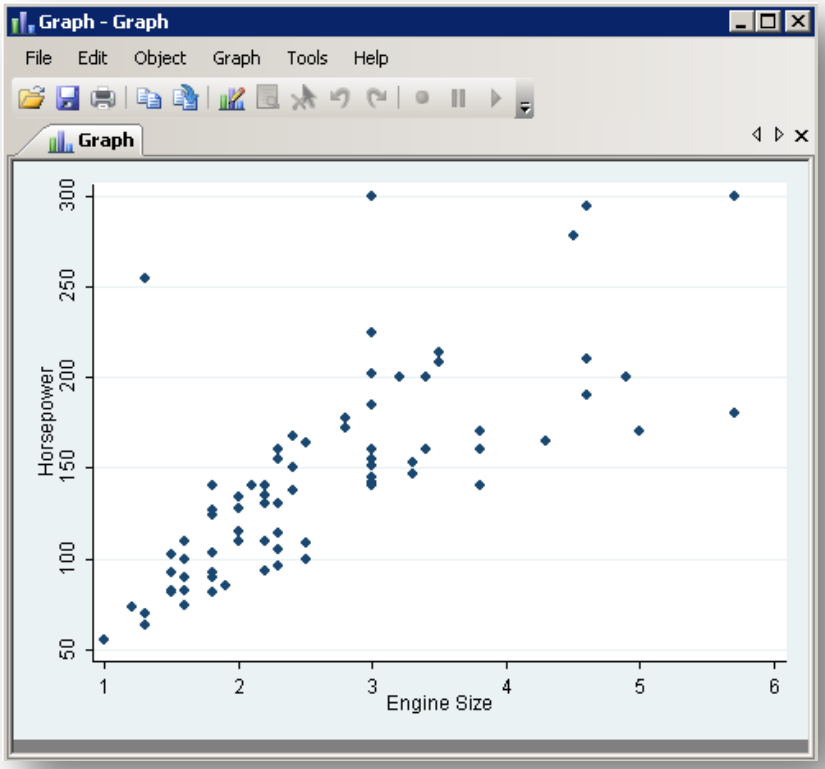


We can see from above that US-made cars have less variation on price, with several expensive outliers. However, the median price of US cars is roughly the same as non-US cars.

Stata 12 has many other ways to graphically display single variables, including pie charts and bar graphs for categorical variables. For a list of all of these options, go to the **Graphics** menu.

For graphically displaying relationships between two variables, go to **Graphics → Two-way Graph…** In the example below, we show the syntax and output for a scatterplot of engine size and horsepower:

```
twoway (scatter Horsepower EngineSize), ytitle(Horsepower)
      xtitle(Engine Size)
```



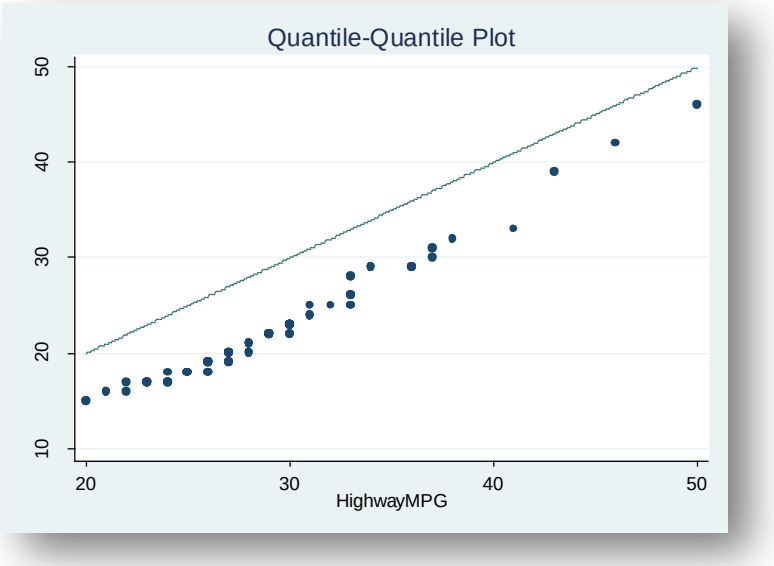
3.4 Bivariate Descriptives

Stata can also quickly and easily provide bivariate descriptive statistics, such as correlations, partial correlations, and covariances. All of these can be found in the **Statistics → Summaries, tables, and tests → Summary and descriptive statistics** menu. Below is an example of a correlation matrix for four variables in our *cars* dataset:

```
. corr CityMPG HighwayMPG EngineSize FuelTank
(obs=92)

             | CityMPG Highwa~G Engine~e FuelTank
-------------+------------------------------------
     CityMPG |  1.0000
  HighwayMPG |  0.9438   1.0000
  EngineSize | -0.7090  -0.6254   1.0000
    FuelTank | -0.8125  -0.7852   0.7586   1.0000
```

You can also visually compare the distribution of two continuous variables to see if they are similar. This could be an important step in many types of analyses, such as ANOVA and non-parametric comparison tests of two or more groups.

```
qqplot CityMPG HighwayMPG
```



From the above plot, we can see that the miles-per-gallon for these cars in the city has a roughly the same shape as on the highway, although there is a "shift," meaning a different mean value. You can see this by the very nearly-linear pattern of the dots in the above graph (indicating a similar shape of the distributions of the two variables), and how they fall below the line in the graph, which is where they would fall if the distributions were positioned over the same mean value.

Section 4: Comparing Means (T-Test, ANOVA, ANCOVA)

4.1 Introduction

Now that you know how to run preliminary descriptive statistics on your data, the next step is inevitably to run statistical tests to determine if your hypotheses are correct or not. This section describes the procedures in Stata that test the equality of means of a continuous variable from two or more groups. The remaining sections of this tutorial dive into more complicated statistical tests.

4.2 One- and Two-Sample T-Tests

A t-test is a useful technique for comparing the mean value of a group against some hypothesized mean (one-sample) or of two separate sets of numbers against each other (two-sample). The result of these tests provides you with a statistic which can be used to determine whether the difference between two means is statistically significant. Two-sample t-tests can be used either to compare two independent groups (known as an independent-samples t-test) or to compare observations from two measurement occasions for the same individuals (a paired comparison t-test).

To conduct a t-test, you must have a continuous variable which is drawn from a normally distributed population (see the previous section for ways to test this). For the examples below, you can alternatively use the **Statistics → Summaries, tables, and tests → Classical tests** menu.

First, we show an example of a one-sample t-test. Below, we test that the mean price for domestic cars is $15,000. Note that we can add "if" conditions to the **`ttest`** command (without that option, we would be testing the price for all cars in the dataset):

```
ttest Price == 15 if Origin == "US"
```

```
One-sample t test

Variable |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
   Price |      48     18.57292    1.128274    7.816914    16.30312    20.84271

    mean = mean(Price)                                        t =   3.1667
Ho: mean = 15                                 degrees of freedom =       47

    Ha: mean < 15               Ha: mean != 15                 Ha: mean > 15
 Pr(T < t) = 0.9986         Pr(|T| > |t|) = 0.0027          Pr(T > t) = 0.0014
```

From this analysis, we see that the mean price of US-made cars is about 18.5 thousand dollars, which is significantly different from our hypothesized mean of 15 thousand dollars (p-value = 0.003). Note that Stata also gives a 95% confidence interval of the mean price of US-made cars by default, and since it does not include our null hypothesis, it also tells us that we can reject it.

When conducting a two-sample t-test, you must test the assumption of equality of variances in the two groups that are being compared. If you have more than two groups that you want to compare, you must use an ANOVA (see next section) and also test that the variances are equal across *all* groups.

Below is an example of a two-sample t-test where we test the difference in city miles-per-gallon between domestic and foreign-made cars. Note that in the output of the **`ttest`** command does not include a test of equal variances, so we must run that first ourselves with the **`sdtest`** command:

```
sdtest CityMPG, by(Origin)
```

```
Variance ratio test

   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
      US |      48    20.95833      .57655    3.994455    19.79846    22.1182
  non-US |      44    23.97727    1.011294    6.708165     21.9378   26.01674
---------+--------------------------------------------------------------------
combined |      92    22.40217    .5879544     5.63946    21.23427   23.57007
------------------------------------------------------------------------------
    ratio = sd(US) / sd(non-US)                                   f =   0.3546
Ho: ratio = 1                                     degrees of freedom =  47, 43

    Ha: ratio < 1               Ha: ratio != 1                 Ha: ratio > 1
  Pr(F < f) = 0.0003         2*Pr(F < f) = 0.0006          Pr(F > f) = 0.9997
```

Since the two-tailed p-value is less than 0.05, we must reject the null hypothesis, which in this case is that the variances are equal. Therefore, we must include the **`unequal`** option at the end of our ttest statement which will adjust the degrees of freedom used in the analysis (Satterthwaite calculation) to correct for unequal variances. If our **`sdtest`** was not significant, we would use the command below without the **`unequal`** at the end:

```
ttest CityMPG, by(Origin) unequal
```

```
Two-sample t test with unequal variances

   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
      US |      48     20.95833      .57655    3.994455    19.79846    22.1182
  non-US |      44     23.97727    1.011294    6.708165     21.9378   26.01674
---------+--------------------------------------------------------------------
combined |      92     22.40217    .5879544     5.63946    21.23427   23.57007
---------+--------------------------------------------------------------------
    diff |           -3.018939    1.164098               -5.341348  -.6965311
------------------------------------------------------------------------------
    diff = mean(US) - mean(non-US)                                t =  -2.5934
Ho: diff = 0                     Satterthwaite's degrees of freedom =  68.8413

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 0.0058         Pr(|T| > |t|) = 0.0116          Pr(T > t) = 0.9942
```

Note that the top of this output reads "with unequal variances," where it would say "with equal variances" if we did not include the **`unequal`** statement in our command. This is a good check if you forget to test for equality of variances prior to running your t-test. From the p-value at the bottom center, we see that there is a significant difference between the city miles-per-gallon for domestic versus foreign cars. We can also see that the 95% confidence interval of the difference of the means does not contain zero.

4.3 ANOVA

You can use a one-way ANOVA if you want to test the difference in a continuous, normally-distributed variable among two or more groups. Similar to t-tests, you must also test the equality of variances across the groups you compare. Luckily, Stata automatically tests for this when you use an ANOVA command, so you do not have to remember to do that ahead of time.

There are two ways to run a one-way ANOVA in Stata. By using the **`oneway`** command, you will get the automatic test of the equality of variances. If you use the more common **`anova`** command, you will not get the assumption test by default. However, the **`oneway`** test does not output the residual sum of squares, which the **`anova`** command does.

Below we test if the weight of cars is equal among all types (compact, midsize, etc.). You can also use the **Statistics → Linear models → ANOVA/MANOVA → Analysis of variance and covariance** menu instead of running the command directly:

```
oneway Weight type
```

```
                        Analysis of Variance
    Source              SS         df      MS            F     Prob > F
------------------------------------------------------------------------
Between groups      24527831.6      5   4905566.31     57.95     0.0000
 Within groups      7279521.44     86   84645.5981
------------------------------------------------------------------------
    Total             31807353     91   349531.352

Bartlett's test for equal variances:  chi2(5) =  11.2319  Prob>chi2 = 0.047
```

The output tells us that the variances among the different types of cars are unequal. However, ANOVA's are somewhat robust against violations of this assumption, and since the p-value is very close to 0.05, we don't see a problem with the analysis (and therefore wouldn't suggest a non-parametric alternative to ANOVA).

The p-value for the ANOVA is <0.0001, meaning that there is a difference in weight among the different types of vehicles. In other words, we can reject the null hypothesis that *all* types of vehicles have equal mean weights. This does not necessarily mean that all types have different means from each other, but that there is at least one type that differs from the rest.

In order to get the marginal means, you must run the **`anova`** command. After running **`anova Weight type`**, you can use the **`margin`** command to get the marginal means of weight for each type of vehicle:

```
. margin type

Adjusted predictions                            Number of obs   =         92

Expression   : Linear prediction, predict()

------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        type |
           1 |   2312.857   63.48813    36.43   0.000     2188.423    2437.292
           2 |   2918.125   72.73479    40.12   0.000     2775.567    3060.683
           3 |   2899.643   77.75676    37.29   0.000     2747.242    3052.043
           4 |   3394.048   63.48813    53.46   0.000     3269.613    3518.482
           5 |   3695.455   87.72146    42.13   0.000     3523.524    3867.385
           6 |   3830.556   96.97972    39.50   0.000     3640.479    4020.632
------------------------------------------------------------------------------
```

To run a two-way ANOVA, which will test differences in two different categorical variables, you must use the **`anova`** command and specify two categorical variables after

the continuous dependent variable. Unlike most other statistical software packages, Stata will not automatically run this test if the categorical variable or variables are formatted as text. Therefore, to add our *Origin* variable to the model, which is coded as "US" or "non-US," we must first create a coded numeric variable that corresponds to those two values:

```
gen OriginDummy = (Origin == "US")
```

Now that we have "Origin_num," we can run the two-way ANOVA (note that by using the "##" in between our two factors, Stata will include both main effects as well as their interaction term):

```
. anova Weight type##OriginDummy

                          Number of obs =      92     R-squared     =  0.7917
                          Root MSE      = 286.009     Adj R-squared =  0.7660

                  Source | Partial SS    df       MS           F     Prob > F
-------------------------+----------------------------------------------------
                   Model |  25181453.4    10  2518145.34       30.78     0.0000
                         |
                    type |  21868541.6     5  4373708.32       53.47     0.0000
            OriginDummy |  1193.60259     1  1193.60259        0.01     0.9042
        type#OriginDummy |  651917.448     4  162979.362        1.99     0.1035
                         |
                Residual |  6625899.64    81  81801.2301
-------------------------+----------------------------------------------------
                   Total |    31807353    91  349531.352
```

From the above output, we can see that the origin of the car is not significant, and neither is the interaction between origin and type. However, *type* is significant (p-value<0.0001), as well as the overall model, which can be found on the top line of the ANOVA table.

4.4 ANCOVA

Suppose you have a continuous variable that you need to control for within your ANOVA procedure. Such a model is referred to as an ANCOVA, since you are adding a covariate, or continuous independent variable, to the model. The way to run an ANCOVA is very simple, but you must remember one important point: you need to tell Stata that a variable in your **anova** statement is continuous or it will treat it as another categorical factor.

You denote continuous independent variables within the **anova** command by placing a "c." in front of them. In the example below, we run the one-way ANOVA where we see if a car's weight varies significantly based on what type it is, but while controlling for the size of its fuel tank:

```
. anova Weight type##c.FuelTank

                          Number of obs =        92     R-squared     =  0.8809
                          Root MSE      =  217.562     Adj R-squared =  0.8646

                  Source |  Partial SS    df       MS           F     Prob > F
-------------------------+----------------------------------------------------
                   Model |  28020685.2    11  2547335.02       53.82     0.0000
                         |
                    type |  525637.514     5  105127.503        2.22     0.0601
                FuelTank |  1993470.03     1  1993470.03       42.12     0.0000
            type#FuelTank |  287408.282     5  57481.6565        1.21     0.3100
                         |
                Residual |  3786667.79    80  47333.3474
-------------------------+----------------------------------------------------
                   Total |    31807353    91  349531.352
```

We can see that type remains very significant (p-value < 0.0001) even when we control for the size of the fuel tank. Note that we tested for the interaction between *type* and *FuelTank*, which we must do whenever we run an ANCOVA. One of the assumptions of an ANCOVA test is that the covariate does not vary among the groups of the categorical factor or factors. Since the interaction term is not significant (p-value=0.31), we see that the assumption is not violated.

If the interaction were significant, we would need to use a different approach to analyze this data, such as a mixed model. However, since it is not significant, we now run the ANCOVA without the interaction term to get our final result:

```
. anova Weight type c.FuelTank

                          Number of obs =        92     R-squared     =  0.8719
                          Root MSE      =    218.93     Adj R-squared =  0.8629

                  Source |  Partial SS    df       MS           F     Prob > F
-------------------------+----------------------------------------------------
                   Model |  27733276.9     6  4622212.82       96.44     0.0000
                         |
                    type |  2327164.84     5  465432.969        9.71     0.0000
                FuelTank |  3205445.36     1  3205445.36       66.88     0.0000
                         |
                Residual |  4074076.08    85  47930.3068
-------------------------+----------------------------------------------------
                   Total |    31807353    91  349531.352
```

Section 5: Linear Regression

5.1 Introduction

Stata 12 has the capability of running a great variety of different types of regression models (linear and non-linear, parametric and non-parametric, etc.). This section focuses on linear regression, both with a single independent variable and with multiple independent variables.

5.2 Simple Linear Regression

Let us model the linear relationship between engine size of the vehicles and their city miles-per-gallon. Below is the code for running the linear regression, but you can alternatively go to **Statistics → Linear models and related → Linear regression**:

```
. regress CityMPG EngineSize
```

Source	SS	df	MS		
Model	1455.02248	1	1455.02248	Number of obs	= 92
Residual	1439.09709	90	15.9899676	F(1, 90)	= 91.00
				Prob > F	= 0.0000
Total	2894.11957	91	31.8035117	R-squared	= 0.5028
				Adj R-squared	= 0.4972
				Root MSE	= 3.9987

CityMPG	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
EngineSize	-3.839187	.402465	-9.54	0.000	-4.638754	-3.039619
_cons	32.62192	1.149601	28.38	0.000	30.33804	34.9058

Stata outputs quite a lot of information for even this simple model. At the top, we see an ANOVA table of the entire model. To the right of that are some fit statistics, including the overall F-test corresponding to the ANOVA table and R-squared. The bottom table presents the estimated coefficients of the independent variable *EngineSize* and the intercept, their standard errors, and the t-statistic and associated p-values. Finally, the table includes a 95% confidence interval for each estimate.

We can interpret these results to say that a vehicle's engine size does significantly impact the city miles-per-gallon. For each additional unit increase in engine size, the vehicle's city miles-per-gallon decreases by roughly 3.84 units.

There are many options available for the **`regress`** command, which are described under **`help regress`**.

5.3 Multiple Linear Regression

Adding more independent variables into your linear model is as simple as listing them in your statement (or adding them in the window if you are using the drop-down menus). For example, let's also consider the horsepower and origin of the vehicles in estimating the city miles-per-gallon.

One drawback to Stata is that it does not automatically create a dummy variable (or set of dummy variables) when you use a categorical independent variable. It will not allow any string values in a **`regress`** command (or most other regression functions or anova procedures, as we saw in Section 4.3). Therefore, you must create your own numeric version of any categorical variable you wish to put in the model, which we show with the following example for *Origin*:

```
gen OriginDummy = (Origin == "US")
```

Now, we can use *OriginDummy* in our model, but because it represents a categorical variable, we tell Stata this by including a "i." in front of it. If this variable had more than two categories, then Stata would output the estimates of each category with respect to the reference category (whichever has the lowest code, usually zero) in the bottom table of the output.

```
. regress CityMPG EngineSize Horsepower i.OriginDummy

      Source |       SS       df       MS              Number of obs =      92
-------------+------------------------------           F(  3,    88) =   36.72
       Model |  1608.95304     3  536.317679            Prob > F      =  0.0000
    Residual |  1285.16653    88  14.6041651           R-squared     =  0.5559
-------------+------------------------------           Adj R-squared =  0.5408
       Total |  2894.11957    91  31.8035117           Root MSE      =  3.8215

------------------------------------------------------------------------------
     CityMPG |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  EngineSize |  -2.272318   .6551444    -3.47   0.001    -3.57428   -.9703563
  Horsepower |  -.0387991   .0120077    -3.23   0.002    -.0626618   -.0149363
1.OriginDu~y |  -.7319677   .9276481    -0.79   0.432    -2.575473    1.111538
       _cons |   34.38241   1.226258    28.04   0.000     31.94548    36.81934
------------------------------------------------------------------------------
```

From the output, we can see that while controlling for engine size and horsepower, the origin of the car does not significantly impact the city miles-per-gallon (p-value = 0.43).

5.4 Marginal Means

A common way to further explore the effect of a categorical independent variable is to look at the marginal means for each group. This is easily done with the **margins** command, which can be run after various types of regression commands and will report on the most recently outputted model:

```
. margins OriginDummy

Predictive margins                              Number of obs   =         92
Model VCE    : OLS

Expression   : Linear prediction, predict()
```

	Margin	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
OriginDummy						
0	22.78407	.6268872	36.34	0.000	21.55539	24.01275
1	22.0521	.5962995	36.98	0.000	20.88338	23.22083

Following what we saw in the original regression output, the city miles-per-gallon do not significantly differ between US and foreign cars. This is evident in that the 95% confidence intervals for the groups overlap with each other.

You can also get the marginal means for continuous independent variables. Although this is usually not very useful in regular linear regressions, it can be in nonlinear regression models, such as logistic, and the command is the same regardless of the type of model you run:

```
margins , at(Horsepower=(120(10)220)) vsquish
```

This command will output the marginal mean city miles-per-gallon of cars with values of *Horsepower* between 120 and 220, in increments of 10. The **vsquish** option just suppresses the empty space in the output and makes it easier to read:

```
. margins , at (Horsepower=(120(10)220)) vsquish

Predictive margins                              Number of obs   =         92
Model VCE    : OLS

Expression   : Linear prediction, predict()
1._at        : Horsepower      =         120
2._at        : Horsepower      =         130
3._at        : Horsepower      =         140
4._at        : Horsepower      =         150
5._at        : Horsepower      =         160
6._at        : Horsepower      =         170
7._at        : Horsepower      =         180
8._at        : Horsepower      =         190
9._at        : Horsepower      =         200
10._at       : Horsepower      =         210
11._at       : Horsepower      =         220
```

	Delta-method Margin	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_at						
1	23.29582	.4850061	48.03	0.000	22.34522	24.24641
2	22.90783	.4280543	53.52	0.000	22.06886	23.7468
3	22.51984	.4000837	56.29	0.000	21.73569	23.30399
4	22.13185	.4071121	54.36	0.000	21.33392	22.92977
5	21.74385	.4474936	48.59	0.000	20.86678	22.62093
6	21.35586	.5134177	41.60	0.000	20.34958	22.36214
7	20.96787	.5964744	35.15	0.000	19.79881	22.13694
8	20.57988	.690509	29.80	0.000	19.22651	21.93326
9	20.19189	.7916189	25.51	0.000	18.64035	21.74344
10	19.8039	.8974159	22.07	0.000	18.045	21.5628
11	19.41591	1.006423	19.29	0.000	17.44336	21.38846

The top portion of the output specifies the values of *Horsepower* at which the predicted means are being calculated and the bottom table contains the actual estimates at each of those intervals. You can see that for each increase of 10 in horsepower, the mean city miles-per-gallon decreases by about 0.39, which is equal to 10 times the coefficient estimate in the original regression output.

Section 6: Conclusion

Hopefully this tutorial has taught you how to run common statistical procedures with Stata 12 and what options are available to test assumptions and make interpretations easier to understand. Stata has the capability of running more complex models, including multilevel models, which is described in our "Multilevel Modeling Tutorial."

If you have any questions on the material presented here, or about other procedures in Stata that might be more appropriate for your data, please feel free to contact us at stats@ssc.utexas.edu. If you have a question about Stata or other statistical software packages, feel free to set up an appointment with one of our consultants by visiting http://ssc.utexas.edu/consulting/free-consulting.