Stata 10 Tutorial 3

TOPIC: **The Basics of OLS Estimation in *Stata***

DATA: **auto1.dta** (the *Stata*-format data file you created in *Tutorial 1*)

or

auto2.dta (the *Stata*-format data file you created in *Tutorial 2*)

TASKS: ***Stata Tutorial 3*** is intended to introduce you to the *Stata* **regress** command for ***OLS (Ordinary Least Squares) estimation of linear regression models***. It also demonstrates several ***post-estimation commands*** that are often used with the **regress** command, and other *Stata* commands that are useful in displaying and using the results of the **regress** command.

- The ***Stata* commands** introduced in this tutorial are:

codebook	Displays properties of variables in the current data set.
regress	Performs OLS estimation of linear regression models.
_b[*varname*]	Contains the ***coefficient estimate*** for the regressor ***varname***.
_se[*varname*]	Contains the ***standard error*** of the coefficient estimate for the regressor ***varname***.
e( )	Saves selected results from most recent **regress** command.
vce	Displays estimated covariance matrix of coefficient estimates.
matrix get	Accesses coefficient estimates and the covariance matrix.
display	Computes and displays the values of algebraic expressions.
scalar	Defines the contents of scalar variables.
scalar list	Lists the names and values of currently-defined scalar variables.
scalar drop	Eliminates previously-defined scalars from memory.
matrix	Defines matrices and performs matrix computations.
matrix list	Lists contents of a vector or matrix.

NOTE: *Stata* commands are *case sensitive*. All *Stata* commands must be typed in the Command window in **lower case letters**.

HELP: *Stata* has an extensive on-line **Help** facility that provides fairly detailed information (including examples) on all *Stata* commands. Students should become familiar with the *Stata* on-line **Help** system. In the course of doing this tutorial, take the time to browse the **Help** information on some of the above *Stata* commands. To access the on-line **Help** for any *Stata* command:

- choose (click on) **Help** from the *Stata* main menu bar
- click on **Stata Command** in the **Help** drop down menu
- type the full name of the *Stata* command in the *Stata* command dialog box and click **OK**

❑ Preparing for Your *Stata* Session

In *Stata Tutorial 1*, you created and saved on your own diskette the *Stata*-format data set **auto1.dta**. In *Stata Tutorial 2*, you created and saved on your own diskette the *Stata*-format data set **auto2.dta**. Either of these two *Stata*-format data sets is completely adequate for doing *Stata Tutorial 3*.

Before beginning your *Stata* session, use Windows Explorer to copy the *Stata*-format data set **auto1.dta** or **auto2.dta** to the *Stata* ***working directory*** on the C:-drive or D:-drive of the computer at which you are working.

- **On the computers in Dunning 350**, the default *Stata* working directory is usually **C:\data**.

- **On the computers in MC B111**, the default *Stata* working directory is usually **D:\courses**.

❑ Start Your *Stata* Session

To start your *Stata* session, double-click on the ***Stata 10*** **icon** in the Windows desktop.

After you double-click the ***Stata 10*** **icon**, you will see the now familiar screen of four *Stata* windows.

❑ Record Your *Stata* Session – log using

To record your *Stata* session, including all the *Stata* commands you enter and the results (output) produced by these commands, make a **.log** file named **351tutorial3.log**. To open (begin) the **.log** file **351tutorial3.log**, enter in the Command window:

```
log using 351tutorial3.log
```

This command opens a text-format file called **351tutorial3.log** in the current *Stata* working directory. Remember that once you have opened the **351tutorial3.log** file, a copy of all the commands you enter during your *Stata* session and of all the results they produce is recorded in that **351tutorial3.log** file.

An alternative way to open the **.log** file **351tutorial3.log** is to click on the **Log** button; click on **Save as type:** and select **`Log (*.log)`**; click on the **File name:** box and type the file name **351tutorial3**; and click on the **Save** button.

❑ Loading a *Stata*-Format Data Set into *Stata* – use

Load, or read, into memory the data set you are using. To load the *Stata*-format data file **auto1.dta** into memory, enter in the Command window:

```
use auto1
```

This command loads into memory the *Stata*-format data set **auto1.dta**.

Alternatively, to load the *Stata*-format data file **auto2.dta** into memory, enter in the Command window:

```
use auto2
```

This command loads into memory the *Stata*-format data set **auto2.dta**.

❑ Summarizing and Viewing the Current Data Set – describe *and* list

To summarize the contents of the current data set, use the **describe** command. Recall that the **describe** command displays a summary of the contents of the current data set in memory, which in this case is either **auto1.dta** or **auto2.dta**.

- To summarize the contents of the current data set in memory, enter in the Command window:

  ```
  describe
  ```

  The display generated by this **describe** command may indicate that the current data set is sorted by the indicator variable **foreign**.

- If the current data set is not sorted by the indicator variable **foreign**, sort it and inspect the results by entering the following commands:

  ```
  sort foreign
  describe
  ```

- To see directly how the observations in the current data set are ordered, enter the command:

  ```
  list foreign price weight mpg
  ```

  Notice that all observations for domestic cars occur before the observations for foreign cars.

❑ Calculating descriptive summary statistics – summarize

Recall that the **summarize** command calculates and displays descriptive summary statistics for some or all of the *numeric variables* in the current data set.

- To calculate and display basic descriptive summary statistics for all variables and all observations in the current data set, enter in the Command window:

  ```
  summarize
  ```

❑ Displaying variable characteristics – codebook

The **codebook** command displays the characteristics, or properties, of any variable(s) in the current data set.

- To display the characteristics of the variable **make**, enter in the Command window:

  ```
  codebook make
  ```

  Examine carefully the screen display: it tells you that **make** is a *string variable* with maximum length of 17 characters, that **make** takes 74 distinct values in the data set, and that there are no missing values for the variable **make**. It also displays some examples of the string values taken by the variable **make**.

- Display the characteristics of some of the *numeric variables* in the data set by entering in the Command window:

  ```
  codebook price mpg weight foreign
  ```

  Again, examine carefully the screen display for each of these *numeric variables*. How is the displayed information for the *indicator variable* **foreign** different from that for the other numeric variables?

❑ OLS estimation of linear regression models – regress

The basic *Stata* command for ***OLS estimation* of linear regression models** is the **regress** command. Subsequent *Stata* tutorials will help you learn how to use and interpret the many features of the **regress** command. This section introduces you to the basic features of the **regress** command.

- To estimate by OLS the simple linear regression model given by the PRE

  $$Y_i = price_i = \beta_0 + \beta_1 weight_i + u_i \qquad (1)$$

  for the full sample of observations in the current data set, enter in the Command window:

```
regress price weight
```

Examine the results of this command. See what elements of the results displayed by the **regress** command you can identify.

- Now estimate by OLS the simple linear regression model given by the PRE

$$\text{price}_i = \beta_0 + \beta_1 \text{mpg}_i + u_i \qquad (2)$$

for the full sample of observations in the current data set. Enter in the Command window:

```
regress price mpg
```

Again, examine the results of this **regress** command. Most of the displayed results of the **regress** command will be meaningless to you now, but you will soon learn what they all mean.

❑ Options to use with the *regress* command

This section introduces you to some **regress** command ***options*** and to some other commands that are often used with the **regress** command.

Basic Syntax of **regress** ***Command***

[by *varname*:] regress *depvar varlist* [if *exp*] [in *range*] [, *options***]**

where

depvar is the user-supplied name of the regressand, or dependent variable, Y_i;

varlist is a list of the user-supplied names of the regressors, or independent variables, X_{1i}, X_{2i}, ..., X_{ki}.

Optional components of the **regress** command are enclosed in square brackets **[ ]**, which are not typed as part of the command.

Two of the ***options*** available with the **regress** command are:

level(#) specifies the confidence level **#**, in percent, to be used for computing confidence intervals of the regression coefficients;

noconstant suppresses the constant term in the regression function; i.e., sets the intercept coefficient β_0 equal to zero.

❑ Computing OLS Regression Equations for Subsets of Sample Observations -- the *if* option on *regress*

The **if *option*** can be used with the **regress** command to compute separate OLS sample regression equations for specified subsamples of observations. For example, suppose you wish to compute separate OLS estimates of regression equation (1) for domestic and foreign cars. Recall that **foreign** is an *indicator variable* that distinguishes between foreign and domestic cars; **foreign** is defined to equal 1 for foreign cars, and 0 for domestic cars.

- To compute OLS estimates of regression equation (1) for the **subsample of *domestic* cars**, enter in the Command window:

```
regress price weight if foreign==0
```

- Similarly, to compute OLS estimates of regression equation (1) for the **subsample of *foreign* cars**, enter in the Command window:

```
regress price weight if foreign==1
```

- An alternative way to estimate separate OLS regression equations for specified subsamples of observations is to use the **bysort *varname*: *option*** in front of the **regress** command. For example, to estimate regression equation (1) separately for the subsamples of foreign and domestic cars, enter the commands:

```
bysort foreign: regress price weight
```

This command does the same thing as the two **regress** commands that immediately preceded it. Note that the **bysort** option checks to see if the current dataset in memory is sorted according to the values of the indicator variable **foreign**: if it is not sorted, then the **bysort** option performs the sort before executing the ensuing **regress** command; if it is already sorted by **foreign**, then *Stata* immediately proceeds to execute the **regress** command.

❑ Computing Confidence Intervals for the Regression Coefficients – level(#)

The **level(*#*)** option on the **regress** command can be used to change the ***confidence level*** used in constructing two-sided confidence intervals for the regression coefficients. The ***#*** is set to the desired confidence level, such as 90 or 99. If the **level(*#*)** option is not specified, the **regress** command computes **two-sided 95 percent confidence intervals** for each regression coefficient; the default value of the confidence level ***#*** is therefore 95 (or $1 - \alpha = 0.95$). To compute two-sided confidence intervals for confidence levels other than 95 percent, use the **level(*#*)** option on the **regress** command.

- To compute ***two-sided 90 percent confidence intervals*** for each regression coefficient, enter *either* of the commands:

```
regress price weight, level(90)
regress, level(90)
```

- To compute ***two-sided 99 percent confidence intervals*** for each regression coefficient, enter *either* of the commands:

```
regress price weight, level(99)
regress, level(99)
```

Compare the two-sided confidence intervals for these three difference confidence levels. For which confidence level are the confidence intervals widest? For which confidence level are the confidence intervals narrowest?

❑ Accessing Coefficient Estimates and Standard Errors – _b[…] *and* _se[…]

<u>Basic Syntax</u>:

_b[*varname*] (or its synonym **_coef[*varname*]**);
_se[*varname*]

where ***varname*** is the user-supplied variable name for one of the regressors in the most recent **regress** command.

- **Accessing coefficient estimates. _b[*varname*]**, or its synonym **_coef[*varname*]**, contains the ***coefficient estimate*** for the regressor ***varname*** in the most recent **regress** command. Thus, **_b[weight]** and **_coef[weight]** both contain the value of the OLS estimate $\hat{\beta}_1$ of the regression coefficient on the regressor **weight** in the previous OLS regression.

- Use either of the following **display** commands to simply display in the Results window the value of the OLS slope coefficient estimate $\hat{\beta}_1$:

```
display _b[weight]
display _coef[weight]
```

- The *Stata* system variable **_cons** is always equal to the number 1, and refers to the intercept coefficient estimate when used with **_b[ ]** and **_coef[ ]**. Thus, **_b[_cons]** and **_coef[_cons]** both contain the value of the OLS estimate $\hat{\beta}_0$ of the intercept coefficient β_0 in the previous OLS regression. To display in the Results window the value of the OLS slope coefficient estimate $\hat{\beta}_0$, enter either of the following commands in the Command window:

```
display _b[_cons]
display _coef[_cons]
```

Note: **_b[ ]** and **_coef[ ]** are equivalent; that is, they are synonyms. Therefore,

_b[_cons] = **_coef[_cons] =** $\hat{\beta}_0$ **=** the OLS intercept coefficient estimate;

b[weight] = ** coef[weight] =** $\hat{\beta}_1$ **=** the OLS slope coefficient estimate for the regressor **weight**.

- **Accessing estimated standard errors. _se[*varname*]** contains the ***estimated standard error*** of the coefficient estimate for the regressor ***varname*** in the most recent **regress** command. Thus, **_se[weight]** contains $\widehat{se}(\hat{\beta}_1)$, the estimated standard error of the OLS slope coefficient estimate $\hat{\beta}_1$. Similarly, **_se[_cons]** contains $\hat{se}(\hat{\beta}_0)$, the estimated standard error of the OLS intercept coefficient estimate $\hat{\beta}_0$.

- Enter the following **display** commands to display in the Results window the values of $\hat{se}(\hat{\beta}_1)$ and $\hat{se}(\hat{\beta}_0)$ from the previous OLS regression:

```
display _se[weight]
display _se[_cons]
```

❑ Saving Coefficient Estimates and Standard Errors – scalar

Stata stores the values of coefficient estimates in **_b[ ]** (or in **_ coef[ ]**) and the values of estimated standard errors in **_se[ ]** only temporarily – that is until another model estimation command such as **regress** is entered.

To save the values of coefficient estimates and their estimated standard errors for subsequent use, you can use **scalar** commands to assign names to these values.

Basic Syntax:

scalar *scalar_name* = *exp*

where ***scalar_name*** is the user-supplied name for the scalar and ***exp*** is an algebraic expression or function.

- The following **scalar** commands name and save the values of the ***OLS coefficient estimates*** $\hat{\beta}_0$ and $\hat{\beta}_1$ from the most recent **regress** command, which performed OLS estimation of regression equation (1). Enter the commands:

```
scalar b0 = _b[_cons]
scalar b1 = _b[weight]
```

- The following **scalar** commands name and save the values of the ***estimated standard errors*** for the OLS coefficient estimates $\hat{\beta}_0$ and $\hat{\beta}_1$. Enter the commands:

```
scalar seb0 = _se[_cons]
scalar seb1 = _se[weight]
```

- You may also wish to generate the ***estimated variances*** of the OLS coefficient estimates $\hat{\beta}_0$ and $\hat{\beta}_1$, which are equal to the squares of the corresponding estimated standard errors. Enter the following **scalar** commands to do this:

```
scalar varb0 = seb0^2
scalar varb1 = seb1^2
```

To display the values of scalar variables, use the **scalar list** command. The **scalar list** command for scalars is the analog of the **list** command for variables.

- To list the values of *all* currently-defined scalars, enter either of the following commands:

```
scalar list _all
scalar list
```

- To list only the values of the scalars **b0** and **b1**, enter the following command:

```
scalar list b0 b1
```

❑ Displaying the variance-covariance matrix of the coefficient estimates – vce

- You can display the ***variance-covariance matrix*** for the OLS coefficient estimates from the most recent **regress** command. This matrix contains the estimated variances of the OLS coefficient estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ in the diagonal cells, and the estimated covariance of $\hat{\beta}_0$ and $\hat{\beta}_1$ in the off-diagonal cells. Enter in the Command window the following two commands:

```
vce
matrix list e(V)
```

Examine the display. The format of the estimated variance-covariance matrix for the coefficient estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ is as follows:

$$\begin{bmatrix} \hat{\text{Var}}(\hat{\beta}_1) & \hat{\text{Cov}}(\hat{\beta}_1, \hat{\beta}_0) \\ \hat{\text{Cov}}(\hat{\beta}_0, \hat{\beta}_1) & \hat{\text{Var}}(\hat{\beta}_0) \end{bmatrix} = \begin{bmatrix} \hat{\text{Var}}(\hat{\beta}_1) & \hat{\text{Cov}}(\hat{\beta}_0, \hat{\beta}_1) \\ \hat{\text{Cov}}(\hat{\beta}_0, \hat{\beta}_1) & \hat{\text{Var}}(\hat{\beta}_0) \end{bmatrix}$$

where the second equality reflects the fact the variance-covariance matrix is symmetric, which in turn follows from the fact that $\text{C}\hat{\text{o}}\text{v}(\hat{\beta}_0, \hat{\beta}_1) = \text{C}\hat{\text{o}}\text{v}(\hat{\beta}_1, \hat{\beta}_0)$. Note that the following definitions are used:

$\text{V}\hat{\text{a}}\text{r}(\hat{\beta}_0)$ = the estimated variance of $\hat{\beta}_0$;
$\text{V}\hat{\text{a}}\text{r}(\hat{\beta}_1)$ = the estimated variance of $\hat{\beta}_1$;
$\text{C}\hat{\text{o}}\text{v}(\hat{\beta}_0, \hat{\beta}_1) = \text{C}\hat{\text{o}}\text{v}(\hat{\beta}_1, \hat{\beta}_0)$ = the estimated covariance of $\hat{\beta}_0$ and $\hat{\beta}_1$.

❑ Saving the coefficient vector and variance-covariance matrix – matrix

To save the entire vector of OLS coefficient estimates and the associated variance-covariance matrix, you can use the **matrix get** or **matrix** command.

Basic Syntax:

matrix *matname* = get(*internal_Stata_matrix_name*)

where ***matname*** is the user-supplied name given to the matrix or vector, and ***internal_Stata_matrix_name*** is the internal name that *Stata* gives to the matrix or vector.

1. The internal name that *Stata* gives to the ***vector of OLS coefficient estimates*** is **_b** or **e(b)**.

2. The internal name that *Stata* gives to the ***estimated variance-covariance matrix*** is **VCE** or **e(V)**.

- **To save the *vector of OLS coefficient estimates*** and give it the name **bvec**, enter the following command:

```
matrix bvec = get(_b)
```

- An alternative way to save the vector of OLS coefficient estimates is to use a **matrix** command and the **e(b)** matrix function. To save the OLS coefficient vector and give it the name **b**, enter the following command:

```
matrix b = e(b)
```

- Use the following **matrix list** commands to display the saved coefficient vectors **bvec** and **b** (which are, of course, identical):

```
matrix list bvec
matrix list b
```

- **To save the *estimated variance-covariance matrix*** and give it the name **V1**, enter the following **matrix get** command:

```
matrix V1 = get(VCE)
```

- Alternatively, a simple **matrix** command and the **e(V)** matrix function can be used to save the estimated variance-covariance matrix and give it the name **V2**. Enter the following **matrix** command:

```
matrix V2 = e(V)
```

- The following **matrix list** commands can be used to display the estimated variance-covariance matrices **V1** and **V2** (which are identical):

```
matrix list V1
matrix list V2
```

❑ Displaying and Saving Selected Regression Results – e( )

Stata temporarily stores selected results from the most recently executed **regress** command in the **e( )** function. The contents of the **e( )** function change each time a new **regress** command is executed.

Let N denote the number of sample observations on which the last **regress** command was executed (here N = 74), and K the total number of estimated regression coefficients (K = 2 for the simple regression model (1)). The following scalars are saved in **e( )** functions after each regress command is executed:

1. number of observations ≡ N
2. explained sum of squares ≡ ESS
3. degrees of freedom for ESS ≡ K−1
4. residual sum of squares ≡ RSS
5. degrees of freedom for RSS ≡ N−K
6. ANOVA F-statistic ≡ F[K−1, N−K]
7. R-squared ≡ R^2
8. adjusted R-squared ≡ $\overline{R}^2$
9. root mean square error = $\hat{\sigma}$

- Enter again the **regress** command for estimating PRE (1):

```
regress price weight
```

- To display all of the saved results for the most recent **regress** command, enter the following command:

```
ereturn list
```

Examine carefully the results of this command. It displays all the results that *Stata* temporarily saves from execution of a **regress** command.

- To ***display*** (but not save) the current contents of individual scalar **e( )** functions for the most recent **regress** command, enter the following **display** commands:

```
display e(N)
display e(mss)
display e(df_m)
display e(rss)
display e(df_r)
display e(F)
display e(r2)
display e(r2_a)
display e(rmse)
```

- To ***save*** the current contents of **e( )** for the most recent **regress** command as named scalars, enter the following **scalar** commands:

```
scalar N = e(N)
scalar ESS = e(mss)
scalar dfESS = e(df_m)
scalar RSS = e(rss)
scalar dfRSS = e(df_r)
scalar Fstat = e(F)
scalar Rsq = e(r2)
scalar adjRsq = e(r2_a)
scalar sigma = e(rmse)
```

- To display the values of the scalars created by the foregoing commands, enter the following **scalar list** command:

```
scalar list N ESS dfESS RSS dfRSS Fstat Rsq adjRsq sigma
```

- You can also use the **scalar** command to save other results of the **regress** command. For example, enter the following commands to create and display some additional scalars for the sample regression equation obtained by OLS estimation of equation (1):

```
scalar K = dfESS + 1
scalar TSS = ESS + RSS
scalar dfTSS = N - 1
scalar list N K TSS ESS RSS dfTSS dfESS dfRSS

scalar Rsq1 = ESS/TSS
scalar Rsq2 = 1 - RSS/TSS
scalar list Rsq Rsq1 Rsq2

scalar sigmasq = sigma^2
scalar sigmasq1 = RSS/dfRSS
scalar sigma1 = sqrt(sigmasq1)
scalar list sigma sigma1 sigmasq sigmasq1
```

- You have just saved a lot of scalars. To display or list *all* of the currently-defined scalars, enter either of the following commands:

```
scalar list
scalar list _all
```

Compare the results of these two **scalar** commands; they should be identical.

- Now use the **scalar drop** command to drop some of the redundant scalars you have created. Enter the command:

```
scalar drop Rsq2 sigmasq1 sigma1
```

❑ Preparing to End Your *Stata* Session

Before you end your *Stata* session, you should do two things.

- First, if you wish to save the current data set (although it is entirely your choice, as you will not need it for future tutorials), use the following **save** command to save the current data set as *Stata*-format data set **auto3.dta**:

```
save auto3
```

- Second, close the **.log** file you have been recording. Enter the command:

```
log close
```

❑ End Your *Stata* Session -- exit

You are now ready to conclude your *Stata* session.

- **To end your *Stata* session**, use the **exit** command. Enter the command:

```
exit    or      exit, clear
```

❑ Cleaning Up and Clearing Out

After returning to Windows, you should copy all the files you have used and created during your *Stata* session to your own portable electronic storage device. These files will be found in the ***Stata* working directory**, which is usually **C:\data** on the computers in Dunning 350. There are two files you will want to be sure you take with you: the *Stata*-format data set **auto3.dta**; and the *Stata* log file **351tutorial3.log**. Use the Windows **copy** command to copy any files you want to keep to your own portable electronic storage device (e.g., flash memory stick) in the E:-drive (or to a diskette in the A:-drive).

Finally, **as a courtesy to other users** of the computing classroom, please delete all the files you have used or created from the *Stata* working directory.