EXST 7014, Lab 05 Multiple Linear Regression: Matrix Algebra and Extra SS

OBJECTIVES

1. Fit multiple regression models using lm in R.
2. Familiarize yourselves with multiple regression using matrix Algebra and Extra SS.
3. Detect multicollinearity in data.

In SLR, only a single dependent variable can be regressed on a single independent variable. multiple regression however, a number of variables can be involved and regressed on one another (model: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$).
The overall test of hypothesis of multiple linear regression is H_0: $\beta_1 = \beta_2 = \cdots = \beta_p = 0$ v.s. H_1: at least one $\beta_i \neq 0$. Rejection of H_0 implies that at least one of the regressors, X_1, X_2, . . ., X_p, contributes significantly to the model. As in SLR, the F-test is used to test this hypothesis. The assumptions for the multiple regression are the same for SLR. Thus, the same sets of analysis, such as residual plot, normality test and diagnostic statistics are used to evaluate the assumptions.

In this lab, we will use various R commands to perform multiple linear regression. You are required to identify various types of sum-of-squares (TypeI, TypeII, TypeIII), and the components in X'X matrix (cross products X'X, X'Y, and Y'Y) and (X'X)-1 matrix (X'X inverse, parameters and SSE) by using lm; to understand that F-Test and T-test give the same results for parameter estimates test of hypothesis

In multiple regression, when two independent variables are highly correlated, the problem occurs because X'X matrix could not be inverted. This problem is called multicollinearity, which could cause large fluctuations of the regression coefficients and inflated variance estimates. Therefore, the regression coefficient estimates are not useful. In this lab, you will also get familiar with the statistics (sequential parameter estimates, variance inflation factor (VIF) and condition index), that evaluate the multicollinearity.

Part I. Housekeeping Statements

Before we dive into the main part of the code it is good to create a pre-amble in which we will load all the necessary packages for R to execute the following tasks. Since we will be graphing the scatterplots we need the library "ggplot2". If you have it installed already great. If not, you can install it using the "packages" tab on the bottom right panel. Click install, and put the name of the package you want on the "install packages" window that pops up. The default setting is installing the packages from the CRAN repository where most "mainstream" packages can be found.

To activate the package, use the following command:

```
library(ggplot2)
```

We will also make use of the library olsrr to handle more complicated regression diagnostics. To activate it use the command:

```
library(olsrr)
```

Dataset
The data is from your textbook, chapter 7, problem 6 and you can attain it through the link: http://www.stat.lsu.edu/exstweb/statlab/datasets/fwdata97/FW07P06.txt.

The data set is from Chapter 8, Problem 3 in your textbook (Table 8.24). It's the results of a test for the strength of asphalt concrete mix. The test consisted of applying a compressive force on the top of different sample specimens. Two responses were collected: the stress and stain at which a sample specimen failed. The factors relate to mixture proportions, rates of speed at which the force was applied, and ambient temperature. Higher values of response variables indict stronger materials.

The variables are:
X1: percent binder (the amount of asphalt in the mixture);
X2: loading rate (the speed at which the force is applied);
X3: the ambient temperature;
Y1: the stress at which the sample specimen failed;
Y2: the strain at which the specimen failed.
We will perform a multiple regression using Y2 as dependent variable and X1, X2 and X3 as independent variables

Download and save the data in the same folder as the R-script you are working on. From the tab "Session" on R-Studio select "Set Working Directory" to "Source File location". This instructs R to look for the datafiles in the same directory as the R-script. Then use the command:

```
asphalt=read.table("data_lab5.txt", header = TRUE, sep = "", dec = ".")
```

The name of the dataset is then asphalt and it is read from the file **data_lab5.txt** using the command **read.table**. The argument **header=TRUE** instructs R to read the first line as the names of the variables corresponding to each column. The argument **sep=""** tells R to use spaces as separators between each column/variable. It can be changed to **","** if one is using comma separated values (CSV) and others. Finally, the **dec="."** forces R to use a dot **(.)** as a separator for decimal points. Again this can be changed to commas, semicolons and more. After you run the line above you should have a dataset named **asphalt** in your data environment on the up-right part of R-studio.

You can view the dataset by either clicking on it, or by using the command:

```
View(ashpalt)
```

Notice that the last two columns have some "." in place of empty values. Also, R thinks those columns are not numeric because of that. The following 2 lines of code will convert the appropriate columns to numbers (numeric) after they convert them first to characters (in order to incorporate the missing cells)

```
asphalt[,5] <-  as.numeric(as.character(asphalt[,5]))
asphalt[,6] <-  as.numeric(as.character(asphalt[,6]))
```

Since in this lab we will only be using this dataset, we can ask R to apply all our commands to that set, so we don't have to specify it repeatedly. The command to do that is:

```
attach(asphalt)
```

Part II. Multiple Linear Regression

R has many different procedures to compute that linear regression but the simples on is the lm method. The command is simply

```
model1 <- lm(y2 ~ x1+x2+x3,  x=T)
```

With this we are creating the object **model1** by invoking the **linear regression model (lm)** between the variables y2 and x1,x2,x3 (the first is the dependent and the others are the independent). The argument **data=ashpalt** tells R which dataset to use in order to find those variables.

The output of lm is very detailed and provides a lot of information, but we need to invoke various "anova" commands on it to obtain the corresponding sums of squares. First the command:

```
an1=anova(model1) # Sequential (type I) SS
```

Computes the classic analysis of variance on our regression and saves the output as an1. Then the command

```
SS1=an1$`Sum Sq`# List of Sums of Squares Type I
```

Creates a vector of the sums of squares (Type I) and saves them with the name SS1. To view it, just type **View(SS1)** or you can view the whole output of the anova analysis by typing **View(an1)**

Similarly we can explore the Type II Sums of squares with the following commands:

```
an2=Anova(model1, type="II") # Will help us compute the (type II) SS
SS2=an2$`Sum Sq`# List of Sums of Squares Type II
View(SS2)
View(an2)
```

And then the type III with:

```
an3=Anova(model1, type="III") # Will help us compute the (type III) SS
SS3=an3$`Sum Sq`# List of Sums of Squares Type II
View(SS3)
View(an3)
```

We can also find the extra SS by subtracting SSRegression of a reduced model from that of a full model. The full model includes all three X's while the reduced model only has X1 and X3. Therefore, the difference of their regression gives the type II SS for X2.
We should also try to analyze both regressions in a nested fashion using the commands:

```
model2 <- lm(y2 ~ x1+x3, x=T) # This is the reduced multilinear regression model.
an4=anova(model1, model2) # This computes the difference between the full model and the reduced one
```

We can view the results using the command **View(an4).**

The following commands create a set of diagnostics tools to check for collinearity amongst the input variables

```
ols_coll_diag(model1) # Collinearity diagnostics table
ols_vif_tol(model1) # VIF computation
ols_plot_obs_fit(model1) # Observed vs Predicted Plot
ols_plot_diagnostics(model1) # A panel of plots for regression diagnostics
```

ols_coll_diag(model1): Generates a number of collinearity diagnostics include condition indices. If the condition index exceeds 30, multicollinearity might be a problem.
ols_vif_tol(model1): the value of VIF is expected to be 1 if the regressors are not correlated. If the value is much greater than 2, serious problems are suggested.
ols_plot_obs_fit(model1) Graphs the observed vs the predicted plot.
The following link contains a lot of information about the package olsrr:
ols_plot_diagnostics(model1) Outputs multiple diagnostics tables that we visited in a previous lab.

For more information on the package olsrr view the following link:

https://cran.r-project.org/web/packages/olsrr

LAB ASSIGNMENT
Your assignment is to perform necessary analysis using R and answer the following questions (Please do not print all the output. Only print the graphs and tables that you think are relevant to your answers).
1. Use the function lm to fit the multiple linear regression model $Y2 = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \varepsilon$. Write down the estimated regression equation. What hypothesis does the F-test in ANOVA table test? What is the conclusion based on the ANOVA table?
2. Are the regression coefficients significant? Are they consistent with the F test in question 1?

3. Suppose that there is a specimen with $X_1 = 9$, $X_2 = 2.0$ and $X_3 = 83$. Estimate the strain of the specimen as well as its 95% confidence interval.
4. What are the assumptions of the fitted model? Evaluate those assumptions by using proper R output.
5. Fit necessary models to find SS (X3|X1, X2) and SS (X3|X2).
6. Is there indication of any possible problem of multicollinearity? Support your answer with proper R output.
7. Examine each F-test of parameter estimates with different type of SS. Which F-test results are identical to the results of T-test for parameter estimates? What is relationship between F-value and T-value?