EXST 7014, Lab 1: Review of R Programming Basics and Simple Linear Regression

OBJECTIVES

1. Prepare a scatter plot of the dependent variable on the independent variable
2. Do a simple linear regression in R
3. Get confidence intervals on the regression coefficients

Simple linear regression (SLR) is a common analysis procedure, used to describe the significant relationship a researcher presumes to exist between two variables: the dependent (or response) variable, and the independent (or explanatory) variable. This lab will familiarize you with how to perform SLR using the lm command in R.

We will first look at the data graphically using a scatter plot to assess the nature of the relationship between the variables. Based on this assessment, we will use SLR to fit a straight line model relating two variables. The line will be fir using least-squares.

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)
```

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 latitude (LAT) and the mean monthly range (RANGE), which is the difference between mean monthly maximum and minimum temperatures, are given for a selected set of US cities. The following program performs a SLR using RANGE as the dependent variable and LAT as the independent variable.

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:

```
fw07p06=read.table("data_lab1.txt", header = FALSE, sep = "", dec = ".")
```

The name of the dataset is then **fw07p06** and it is read from the file **data_lab1.txt** using the command **read.table**. The argument **header=FALSE** instructs R to see the first line of the table (the header) as yet another datapoint. If it is set to **header=TRUE** then R will read the first line as

the names of the variables corresponding to each column. Our dataset does not have names for the columns and hence we choose header to be false and we will then create the names of the columns ourselves.
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 **fw07p06** in your data environment on the up right part of R-studio.

To set the variable names we use the command **colnames** as follows:

```
colnames(fw07p06)=c("CITY","STATE","LAT","RANGE")
```

This forces R to rename (or just name) the 4 columns of the dataset using the labels inside the c() argument. Don't forget the c() is used to create an ordered list of elements or a vector.

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

```
View(fw07p06)
```

Creating a Scatter Plot
When performing a regression analysis, it is always advisable to look at scatter plots of the data in order to get an idea of the type of relationship that exists between the response variable and the explanatory variables. The following commands will create a scatterplot between the two variables.

```
plot1=ggplot(fw07p06, aes(x = LAT, y = RANGE)) +
        geom_point()+
        theme_classic()
plot1
ggsave("Scatter1.pdf",plot1)
```

Let's explore what each line does. The first one utilizes the command ggplot that is used to create good plots using R. Inside the argument of ggplot, we define the dataset which we will use to create the plot to be the dataframe fw07p06 we created earlier. Then we need to define the two variables in the aes (short for aesthetics) argument. The x axis for us will be the LAT variable (independent) and the y axis will be the RANGE variable (dependent).

The next line instructs R to create the pairs of values as geometrical points (geom_point) there are many features in the geom_point that are left to the user, like shape, size and color of the points. Check the following link for more information:

http://www.cookbook-r.com/Graphs/Scatterplots_(ggplot2)/

The Theme_classic() argument creates the plot with the default settings for ggplot 2. Again the possibilities of customization are endless. Examples of the different themes can be found here:

https://ggplot2.tidyverse.org/reference/ggtheme.html

As you might have noticed in the first line, we named our plot **"plot1"**. In order for us to see the plot we need to call it by its name in the script hence the line plot1. Finally, the command **ggsave(Scatter1.pdf,plot1)** saves the plot we created as a pdf file named Scatter1 in the same folder as the script we are working on. This is very handy if one wants to use that for another document. You can also save it as a jpg using the command **ggsave(Scatter1.jpg,plot1)**
Note that the "+" at the end of each line inside the ggplot argument are needed in order for R to consider all three of the lines as one thing.

The statement above will create a scatter plot of RANGE vs. LAT. The graph is not fancy, but is sufficient for getting an idea of how RANGE and LAT are related.
To create more professional graphics, you can explore the R cookbook and ggplot2 in depth here:

http://www.cookbook-r.com/

Fitting the Least-Squares Regression line Using SAS

Based on the scatter plot produced above, we will assume that an appropriate regression model relating RANGE and LAT is the liner model given by:

$$y = \beta_0 + \beta_1 \chi + \varepsilon$$

where Y is the RANGE, X is the LAT, and ε is a random error term that is normally distributed with mean 0 and unknown variances σ^2. β_0 is the estimate of Y-intercept, and β_1 is the estimate of the slope coefficient.

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

```
model1 <- lm(RANGE ~ LAT,  data=fw07p06)
```

With this we are creating the object **model1** by invoking the linear regression **model (lm)** between the variables RANGE and LAT (the first is the dependent and the second is the independent). The argument **data=fw07p06** tells R where to find those variables.

The output of lm is very detailed and provides more information than we will be using in this lab. For this lab, we want to focus on the table of parameter estimates, and the coefficient of determination, denoted by R^2, that is the measure of how well the Least-Squares line fits the data. In particular, R^2 gives the proportion of the variability in the dependent variable that is accounted by the Least-Squares line. The command:

```
summary(model1)
```

Gives us the following table:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -6.4793     5.5481  -1.168    0.249
LAT           0.7515     0.1438   5.228 4.79e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The coefficient of LAT (corresponding to β_1) is 0.7515 and the constant term, β_0 is then -6.4793. To find the R squared value we can look at the next output of the summary :

```
Residual standard error: 5.498 on 43 degrees of freedom
Multiple R-squared:  0.3886,   Adjusted R-squared:  0.3744
F-statistic: 27.33 on 1 and 43 DF,  p-value: 4.786e-06
```

The coefficient of determination is labeled Multiple R-squared in the output of summary lm. Also, if one works with more complicated models, the adjusted R-square is preferable, which is similar to R-square but penalizes the use of many explanatory variables. More on that later.
Notice that summary is a very useful generic R command that we will use for other processes as well.

To get the 100(1-α)% upper and lower confidence limits for the parameter estimates we use the command:

```
confint(model1)
```

Which yields

```
                2.5 %   97.5 %
(Intercept) -17.6681689 4.709508
LAT           0.4616029 1.041418
```

By default, the 95% limits are computed but we can change that using the following argument:

```
confint(model1,level = 0.90)
```

which yields:

```
                   5 %      95 %
(Intercept) -15.8061025 2.8474418
LAT           0.5098499 0.9931715
```

To finish up the analysis let’s beef up the plot we did earlier and include the regression line computed above with the confidence intervals using the code:

```
plot2=ggplot(fw07p06, aes(x = LAT, y = RANGE)) +
  geom_point()+
  geom_smooth(method = "lm",se=TRUE)+
  theme_classic()
plot2
ggsave("Scatter2.pdf",plot2)
```

As you can see we added another argument in ggplot called geom_smooth() which asks R to create a smooth curve based on a specific method to approximate the points. In this case the method we use is lm (linear model). The argument **se=TRUE** lets R know that we need the models with coefficients in the confidence intervals. Again by default this is set to 95%.

LAB ASSIGNMENT

1. Produce a scatter plot to show the relationship between RANGE and LAT. What is your observation?
2. Use lm to fit the linear model: $y = \beta_0 + \beta_1\chi + \varepsilon$. Explain briefly your findings, which should include the parameter estimates, the interpretation of the parameters and appropriate hypothesis test.
3. Write the estimated regression function.
4. Fine the confidence interval for the intercept and the slope coefficients.
5. What proportion of the variability in the dependent variable RANGE is accounted for by LAT through the regression line?

*Remember to attach your R script for the lab report.