**IST 687 PREP EXERCISE 08**

**Student Name: Bhavish Kumar**

**Prep Exercise No: 08**

**Date Due: 16th October 2019**

1. **In a paragraph or two, explain the concept of linear modeling and its uses within data analysis.**

Linear Modelling is used to predict the value of a numerical variable which is called a dependent variable based on the values of other variables which are known as the explanatory variables. A Linear equation of the form y = a+b1x1+b2x2+……. is formulated between the dependent variable whose value needs to be predicted and the explanatory variables. The equation is formed using historical/past data for the dependent variable and the explanatory variable. This equation is then used to predict the numeric value for the dependent variable based on the values of the explanatory variable. The model also helps us understand how the variation in one explanatory variable impacts the value of the dependent variable. It is very useful in data analysis and real life as with the help of linear modelling we can formulate an equation and accurately predict the value of the dependent variable when we know the values of our explanatory variables. For example, if we need to predict the selling price of a house based on the area in square feet, age and # beds and #baths, we can build a linear model using past data and then plug in the available values for area in square feet, age, #beds and #baths to accurately predict the selling price of the house.

1. **Getting Ready: Read data from a URL directly into a data frame, the data at the end of the URL is an excel file.**
   1. Read in data from the following URL into a data frame  
      <http://college.cengage.com/mathematics/brase/understandable_statistics/7e/students/datasets/mlr/excel/mlr01.xls>

urlread <- "http://college.cengage.com/mathematics/brase/understandable\_statistics/7e/students/datasets/mlr/excel/mlr01.xls"

install.packages("readxl")

library("readxl")

df\_url <- read.csv(urlread, stringsAsFactors = FALSE) ### Reading directly from url

View(df\_url)

* 1. Use the ‘download.file’ command to download the excel spreadsheet, and store it in a temporary local file

download.file(urlread, destfile = "webexcel.xls") ### downloading file into directory and storing as csv

* 1. Use the read\_excel command from the readxl package (you might need to install and library readxl). Store the data in the dataframe ‘df’. *(If you receive an error in this step related to libxls, please feel free to manually download the file from the web site and then use read\_excel)*

df<-read\_excel("webexcel.xls") ## reading from excel into dataframe

1. **Examine your dataframe using the View() and str() functions.**
   1. Verify that there are 4 columns within your dataframe, use the following URL to identify what each column represents. List the column names and their respective representations below.   
      <http://college.cengage.com/mathematics/brase/understandable_statistics/7e/students/datasets/mlr/frames/frame.html>

View(df)

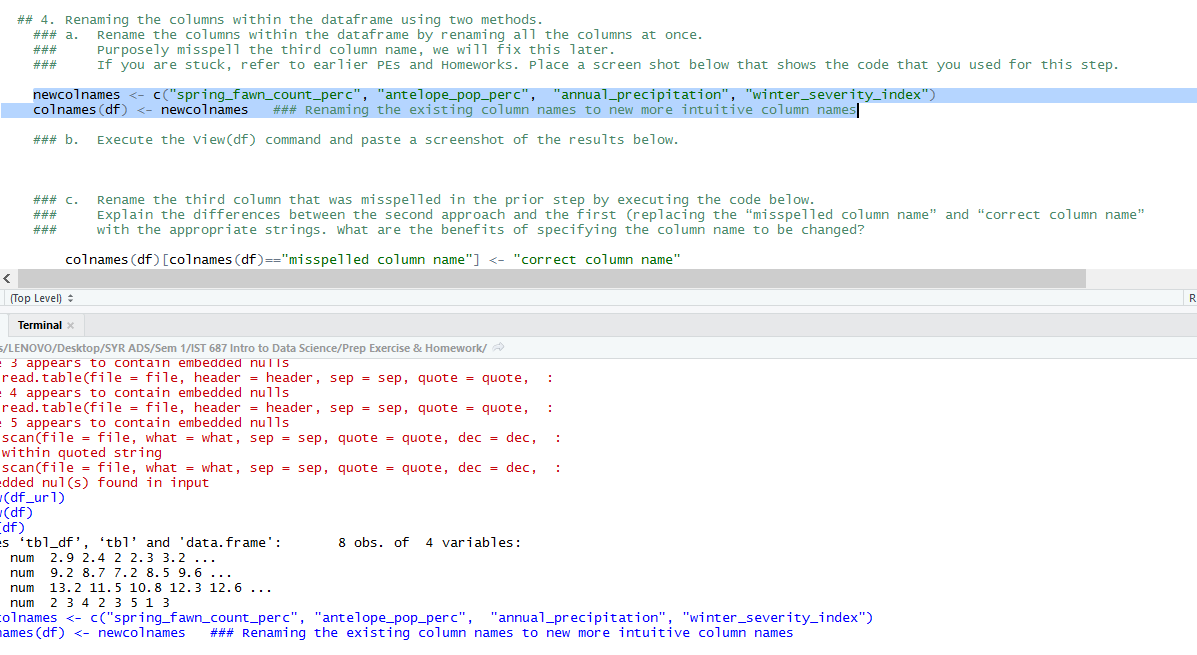
str(df)

### The data contains 4 columns namely X1, X2, X3 and X4 where X1 is the spring fawn count/100, X2 is size of adult antelope population/100, X3 is annual precipitation (inches), X4 is winter severity index (1=mild,5=severe)

1. **Renaming the columns within the dataframe using two methods.**
   1. Rename the columns within the dataframe by renaming all the columns at once. Purposely misspell the third column name, we will fix this later. If you are stuck, refer to earlier PEs and Homeworks. Place a screen shot below that shows the code that you used for this step.

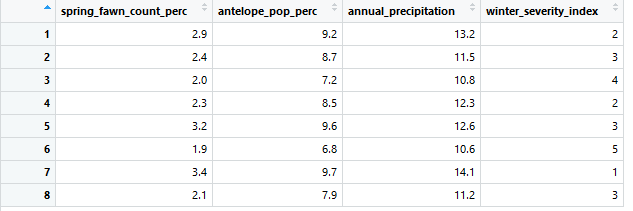
newcolnames <- c("spring\_fawn\_count\_perc", "antelope\_pop\_perc", "annual\_precipitation", "winter\_severity\_index")

colnames(df) <- newcolnames ### Renaming the existing column names to new more intuitive column names



* 1. Execute the View(df) command and paste a screenshot of the results below.

View(df)



* 1. Rename the third column that was misspelled in the prior step by executing the code below. Explain the differences between the second approach and the first (replacing the “misspelled column name” and “correct column name” with the appropriate strings. What are the benefits of specifying the column name to be changed?

colnames(df)[colnames(df)=="annual\_precipitation"] <- "annual\_preci"

### In the first approach we change all the column names at once without checking the existing column names and hence we cannot be sure if the existing names are correcct or incorrecct.

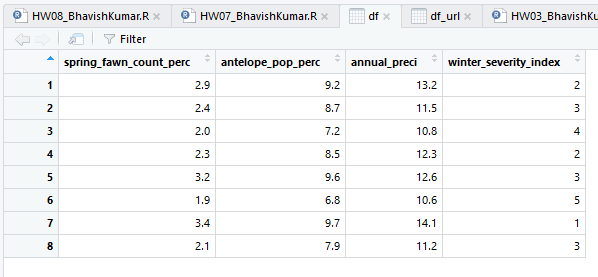
### In the second approach we are changing the column names of the incorrect one with the corresponding correct column name.

### The benefit of the second method is that we can reduce the margin of error by specifying the condition to check if the current column name is incorrect and only then changing it to the correct name thereby reducing the chance of naming a column incorrectly.

### This way we can ensure that the order in which we change the column names are correct.

1. **Verify that your changes are present within the dataframe using the View command.**
   1. Attach a screenshot of your correctly named dataframe third column below.

View(df)



Correctly named the 3rd column to annual\_preci

1. **Define a bivariate plot (i.e., explain what is a bivariate plot) and then explain how you would create a bivariate plot within R.**

A bivariate plot is a plot between two variables for example a scatter plot or a time series plot.

Using the plot function we can create a bi variate plot in R. We need to provide 2 vector inputs to the plot function so that it generates a plot between the 2 variables.

For example: plot(df$spring\_fawn\_count\_perc, df$antelope\_pop\_perc)

In the above example we are inputting two vectors which are the columns of the df dataframe to generate a scatter plot between the 2 variables.

1. **Explain the significance of the R-Squared value of a regression model.**

The R-squared value is a measure of how good or bad the model is, i.e. it tells us how much of the variation in dependent variable is explained by the variation in independent variables. For a simple linear regression model involving only 1 independent variable, the R square value is the square of the correlation coefficient between the 2 variables. Higher the R square the better is fit of the linear regression model and hence better the model. Hence higher R square value is preferred. Increase in R square reduces the sum of squared errors between the actual value and the predicted value of the dependent variable.