# Prep Exercise (PE08) Lining up our models (Linear Modeling)

### General Instructions

1. For this exercise you will answer all of the questions in this document and turn it in to Blackboard.

General Directions:

* Before attempting the Prep Exercise, read through Chapter 16 of *An Introduction to Data Science* and execute the code provided within the readings to familiarize yourself with the code and output.
* Getting Started: The chapter on linear models (“Lining Up Our Models”) introduces linear predictive modeling using the tool known as multiple regression. The term “multiple regression” has an odd history, dating back to an early scientific observation of a phenomenon called “regression to the mean.” These days, multiple regression is just an interesting name for using a simple linear modeling technique to measure the connection between one or more predictor variables and an outcome variable. In this week’s homework, you will use an open data set to predict “spring fawn count” from the three other variables in the data set. The term “fawn” refers to a newly-born antelope.
* # IST 687, Standard Homework Heading
* #
* # Student name: Thadhani Hitesh Chandrakumar
* # Homework number: PE08
* # Date due: Wed 16th Oct 2019 11:59PM
* #
* # Attribution statement: (choose the statements that are true)
* # 1. I did this work by myself, with help from the book and the professor
* # 2. I did this homework with help from the book and the professor and these Internet sources: www.stackoverflow.com
* # 3. I did this homework with coaching from <Name of another student> but did not cut and paste any code

### Prep Exercise

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

**Linear modeling is used to make prediction or find the correlation between the variables. How changing 1 variable affects the other. The basic idea is to make a model which fits a straight line through the set of points of the 2 variables where dependent variable is explained by explanatory/independent variable. The straight line is drawn between the data points such that there is minimum distance between the line and the individual points and maximum points fall on the line. This is modeled in mathematical terms as a equation**

**Y= MX+C**

**Where Y is the predicted value of dependent variable,**

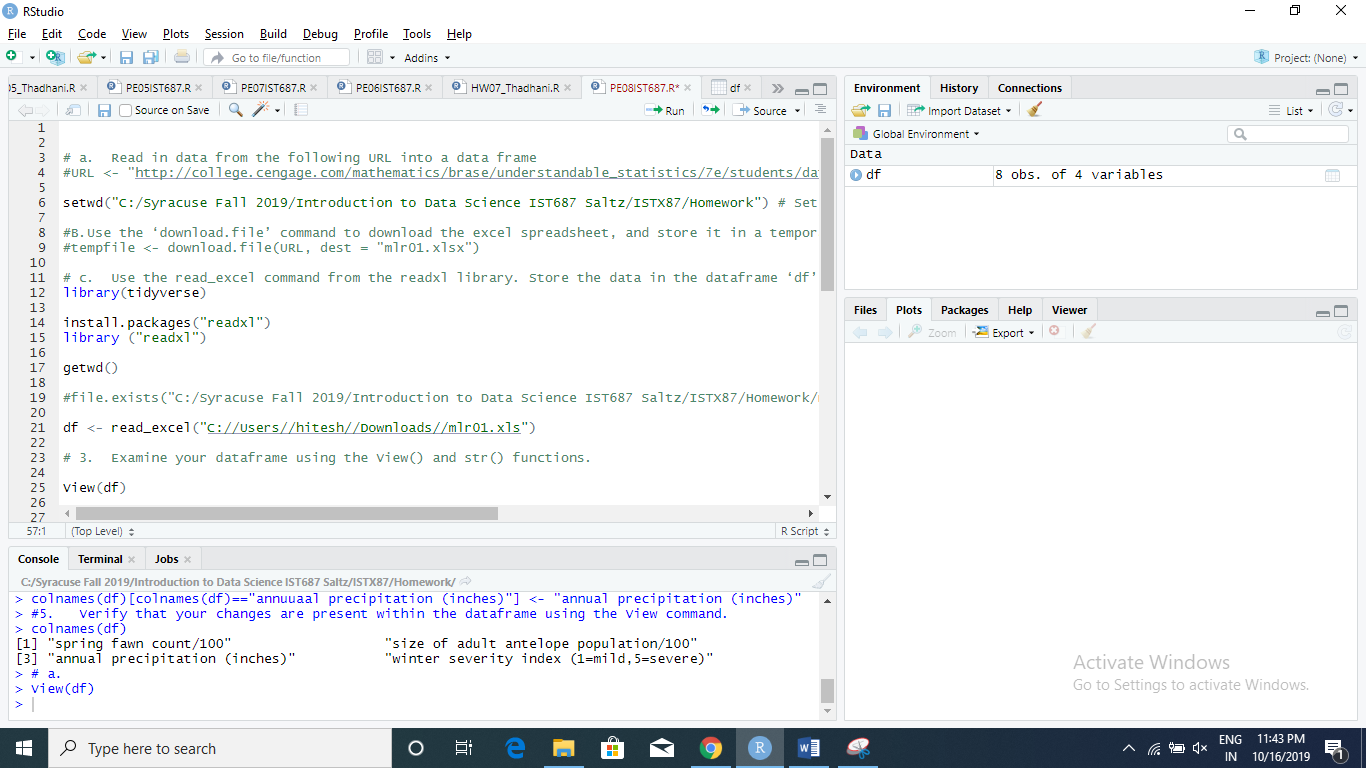
**M is the slope of the line**

**X is the independent variable**

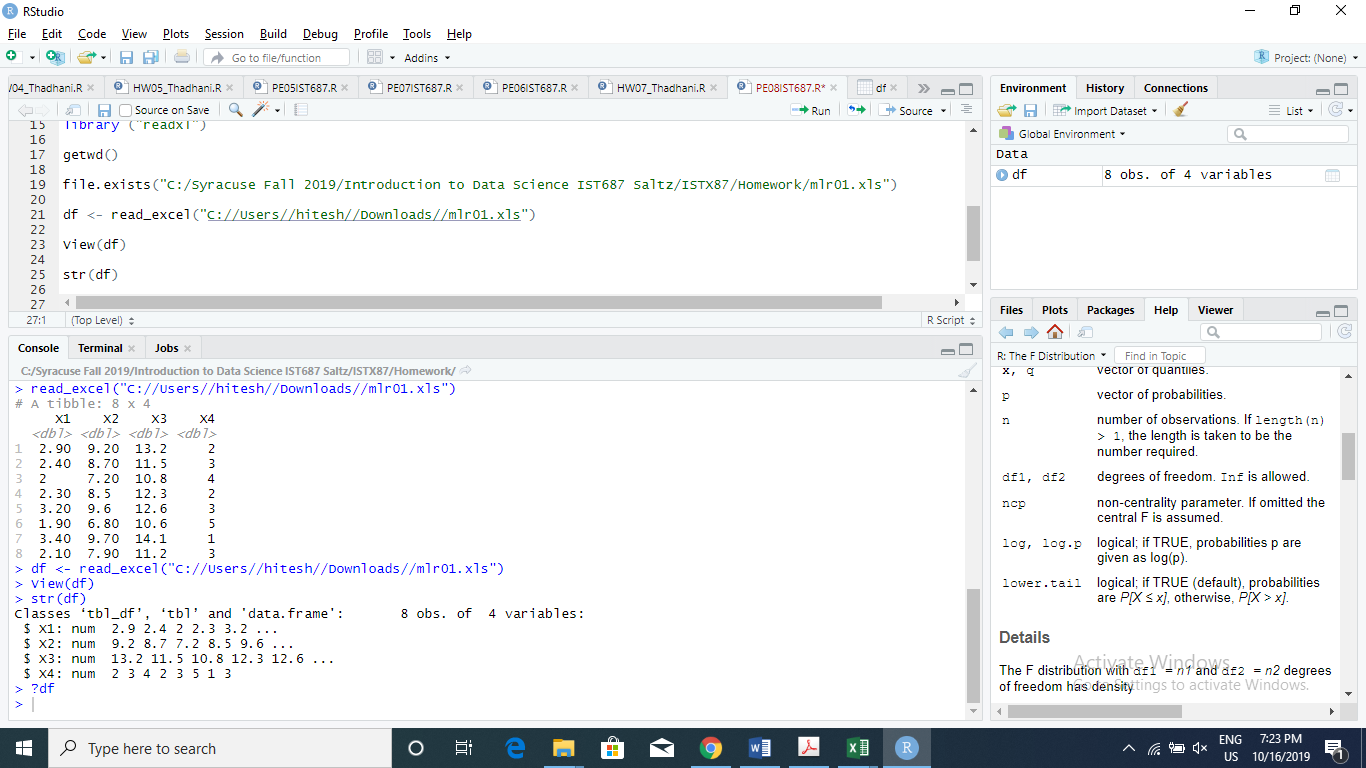
**C is the intercept/ constant**

**If we have a linear regression model where the coefficients of M, X and C are known we can find out the predicted value of Y. This linear modeling is helpful in data analysis as it gives how the variables are correlated between each other and it becomes very easy to predict/calculate the values depending upon many factors/variables. We can use linear models designed using this to answer business questions like how the price of a house depends on area, number of bedrooms, number of bathrooms etc. and find out hidden insights which can be helpful to make business critical decisions.**

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

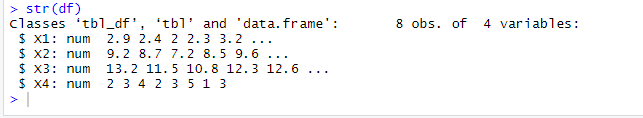


* 1. Use the read\_excel command from the readxl library (you might need to install and library readxl). Store the data in the dataframe ‘df’.



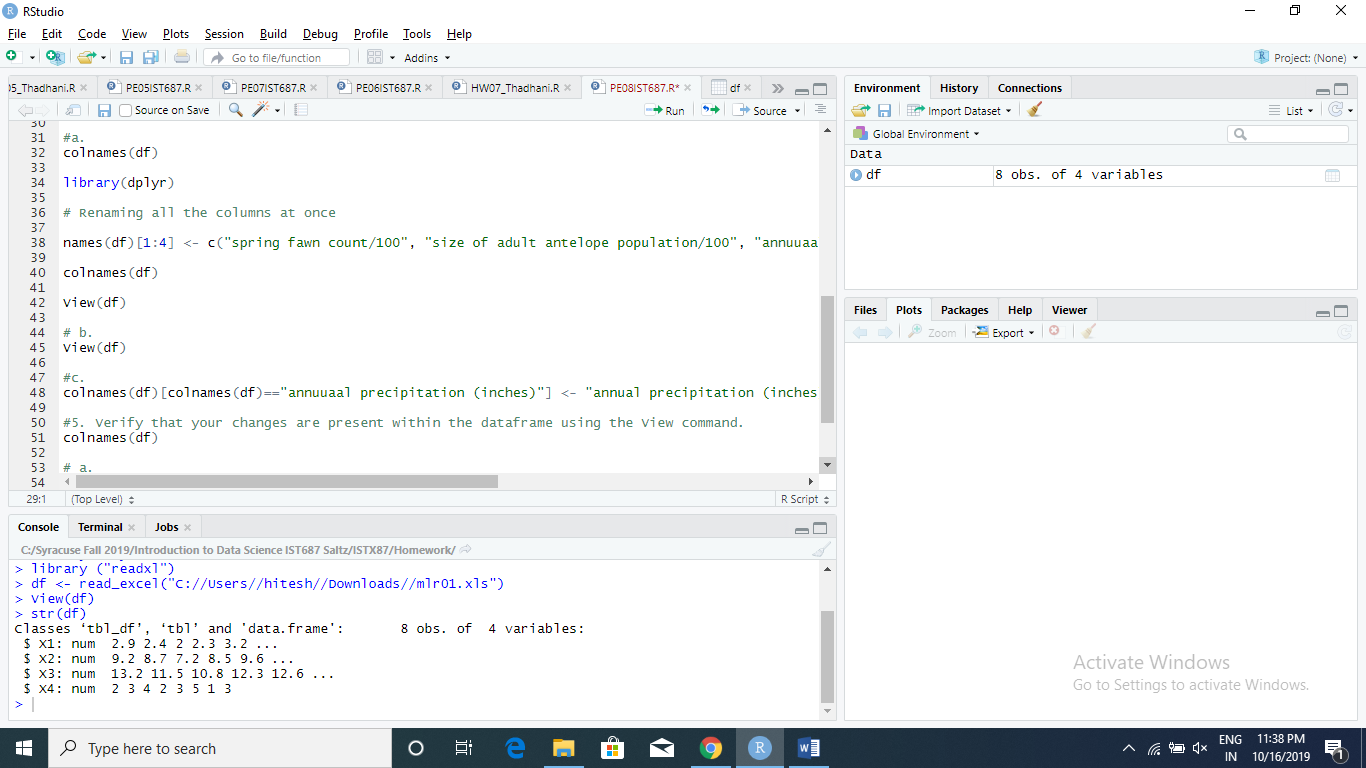
1. **Examine your dataframe using the View() and str() functions.**
   1. Verify that there are 4 columns within your dataframe, use the following URL to identify what each column represents. List the column names and their respective representations below.   
      <http://college.cengage.com/mathematics/brase/understandable_statistics/7e/students/datasets/mlr/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

**Thunder Basin Antelope Study**  
  
The data (X1, X2, X3, X4) are for each year.  
X1 = spring fawn count/100  
X2 = size of adult antelope population/100  
X3 = annual precipitation (inches)  
X4 = winter severity index (1=mild,  
5=severe)

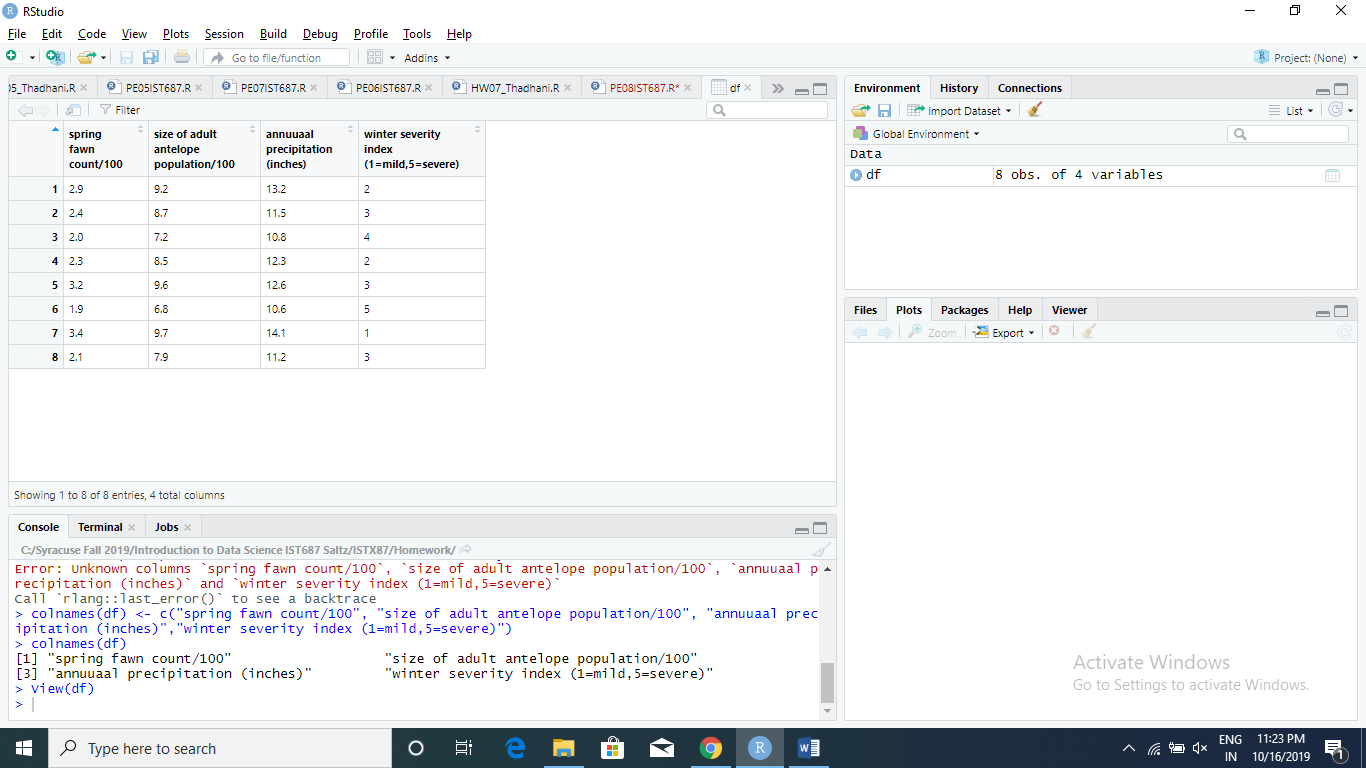


1. **Renaming the columns within the dataframe using two methods.**
   1. Rename the columns within the dataframe by renaming all the columns at once. Purposely misspell the third column name, we will fix this later. If you are stuck, refer to earlier PEs and Homeworks. Place a screen shot below that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

names(df)[1:4] <- c("spring fawn count/100", "size of adult antelope population/100", "annuuaal precipitation (inches)","winter severity index (1=mild,5=severe)")



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

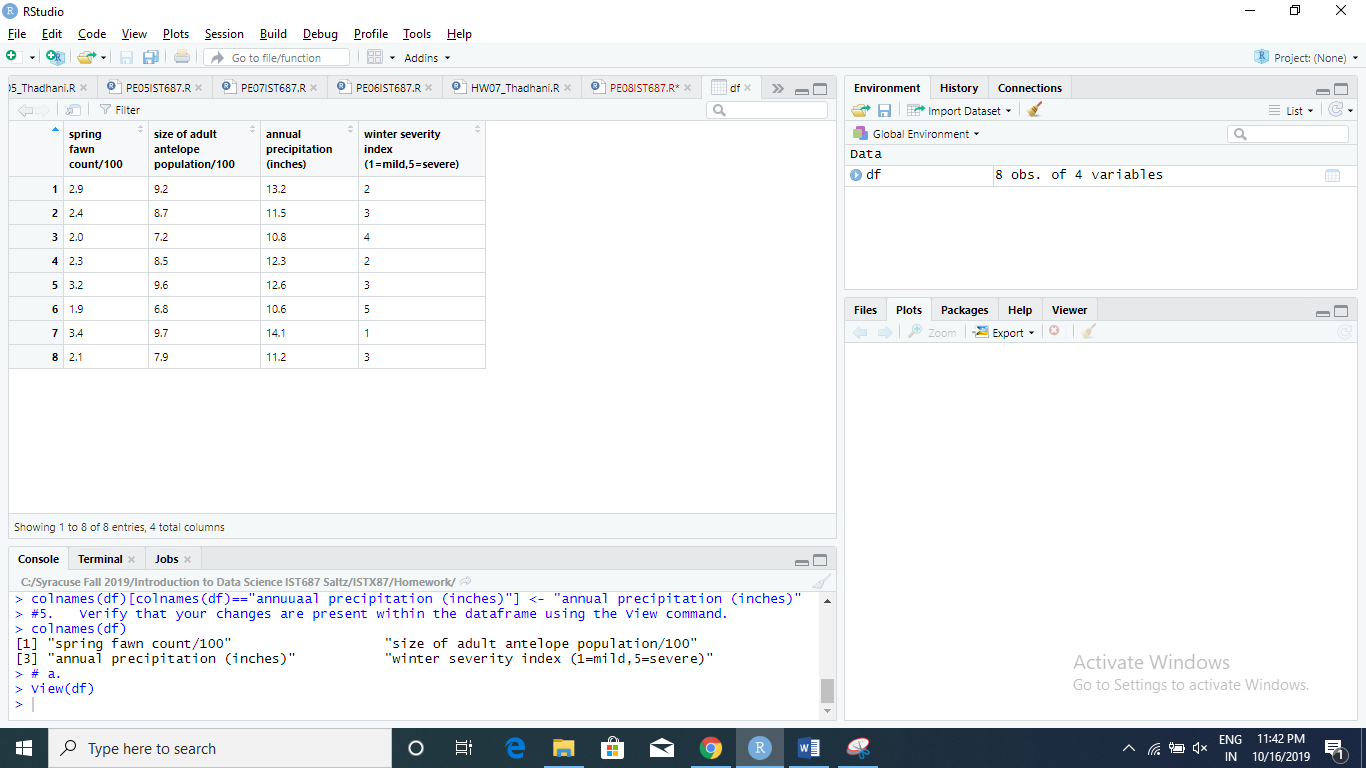


* 1. Rename the third column that was misspelled in the prior step by executing the code below. Explain the differences between the second approach and the first (replacing the “misspelled column name” and “correct column name” with the appropriate strings. What are the benefits of specifying the column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

**colnames(df)[colnames(df)=="misspelled column name"] <- "correct column name"**

The benefit of using this approach is that here we are dealing with only 1 column and are changing that column name only. When there are multiple columns this approach has advantage over the one which was earlier in part a as we can interchange or by mistake rename some other column and that creates problems with the whole dataset**.** If a column name is changed wrongly by using all at once approach we will lose to the meaning to the dataset and won’t be able to make out correct interpretations.

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



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.**

**Bivariate plot is plot of 2 variables where one variable is plotted on X-Axis and the other on Y-Axis. It is also called as Scatterplot. The dependent variable is plotted on Y-Axis whereas the explanatory variable is plotted on X-Axis. The bivariate plot is created in R using**

**Plot(X-Axis Variable, Y-Axis Variable)**

**Plot(state$stateName,state$MeanIncome)**

**Here Meanincome is dependent on the stateName variable which can be plotted as scatter plot.**

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

**R-squared value of a regression model predicts how good the linear model is to predict the dependent variable based on independent variable. Suppose we have age and salary as 2 variables where age has an effect on salary. So age and salary are linearly dependent with salary predicted based on the age. Here R-squared value of the regression performed on the 2 variables will be able to predict the variability of the model and how good the model is while predicting salary based on age as explanatory variable. R-squared closer to 1 the better the linear model is in predicting the value of 1 variable based on another. R-squared takes values from 0 to 1.**

1. **List any additional resources you used here.**
2. **Be sure to save your R file as this will become the starting code for your homework.**

***You must submit all Prep Exercises to blackboard prior to the deadline specified for each assignment.*** PE assignments are due on the evening prior to the lecture class. Late PE assignments will not be accepted for credit.

*Materials provided by the instructors of this course are copyrighted 2018 by Jeffrey Saltz and Jeffrey Stanton, or by the respective instructor that produced them. These materials are provided to enrolled students for their exclusive use during the semester of instruction. Uploading instructor-produced materials to Internet sites (e.g., Coursehero) without permission is considered a breach of professional ethics and will be reported as a possible academic integrity violation.*