**IST 687 PREP EXERCISE 02**

**Student Name: Bhavish Kumar**

**Prep Exercise No: 02**

**Date Due: 4th September 2019**

1. **Create a new dataframe within R Studio using the ‘USArrests’ dataset (‘USArrests’ is preloaded into R Studio). ﻿**This data set contains statistics, in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973. Also given is the percent of the population living in urban areas.
   1. Creating a dataframe using a given dataset:

**myArrests <- USArrests**

1. **Verify that your new dataframe was properly populated with the USArrests dataset.**
   1. Viewing the contents of a dataframe:

**View(myArrests)**

* 1. List the attributes of the dataframe:

**attributes(myArrests)**

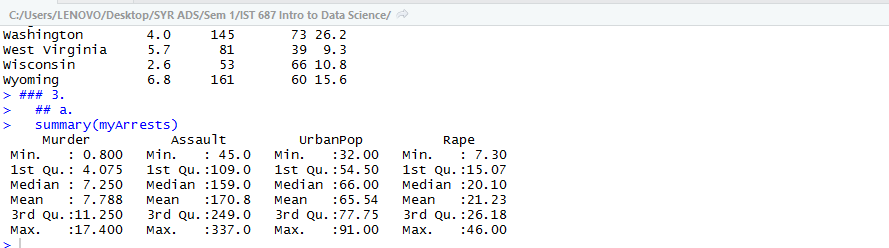
* 1. Describe the instances within the dataframe:

**print.data.frame(myArrests)**

1. **Obtain a summary of the newly created dataframe using one R Studio command. Summary of the dataframe includes Min, 1st Quartile, Median, Mean, 3rd Quartile, and Max.** 
   1. Summarize your new dataset:

**summary(myArrests)**

* 1. Place a screen shot below of the output from summary(myArrests)



* 1. Why is the Summary command useful to a data scientist?

Summary command is useful for data scientists because it is one single command that produces the minimum & maximum values, 1st & 3nd quartiles, mean and median of all the numerical variables, which helps a data scientist understand the range, interquartile range and the central values of the data which can help detect outliers and understand the general distribution.

1. **Now create your own dataframe with information about your family!**
   1. Create a column of names for your family dataframe:

**myFamilyNames <- c("Mom", "Dad", "Brother", "Sister")**

* 1. Create a vector for family member ages, entering the ages in the same order that would correspond to your family members, i.e. if you entered mom first in the previous step then your mom’s age should be the first entered within the vector.

**myFamilyAges <- c("48", "52" ,"9", "20")**

* 1. Add a eye color attribute to your dataframe by defining each family members eye color in a vector:

**myFamilyEyeColor <-c("black", "brown", "brown", "black")**

* 1. Combine the vectors into a dataframe:

**myFamily <- data.frame(myFamilyNames, myFamilyAges, myFamilyEyeColor)**

* 1. Run the structure (**str()**) command on your newly created dataframe and record your observations of what the command did:

The command produced the structure of the dataframe, i.e. it lists down the number of columns (variables) and the number of rows (observations).

data.frame': 4 obs. of 3 variables:

$ myFamilyNames : Factor w/ 4 levels "Brother","Dad",..: 3 2 1 4

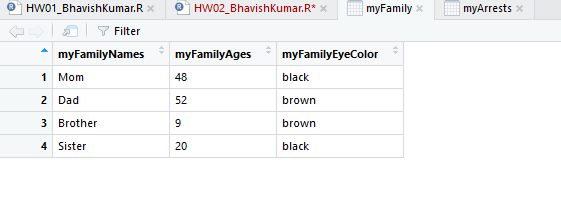
$ myFamilyAges : Factor w/ 4 levels "20","48","52",..: 2 3 4 1

$ myFamilyEyeColor: Factor w/ 2 levels "black","brown": 1 2 2 1

* 1. Now run the (**View()**)command. How does the output of the View command correlate with the structurecommand?

The view command yields the entire dataframe and displays it in a structured manner, i.e. in the form of rows and columns. Whereas the structure command str does not display the entire dataframe but instead it only tells the number of rows and columns along with some example values.

* 1. Place a screenshot below of the output from the View() command.



1. **Removing Rows and Columns within a dataframe.**
   1. Remove the third row in your newly created dataframe:

**myFamily <- myFamily[-3,]**

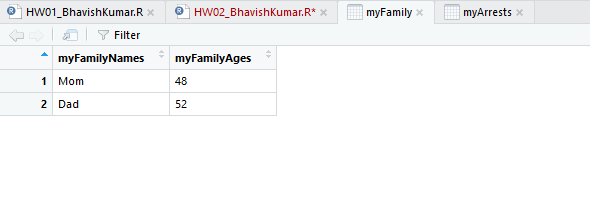
* 1. Remove the eye color column in your newly created dataframe:

**myFamily <- myFamily[,-3]**

* 1. Assuming a data scientist wishes to analyze quantitative data within the dataframe, how is removing the eye color column an example of munging?

Since Eye color is a Categorical variable (Qualitative data), it plays no role in quantitative data analysis. For a data scientist who wishes to analyze quantitative data, the eye color column is of no use. Hence by getting rid of that column is a part of data cleaning/munging process to aid the analysis on clean data.

* 1. Place a screenshot below of your final dataframe. (Hint: use the View() command)



**List any additional resources you used here.**