Executive Summary Report 1

28th September 2021

HARSHIT GAUR

NUID : 001093079

1. *Age ~ Weight* data set

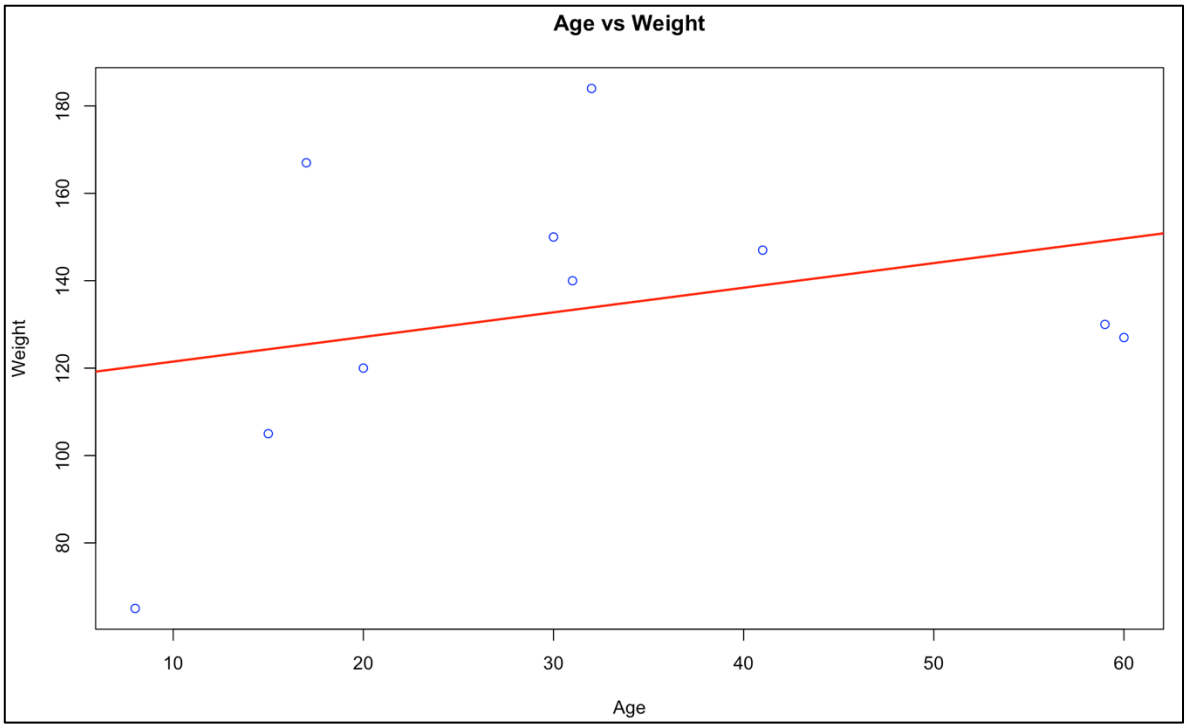

Figure 1.1

```
[1] "Median of weight : 135"
```

Figure 1.2

```
> cor(age, weight)
[1] 0.3009934
```

Figure 1.2

```
> cor(age, weight)
[1] 0.4163029
```

Figure 1.3

- The graph depicts that weight is linearly increasing with respect to age with a correlation value of 0.3009934.
- After replacing the 7th element in the '*age'* vector from 60 to 26, the correlation remains positive with a value of 0.4163029.

2. Data modification in the '*age*' vector

```
> str(age)
 num [1:10] 8 20 17 32 59 41 60 31 15 30
> #STEP 6: Deleted the 7th element from 'age' vector
> age <- age[-7]
> #STEP 7: Insert 26 as 7th element in 'age' vector
> age <- append(age, 26, 6)
> str(age)
 num [1:10] 8 20 17 32 59 41 26 31 15 30
```

Figure 2.1

- We can perform modifications like deleting and adding an element using functions. Referencing to the above screenshot, we have modified '*age'* vector by replacing the 7th element from 60 to 26.

3. Creation of '*color'* vector and representation of its structure

```
> str(color)
 chr [1:3] "Red" "Green" "Blue"
```

Figure 3.1

4. Matrix ranging from 1 to 15

```
#STEP 9: Creating a 5x3 matrix ranging from 1 to 15
rNames <- paste("R", seq.int(1:5), sep = "")
cNames <- paste("C", seq.int(1:3), sep = "")
intMatrix <- matrix(1:15, nrow = 5, ncol = 3, byrow
= TRUE, dimnames = list(rNames, cNames))
View(intMatrix)
```

Figure 4.1

	C1	C2	C3
R1	1	2	3
R2	4	5	6
R3	7	8	9
R4	10	11	12
R5	13	14	15

Figure 4.2

- Matrix operations can be performed in R language by providing a range of a data set along with other parameters related to number of rows, number of columns, matrix filling property.

5. Data frame in R

	age	weight
1	8	65
2	20	120
3	17	167
4	32	184
5	59	130
6	41	147
7	26	127
8	31	140
9	15	105
10	30	150

Figure 5.1

```
> #STEP 11: Displaying the structure of 'people' data frame
> str(people)
'data.frame':   10 obs. of  2 variables:
 $ age   : num  8 20 17 32 59 41 26 31 15 30
 $ weight: num  65 120 167 184 130 147 127 140 105 150
```

Figure 5.2

```
> #STEP 12: Displaying the summary of 'people' data frame
> summary(people)
      age             weight
 Min.   : 8.00   Min.   : 65.0
 1st Qu.:17.75   1st Qu.:121.8
 Median :28.00   Median :135.0
 Mean   :27.90   Mean   :133.5
 3rd Qu.:31.75   3rd Qu.:149.2
 Max.   :59.00   Max.   :184.0
```

Figure 5.3

- Matrix operations can be performed in R language by providing a range of a data set and other parameters related to number of rows, number of columns, row-major property, etc.
- We can infer the below table from the data frame :

Statistics	Age	Weight
Minimum Value	8	65
Maximum Value	59	184
Median	28	135
Mean	27.90	133.5
Standard Deviation	14.59141	33.18383
1st Quartile Value (Q1)	17.75	121.8
3rd Quartile Value (Q3)	31.75	149.2

- The means of both the data sets are near to their medians which means that outliers will not affect the graph extensively.

6. Student.csv data set

```
> colnames(student)          #NOTE: Display the variable names of the data set.
[1] "StudentID"      "First"          "Last"           "Math"           "Science"        "Social.Studies"
```

Figure 6.1

- The data set contains records of 4 students along with their marks in 3 subjects.

7. Summary

7.1. We can perform various statistical operations using the R language and RStudio and find out the statistics of the sample data.

7.2. There is a linear relationship between the data sets of '*age'* and '*weight*' with a positive correlation.

7.3. We can find the statistics (mean, median, standard deviation, quartile values, maximum, minimum) of the '*age'* and '*weight*' data sets and based on them can infer the impacts of the outliers present in the data sets (explained in the previous points).

7.4. We can infer from Student.csv data set that it contains **discrete numerical data** in the subject variables and using them can find out the statistics (mean, median, standard-deviation, quartile values, maximum, minimum) of the marks scored by all the 4 students in every subject. Screenshots have been attached for reference for the same.

7.5 We can also infer that one student has either missed or not appeared or not eligible for a subject (Science) where no data is available for his/her marks.

7.6 Since, we do not have any population from where this Student.csv sample was taken from, we cannot perform inferential analysis and were only able to perform descriptive analysis on the sample.

```
summary(student$Math)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 65.0    72.5    82.5    80.0    90.0    90.0
```

Figure 7.1

```
summary(student$Social.Studies)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
67.00   69.25   75.00   77.25   83.00   92.00
```

Figure 7.2

```
summary(student$Science)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
75.00   77.50   80.00   83.33   87.50   95.00       1
```

Figure 7.3

BIBLIOGRAPHY

1. *R in Action, Second Edition*. (2021). Manning Publications. https://www.manning.com/books/r-in action-second-edition

2. *R Tutorial*. (2020). R Tutorials Point. https://www.tutorialspoint.com/r/index.html

3. *abline R function : An easy way to add straight lines to a plot using R software - Easy Guides - Wiki - STHDA*. (2021). STHDA. http://www.sthda.com/english/wiki/abline-r-function-an-easy-way-to-add-straight-lines-to-a-plot-using-r-software

4. GeeksforGeeks. (2020, June 25). *Get a List of Numbers in the Specified Range in R Programming - seq.int() Function*. https://www.geeksforgeeks.org/get-a-list-of-numbers-in-the-specified-range-in-r-programming-seq-int-function/

APPENDIX

```
#---------------------- GAUR_HARSHIT_M1_PROJECT1 ----------------------#

#STEP 1: Printing my name.
print("HARSHIT GAUR")

#STEP 2: Installing the 'vcd' package.
install.packages("vcd")

#STEP 3: Importing the 'vcd' package
library("vcd")

#STEP 4: Plot age ~ weight scatter plot.
#Note: Defining 'age' vector
age <- c(8,20,17,32,59,41,60,31,15,30)
#Note: Defining 'weight' vector
weight <- c(65,120,167,184,130,147,127,140,105,150)
#Note: Plot a Scatter plot for age ~ weight
plot(age, weight, main = "Age vs Weight", xlab = "Age", ylab = "Weight", col = "BLUE")
abline(lm(weight~age), col="red", lwd=2)

#Note: Finding correlation between age and weight
cor(age, weight)

#STEP 5: Median of the 'weight' vector
paste("Median of weight :", median(weight))

str(age)
#STEP 6: Deleted the 7th element from 'age' vector
age <- age[-7]
#STEP 7: Insert 26 as 7th element in 'age' vector
age <- append(age, 26, 6)
str(age)

#STEP 8: Creating a 'color' vector
color <- c("Red", "Green", "Blue")
str(color)

#STEP 9: Creating a 5x3 matrix ranging from 1 to 15
rNames <- paste("R", seq.int(1:5), sep = "")
cNames <- paste("C", seq.int(1:3), sep = "")
intMatrix <- matrix(1:15, nrow = 5, ncol = 3, byrow = TRUE, dimnames = list(rNames, cNames))
View(intMatrix)

#STEP 10: Creating a 'people' data frame using 'age' and 'weight' vectors
people <- data.frame(age, weight)
View(people)

#STEP 11: Displaying the structure of 'people' data frame
str(people)
#STEP 12: Displaying the summary of 'people' data frame
summary(people)

#STEP 13: Import 'Student.csv' data set
#Note: Change the working directory as per the file's location.
setwd("/Users/HarshitGaur/Documents/Northeastern University/MPS Analytics/ALY 6000/Class 1")
student <- read.csv("Student.csv", header = TRUE)

#STEP 14: Display the variable names of the data set.
colnames(student)

#Note: Find summary of 'Student - Math'
summary(student$Math)

#Note: Find summary of 'Student - Science'
summary(student$Science)

#Note: Find summary of 'Student - Social Science'
summary(student$Social.Studies)
```