Project 1 – R Practice

CPS Graduate Analytics program, NorthEastern University, Seattle

ALY 6000: Introduction to Data Analytics

Professor: SELCUK BARAN

Submitted By

Syed Tanveer

Introduction: Data Analysis is the process of systematically applying statistical and/or logical techniques to describe and illustrate, condense and recap, and evaluate data.

Overview: In this project the initial practice using R studio is performed and the required outputs are fetched.

Key Findings: All the functions used in the R script and outputs obtained as the result gave clear insight about how things are processed and worked out.

The illustrated explanations of the functions are below:

Explanation

```
#Name:Syed Tanveer Date:02-10-2023 Class:

cat("\014") # clears console

rm(list = ls()) # clears global environment

try(dev.off(dev.list()["RStudioGD"]), silent = TRUE) # clears plots

try(p_unload(p_loaded(), character.only = TRUE), silent = TRUE)

#clears packages

options(scipen = 100)

# disables scientific notation for entire R session

#1

123 * 453 #the multiplication of these functions here is 55719 in the output

5^2 * 40 #5 square that is 25 multiplies 40 that is 1000 which reflects on output

TRUE & FALSE #here it gives the false for true and false notation

TRUE | FALSE #here for true or false the output is true

75 %% 10 #5

75 / 10 #7.5

#2

first_vector <- c(17,12,-33,5) # here the the first vectors will be 17 12 -33 5

first_vector
```

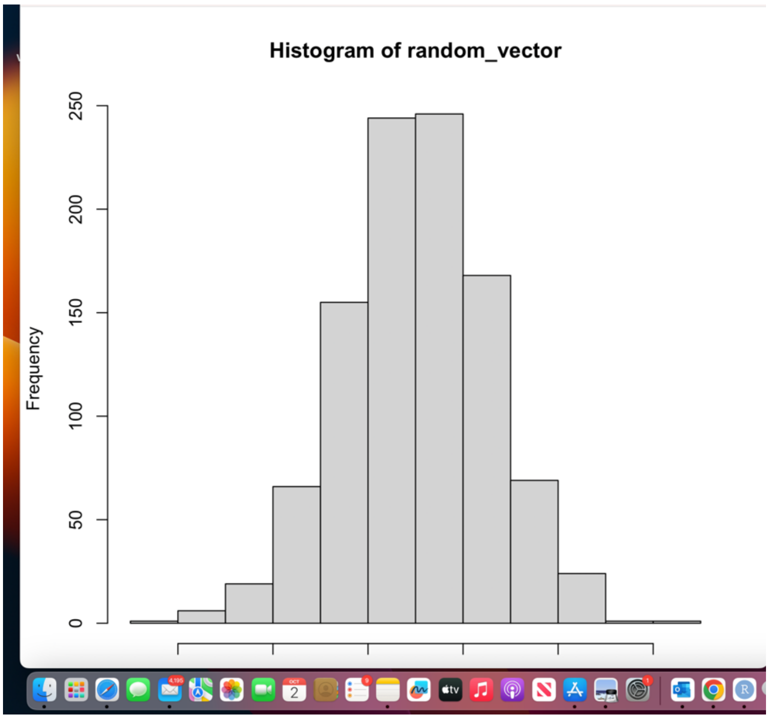
```
#3

counting_by_fives <- c(5,10,15,20,25,30,35)# here the counting by fives vectors are assigned

counting_by_fives

#4

second_vector <- seq(from = 10, to = 30, by = 2) # the output here will be the sequence of numbers which are even between 10 and 30 that is 10 12 14 16 18 20 22 24 26 28 30.

second_vector

#5

counting_by_fives_with_seq <- seq(from = 5, to = 35, by = 5) #the multiples of 5 between 5b and 35 are created as sequences and assigned to the variables and o/p is 5 10 15 20 25 30 35

counting_by_fives_with_seq

#6

third_vector <- rep(first_vector, times = 10) #first is repeated 10 times and assigned as third vector

third_vector

#7

rep_vector <- rep(0,20) #here 0 are created 20 times and result is stored in rep vector

rep_vector

#8

fourth_vector <- (10:1) # the values 10 to 1 in descending orders are created as fourth vector

fourth_vector

#9

counting_vector <- (5:15) #counting from 5 to 15 is done and given as o/p

counting_vector

#10

grades <- c(96, 100, 85, 92, 81, 72) #these values are stored in grades

grades
```

```
#11

bonus_points_added <- grades+3 # here the bonus points 3 is added to the stored grades and the result is this way 99 103 88 95 84 75

bonus_points_added

#12

one_to_one_hundred <- 1:100 # 1 to 100 is stored in the variables attached

one_to_one_hundred

#13

reverse_numbers <- seq(from = 100, to = -100, by = -3) #the sequence of 100 to -100 is obtained with -3 intervals

reverse_numbers

#14

second_vector + 20 #here 20 is added to the sequence of the second vector that is 30 32 34 36 38 40 42 44 46 48 50

second_vector * 20 # here 20 is multiplied with the sequence of the second vector that is  200 240 280 320 360 400 440 480 520 560 600

second_vector >= 20 # here the second vector values should be greater than or equal to 20 that is o/p fetched: FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE

second_vector != 20 here the second vector values should benot equal to 20 that is TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE

#15

total <- sum(one_to_one_hundred) #sum of one to one hundred is 5050

total

#16

average_value <- mean(one_to_one_hundred) # mean of one to one hundred is 50.5( 5050/100)

average_value

#17
```

```
median_value <- median(one_to_one_hundred) #median of one to one hundred is 50.5

median_value

#18

max_value <- max(one_to_one_hundred) #max value of one to one hundred is 100

max_value

#19

min_value <- min(one_to_one_hundred) #minimum of one to one hundred is 1

min_value

#20

first_value <- second_vector[1] #first value of second vector 10

first_value

#21

first_three_values <- second_vector[1:3] # first three values of second vector 10 1214

first_three_values

#22

vector_from_brackets <- second_vector[c(1, 5, 10, 11)] #1st 5th 10th and 11th elements of
second vector are 10 18 28 30

vector_from_brackets

#23

vector_from_boolean_brackets <- first_vector[c(FALSE, TRUE, FALSE, TRUE)] # the true
values statements of first vector are 12 5

vector_from_boolean_brackets

#24

second_vector >= 20 # FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
TRUE  TRUE, only the true obtained elements values of second vector are greater than or
equal to 20

#25
```

```
ages_vector <- seq(from = 10, to = 30, by = 2) #values from 10 to 30 with even numbers that is 10 12 14 16 18 20 22 24 26 28 30

ages_vector

#26

ages_vector[ages_vector >= 20] #in the ages vector the values greater than equal to 20 are 20 22 24 26 28 30

#27

lowest_grades_removed <- grades[grades >= 85]# lower than 85 grades are removed from grades 96 100  85  92

lowest_grades_removed

#28

middle_grades_removed <- grades[-c(3, 4)] #3rd and 4th element of grade are removed that is 96 100  81  72

middle_grades_removed



#29

fifth_vector <- second_vector[-c(5, 10)] #5th and 10th element of second vector are removed that is the o/p is 10 12 14 16 20 22 24 26 30

fifth_vector

#30

set.seed(5)

random_vector <- runif(n = 10, min = 0, max = 1000) #uniform distribution for 10 numbers are obtained-- 200.2145 685.2186 916.8758 284.3995 104.6501 701.0575 527.9600 807.9352 956.5001 110.4530

random_vector

#31

sum_vector <- sum(random_vector) # sum of random vector 5295.264

sum_vector

#32
```

```
cumsum_vector <- cumsum(random_vector) #cummulative sum of random vector 200.2145
885.4330 1802.3088 2086.7083 2191.3584 2892.4159 3420.3759 4228.3111 5184.8112

[10] 5295.2642

cumsum_vector

#33

mean_vector <- mean(random_vector) # mean of random vector 529.5264

mean_vector

#34

sd_vector <- sd(random_vector) #standard deviation of random vector 331.3606

sd_vector

#35

round_vector <- round(random_vector) #rounding the values of random vector 200 685 917
284 105 701 528 808 957 110

round_vector

#36

sort_vector <- sort(random_vector) #sorting the random vectors

sort_vector

#37

set.seed(5)

random_vector <- rnorm(n=1000, mean = 50, sd = 15) #normal distribution of random
vectors which are 1000 in numbers with 50 mean and sd of 15

random_vector

#38

histogram <- hist(random_vector)
```



```
#39

#Downloaded the datafile ds_salaries.csv from Canvas

#40

library(pacman)

p_load(tidyverse)

#41

first_dataframe <- read_csv("ds_salaries.csv")

#42

head(first_dataframe) #it generated the table of data frame 6 rows 12 columns

head(first_dataframe, n = 7) #table of  7rows 12 columns

names(first_dataframe) #names of the dataframe are listed

smaller_dataframe <- select(first_dataframe, job_title, salary_in_usd)

smaller_dataframe #job title and salary in usd are the column

better_smaller_dataframe <- arrange(smaller_dataframe,

                              desc(salary_in_usd))#arrainged in highest to lowest salary range w.r.t
to job title
```

```
better_smaller_dataframe
better_smaller_dataframe <- filter(smaller_dataframe, salary_in_usd >
                                     80000) #salary greater than 80k is listed
better_smaller_dataframe
better_smaller_dataframe <-
  mutate(smaller_dataframe, salary_in_euros = salary_in_usd * .94)
better_smaller_dataframe #the equals of salary in usd and euro are listed
better_smaller_dataframe <- slice(smaller_dataframe, 1, 1, 2, 3, 4, 10,
                                  1)  #
better_smaller_dataframe
ggplot(better_smaller_dataframe) +
  geom_col(mapping = aes(x = job_title, y = salary_in_usd), fill =
             "blue") +
  xlab("Job Title") +
  ylab("Salary in US Dollars") +
  labs(title = "Comparison of Jobs ") +
  scale_y_continuous(labels = scales::dollar) +
  theme(axis.text.x = element_text(angle = 50, hjust = 1))
#testing the solution
library(pacman)
p_load(testthat)
test_file("project1_tests.R")
```

Conclusion/Recommendations:

The explanation of each line of the code is given above.

Works Cited

https://apastyle.apa.org/style-grammar-guidelines/paper-format/title-page
https://www.google.com/search