Analytical Questions – Day 5

K.RITHIK VASAN
191921033

1. Write a R program to Create the following details
   a. x= sample(-50:50, 10, replace=TRUE).and print the value of x

      x= sample(-50:50, 10, replace=TRUE)

      op/

      [1] 1 -37 -14 -43 -35 34 -9 10 -16 -34

   b. To create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 50 and sum of numbers from 20 to 50.

      v <- c(20:50)

      > mean(v)

      [1] 35

      sum(v)

2. To create an array of two 3x3 matrices each with 3 rows and 3 columns from two given two vectors.vector1 = c(1,3,4,5) and vector2 = c(10,11,12,13,14,15)
   a. Print vector1, vector2
      v1 <- c(1,3,4,5)

      v2 = c(10,11,12,13,14,15)
   b. Print new array

      x <- array(c(v1,v2),dim=c(3,3))

3. Write a R program to merge two given lists into one list. n1 = list (1,2,3) c1 = list("Raja", "Rani", "Prince")

   n1 = list (1,2,3)

   > c1 = list("Raja",

+ "Rani", "Prince")
> li <- merge(n1,c1)
> li

i) Write a R program to convert a given list to vector.n1 = list (1,2,3)c1 = list(4,5,6)

n1 = list (1,2,3)

> c1 = list(4,5,6)

> v <- unlist(n1)

> v2 <- unlist(c1)

> v

4. Consider A=matrix(c(2,0,1,3),ncol=2) and B=matrix(c(5,2,4,-1),ncol=2).3

a) Find A + B b) Find A – B c) Find A * B d) Find 3A + 3B

> A=matrix(c(2,0,1,3),ncol=2)

> A

[,1] [,2]

[1,] 2 1

[2,] 0 3

> B=matrix(c(5,2,4,-1),ncol=2)

> A+B

[,1] [,2]

[1,] 7 5

[2,] 2 2

> A-B

[,1] [,2]

```
[1,]   -3   -3

[2,]   -2    4

> A*B

     [,1] [,2]

[1,]   10    4

 [2,]    0  -3

> 3*A+3*B

     [,1] [,2]

[1,]  21   15

[2,]    6    6

>
```

5. Write a nested loop, where the outer for() loop increments “a” 3 times, and the inner for()loop increments “b” 3times. The breakstatement exits the inner for() loop after 2 incrementations. The nested loop prints the values of variables, “a” and “b“.

6. (a) Suppose we have a fruit basket with 20 apples. Store the number of apples in a variable *my_apples.*
(b) Every tasty fruit basket needs oranges, so we decide to add six oranges. As a data analyst , the reflex is to immediately create a variable *my_oranges* and assign the value 6 to it. Next , calculate howmany pieces of fruit we have in total in the variable *my_fruit.*

```
my_apples <- 20

my_oranges <- 6

my_fruits <- my_apples +my_oranges

my_fruits
```

7. Perform the following operations using R:

a. Initialize 3 character variables named *age,employed* and *salary.*

b. Transform *age* to numeric type and store in the variable *age_clean.*

c. Initialize *employed_clean* with the result obtained by converting *employed* to logical type.

d. Convert the respondent's salary to a numeric and store it in the variable *salary_clean.*

```
age <-  "30"

employed <- "TRUE"

salary <- "50000"

age_clean <- as.numeric(age)

employed_clean <- as.logical(employed)

salary_clean <- as.numeric(salary)
```

8.Create the following vectors in R.

a = (5,10, 15, 20, ..., 160)

b = (87, 86, 85, ..., 56)

Use vector arithmetic to multiply these vectors and call the result d. Select subsets of d to identify the following.

a) What are the 19th, 20th, and 21st elements of d?

```
> a=seq(from=5,to=160,by=5)
> a
 [1]  5  10  15  20  25  30  35  40  45  50  55  60  65  70  75  80  85  90  95
[20] 100 105 110 115 120 125 130 135 140 145 150 155 160
> b=seq(from=87,to=56,by=-1)
> b
 [1] 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63
[26] 62 61 60 59 58 57 56
> d <- a*b
> d[19]
[1] 6555
> d[20]
[1] 6800
> d[21]
[1] 7035
```

b) What are all of the elements of d which are less than 2000?

```
> sort(b)

 [1] 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

[26] 81 82 83 84 85 86 87

> d=a*b

> for(i in d){
```

```
+ if(i<2000){
+ li<- list.append(i)
+ }
+ else{
+ break
+ }
+ }
```

C. How many elements of d are greater than 6000?

```
> for(i in d){
+ if(i>6000){
+ li<- list.append(i)
+ }
}
```

9. You have an employee data-set, which comprises of two columns->"name" and designation", add a third column which would indicate the current date and time.

This is the employee data-set:

name	designation
John	CTO
Sam	CEO
Raj	SDE
Amy	COO
Anne	Analyst

v1

Output:

```
  name designation                  date()
1 John         CTO Tue Dec 19 12:13:58 2017
2  Sam         CEO Tue Dec 19 12:13:58 2017
4  Raj         SDE Tue Dec 19 12:13:58 2017
5  Amy         COO Tue Dec 19 12:13:58 2017
7 Anne     Analyst Tue Dec 19 12:13:58 2017
```

10. Implement a multiplication game. A while loop that gives the user two random numbers from 2 to 12 and asks the user to multiply them. Only exit the loop after five correct answers. Try using as.integer(readline())

```
correct_answers <- 0
```

```
while(correct_answers < 5) {
  num1 <- sample(2:12, 1)
  num2 <- sample(2:12, 1)

  cat(paste("What is", num1, "x", num2, "? "))
  user_answer <- as.integer(readline())
  if(user_answer == num1 * num2) {
  correct_answers <- correct_answers + 1
  cat("Correct!\n")
  } else {
    cat("Incorrect. The correct answer is", num1 * num2, "\n")
  }
}
cat("Congratulations! You got 5 correct answers.\n")
```

11. Create a Attendance sheet of the course “R Programming”.All are present for the course and total strength of the students is 30. There are 15 male students register number from 191611258 to 191611272 and 15 female students of Register number from 191611273 to 191611287. Use data frames to create the Attendance Sheet.(Refer the Sample attendance sheet for 6 students is given below)

Sample Attendance Sheet

```
regno gender attendance
1 191611258   MALEPRESENT
2 191611259   MALEPRESENT
3 191611260   MALEPRESENT
4 191611261 FEMALE        PRESENT
5 191611262 FEMALE        PRESENT
6 191611263 FEMALE        PRESENTK
```

```
male_regno <- seq(from = 191611258, to = 191611272)
female_regno <- seq(from = 191611273, to = 191611287)
```

```
attendance_df <- data.frame(regno = c(male_regno, female_regno),
gender = c(rep("MALE", length(male_regno)), rep("FEMALE", length(female_regno))),
attendance = rep("ABSENT", 30))
attendance_df$attendance <- "PRESENT"
attendance_df
```

12. Create two vectors named v and w with the following contents:
    v :21,55,84,12,13,15
    w : 9,44,22,33,14,35

A) Print the length of the vectors
B) Print all elements of the vectors
C) Print the sum of the elements in each vector.
D)Find the mean of each vector. (Use R's mean() function)
E) Add vectors v and w.
F) Multiply vectors v and w.
G) In vector v select all elements that are greater than 2.
H) In vector w select all elements that are less than 20.

```
v <- c(21, 55, 84, 12, 13, 15)
w <- c(9, 44, 22, 33, 14, 35)
cat("Length of v:", length(v), "\n")
cat("Length of w:", length(w), "\n")
cat("Elements of v:", v, "\n")
cat("Elements of w:", w, "\n")
cat("Sum of elements in v:", sum(v), "\n")
cat("Sum of elements in w:", sum(w), "\n")
cat("Mean of v:", mean(v), "\n")
cat("Mean of w:", mean(w), "\n")
cat("v + w:", v + w, "\n")
cat("v * w:", v * w, "\n")
cat("Elements of v greater than 2:", v[v > 2], "\n")
cat("Elements of w less than 20:", w[w < 20], "\n")
```

13. lapply function is applied to all elements of the input and it returns a list and saaply function is

applied to all elements of the input and it returns a vector. Demonstrate the use of sapply and lapply with the following vector.

movies<- c("SPYDERMAN","BATMAN","VERTIGO","CHINATOWN")

Convert these elements of vector into lowercase letters.

movies <- c("SPYDERMAN", "BATMAN", "VERTIGO", "CHINATOWN")

```
> l <-lapply(movies,tolower)

> l

[[1]]

[1] "spyderman"

[[2]]

[1] "batman"

[[3]]

[1] "vertigo"

[[4]]

[1] "chinatown"

> s <- sapply(movies,tolower)

> s

  SPYDERMAN      BATMAN     VERTIGO  CHINATOWN

"spyderman"    "batman"  "vertigo" "chinatown"

>
```

14. Create dataframe dataframe1 with the following vectors,

Mark1=c(35,45,67)

Mark2=c(56,89,99)

Mark3=c(78,75,83)

Use sapply and lapply function to find minimum marks ,maximum mark and average of all marks

```
> Mark1=c(35,45,67)

> Mark2=c(56,89,99)

> Mark3=c(78,75,83)

> df <- data.frame(Mark1,Mark2,Mark3)
```

```
> sapply(df,min)
Mark1 Mark2 Mark3
   35    56    75
> sapply(df,max)
Mark1 Mark2 Mark3
   67    99    83
> sapply(df,avg)
Error in match.fun(FUN) : object 'avg' not found
> sapply(df,mean)
   Mark1    Mark2    Mark3
49.00000 81.33333 78.66667
>    1apply(df,min)
Error: unexpected symbol in "1apply"
> lapply(df,mean)
$Mark1
[1] 49

$Mark2
[1] 81.33333

$Mark3
[1] 78.66667
> lapply(df,max)
$Mark1
[1] 67
$Mark2
```

[1] 99

$Mark3

[1] 83

>

15. Write a R Program :

a. To find the multiplication table (from 1 to 10)

```
for (i in 1:10) {
for (j in 1:10) {
cat(i*j, "\t")
}
cat("\n")
}
```

b. To find factorial of number

```
factorial <- function(n) {
if (n == 0) {
return (1)
} else {
return (n * factorial(n-1))
}
}
```

c. To check if the input number is odd or even

```
is_odd_or_even <- function(n) {
if (n %% 2 == 0) {
return ("Even")
} else {
return ("Odd")
}
}
```

d. To check if the input number is prime or not

```
is_prime <- function(n) {
if (n < 2) {
return ("Not Prime")
}
for (i in 2:(n-1)) {
if (n %% i == 0) {
return ("Not Prime")
}
}
return ("Prime")
}
```

e. To find sum of natural numbers up-to 10, without formula using loop statement

```
sum <- 0
```

```
for (i in 1:10) {

sum <- sum + i

}

cat("Sum of Natural Numbers up to 10:", sum)
```

16. a. Create a data frame from four given vectors. name =c ('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas')
'Jonas') score = c (12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19) attempts =c
(1, 3, 2, 3, 2, 3, 1, 1, 2, 1) qualify = c ('yes', 'no', 'yes', 'no', 'no', 'yes',
'yes', 'no', 'no', 'yes')

```
name <- c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael',
'Matthew', 'Laura', 'Kevin', 'Jonas')
score <- c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19)
attempts <- c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1)
qualify <- c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')

df <- data.frame(name, score, attempts, qualify)
df
```

b. Write a R program to extract first two rows from a given data frame.

```
first_two_rows <- df[1:2, ]
first_two_rows
```

c. Write a R program to extract 3rd and 5th rows with 1st and 3rd columns from a given data frame

```
third_and_fifth_rows <- df[c(3, 5), c(1, 3)]
third_and_fifth_rows
```

d. Find the average score with respect to first, second, and third attempts. Don't use any special in build function for this task.

```
sums <- c(0, 0, 0)
for (i in 1:nrow(df)) {
  sums[df$attempts[i]] <- sums[df$attempts[i]] + df$score[i]
}

averages <- sums / table(df$attempts)

averages
```

e. Write a R program to create a list containing a vector, a matrix and a list and give names to the elements in the list. Access and print the first and second element of the list

```
vec <- c(1, 2, 3)
mat <- matrix(1:9, nrow=3)
lst <- list(a=1, b=2, c=3)


my_list <- list(my_vector=vec, my_matrix=mat, my_list=lst)
```

```
# Print the first and second elements of the list
print(my_list$my_vector)
print(my_list$my_matrix)
```