***ITA0448 – STATISTICS WITH R PROGRMMING***

***NAME : PRADEEP B***

***REG NO : 192121092***

***DAY-5***

***ASSESMENT***

**1. Write a R program to Create the following details**

**a. x= sample(-50:50, 10, replace=TRUE).and print the value of x**

**INPUT**

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

print("Content of the vector:")

print("10 random integer values between -50 and +50:")

print(v)

**OUTPUT**

[1] "Content of the vector:"

[1] "10 random integer values between -50 and +50:"

[1] 31 -13 -21 42 49 -39 20 12 39 -2

**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.**

**INPUT**

print("Sequence of numbers from 20 to 50:")

print(seq(20,50))

print("Mean of numbers from 20 to 50:")

print(mean(20:50))

print("Sum of numbers from 20 to 50:")

print(sum(20:50))

**OUTPUT**

[1] "Sequence of numbers from 20 to 50:"

[1] 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

[1] "Mean of numbers from 20 to 50:"

[1] 35

[1] "Sum of numbers from 20 to 50:"

[1] 1085

**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**

**b. Print new array**

**INPUT**

print("Two vectors of different lengths:")

v1 = c(1,3,4,5)

v2 = c(10,11,12,13,14,15)

print(v1)

print(v2)

result = array(c(v1,v2),dim = c(3,3,2))

print("New array:")

print(result)

**OUTPUT**

[1] "Two vectors of different lengths:"

[1] 1 3 4 5

[1] 10 11 12 13 14 15

[1] "New array:"

, , 1

[,1] [,2] [,3]

[1,] 1 5 12

[2,] 3 10 13

[3,] 4 11 14

, , 2

[,1] [,2] [,3]

[1,] 15 4 11

[2,] 1 5 12

[3,] 3 10 13

**3. Write a R program to merge two given lists into one list. n1 = list (1,2,3) c1 = list(&quot;Raja&quot;,**

**&quot;Rani&quot;, &quot;Prince&quot;)**

**INPUT**

n1 = list(1,2,3)

c1 = list("Raja", "Rani", "Prince")

print("Original lists:")

print(n1)

print(c1)

print("Merge the said lists:")

mlist = c(n1, c1)

print("New merged list:")

print(mlist)

**OUTPUT**

[1] "Merge the said lists:"

[1] "New merged list:"

[[1]]

[1] 1

[[2]]

[1] 2

[[3]]

[1] 3

[[4]]

[1] "Raja"

[[5]]

[1] "Rani"

[[6]]

[1] "Prince"

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

INPUT

n1 = list(1,2,3)

c1 = list(4,5,6)

print("Original lists:")

print(n1)

print(c1)

print("Convert the lists to vectors:")

v1 = unlist(n1)

v2 = unlist(c1)

print(v1)

print(v2)

print("Add two vectors:")

v = v1 + v2

print("New vector:")

print(v)

**OUTPUT**

[1] "Original lists:"

[[1]]

[1] 1

[[2]]

[1] 2

[[3]]

[1] 3

[[1]]

[1] 4

[[2]]

[1] 5

[[3]]

[1] 6

[1] "Convert the lists to vectors:"

[1] 1 2 3

[1] 4 5 6

[1] "Add two vectors:"

[1] "New vector:"

[1] 5 7 9

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

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

a) INPUT

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

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

A+B

**OUTPUT**

## [,1] [,2]

## [1,] 7 5

## [2,] 2 2

b) INPUT

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

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

A-B

**OUTPUT**

## [,1] [,2]

## [1,] -3 -3

## [2,] -2 4

c) INPUT

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

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

A\*B

**OUTPUT**

[,1] [,2]

[1,] 10 4

[2,] 0 -3

d) INPUT

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

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

3A + 3B

**5. Write a nested loop, where the outer for() loop increments “a” 3 times, and the inner for() loop**

**increments “b” 3 times. The break statement exits the inner for() loop after 2 incrementations. The**

**nested loop prints the values of variables, “a” and “b“.**

**INPUT**

> for(a in 1:3){

+ for(b in 1:3){

+ if(b > 2) {

+ break

+ }

+ print(paste("a =", a, "b =", b))

+ }

+ }

**OUTPUT**

[1] "a = 1 b = 1"

[1] "a = 1 b = 2"

[1] "a = 2 b = 1"

[1] "a = 2 b = 2"

[1] "a = 3 b = 1"

[1] "a = 3 b = 2"

**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 how**

**many pieces of fruit we have in total in the variable my\_fruit.**

**INPUT**

# Assign a value to the variables my\_apples and my\_oranges

my\_apples <- 5

my\_oranges <- 6

# Add these two variables together

my\_apples + my\_oranges

# Create the variable my\_fruit

my\_fruit <- my\_apples + my\_oranges

**OUTPUT**

> # Assign a value to the variables my\_apples and my\_oranges

> my\_apples <- 5

> my\_oranges <- 6

>

> # Add these two variables together

> my\_apples + my\_oranges

[1] 11

>

> # Create the variable my\_fruit

> my\_fruit <- my\_apples + my\_oranges

> # Assign a value to the variables my\_apples and my\_oranges

> my\_apples <- 5

> my\_oranges <- 6

>

> # Add these two variables together

> my\_apples + my\_oranges

[1] 11

>

> # Create the variable my\_fruit

> my\_fruit <- my\_apples + my\_oranges

>

> # see result

> my\_fruit

[1] 11

**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.**

**INPUT**

> # Part (a)

> age <- "25"

> employed <- "TRUE"

> salary <- "$5000"

>

> # Part (b)

> age\_clean <- as.numeric(age)

> age\_clean

> # Part (c)

> employed\_clean <- as.logical(employed)

> employed\_clean

> # Part (d)

> salary\_clean <- as.numeric(gsub("[^0-9]", "", salary))

> salary\_clean

**OUTPUT**

[1] 25

[1] TRUE

[1] 5000

**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?**

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

**(c) How many elements of d are greater than 6000?**

**INPUT**

a<-seq(from=5 ,to=160, by=5) # Create a vector a

print(a)

length(a)

b<-seq(from=87,to=56,by=-1)# Create a vector b

print(b)

length(b)

d<-a\*b # Use vector arithmetic to multiply these vectors and call the result ‘d’.

print(d)

d<-a\*b # Use vector arithmetic to multiply these vectors and call the result ‘d’.

print(d)

j<-d<2000 #What are all of the elements of d which are less than 2000?

d[j]

k<-d>6000 # How many elements of d are greater than 6000?

length(d[k])

**OUTPUT**

## [1] 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85

## [18] 90 95 100 105 110 115 120 125 130 135 140 145 150 155 160

## [1] 32

## [1] 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65

## [24] 64 63 62 61 60 59 58 57 56

## [1] 32

## [1] 435 860 1275 1680 2075 2460 2835 3200 3555 3900 4235 4560 4875 5180

## [15] 5475 5760 6035 6300 6555 6800 7035 7260 7475 7680 7875 8060 8235 8400

## [29] 8555 8700 8835 8960

## [1] 6555 6800 7035

## [1] 435 860 1275 1680

## [1] 16

**9. You have an employee data-set, which comprises of two columns-&gt;”name” and designation”, add**

**a third column which would indicate the current date and time.**

**This is the employee data-set:**

**INPUT**

# create the employee data-set

employee <- data.frame(

name = c("john", "sam", "raj", "amy", "anne"),

designation = c("ceo", "ceo", "sde", "coo", "analyst")

)

# add the current date and time as the third column

employee$datetime <- Sys.time()

# print the updated employee data-set

print(employee)

**OUTPUT**

name designation datetime

1 john ceo 2023-03-24 14:25:12

2 sam ceo 2023-03-24 14:25:12

3 raj sde 2023-03-24 14:25:12

4 amy coo 2023-03-24 14:25:12

5 anne analyst 2023-03-24 14:25:12

**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())**

**INPUT**

> correct\_answers <- 0

> while (correct\_answers < 5) {

+ num1 <- sample(2:12, 1)

+ num2 <- sample(2:12, 1)

+ cat("What is", num1, "\*", num2, "?\n")

+ answer <- as.integer(readline())

+ if (answer == num1 \* num2) {

+ cat("Correct!\n")

+ correct\_answers <- correct\_answers + 1

+ } else {

+ cat("Incorrect. The correct answer is", num1 \* num2, "\n")

+ }

+ }

What is 12 \* 9 ?

cat("Congratulations, you answered 5 questions correctly!")

**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)**

**INPUT**

> # Create a data frame for male students

> male\_regno <- 191611258:191611272

> male\_attendance <- rep("PRESENT", 15)

> male\_df <- data.frame(regno = male\_regno, gender = "MALE", attendance = male\_attendance)

> # Create a data frame for female students

> female\_regno <- 191611273:191611287

> female\_attendance <- rep("PRESENT", 15)

> female\_df <- data.frame(regno = female\_regno, gender = "FEMALE", attendance = female\_attendance)

> # Combine the two data frames

> attendance\_sheet <- rbind(male\_df, female\_df)

> # Print the attendance sheet

> print(attendance\_sheet)

OUTPUT

regno gender attendance

1 191611258 MALE PRESENT

2 191611259 MALE PRESENT

3 191611260 MALE PRESENT

4 191611261 MALE PRESENT

5 191611262 MALE PRESENT

6 191611263 MALE PRESENT

7 191611264 MALE PRESENT

8 191611265 MALE PRESENT

9 191611266 MALE PRESENT

10 191611267 MALE PRESENT

11 191611268 MALE PRESENT

12 191611269 MALE PRESENT

13 191611270 MALE PRESENT

14 191611271 MALE PRESENT

15 191611272 MALE PRESENT

16 191611273 FEMALE PRESENT

17 191611274 FEMALE PRESENT

18 191611275 FEMALE PRESENT

19 191611276 FEMALE PRESENT

20 191611277 FEMALE PRESENT

21 191611278 FEMALE PRESENT

22 191611279 FEMALE PRESENT

23 191611280 FEMALE PRESENT

24 191611281 FEMALE PRESENT

25 191611282 FEMALE PRESENT

26 191611283 FEMALE PRESENT

27 191611284 FEMALE PRESENT

28 191611285 FEMALE PRESENT

29 191611286 FEMALE PRESENT

30 191611287 FEMALE PRESENT

**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&#39;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.**

**INPUT**

# Create the vectors v and w

> v <- c(21, 55, 84, 12, 13, 15)

> w <- c(9, 44, 22, 33, 14, 35)

> # A) Print the length of the vectors

> cat("Length of v:", length(v), "\n")

Length of v: 6

> cat("Length of w:", length(w), "\n\n")

Length of w: 6

> # B) Print all elements of the vectors

> cat("Elements of v:", v, "\n")

Elements of v: 21 55 84 12 13 15

> cat("Elements of w:", w, "\n\n")

Elements of w: 9 44 22 33 14 35

> # C) Print the sum of the elements in each vector

> cat("Sum of v:", sum(v), "\n")

Sum of v: 200

> cat("Sum of w:", sum(w), "\n\n")

Sum of w: 157

> # D) Find the mean of each vector

> cat("Mean of v:", mean(v), "\n")

Mean of v: 33.33333

> cat("Mean of w:", mean(w), "\n\n")

Mean of w: 26.16667

> # E) Add vectors v and w

> cat("Sum of v and w:", v + w, "\n\n")

Sum of v and w: 30 99 106 45 27 50

> # F) Multiply vectors v and w

> cat("Product of v and w:", v \* w, "\n\n")

Product of v and w: 189 2420 1848 396 182 525

> # G) In vector v select all elements that are greater than 2

> cat("Elements of v greater than 2:", v[v > 2], "\n\n")

Elements of v greater than 2: 21 55 84 12 13 15

> # H) In vector w select all elements that are less than 20

> cat("Elements of w less than 20:", w[w < 20], "\n\n")

Elements of w less than 20: 9 14

**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&lt;- c(&quot;SPYDERMAN&quot;,&quot;BATMAN&quot;,&quot;VERTIGO&quot;,&quot;CHINATOWN&quot;)**

**Convert these elements of vector into lowercase letters.**

**INPUT**

# create the vector

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

# use lapply to convert the elements to lowercase letters

lapply(movies, tolower)

# use sapply to convert the elements to lowercase letters

sapply(movies, tolower)

**OUTPUT**

# lapply output

[[1]]

[1] "spyderman"

[[2]]

[1] "batman"

[[3]]

[1] "vertigo"

[[4]]

[1] "chinatown"

# sapply output

[1] "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**

**INPUT**

# create the data-frame with the given vectors

dataframe1 <- data.frame(

Mark1 = c(35, 45, 67),

Mark2 = c(56, 89, 99),

Mark3 = c(78, 75, 83)

)

# use sapply to find minimum, maximum, and average marks

sapply(dataframe1, min) # minimum marks for each subject

sapply(dataframe1, max) # maximum marks for each subject

sapply(dataframe1, mean) # average marks for each subject

# use lapply to find minimum, maximum, and average marks

lapply(dataframe1, min) # minimum marks for each subject

lapply(dataframe1, max) # maximum marks for each subject

lapply(dataframe1, mean) # average marks for each subject

**OUTPUT**

# sapply output

Mark1 Mark2 Mark3

35 56 75

Mark1 Mark2 Mark3

67 99 83

Mark1 Mark2 Mark3

46.00000 87.66667 78.66667

# lapply output

$Mark1

[1] 35

$Mark2

[1] 56

$Mark3

[1] 75

$Mark1

[1] 67

$Mark2

[1] 99

$Mark3

[1] 83

$Mark1

[1] 46

$Mark2

[1] 87.66667

$Mark3

[1] 78.66667

**15. Write a R Program :**

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

**b. To find factorial of number**

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

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

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

**CODE:**

# Using a loop to calculate the sum of natural numbers

> sum <- 0

> for(i in 1:10) {

+ sum <- sum + i

+ }

> # Print the result

> cat("The sum of natural numbers up to 10 is:", sum)

The sum of natural numbers up to 10 is: 55

# Using a loop to calculate the factorial

> factorial <- function(n) {

+ result <- 1

+ for(i in 1:n) {

+ result <- result \* i

+ }

+ return(result)

+ }

> # Example usage:

> factorial(5) # Returns 120

[1] 120

> # Using modulo operator to check if number is even or odd

> is\_even <- function(n) {

+ if(n %% 2 == 0) {

+ return(TRUE)

+ } else {

+ return(FALSE)

+ }

+ }

> # Example usage:

> is\_even(4) # Returns TRUE

[1] TRUE

> is\_even(5) # Returns FALSE

[1] FALSE

> # Using a loop to check if number is prime

> is\_prime <- function(n) {

+ if(n == 1) {

+ return(FALSE)

+ }

+ for(i in 2:(n-1)) {

+ if(n %% i == 0) {

+ return(FALSE)

+ }

+ }

+ return(TRUE)

+ }

> # Example usage:

> is\_prime(7) # Returns TRUE

[1] TRUE

> is\_prime(8) # Returns FALSE

[1] FALSE

> # Using a loop to calculate the sum of natural numbers

> sum <- 0

> for(i in 1:10) {

+ sum <- sum + i

+ }

> # Print the result

> cat("The sum of natural numbers up to 10 is:", sum)

The sum of natural numbers up to 10 is: 55

**16. a. Create a data frame from four given vectors.**

name =c (&#39;Anastasia&#39;, &#39;Dima&#39;, &#39;Katherine&#39;, &#39;James&#39;, &#39;Emily&#39;, &#39;Michael&#39;, &#39;Matthew&#39;, &#39;Laura’, &#39;Kevin&#39;,

&#39;Jonas&#39;)

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 (&#39;yes&#39;, &#39;no&#39;, &#39;yes&#39;, &#39;no&#39;, &#39;no&#39;, &#39;yes&#39;, &#39;yes&#39;, &#39;no&#39;, &#39;no&#39;, &#39;yes&#39;)

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

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

d. Find the average score with respect to first, second, and third attempts. Don’t use any special in

build function for this task.

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

**CODE:**

a)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')

dataframe1 <- data.frame(name, score, attempts, qualify)

b)first\_two\_rows <- dataframe1[1:2,]

c)third\_fifth\_rows <- dataframe1[c(3,5), c(1,3)]

d)# Using for loop

avg\_score <- numeric(3)

for(i in 1:3) {

avg\_score[i] <- mean(dataframe1$score[dataframe1$attempts == i])

}

# Using sapply function

avg\_score <- sapply(1:3, function(x) mean(dataframe1$score[dataframe1$attempts == x]))

e)my\_list <- list(vec = c(1,2,3), mat = matrix(1:6, nrow = 2), lst = list("a", "b", "c"))

names(my\_list) <- c("vector", "matrix", "list")

# Accessing the first and second elements

print(my\_list$vector)

print(my\_list$matrix)