|  |  |  |  |
| --- | --- | --- | --- |
| **Register number** | 192124100 | **Student Name** | MARY SUSHMIJA X M |
| **Course code** | **ITA04** | **Course Name** | **STATISTICS WITH R PROGRAMMING** |

1. Write a R program to Create the following details
   1. x= sample(-50:50, 10, replace=TRUE).and print the value of x
   2. 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.

**PROGRAM:**

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

> print(x)

[1] -5 -34 -9 32 29 44 30 -28 18 14

> seq <- seq(20,50)

> mean\_seq <- mean(seq)

> print(mean\_seq)

[1] 35

> sum\_seq <- sum(seq)

> print(sum\_seq)

[1] 1085

1. 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)
   1. Print vector1, vector2
   2. Print new array

**PROGRAM:**

> vector1 <- c(1,3,4,5)

> vector2 <- c(10,11,12,13,14,15)

> matrix1 <- matrix(vector1[1:9], nrow=3, ncol=3)

> matrix2 <- matrix(vector1[1:9], nrow=3, ncol=3)

> my\_array <- array(c(matrix1,matrix2), dim=c(3,3,2))

> print(vector1)

[1] 1 3 4 5

> print(vector2)

[1] 10 11 12 13 14 15

> print(my\_array)

, , 1

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

[1,] 1 5 NA

[2,] 3 NA NA

[3,] 4 NA NA

, , 2

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

[1,] 1 5 NA

[2,] 3 NA NA

[3,] 4 NA NA

1. Write a R program to merge two given lists into one list. n1 = list (1,2,3) c1 = list("Raja", "Rani", "Prince")
2. Write a R program to convert a given list to vector.n1 = list (1,2,3)c1 = list(4,5,6)

**PROGRAM:**

> n1 <- list(1,2,3)

> c1 <- list("raja", "rani", "prince")

> merged\_list <- c(n1, c1)

> print(merged\_list)

[[1]]

[1] 1

[[2]]

[1] 2

[[3]]

[1] 3

[[4]]

[1] "raja"

[[5]]

[1] "rani"

[[6]]

[1] "prince"

> n2 <- list(1,2,3)

> c2 <- list(4,5,6)

> combined\_list <- c(n2,c2)

> vector\_from\_list <- unlist(combined\_list)

> print(vector\_from\_list)

[1] 1 2 3 4 5 6

1. Consider A=matrix(c(2,0,1,3),ncol=2) and B=matrix(c(5,2,4,-1), ncol=2).
2. Find A + B b) Find A – B c) Find A \* B d) Find 3A + 3B

**PROGRAM:**

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

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

> A\_plus\_B <- A + B

> print(A\_plus\_B)

[,1] [,2]

[1,] 7 5

[2,] 2 2

> A\_minus\_B <- A - B

> print(A\_minus\_B)

[,1] [,2]

[1,] -3 -3

[2,] -2 4

> A\_times\_B <- A%\*%B

> print(A\_times\_B)

[,1] [,2]

[1,] 12 7

[2,] 6 -3

> three\_A\_plus\_three\_B <- 3 \* A + 3 \* B

> print(three\_A\_plus\_three\_B)

[,1] [,2]

[1,] 21 15

[2,] 6 6

1. 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“.

**PROGRAM:**

> for (a in 1:3) {

+ for (b in 1:3) {

+ if (b > 2) {

+ break

+ }

+ cat("a =", a, "b =", b, "\n")

+ }

+ }

a = 1 b = 1

a = 1 b = 2

a = 2 b = 1

a = 2 b = 2

a = 3 b = 1

a = 3 b = 2

1. (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.*

**PROGRAM:**

> my\_apples <- 20

> my\_oranges <- 6

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

> cat("number of apples:",my\_apples,"\n")

number of apples: 20

> cat("numbers of oranges:",my\_oranges,"\n")

numbers of oranges: 6

> cat("total fruits:",my\_fruit,"\n")

total fruits: 26

1. Perform the following operations using R:
   1. Initialize 3 character variables named *age,employed* and *salary.*
   2. Transform *age* to numeric type and store in the variable *age\_clean*.
   3. Initialize *employed\_clean* with the result obtained by converting *employed* to logical type.
   4. Convert the respondent’s salary to a numeric and store it in the variable *salary\_clean.*

**PROGRAM:**

> age <- "25"

> employed <- "yes"

> salary <- "$50,000"

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

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

> salary\_clean <- as.numeric(gsub("\\$|,","",salary))

> age\_clean

[1] 25

> employed\_clean

[1] NA

> salary\_clean

[1] 50000

1. 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.

1. What are the 19th, 20th, and 21st elements of d?
2. What are all of the elements of d which are less than 2000?
3. How many elements of d are greater than 6000?

**PROGRAM**:

> a <- seq(from = 5, to =160, by =5)

> b <- seq(from =87, to =56, by =-1)

> d <- a \* b

> d[19:21]

[1] 6555 6800 7035

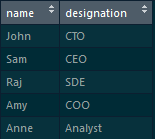
> d[d < 2000]

[1] 435 860 1275 1680

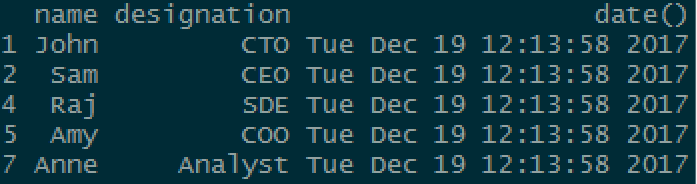
> sum(d > 6000)

1. 16
2. 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:



Output:



**PROGRAM:**

> name <- c("John", "Mary", "Paul", "Lisa", "David")

> designation <- c("Manager", "Supervisor", "Engineer", "Analyst", "Technician")

> employee\_df <- data.frame(name, designation)

> employee\_df$date\_time <- Sys.time()

> employee\_df

name designation date\_time

1 John Manager 2023-03-24 14:35:22

2 Mary Supervisor 2023-03-24 14:35:22

3 Paul Engineer 2023-03-24 14:35:22

4 Lisa Analyst 2023-03-24 14:35:22

5 David Technician 2023-03-24 14:35:22

1. 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())

**PROGRAM**:

+ correct\_answer <-0

+ while (correct\_answer < 5){

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

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

+ cat("what is", num1,"times", num2, "? ")

+ user\_answer <- as.integer(readline())

+ if (user\_answer == num1 \* num2) {

+ cat("correct!\n")

+ correct\_answer <- correct\_answer + 1

+ } else {

+ cat("incorrect.the correct answer is", num1 \* num2, "\n")

+ }

+ }

+ cat("congratulations! you answered 5 questions correctly.\n")

1. 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)

*S ample Attendance Sheet*

regno gender attendance

1. 191611258 MALE PRESENT
2. 191611259 MALE PRESENT
3. 191611260 MALE PRESENT
4. 191611261 FEMALE PRESENT
5. 191611262 FEMALE PRESENT
6. 191611263 FEMALE PRESENT

**PROGRAM:**

> male\_regno <- seq(191611258, 191611272)

> female\_regno <- seq(191611273, 191611287)

> attendance <- data.frame(regno = c(male\_regno, female\_regno),

+ gender = c(rep("MALE", 15), rep("FEMALE", 15)),

+ attendance = rep("PRESENT", 30))

> attendance

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

1. 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

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

**PROGRAM:**

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

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

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

Length of v: 6

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

Length of w: 6

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

v: 21 55 84 12 13 15

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

w: 9 44 22 33 14 35

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

Sum of v: 200

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

Sum of w: 157

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

Mean of v: 33.33333

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

Mean of w: 26.16667

> vw\_sum <- v + w

> cat("v + w:", vw\_sum, "\n")

v + w: 30 99 106 45 27 50

> vw\_prod <- v \* w

> cat("v \* w:", vw\_prod, "\n")

v \* w: 189 2420 1848 396 182 525

> v\_gt\_2 <- v[v > 2]

> cat("Elements in v greater than 2:", v\_gt\_2, "\n")

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

> w\_lt\_20 <- w[w < 20]

> cat("Elements in w less than 20:", w\_lt\_20, "\n")

Elements in w less than 20: 9 14

1. 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.

**PROGRAM:**

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

> lower\_list <- lapply(movies, tolower)

> print(lower\_list)

[[1]]

[1] "spyderman"

[[2]]

[1] "batman"

[[3]]

[1] "vertigo"

[[4]]

[1] "chinatown"

> lower\_vec <- sapply(movies, tolower)

> print(lower\_vec)

SPYDERMAN BATMAN VERTIGO CHINATOWN

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

1. 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

**PROGRAM:**

> Mark1 <- c(35, 45, 67)

> Mark2 <- c(56, 89, 99)

> Mark3 <- c(78, 75, 83)

> dataframe1 <- data.frame(Mark1, Mark2, Mark3)

> lapply(dataframe1, function(x) c(min(x), max(x), mean(x)))

$Mark1

[1] 35 67 49

$Mark2

[1] 56.00000 99.00000 81.33333

$Mark3

[1] 75.00000 83.00000 78.66667

> sapply(dataframe1, function(x) c(min(x), max(x), mean(x)))

Mark1 Mark2 Mark3

[1,] 35 56.00000 75.00000

[2,] 67 99.00000 83.00000

[3,] 49 81.33333 78.66667

1. Write a R Program :
   1. To find the multiplication table (from 1 to 10)
   2. To find factorial of number
   3. To check if the input number is odd or even
   4. To check if the input number is prime or not
   5. To find sum of natural numbers up-to 10, without formula using loop statement

**PROGRAM:**

> for(i in 1:1) {

+ cat(paste("Multiplication table of", i, "\n"))

+ for(j in 1:1) {

+ cat(paste(i, "x", j, "=", i\*j, "\n"))

+ }

+ cat("\n")

+ }

Multiplication table of 1

1 x 1 = 1

> factorial <- function(n) {

+ if(n < 0) {

+ return(NA)

+ } else if(n == 0) {

+ return(1)

+ } else {

+ return(n \* factorial(n-1))

+ }

+ }

> is\_odd\_or\_even <- function(n) {

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

+ return("Even")

+ } else {

+ return("Odd")

+ }

+ }

> is\_prime <- function(n) {

+ if(n <= 1) {

+ return(FALSE)

+ } else if(n <= 3) {

+ return(TRUE)

+ } else if(n %% 2 == 0 || n %% 3 == 0) {

+ return(FALSE)

+ }

+

+ i <- 5

+ while(i \* i <= n) {

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

+ return(FALSE)

+ }

+ i <- i + 6

+ }

+

+ return(TRUE)

+ }

> sum\_of\_numbers <- 0

> for(i in 1:10) {

+ sum\_of\_numbers <- sum\_of\_numbers + i

+ }

> print(sum\_of\_numbers)

[1] 55

1. a. Create a data frame from four given vectors.

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')

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

**PROGRAM:**

> 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)

> first\_two\_rows <- df[1:2,]

> subset\_df <- df[c(3,5), c(1,3)]

> first\_attempt <- subset(df, attempts == 1)

> second\_attempt <- subset(df, attempts == 2)

> third\_attempt <- subset(df, attempts == 3)

> avg\_score\_1 <- sum(first\_attempt$score) / nrow(first\_attempt)

> avg\_score\_2 <- sum(second\_attempt$score) / nrow(second\_attempt)

> avg\_score\_3 <- sum(third\_attempt$score) / nrow(third\_attempt)

> vector <- c(1, 2, 3)

> matrix <- matrix(1:9, nrow = 3, ncol = 3)

> nested\_list <- list(a = 4, b = 5)

> main\_list <- list(vector, matrix, nested\_list)

> names(main\_list) <- c('vector', 'matrix', 'nested\_list')

> print(main\_list[[1]])

[1] 1 2 3

> print(main\_list[[2]])

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

[1,] 1 4 7

[2,] 2 5 8

[3,] 3 6 9