SAVITRIBAI PHULE PUNE UNIVERSITY

S. Y. B. B. A. (C.A.) SEMESTER III

(CBCS 2019 PATTERN)

LAB BOOK

NAME : LALIT DEVIDAS PATIL

COLLEGE NAME: SINHGAD COLLEGE OF ARTS & COMMERCE WARJE PUNE

ROLL NO : 106 DIVISION:B SEAT NO:

1	3	2	8	5

ACADEMIC YEAR : 2023-24

Certificate

This is to certify that

Mr. PATIL LALIT DEVIDAS

Seat Number_____of S.Y.BBA(CA) Sem- III has Successfully completed Laboratory course (BIG DATA) in the Year . He has scored mark out of 10 (For Lab Book).

Subject Teacher **H.O.D./Coordinator**

Internal Examiner **External Examiner**

Slip 11

Q1- Write a script in R to create two vectors of different lengths and give these vectors as input to array and print addition and subtraction of those matrices.

```
vector1 <- c(1, 2, 3, 4)
vector2 <- c(5, 6, 7)
len1 <- length(vector1)
len2 <- length(vector2)
max_len <- max(len1, len2)
vector1 <- c(vector1, rep(NA, max_len - len1))
vector2 <- c(vector2, rep(NA, max_len - len2))
matrix1 <- array(vector1, dim = c(1, max_len))
matrix2 <- array(vector2, dim = c(1, max_len))
cat("Matrix 1:\n", matrix1, "\n\n")
cat("Matrix 2:\n", matrix2, "\n\n")
addition_matrix <- matrix1 + matrix2
subtraction_matrix <- matrix1 - matrix2
cat("Addition Matrix:\n", addition_matrix, "\n\n")
cat("Subtraction Matrix:\n", subtraction_matrix, "\n")
```

```
Matrix 1:
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4

Matrix 2:
     [,1] [,2] [,3] [,4]
[1,]    5    6    7   NA

Addition Matrix:
     [,1] [,2] [,3] [,4]
[1,]    6    8   10   NA

Subtraction Matrix:
     [,1] [,2] [,3] [,4]
[1,]   -4   -4   -4   NA
```

Slip 12

Q2- Write an R Program to calculate Multiplication Table.

```
calculateMultiplicationTable <- function(num) {
  cat(paste("Multiplication Table for", num, ":\n"))
  for (i in 1:10) {
    result <- num * i
    cat(paste(num, "*", i, "=", result, "\n"))
  }}
num <- 5
calculateMultiplicationTable(num)
```

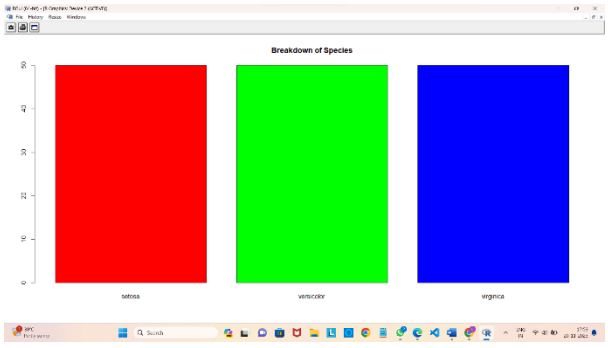
```
Multiplication Table for 5 :
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

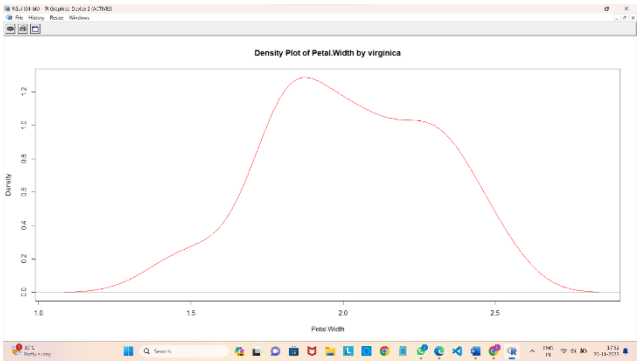
Slip 13

Q3 - Consider the inbuilt iris dataset

i) Create a variable “y” and attach to it the output attribute of the “iris” dataset.
ii) Create a barplot to breakdown your output attribute.
iii) Create a density plot matrix for each attribute by classvalue.

```
data(iris)
y <- iris$Species
barplot(table(y), col = rainbow(length(unique(y))), main =
"Breakdown of Species")
par(mfrow = c(2, 2))
for (i in levels(y)) {
  for (j in 1:4) {
    den <- density(iris[iris$Species == i, j], na.rm = TRUE)
    plot(den, main = paste("Density Plot of", names(iris)[j], "by", i),
         xlab = names(iris)[j], col = rainbow(length(levels(y))))
  }
}
```





Slip 14

Q4- Write an R program to concatenate two given factor in a single factor and display in descending order.

```
concatenateAndDisplay <- function(factor1, factor2) {
  concatenated_factor <- factor(c(factor1, factor2))
  sorted_factor <-
factor(concatenated_factor[order(concatenated_factor, decreasing =
TRUE)])
  print(sorted_factor)
}
factor1 <- factor(c("A", "B", "C"))
factor2 <- factor(c("D", "E", "F"))
concatenateAndDisplay(factor1, factor2)
```

```
[1] F E D C B A
Levels: F E D C B A
```

Slip 15

Q5- Write an R program to extract the five of the levels of factor created from a random sample from the LETTERS.

```
set.seed(123)
sample_letters <- sample(LETTERS, 20, replace = TRUE)
letter_factor <- factor(sample_letters)
selected_levels <- levels(letter_factor)[1:5]
cat("Original Factor:\n", letter_factor, "\n\n")
cat("Selected Levels:\n", selected_levels, "\n")
```

```
Original Factor:
 N M N V F M V D I C N B F B C C B D V N N

Selected Levels:
 B C D F I
```

Slip 16

Q6- Consider the inbuilt mtcar dataset
i) Subset the vector, "mtcars[,1]", for values greater than "15.0".
ii) Subset "airquality" for "Ozone" greater than "28", or "Temp" greater than "70". Return the first five rows.
iii) Subset "airquality" for "Ozone" greater than "100". Select the columns "Ozone", "Temp", "Month" and "Day" only.

```
mtcars_subset <- mtcars[mtcars[, 1] > 15.0, ]
airquality_subset <- airquality[airquality$Ozone > 28 |
airquality$Temp > 70, ][1:5, ]
airquality_subset_100 <- airquality[airquality$Ozone > 100,
c("Ozone", "Temp", "Month", "Day")]
cat("i) Subset of mtcars for values greater than 15.0:\n")
print(mtcars_subset)
cat("\nii) Subset of airquality for Ozone greater than 28 or Temp
greater than 70 (first five rows):\n")
print(airquality_subset)
cat("\niii) Subset of airquality for Ozone greater than 100 (selected
columns):\n")
print(airquality_subset_100)
```

```
i) Subset of mtcars for values greater than 15.0:
                mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1

ii) Subset of airquality for Ozone greater than 28 or Temp greater than 70 (first fi
  Ozone Solar.R Wind Temp Month Day
2    36     118  8.0   72     5   2
5    28       0 14.9   66     5   5
6    23      67  6.9   65     5   6
9    19     299  8.6   74     5   9
12   16     256  9.7   71     5  12

iii) Subset of airquality for Ozone greater than 100 (selected columns):
    Ozone Temp Month Day
62    168   74     6   1
116   122   79     8  25
```

Slip 17

Q7- Write an R Program to calculate Decimal into binary of a given number.

```
decimal_to_binary <- function(decimal_num) {
  binary_num <- vector(mode = "character", length = 0)
  if (decimal_num == 0) {
    binary_num <- "0"
  } else {
    while (decimal_num > 0) {
      binary_num <- c(as.character(decimal_num %% 2), binary_num)
      decimal_num <- decimal_num %/% 2
    }
}
  result <- paste(binary_num, collapse = "")
  return(result)
}
decimal_number <- 25
binary_representation <- decimal_to_binary(decimal_number)
cat("Decimal:", decimal_number, "\n")
cat("Binary:", binary_representation, "\n")
```

```
Decimal: 25
Binary: 11001
```

Slip 18

Q8- Write an R program to create three vectors a,b,c with 3 integers. Combine the three vectors to become a 3×3 matrix where each column represents a vector. Print the content of the matrix.

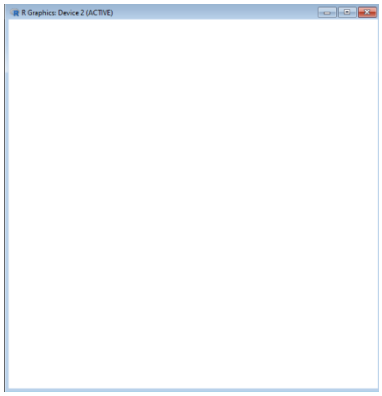
```
a <- c(1, 2, 3)
b <- c(4, 5, 6)
c <- c(7, 8, 9)
matrix_combined <- cbind(a, b, c)
print(matrix_combined)
```

```
     a b c
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9
```

Slip 19

Q9- Write an R program to draw an empty plot and an empty plot specify the axes limits of the graphic.

```
plot.new()
plot.new()
plot.window(xlim = c(0, 10), ylim = c(0, 20))
```



Slip 20

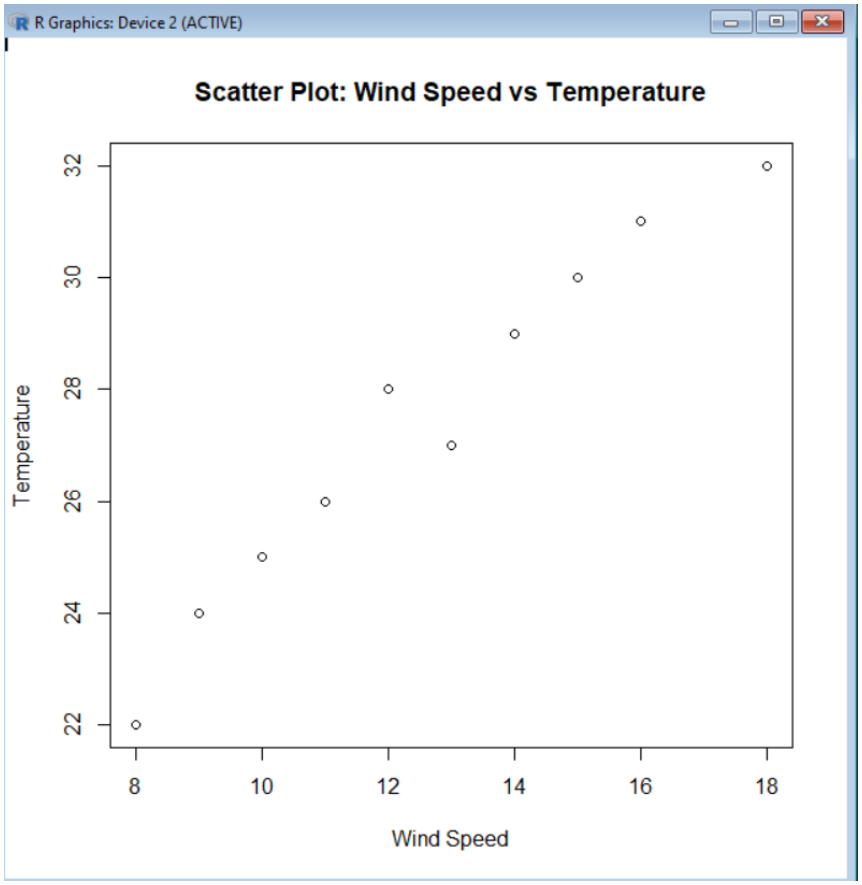
Q10- Consider Weather dataset

i) Selecting using the column number

ii) Selecting using the column name

iii) Make a scatter plot to compare Wind speed and temperature

```
weather <- data.frame(
  Date = seq(as.Date("2022-01-01"), as.Date("2022-01-10"), by = "1 day"),
  Temperature = c(25, 28, 22, 30, 32, 29, 27, 26, 24, 31),
  WindSpeed = c(10, 12, 8, 15, 18, 14, 13, 11, 9, 16)
)
selected_column_by_number <- weather[, 2]
selected_column_by_name <- weather$Temperature
plot(weather$WindSpeed, weather$Temperature,
     xlab = "Wind Speed", ylab = "Temperature",
     main = "Scatter Plot: Wind Speed vs Temperature")
```



Slip 22

Q12- Write an R program to print the numbers from 1 to 100 and print "SY" for multiples of 3, print "BBACA" for multiples of 5, and print "SYBBACA" for multiples of both.

```
for (i in 1:100) {
  if (i %% 3 == 0 && i %% 5 == 0) {
    print("SYBBACA")
  } else if (i %% 3 == 0) {
    print("SY")
  } else if (i %% 5 == 0) {
    print("BBACA")
  } else {
    print(i)
  }
}
```

```
[1] 1
[1] 2
[1] "SY"
[1] 4
[1] "BBACA"
[1] "SY"
[1] 7
[1] 8
[1] "SY"
[1] "BBACA"
[1] 11
[1] "SY"
[1] 13
[1] 14
[1] "SYBBACA"
[1] 16
[1] 17
[1] "SY"
[1] 19
[1] "BBACA"
[1] "SY"
```

Slip 23

Q13- Write a script in R to create two vectors of different lengths and give these vectors as input to array and print second row of second matrix of the array.

```
vector1 <- c(1, 2, 3)
vector2 <- c(4, 5, 6, 7)
array_result <- array(c(vector1, vector2), dim = c(3, 2, 2))
print(array_result)
print(array_result[2, , 2])
```

```
, , 1

     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6

, , 2

     [,1] [,2]
[1,]    0    7
[2,]    0    0

[1] 0 0
```

Slip 24

Q14- Write a script in R to create two vectors of different lengths and give these vectors as input to array and print Multiplication of those matrices.

```
vector1 <- c(1, 2, 3)
vector2 <- c(4, 5, 6, 7)
array_result <- array(c(vector1, vector2), dim = c(3, 2, 2))
print(array_result)
result_matrix <- array_result[,,1] %*% array_result[,,2]
print(result_matrix)
```

```
, , 1

     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6

, , 2

     [,1] [,2]
[1,]    0    7
[2,]    0    0

     [,1] [,2]
[1,]    8   29
[2,]   15   40
[3,]   18   51
```

Slip 25

Q15- Write an R program to create a list of elements using vectors, matrices and a functions. Print the content of the list.

```
vector1 <- c(1, 2, 3)
vector2 <- c("a", "b", "c")
matrix1 <- matrix(1:9, nrow = 3, ncol = 3)
matrix2 <- matrix(c("x", "y", "z"), nrow = 1, ncol = 3)
multiply_by_two <- function(x) {
  return(x * 2)
}
my_list <- list(
  numeric_vector = vector1,
  character_vector = vector2,
  numeric_matrix = matrix1,
  character_matrix = matrix2,
  custom_function = multiply_by_two
)
print(my_list)
```

```
$numeric_vector
[1] 1 2 3

$character_vector
[1] "a" "b" "c"

$numeric_matrix
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

$character_matrix
     [,1] [,2] [,3]
[1,] "x"  "y"  "z"

$custom_function
function(x) {
  return(x * 2)
}
```

Slip 26

Q16- Write a script in R to create an array, passing in a vector of values and a vector of dimensions. Also provide names for each dimension.

```
values <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
dimensions <- c(3, 2, 2)
dimnames_list <- list(
  c("row1", "row2", "row3"),
  c("col1", "col2"),
  c("depth1", "depth2")
)
my_array <- array(values, dim = dimensions, dimnames = dimnames_list)
print(my_array)
```

```
, , depth1

     col1 col2
row1    1    4
row2    2    5
row3    3    6

, , depth2

     col1 col2
row1    7   10
row2    8   11
row3    9   12
```

Slip 27

Q17- Write an R Program to calculate binary into Decimal of a given number.

```
binary_to_decimal <- function(binary_number) {
  decimal_result <- 0
  binary_digits <- rev(as.numeric(strsplit(as.character(binary_number), "")[[1]]))
  for (i in seq_along(binary_digits)) {
    decimal_result <- decimal_result + binary_digits[i] * 2^(i - 1)
  }
  return(decimal_result)}
binary_number <- "1101"
decimal_result <- binary_to_decimal(binary_number)
cat("Decimal equivalent of", binary_number, "is:", decimal_result, "\n")
```

```
Decimal equivalent of 1101 is: 13
```

Slip 28

Q18- Write an R program to convert a given matrix to a list and print list in ascending order.

```
my_matrix <- matrix(c(5, 2, 8, 1, 7, 3, 4, 6, 9), nrow = 3, byrow = TRUE)
my_list <- as.list(my_matrix)
flat_list <- unlist(my_list)
sorted_list <- sort(flat_list)
cat("Original Matrix:\n", my_matrix, "\n\n")
cat("List in Ascending Order:\n", sorted_list, "\n")
```

```
Original Matrix:
 5 1 4 2 7 6 8 3 9

> cat("List in Ascending Order:\n", sorted_list, "\n")
List in Ascending Order:
 1 2 3 4 5 6 7 8 9
>
```

Slip 29

Q19- Write a script in R to create a list of students and perform the following

1) Give names to the students in the list.

2) Add a student at the end of the list.

3) Remove the firstStudent.

4) Update the second last student.

```
students <- list(
  student1 = c("Sagar", 25, " Mathematics "),
  student2 = c("Aditya", 22, " Biology "),
  student3 = c("Rohit", 24, "Physics")
)
names(students) <- c("firstStudent", "secondStudent",
"thirdStudent")
new_student <- c("Hitesh", 23, "Chemistry")
students <- c(students, list(newStudent = new_student))
students <- students[-which(names(students) == "firstStudent")]
students[which(names(students) == "secondStudent")] <- c("Lalit",
26, "Computer Science")
print(students)
```

```
$secondStudent
[1] "Lalit"            "26"               "Computer Science"

$thirdStudent
[1] "Rohit"   "24"      "Physics"

$newStudent
[1] "Hitesh"   "23"       "Chemistry"
```

Slip 30

Q20- Write an R program to sort a list of 10 strings in ascending and descending order.

```
string_list <- c("apple", "orange", "banana", "grape", "kiwi", "melon",
"pear", "cherry", "pineapple", "strawberry")
ascending_order <- sort(string_list)
descending_order <- sort(string_list, decreasing = TRUE)
cat("Ascending Order:\n", ascending_order, "\n\n")
cat("Descending Order:\n", descending_order, "\n")
```

```
Ascending Order:
 apple    banana    cherry     grape      kiwi     melon   orange      pear pineapple

Descending Order:
 strawberry pineapple      pear   orange     melon      kiwi     grape    cherry    ba
```