pp
Experiment No: 1
Experiment Name: To perform the basic mathematical operations in R programming.
Name:
Roll No: -

R Script: -

```
#Assignment
s <- 10
print(s)

#case sensitive
a <- 5
b <- 2
c <- 1
print(a+B+c)

#Arithmetic Operations
#creation of vector

a <- c (10, 20, 30, 40, 50)
b <- c (1, 2, 3, 4, 5)
print(a)
print(b)

#Addition of two vector
Result <- a+b
print(Result)

#Subtraction of two vector
Result <- a-b
print(Result)

#Multiplication of two vector
Result <- a*b
print(Result)

#Division of two vector
Result <- a/b
print(Result)

#options ()
1/7#by default it will show 7 digits output.

options (digits = 3)#by using this it will show only 3 digits after decimal point
1/7

#Miscellaneous Mathematical functions
```

```
x<-20
abs(x)  #Absolute Value
sqrt(x) #square root
exp(x)  #exponential transformation
log(x)  #logarithmic transformation
cos(x)  #cosine and other trigonometric transformation

#infinite and Nan Number
y<-5
z<-6

ls()  #List all object
exists("y")  #identify R object with 'y' name
rm(y)        #remove object.
rm(y,z)      #remove multiple object.
rm(list=ls()) #remove everything on working environment.
```

OUTPUT-

```
> #Assignment
> s <- 10
> print(s)
[1] 10
>
> #case sensitive
> a <- 5
> b <- 2
> c <- 1
> print(a+B+c)
Error in print(a + B + c) : object 'B' not found
>
> #Arithmetic Operations
> #creation of vector
>
> a <- c(10, 20, 30, 40, 50)
> b <- c(1, 2, 3, 4, 5)
> print(a)
[1] 10 20 30 40 50
> print(b)
[1] 1 2 3 4 5
>
> #Addition of two vector
> Result <- a+b
> print(Result)
[1] 11 22 33 44 55
>
> #Subtraction of two vector
> Result <- a-b
> print(Result)
[1]  9 18 27 36 45
```

```
>
> #Multiplication of two vector
> Result <- a*b
> print(Result)
[1]  10  40  90 160 250
>
> #Division of two vector
> Result <- a/b
> print(Result)
[1] 10 10 10 10 10
>
> #options()
> 1/7#by default it will show 7 digits output.
[1] 0.1428571
>
> options(digits = 3)#by using this it will show only 3 digits after decimal point
> 1/7
[1] 0.143
>
> #Miscellaneous Mathematical functions
>
> x<-20
> abs(x)  #Absolute Value
[1] 20
> sqrt(x) #square root
[1] 4.47
> exp(x)  #exponential transformation
[1] 4.85e+08
> log(x)  #logarithmic transformation
[1] 3
> cos(x)  #cosine and other trigonometric transformation
[1] 0.408
>
> #infinite and Nan Number
> y<-5
> z<-6
>
> ls()  #List all object
[1] "a"      "b"      "c"      "Result" "s"      "x"      "y"      "z"
> exists("y")   #identify R object with 'y' name
[1] TRUE
> rm(y)         #remove object.
> rm(y,z)       #remove multiple object.
Warning message:
In rm(y, z) : object 'y' not found
> rm(list=ls()) #remove everything on working environment.
```

Experiment No: 2

Experiment Name: Write program for creating and manipulating R objects in R-Vectors, Matrices, Array, Dataframes, List.

Name:

Roll No:-

Atomic Vector:-

Numeric vector: -

```
num_vec<-c(10.1, 10.2, 33.2)
num_vec
```

Integer vector: -

```
num<-c(2L,6L,4L,9L)
num
```

Character vector: -

```
fruits<-c("Mango","apple","papaya")
print(fruits)
```

Logical vector: -

```
a<-as.integer(20)
b<-as.integer(10)
log_vec<-c(a<b,b<a,a>b,b>a)
log_vec
```

Operations on Vector: -

1)combining vectors:

```
data_vec<-c(names,num)
data_vec
```

2)Arithmetic operations:

```
a<-c(1,3,5,7)
b<-c(2,4,6,8)
a+b
a-b
a*b
a/b
```

3)Logical Index vector:

```
z<-c(1,2,3,4,5,6)
z[c(TRUE,FALSE,TRUE,TRUE,FALSE,TRUE)]
```

4)Numeric Index: -

```
q<-c("shubham","arpita","nishka","gunjan","vaishali","sumit")
q[2]
q[-4]
q[15]
```

5)Duplicate Index: -

```
q<-c("shubham","arpita","nishka","gunjan","vaishali","sumit")
```

```
q[c(2,4,4,3)]
```

6)Range Indexes: -

```
q<-c("shubham","arpita","nishka","gunjan","vaishali","sumit")
b<-q[2:5]
b
```

7)out-of-order Indexes: -

```
q<-c("shubham","arpita","nishka","gunjan","vaishali","sumit")
q[c(2,1,3,4,5,6)]
```

8)Named vectors members: -

```
z=c("Roshani","Kawale")
z
names(z)=c("FirstName","LastName")
z
z["FirstName"]
```

OUTPUT: -

```
> #Numeric vector: -
>   num_vec<-c(10.1, 10.2, 33.2)
>   num_vec
[1] 10.1 10.2 33.2
>
> #Integer vector: -
>   num<-c(2L,6L,4L,9L)
>   num
[1] 2 6 4 9
>
> #Character vector: -
>   fruits<-c("Mango","apple","papaya")
>   print(fruits)
[1] "Mango"  "apple"  "papaya"
>
> #Logical vector: -
>   a<-as.integer(20)
>   b<-as.integer(10)
>   log_vec<-c(a<b,b<a,a>b,b>a)
>   log_vec
[1] FALSE  TRUE  TRUE FALSE
>
> #Operations on Vector: -
> #1)combining vectors:
>   data_vec<-c(names,num)
>   data_vec
[[1]]
function (x)  .Primitive("names")

[[2]]
[1] 2

[[3]]
```

```
[1] 6

[[4]]
[1] 4

[[5]]
[1] 9

>
> #2)Arithmetic operations:
>  a<-c(1,3,5,7)
>  b<-c(2,4,6,8)
>  a+b
[1]  3  7 11 15
>  a-b
[1] -1 -1 -1 -1
>  a*b
[1]  2 12 30 56
>  a/b
[1] 0.5000000 0.7500000 0.8333333 0.8750000
>
> #3)Logical Index vector:
>  z<-c(1,2,3,4,5,6)
>  z[c(TRUE,FALSE,TRUE,TRUE,FALSE,TRUE)]
[1] 1 3 4 6
>
>
> #4)Numeric Index: -
>  q<-c("shubham","arpita","nishka","gunjan","vaishali","sumit")
>  q[2]
[1] "arpita"
>  q[-4]
[1] "shubham"  "arpita"   "nishka"   "vaishali" "sumit"
>  q[15]
[1] NA
>
> #5)Duplicate Index: -
>  q<-c("shubham","arpita","nishka","gunjan","vaishali","sumit")
>  q[c(2,4,4,3)]
[1] "arpita" "gunjan" "gunjan" "nishka"
>
> #6)Range Indexes: -
>  q<-c("shubham","arpita","nishka","gunjan","vaishali","sumit")
>  b<-q[2:5]
>  b
[1] "arpita"   "nishka"   "gunjan"   "vaishali"
>
> #7)out-of-order Indexes: -
>  q<-c("shubham","arpita","nishka","gunjan","vaishali","sumit")
>  q[c(2,1,3,4,5,6)]
[1] "arpita"   "shubham"  "nishka"   "gunjan"   "vaishali" "sumit"
>
> #8)Named vectors members: -
```

```
> z=c("Roshani","Kawale")
> z
[1] "Roshani" "Kawale"
> names(z)=c("FirstName","LastName")
> z
FirstName  LastName
"Roshani"  "Kawale"
> z["FirstName"]
FirstName
"Roshani"
```

```
#creation of matrix:-
        P <- matrix(c(5:16), nrow = 4, byrow = TRUE)
        print(P)

        Q <- matrix(c(3:14), nrow = 4, byrow = FALSE)
        print(Q)

#operations on Matrix:-
#1)Addition:-
        sum<-P+Q
        print(sum)
#2)Subtraction:-
        sub<-P-Q
        print(sub)
#3)Multiplication(*):-
        mult<-P*Q
        print(mult)
#4)Multiplication(by constant):-
        mult<-P*5
        print(mult)
#5)Division:-
        div<-P/Q
        div
```

OUTPUT:-

```
#creation of matrix:-
>       P <- matrix(c(5:16), nrow = 4, byrow = TRUE)
>       print(P)
     [,1] [,2] [,3]
[1,]    5    6    7
[2,]    8    9   10
[3,]   11   12   13
[4,]   14   15   16
>
>       Q <- matrix(c(3:14), nrow = 4, byrow = FALSE)
>       print(Q)
     [,1] [,2] [,3]
[1,]    3    7   11
[2,]    4    8   12
[3,]    5    9   13
[4,]    6   10   14
>
```

```
> #operations on Matrix:-
> #1)Addition:-
>         sum<-P+Q
>         print(sum)
     [,1] [,2] [,3]
[1,]    8   13   18
[2,]   12   17   22
[3,]   16   21   26
[4,]   20   25   30
> #2)Subtraction:-
>         sub<-P-Q
>         print(sub)
     [,1] [,2] [,3]
[1,]    2   -1   -4
[2,]    4    1   -2
[3,]    6    3    0
[4,]    8    5    2
> #3)Multiplication(*):-
>         mult<-P*Q
>         print(mult)
     [,1] [,2] [,3]
[1,]   15   42   77
[2,]   32   72  120
[3,]   55  108  169
[4,]   84  150  224
> #4)Multiplication(by constant):-
>         mult<-P*5
>         print(mult)
     [,1] [,2] [,3]
[1,]   25   30   35
[2,]   40   45   50
[3,]   55   60   65
[4,]   70   75   80
> #5)Division:-
>         div<-P/Q
>  div
         [,1]     [,2]      [,3]
[1,] 1.666667 0.8571429 0.6363636
[2,] 2.000000 1.1250000 0.8333333
[3,] 2.200000 1.3333333 1.0000000
[4,] 2.333333 1.5000000 1.1428571
>
```

```
#creation of Arrays:-
vec1 <-c(1,3,5)
vec2 <-c(10,11,12,13,14,15)
res <- array(c(vec1,vec2),dim=c(3,3,2))
print(res)

#Naming Of Arrays
col_names <- c("Col1","Col2","Col3")
row_names <- c("Row1","Row2","Row3")
matrix_names <- c("Matrix1","Matrix2")
```

```
res <-
array(c(vec1,vec2),dim=c(3,3,2),dimnames=list(row_names,col_names,matrix_names))
print(res)
```

OUTPUT: -

```
#creation of Arrays:-
> vec1 <-c(1,3,5)
> vec2 <-c(10,11,12,13,14,15)
> res <- array(c(vec1,vec2),dim=c(3,3,2))
> print(res)
, , 1

     [,1] [,2] [,3]
[1,]    1   10   13
[2,]    3   11   14
[3,]    5   12   15

, , 2

     [,1] [,2] [,3]
[1,]    1   10   13
[2,]    3   11   14
[3,]    5   12   15

>
> #Naming Of Arrays
> col_names <- c("Col1","Col2","Col3")
> row_names <- c("Row1","Row2","Row3")
> matrix_names <- c("Matrix1","Matrix2")
> res <-
array(c(vec1,vec2),dim=c(3,3,2),dimnames=list(row_names,col_names,matrix_names))
> print(res)
, , Matrix1

     Col1 Col2 Col3
Row1    1   10   13
Row2    3   11   14
Row3    5   12   15

, , Matrix2

     Col1 Col2 Col3
Row1    1   10   13
Row2    3   11   14
Row3    5   12   15
```

#Creation Of DataFrame

```
stud.data<- data.frame(
   student_id = c (1:5),
   student_name = c("Shubham","Arpita","Nishka","Gunjan","Sumit"),
```

```
    class = c("MBA","MCA","MBA","IMCA","MCA"),
    roll_no=c(20,45,78,12,50)
)
print(stud.data)
```

#Operations on DataFrame:-

1)Extracting specific columns from a data frame

```
final <- data.frame(stud.data$student_id,stud.data$class)
print(final)
```

2)Modification of DataFrame:

#Adding row in the data frame

```
x <- list(6,"Vaishali","IMCA",15)
rbind(stud.data,x)
```

#Adding column in the data frame

```
y <- c("Moradabad","Lucknow","Etah","Sambhal","Khurja")
cbind(stud.data,city=y)
```

#Delete rows from data frame

```
stud.data<-stud.data[-1,]
print(stud.data)
```

#Delete column from the data frame

```
stud.data$roll_no<-NULL
print(stud.data)
```

OUTPUT:-

```
#Creation Of DataFrame
> stud.data<- data.frame(
+     student_id = c (1:5),
+     student_name = c("Shubham","Arpita","Nishka","Gunjan","Sumit"),
+     class = c("MBA","MCA","MBA","IMCA","MCA"),
+     roll_no=c(20,45,78,12,50)
+ )
> print(stud.data)
  student_id student_name class roll_no
1          1     Shubham   MBA      20
2          2      Arpita   MCA      45
3          3      Nishka   MBA      78
4          4      Gunjan  IMCA      12
5          5       Sumit   MCA      50
> #operations on DataFrame:-
> #Extracting specific columns from a data frame
> final <- data.frame(stud.data$student_id,stud.data$class)
> print(final)
  stud.data.student_id stud.data.class
1                    1             MBA
2                    2             MCA
3                    3             MBA
4                    4            IMCA
```

```
5                5          MCA
> #Modification of DataFrame
> #Adding row in the data frame
> x <- list(6,"Vaishali","IMCA",15)
> rbind(stud.data,x)
  student_id student_name class roll_no
1          1     Shubham   MBA      20
2          2      Arpita   MCA      45
3          3      Nishka   MBA      78
4          4      Gunjan  IMCA      12
5          5       Sumit   MCA      50
6          6    Vaishali  IMCA      15
> #Adding column in the data frame
> y <- c("Moradabad","Lucknow","Etah","Sambhal","Khurja")
> cbind(stud.data,city=y)
  student_id student_name class roll_no      city
1          1     Shubham   MBA      20 Moradabad
2          2      Arpita   MCA      45   Lucknow
3          3      Nishka   MBA      78      Etah
4          4      Gunjan  IMCA      12   Sambhal
5          5       Sumit   MCA      50    Khurja
> #Delete rows from data frame
> stud.data<-stud.data[-1,]
> print(stud.data)
  student_id student_name class roll_no
2          2      Arpita   MCA      45
3          3      Nishka   MBA      78
4          4      Gunjan  IMCA      12
5          5       Sumit   MCA      50
> #Delete column from the data frame
> stud.data$roll_no<-NULL
> print(stud.data)
  student_id student_name class
2          2      Arpita   MCA
3          3      Nishka   MBA
4          4      Gunjan  IMCA
5          5       Sumit   MCA
>
```

#creation of List:-

```
list_1<-list("Shubham","Arpita","Vaishali")
list_1
list_data<-list("Shubham","Arpita",c(1,2,3,4,5),TRUE,FALSE,22.5,12L)
print(list_data)
```

#Operation on lists:-

1)Giving name to list:-

```
list_data <- list(c("Shubham","Nishka","Gunjan"), matrix(c(40,80,60,70,90,80), nrow = 2),
   list("BCA","MCA","B.tech"))
names(list_data) <- c("Students", "Marks", "Course")
```

list_data

2)Accessing elements using index:-

```
print(list_data[1])
```

3)Accessing elements using names:-

```
print(list_data["Students"])
print(list_data$Marks)
```

4)Merging Lists:-

```
Even_list <- list(2,4,6)
Odd_list <- list(1,3,5)
# Merging the two lists.
merged.list <- list(Even_list,Odd_list)
print(merged.list)
```

OUTPUT: -

```
#creation of List:-
> list_1<-list("Shubham","Arpita","Vaishali")
> list_1
[[1]]
[1] "Shubham"

[[2]]
[1] "Arpita"

[[3]]
[1] "Vaishali"

> list_data<-list("Shubham","Arpita",c(1,2,3,4,5),TRUE,FALSE,22.5,12L)
> print(list_data)
[[1]]
[1] "Shubham"

[[2]]
[1] "Arpita"

[[3]]
[1] 1 2 3 4 5

[[4]]
[1] TRUE

[[5]]
[1] FALSE

[[6]]
[1] 22.5

[[7]]
[1] 12

> #Operation on lists:-
```

```
> #1)Giving name to list:-
> list_data <- list(c("Shubham","Nishka","Gunjan"), matrix(c(40,80,60,70,90,80), nrow = 2),
+     list("BCA","MCA","B.tech"))
> names(list_data) <- c("Students", "Marks", "Course")
> list_data
$Students
[1] "Shubham" "Nishka"  "Gunjan"

$Marks
     [,1] [,2] [,3]
[1,]   40   60   90
[2,]   80   70   80

$Course
$Course[[1]]
[1] "BCA"

$Course[[2]]
[1] "MCA"

$Course[[3]]
[1] "B.tech"


> #2)Accessing elements using index:-
> print(list_data[1])
$Students
[1] "Shubham" "Nishka"  "Gunjan"

> #3)Accessing elements using names:-
> print(list_data["Students"])
$Students
[1] "Shubham" "Nishka"  "Gunjan"

> print(list_data$Marks)
     [,1] [,2] [,3]
[1,]   40   60   90
[2,]   80   70   80
> #4)Merging Lists:-
> Even_list <- list(2,4,6)
> Odd_list <- list(1,3,5)
> # Merging the two lists.
> merged.list <- list(Even_list,Odd_list)
> print(merged.list)
[[1]]
[[1]][[1]]
[1] 2

[[1]][[2]]
[1] 4

[[1]][[3]]
[1] 6
```

```
[[2]]
[[2]][[1]]
[1] 1

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

[[2]][[3]]
[1] 5
```

Experiment No: 3
Experiment Name: Write program to demonstrate Loops & Vectorization of Missing Values.
Name:
Roll No:-

#using for loop

```
week<-c('Sunday',
        'Monday',
        'Tuesday',
        'Wednesday',
        'Thursday',
        'Friday',
        'saturday')
for(day in week)
{
 print(day)
}
```

#using while loop

```
val=1
while(val<=5)
{
 print(val)
 val=val+1
}
```

#using repeat loop

```
val=1
repeat
{
 print(val)
```

```
  val=val+1
  if(val>5)
  {
   break
  }
}
```

OUTPUT :

For-loop

```
[1] "Sunday"
[1] "Monday"
[1] "Tuesday"
[1] "Wednesday"
[1] "Thursday"
[1] "Friday"
[1] "saturday"
```

While-loop

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

Repeat-loop

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

Experiment No: 4
Experiment Name: Demonstrate Importing and exporting data.
Name:
Roll No:-

#IMPORT

```
getwd()
#Importing csv file.

path<-"C:/Users/Leena/OneDrive/Documents/candidate-elimination.csv"
content<-read.csv(path)
print(content)

#Importing Text file.

x<-read.table("C:/Users/Leena/OneDrive/Documents/file.txt",header=FALSE)
print(x)

#Importing CSV file using csv2.

x<-read.csv2("C:/Users/Leena/OneDrive/Documents/candidate-elimination.csv")
print(x)
```

OUTPUT

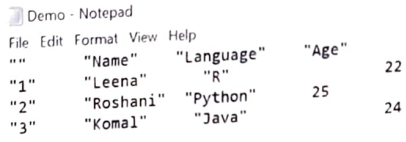
```
  SKY TEMP  HUMID   WIND WATER FOREST OUTPUT
1 sunny warm normal strong  warm   same    yes
2 sunny warm   high strong  warm   same    yes
3 rainy cold   high strong  warm change     no
4 sunny warm   high strong  cool change     yes

SKY.TEMP.HUMID.WIND.WATER.FOREST.OUTPUT
1  sunny,warm,normal,strong,warm,same,yes
2    sunny,warm,high,strong,warm,same,yes
3   rainy,cold,high,strong,warm,change,no
4  sunny,warm,high,strong,cool,change,yes


 SKY.TEMP.HUMID.WIND.WATER.FOREST.OUTPUT

1  sunny,warm,normal,strong,warm,same,yes
2    sunny,warm,high,strong,warm,same,yes
3   rainy,cold,high,strong,warm,change,no
4  sunny,warm,high,strong,cool,change,yes
```

 # EXPORT

1).Export a data frame to a text file using write.table().

```
df=data.frame(
  "Name"=c("Leena","Roshani","Komal"),
  "Language"=c("R","Python","Java"),
  "Age"=c(22,25,24)
)
write.table(df,
            file="Demo.txt",
            sep = "\t",
            row.names = TRUE,
            col.names = NA)
```

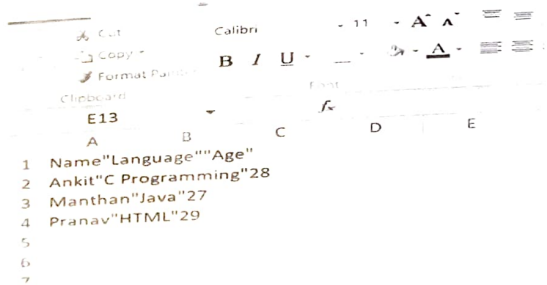
OUTPUT :



2).**Exporting Data to a csv file.**

```
df=data.frame(
  "Name"=c("Ankit","Manthan","Pranav"),
  "Language"=c("C Programming","Java","HTML"),
  "Age"=c(28,27,29)
)
write.table(df,
            file="myFile.csv",
            sep = "\t",
            row.names = FALSE)
```

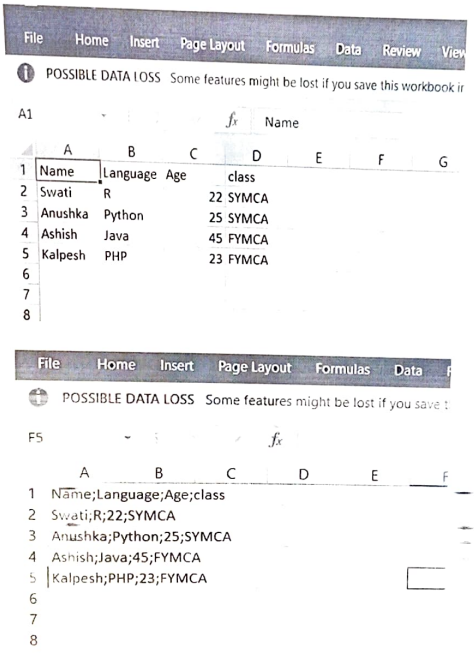
OUTPUT :



3)Exporting data to a csv2 file

```
library(readr)
df2=data.frame(
 "Name"=c("Swati","Anushka","Ashish","Kalpesh"),
 "Language"=c("R","Python","Java","PHP"),
 "Age"=c(22,25,45,23),
 "class"=c("SYMCA","SYMCA","FYMCA","FYMCA")
)
write_csv(df2,path="Demo2.csv")
write_csv2(df2,path="Demo3.csv")
```

OUTPUT



4)Exporting data using write_tsv()function

```
getwd()
#library(readr)
df2=data.frame(
  "Name"=c("Swati","Anushka","Ashish","Kalpesh"),
  "Language"=c("R","Python","Java","PHP"),
  "Age"=c(22,25,45,23),
  "class"=c("SYMCA","SYMCA","FYMCA","FYMCA")
)
write_tsv(df2,path="pract5.txt")
```

OUTPUT

*pract5.txt - Notepad

File Edit Format View Help

Name	Language	Age	class
Swati	R	22	SYMCA
Anushka	Python	25	SYMCA
Ashish	Java	45	FYMCA
Kalpesh	PHP	23	FYMCA

Experiment No: 5
Experiment Name: Write program for validating and exploring data manipulation (Summarizing, Sorting, Subsetting, Merging, Joining).
Name:
Roll No:-

1)Summarizing:-

```
#create a data frame
data1<-data.frame(player=c('A','B','c','D','E'),
                  runs=c(100,200,105,50,90),
                  wickets=c(15,20,8,5,8)
                  )
data1
#summarize method
summarize(data1,sum(runs),mean(runs),mode(wickets))
```

```
#summarize(data1)
```

---

2)Sorting:-

```
#creating data frame
dataBook=data.frame(Customers=c("Ruhi","James","Heera","Shubham","Joe","Priya"),
                    Products=c("ProdA","ProdB","ProdC","ProdD","ProdE","prodF"),
                    Salary=c(500,600,450,700,300,400))
dataBook
#sorting the data frame in ascending order
arrange(dataBook,Salary)
#sorting the data frame in descending order
dataBook%>%arrange(desc(Salary))
```

---

3)Subsetting:-

```
#Subsetting in R using []operator:
#create vector
x<-1:15
cat("Original vector:",x,"\n")
#subsetting vector:
cat("First 5 values of vector:",x[1:5],"\n")
cat("Without values present at index 1,2and 3",x[-c(1,2,3),"\n"])
#Subsetting in R using [[]]operator:
#create list:
ls<-list(a=1,b=2,c=10,d=20)
cat("Original List:\n")
print(ls)
#select first element of list:
cat("Element of list:",ls[[3]],"\n")
#Subsetting using c() function:
ls2<-list(a=list(x=1,y="students"),b=1:10)
ls2
```

```
cat("Using c() function:\n")
//print(ls2[[c(1,2)]])
//print(ls2[[1]][[2]])
#Subsetting Using $ operator:
ls3<-list(a="Roshani",b=1,c="Hello")
ls3
cat("Using $ operator:\n")
print(ls3$a)
```

---

4)Merging: -

```
#Merge DataFrames by Row Names:-
data_frame1<-data.frame(No=c(1:5),
                        Name=letters[1:5],
                        Salary=c(200,200,300,NA,300)
                        )
data_frame1

data_frame2<-data.frame(No=c(6:8),
                        Name=letters[8:10],
                        Salary=c(400,350,NA)
                        )
data_frame2

data_frame_merge<-merge(data_frame1,data_frame2,by='row.names',all=TRUE)
print("Merged DataFrame")
print(data_frame_merge)
```

---

5)Joining:-

```
#Using Inner join:-
data1<-data.frame(ID=c(1:5))
data2<-data.frame(ID=c(4:8))
inner_join(data1,data2,by="ID")
```

```
#Using Left join:-
data1<-data.frame(ID=c(1:5),
                  Name=c("Rutuja","Lokesh","Ram","Purvi","Nita"))
data2<-data.frame(ID=c(4:8),
                  Marks=c(70,85,80,90,75))
left_join(data1,data2,by="ID")
```

OUTPUT: -

```
#1)Summarizing:-
> #create a data frame
> data1<-data.frame(player=c('A','B','c','D','E'),
+                   runs=c(100,200,105,50,90),
+                   wickets=c(15,20,8,5,8)
+                   )
> data1
  player runs wickets
1      A  100      15
2      B  200      20
3      c  105       8
4      D   50       5
5      E   90       8
> #summarize method
> summarize(data1,sum(runs),mean(runs),mode(wickets))
  sum(runs) mean(runs) mode(wickets)
1       545       109       numeric
#----------------------------------------------------------
#2)Sorting:-
#creating data frame
ataBook=data.frame(Customers=c("Ruhi","James","Heera","Shubham","Joe","Priya"),
                   Products=c("ProdA","ProdB","ProdC","ProdD","ProdE","prodF"),
```

```
                    Salary=c(500,600,450,700,300,400))
> dataBook
  Customers Products Salary
1      Ruhi    ProdA    500
2     James    ProdB    600
3     Heera    ProdC    450
4   Shubham    ProdD    700
5       Joe    ProdE    300
6     Priya    prodF    400
> #sorting the data frame in ascending order
> arrange(dataBook,Salary)
  Customers Products Salary
1       Joe    ProdE    300
2     Priya    prodF    400
3     Heera    ProdC    450
4      Ruhi    ProdA    500
5     James    ProdB    600
6   Shubham    ProdD    700
> #sorting the data frame in descending order
> dataBook%>%arrange(desc(Salary))
  Customers Products Salary
1   Shubham    ProdD    700
2     James    ProdB    600
3      Ruhi    ProdA    500
4     Heera    ProdC    450
5     Priya    prodF    400
6       Joe    ProdE    300
> #-----------------------------------------------------------------
> #3)Subsetting:-
> #Subsetting in R using []operator:
> #create vector
> x<-1:15
```

```
                    Salary=c(500,600,450,700,300,400))
+
> dataBook
  Customers Products Salary
1      Ruhi    ProdA    500
2     James    ProdB    600
3     Heera    ProdC    450
4   Shubham    ProdD    700
5       Joe    ProdE    300
6     Priya    prodF    400
> #sorting the data frame in ascending order
> arrange(dataBook,Salary)
  Customers Products Salary
1       Joe    ProdE    300
2     Priya    prodF    400
3     Heera    ProdC    450
4      Ruhi    ProdA    500
5     James    ProdB    600
6   Shubham    ProdD    700
> #sorting the data frame in descending order
> dataBook%>%arrange(desc(Salary))
  Customers Products Salary
1   Shubham    ProdD    700
2     James    ProdB    600
3      Ruhi    ProdA    500
4     Heera    ProdC    450
5     Priya    prodF    400
6       Joe    ProdE    300
> #-------------------------------------------------------------------------
> #3)Subsetting:-
> #Subsetting in R using []operator:
> #create vector
> x<-1:15
```