## PRACTICAL NO:1

AIM: Write a program to find sum,mean and product of a vector ignore NA or NULL.

CODE:

```
> x<-c(10,11,12,13,14)

> print(x)

[1] 10 11 12 13 14

> sum(x)

[1] 60

> mean(x)

[1] 12

> prod(x)

[1] 240240

> x<-c(1,2,NA,5,6,NA)

> print(x)

[1]  1  2 NA  5  6 NA

> sum(x)

[1] NA

> sum(x,na.rm=TRUE)

[1] 14

> mean(x,na.rm=TRUE)

[1] 3.5

> prod(x,na.rm=TRUE)

[1] 60

>
```

## PRACTICAL NO:2

AIM:Write a program to create a vector using colon(:) operator and sequence () functions.

CODE:

```
> x<-c(1:15)

> print(x)
```

```
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15

> y=seq(1,5,by=0.5)

> print(y)

[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0

> z=seq(1,5,length.out=5)

> print(z)

[1] 1 2 3 4 5

> a=seq(20,5,from=6)

> print(a)

[1]  6 11 16

> b=seq(1,3,to=15)

> print(b)

[1]  1  4  7 10 13

> c=seq(from=5,to=13)

> print(c)

[1]  5  6  7  8  9 10 11 12 13
```

## PRACTICAL NO:5

AIM: Write a R program to create an ordered factor from data consisting of names of months.

CODE:

```
> months=c("March","April","January","September","October","April","December","July")

> months

[1] "March"    "April"    "January"  "September" "October"  "April"    "December" "July"

> #CREATING A FACTOR

> f=factor(months)

> f

[1] March     April     January   September October   April     December  July

Levels: April December January July March October September

> #CREATING ORDERED FACTOR
```

```
>
nf=factor(f,levels=c("January","March","April","July","September","October","November","Decem
ber"))

> nf

[1] March     April     January  September October  April     December July

Levels: January March April July September October November December

> table(nf)

nf

 January    March    April    July September  October November December

     1         1        2        1        1        1        0        1

>
```

## PRACTICAL NO:6

AIM: Write a program to create a data frame from four vectors.

CODE:

```
>df1=data.frame(emp_id=c(1,2,3,4,5),emp_name=c("Akash","Vikas","Raj","Alok","Ajay"),emp_lo
cation=c("Andheri","Dadar","Virar","Sion","Vasai"),emp_contact=c(12345,21333,98765,34567,5
6784))

> print(df1)
```

| | emp_id | emp_name | emp_location | emp_contact |
|---|---|---|---|---|
| 1 | 1 | Akash | Andheri | 12345 |
| 2 | 2 | Vikas | Dadar | 21333 |
| 3 | 3 | Raj | Virar | 98765 |
| 4 | 4 | Alok | Sion | 34567 |
| 5 | 5 | Ajay | Vasai | 56784 |

## PRACTICAL NO:7

AIM: Write a program to extract 3rd and 5th row with 1st and 3rd column from a given data frame.

CODE:

```
>df1=data.frame(emp_id=c(1,2,3,4,5),emp_name=c("Akash","Vikas","Raj","Alok","Ajay"),emp_lo
cation=c("Andheri","Dadar","Virar","Sion","Vasai"),emp_contact=c(12345,21333,98765,34567,5
6784))
```

> print(df1)

| | emp_id | emp_name | emp_location | emp_contact |
|---|---|---|---|---|
| 1 | 1 | Akash | Andheri | 12345 |
| 2 | 2 | Vikas | Dadar | 21333 |
| 3 | 3 | Raj | Virar | 98765 |
| 4 | 4 | Alok | Sion | 34567 |
| 5 | 5 | Ajay | Vasai | 56784 |

> new=df1[c(3,5),c(1,3)]

> print(new)

| | emp_id | emp_location |
|---|---|---|
| 3 | 3 | Virar |
| 5 | 5 | Vasai |

>

## PRACTICAL NO:8

AIM: Write a R program to drop row/column by number from a given data frame.

CODE:

### DROP A COLUMN

>
```
df1=data.frame(emp_id=c(1,2,3,4,5),emp_name=c("Akash","Vikas","Raj","Alok","Ajay"),emp_loc
ation=c("Andheri","Dadar","Virar","Sion","Vasai"),emp_contact=c(12345,21333,98765,34567,56
784))
```

> print(df1)

| | emp_id | emp_name | emp_location | emp_contact |
|---|---|---|---|---|
| 1 | 1 | Akash | Andheri | 12345 |
| 2 | 2 | Vikas | Dadar | 21333 |
| 3 | 3 | Raj | Virar | 98765 |
| 4 | 4 | Alok | Sion | 34567 |
| 5 | 5 | Ajay | Vasai | 56784 |

```
> cat("Drop a column")
```

Drop a column>

```
> df1[2]<-NULL
```

```
> print(df1)
```

|   | emp_id | emp_location | emp_contact |
|---|--------|--------------|-------------|
| 1 | 1      | Andheri      | 12345       |
| 2 | 2      | Dadar        | 21333       |
| 3 | 3      | Virar        | 98765       |
| 4 | 4      | Sion         | 34567       |
| 5 | 5      | Vasai        | 56784       |

>

## DROP A ROW

```
> df1[-3,]
```

|   | emp_id | emp_location | emp_contact |
|---|--------|--------------|-------------|
| 1 | 1      | Andheri      | 12345       |
| 2 | 2      | Dadar        | 21333       |
| 4 | 4      | Sion         | 34567       |
| 5 | 5      | Vasai        | 56784       |

>

## PRACTICAL NO:9

AIM:Write a program to create inner,outer,left,right join image from given two data frame.

CODE:

```
>
df1=data.frame(Student_Id=c(1:6),languages=c("Java","C","C++","Python","Ruby","JavaScript")
)
```

```
> df2=data.frame(Student_Id=c(2,1,4,3,7,9),Career=c("System Programmer","Backend
Developer","Data Scientist","Game Developer","Network Engineer","App Developer"))
```

```
> df1
```

|   | Student_Id | languages |
|---|---|---|
| 1 | 1 | Java |
| 2 | 2 | C |
| 3 | 3 | C++ |
| 4 | 4 | Python |
| 5 | 5 | Ruby |
| 6 | 6 | JavaScript |

```
> df2
```

|   | Student_Id | Career |
|---|---|---|
| 1 | 2 | System Programmer |
| 2 | 1 | Backend Developer |
| 3 | 4 | Data Scientist |
| 4 | 3 | Game Developer |
| 5 | 7 | Network Engineer |
| 6 | 9 | App Developer |

```
> #INNER JOIN
> innerJoin=merge(df1,df2,all=FALSE)
> innerJoin
```

|   | Student_Id | languages | Career |
|---|---|---|---|
| 1 | 1 | Java | Backend Developer |
| 2 | 2 | C | System Programmer |
| 3 | 3 | C++ | Game Developer |
| 4 | 4 | Python | Data Scientist |

```
> #OUTER JOIN
> outerJoin=merge(df1,df2,all=TRUE)
> outerJoin
```

|   | Student_Id | languages | Career |
|---|---|---|---|
| 1 | 1 | Java | Backend Developer |
| 2 | 2 | C | System Programmer |
| 3 | 3 | C++ | Game Developer |

| | | | |
|---|---|---|---|
| 4 | 4 | Python | Data Scientist |
| 5 | 5 | Ruby | <NA> |
| 6 | 6 | JavaScript | <NA> |
| 7 | 7 | <NA> | Network Engineer |
| 8 | 9 | <NA> | App Developer |

> #LEFT JOIN

> leftJoin=merge(df1,df2,all.x=T)

> leftJoin

| | Student_Id | languages | Career |
|---|---|---|---|
| 1 | 1 | Java | Backend Developer |
| 2 | 2 | C | System Programmer |
| 3 | 3 | C++ | Game Developer |
| 4 | 4 | Python | Data Scientist |
| 5 | 5 | Ruby | <NA> |
| 6 | 6 | JavaScript | <NA> |

> #RIGHT JOIN

> RightJoin=merge(df1,df2,all.d2=TRUE)

> RightJoin

| | Student_Id | languages | Career |
|---|---|---|---|
| 1 | 1 | Java | Backend Developer |
| 2 | 2 | C | System Programmer |
| 3 | 3 | C++ | Game Developer |
| 4 | 4 | Python | Data Scientist |

>

## PRACTICAL NO:10

AIM: Write a R program to create a data frame using two given vectors and display the duplicated elements and unique rows of a data frame.

CODE:

## DUPLICATED ELEMENTS

```
>
df1=data.frame(emp_id=c(1,2,3,4,5),emp_name=c("Akash","Vikas","Raj","Alok","Ajay"),emp_loc
ation=c("Andheri","Dadar","Virar","Sion","Vasai"),emp_contact=c(12345,21333,98765,34567,56
784))

> print(df1)
```

| | emp_id | emp_name | emp_location | emp_contact |
|---|---|---|---|---|
| 1 | 1 | Akash | Andheri | 12345 |
| 2 | 2 | Vikas | Dadar | 21333 |
| 3 | 3 | Raj | Virar | 98765 |
| 4 | 4 | Alok | Sion | 34567 |
| 5 | 5 | Ajay | Vasai | 56784 |

```
> print(duplicated (df1))
```

[1] FALSE FALSE FALSE FALSE FALSE


## UNIQUE ELEMENTS

```
>
df1=data.frame(emp_id=c(1,2,3,4,5),emp_name=c("Akash","Vikas","Raj","Alok","Ajay"),emp_loc
ation=c("Andheri","Dadar","Virar","Sion","Vasai"),emp_contact=c(12345,21333,98765,34567,56
784))

> print(df1)
```

| | emp_id | emp_name | emp_location | emp_contact |
|---|---|---|---|---|
| 1 | 1 | Akash | Andheri | 12345 |
| 2 | 2 | Vikas | Dadar | 21333 |
| 3 | 3 | Raj | Virar | 98765 |
| 4 | 4 | Alok | Sion | 34567 |
| 5 | 5 | Ajay | Vasai | 56784 |

```
> print(unique(df1))
```

| | emp_id | emp_name | emp_location | emp_contact |
|---|---|---|---|---|
| 1 | 1 | Akash | Andheri | 12345 |
| 2 | 2 | Vikas | Dadar | 21333 |
| 3 | 3 | Raj | Virar | 98765 |

| 4 | 4 | Alok | Sion | 34567 |
| 5 | 5 | Ajay | Vasai | 56784 |

>

PRACTICAL NO:11

AIM: Implement matrices addition,subtraction,multiplication and division.

CODE:

**Addition of Matrix**

```
 > M1=matrix(c(15,76,23,45,78,98,34,54,67),nrow=3)
> print(M1)
     [,1] [,2] [,3]
[1,]   15   45   34
[2,]   76   78   54
[3,]   23   98   67
> M2=matrix(c(56,90,98,12,43,69,92,47,47),nrow=3)
> print(M2)
     [,1] [,2] [,3]
[1,]   56   12   92
[2,]   90   43   47
[3,]   98   69   47
> result=M1+M2
> cat("Addition of Matrix")
Addition of Matrix>
> print(result)
     [,1] [,2] [,3]
[1,]   71   57  126
[2,]  166  121  101
[3,]  121  167  114
>
```

**Subtraction of Matrix**

```
> M1=matrix(c(15,76,23,45,78,98,34,54,67),nrow=3)
> print(M1)
     [,1] [,2] [,3]
[1,]   15   45   34
[2,]   76   78   54
[3,]   23   98   67
> M2=matrix(c(56,90,98,12,43,69,92,47,47),nrow=3)
> print(M2)
     [,1] [,2] [,3]
[1,]   56   12   92
[2,]   90   43   47
[3,]   98   69   47
> cat("Subtraction of Matrix")
Subtraction of Matrix>
> result=M1-M2
> print(result)
     [,1] [,2] [,3]
[1,]  -41   33  -58
[2,]  -14   35    7
[3,]  -75   29   20
>
```

## Multiplication of Matrix

```
> M1=matrix(c(15,76,23,45,78,98,34,54,67),nrow=3)
> print(M1)
     [,1] [,2] [,3]
[1,]   15   45   34
[2,]   76   78   54
[3,]   23   98   67
> M2=matrix(c(56,90,98,12,43,69,92,47,47),nrow=3)
```

```
> print(M2)
     [,1] [,2] [,3]
[1,]   56   12   92
[2,]   90   43   47
[3,]   98   69   47
> cat("Multiplication of Matrix")
Multiplication of Matrix>
> result=M1*M2
> print(result)
     [,1] [,2] [,3]
[1,]  840  540 3128
[2,] 6840 3354 2538
[3,] 2254 6762 3149
>
```

**Division of Matrix**

```
> M1=matrix(c(15,76,23,45,78,98,34,54,67),nrow=3)
> print(M1)
     [,1] [,2] [,3]
[1,]   15   45   34
[2,]   76   78   54
[3,]   23   98   67
> M2=matrix(c(56,90,98,12,43,69,92,47,47),nrow=3)
> print(M2)
     [,1] [,2] [,3]
[1,]   56   12   92
[2,]   90   43   47
[3,]   98   69   47
> result=M1/M2
> print(result)
```

```
          [,1]      [,2]      [,3]
[1,] 0.2678571 3.750000 0.3695652
[2,] 0.8444444 1.813953 1.1489362
[3,] 0.2346939 1.420290 1.4255319
>
```