Vidya Pratishthan’s Arts Science and Commerce College, Baramati

Department of B.B.A.(Computer Applications)

S.Y.B.B.A.(C.A) Semester-III (CBCS 2019 Pattern)
Subject: Bigdata(Practical Slips)

Slip No-1

Write a R program to find the maximum and the minimum value of a given vector.

```
nums = c(10, 20, 30, 40, 50, 60)
print(‘Original vector:’)
print(nums)
print(paste("Maximum value of the said vector:",max(nums)))
print(paste("Minimum value of the said vector:",min(nums)))
```

```
nums = c(10, 20, 30, 40, 50, 60)
> print("Original vector:")
[1] "Original vector:"
> print(nums)
[1] 10 20 30 40 50 60
> print(paste("Maximum value of the said vector:",max(nums)))
[1] "Maximum value of the said vector: 60"
> print(paste("Minimum value of the said vector:",min(nums)))
[1] "Minimum value of the said vector: 10"
```

Slip No-2

Write a R program to sort a Vector in ascending and descending order.

```
x = c(10, 20, 30, 25, 9, 26)
print("Original Vectors:")
print(x)
print("Sort in ascending order:")
print(sort(x))
print("Sort in descending order:")
print(sort(x, decreasing=TRUE))
```

Output:

```
print("Original Vectors:")
[1] "Original Vectors:"
> print(x)
[1] 10 20 30 25  9 26
> print("Sort in ascending order:")
[1] "Sort in ascending order:"
> print(sort(x))
[1]  9 10 20 25 26 30
> print("Sort in descending order:")
[1] "Sort in descending order:"
> print(sort(x, decreasing=TRUE))
[1] 30 26 25 20 10  9
```

Slip No-3

Write a R program to compare two data frames to find the row(s) in first data frame that are not present in second data frame.

```
df_90 = data.frame(
  "item" = c("item1", "item2", "item3"),
  "Jan_sale" = c(12, 14, 12),
  "Feb_sale" = c(11, 12, 15),
  "Mar_sale" = c(12, 14, 15)
)
df_91 = data.frame(
  "item" = c("item1", "item2", "item3"),
  "Jan_sale" = c(12, 14, 12),
  "Feb_sale" = c(11, 12, 15),
  "Mar_sale" = c(12, 15, 18))
print("Original Dataframes:")
print(df_90)
print(df_91)
print("Row(s) in first data frame that are not present in second data frame:")
print(setdiff(df_90,df_91))
```

Output:-

```
[1] "Original Dataframes:"
   item Jan_sale Feb_sale Mar_sale
1 item1       12       11       12
2 item2       14       12       14
3 item3       12       15       15
   item Jan_sale Feb_sale Mar_sale
1 item1       12       11       12
2 item2       14       12       15
3 item3       12       15       18
[1] "Row(s) in first data frame that are not present in second data frame:"
  Mar_sale
1       12
2       14
3       15
```

Slip No-4

Write an R program to extract first 10 English letter in lower case and last 10 letters in upper case and extract letters between 22nd to 24th letters in upper case.

```
print("First 10 letters in lower case:")
t = head(letters, 10)
print(t)
print("Last 10 letters in upper case:")
t = tail(LETTERS, 10)
print(t)
print("Letters between 22nd to 24th letters in upper case:")
e = tail(LETTERS[22:24])
print(e)
```

Output:-

```
[1] "First 10 letters in lower case:"
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"
[1] "Last 10 letters in upper case:"
 [1] "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
[1] "Letters between 22nd to 24th letters in upper case:"
[1] "V" "W" "X"
```

Slip No-5

Write an R program to find Sum, Mean and Product of a Vector.

```
x = c(10, 20, 30)

print("Sum:")

print(sum(x))

print("Mean:")

print(mean(x))

print("Product:")

print(prod(x))
```

Output:-

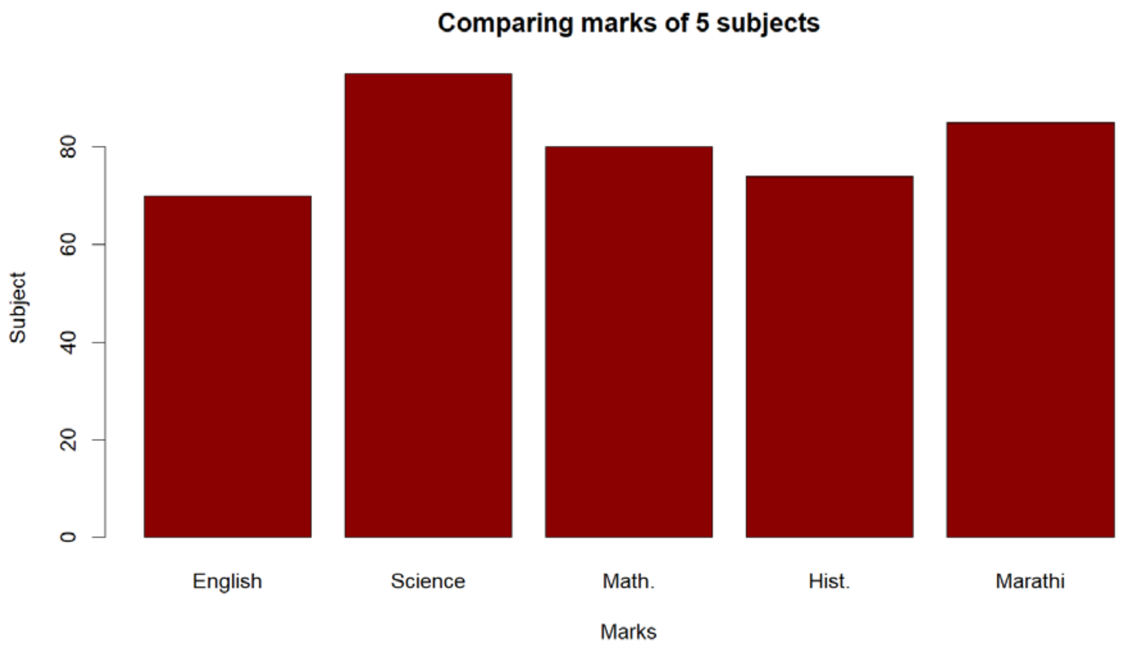
```
[1] "Sum:"
[1] 60
[1] "Mean:"
[1] 20
[1] "Product:"
[1] 6000
```

Slip No-6

Write an R program to create a simple bar plot of five subject’s marks

```
marks = c(70, 95, 80, 74,85)
barplot(marks,
main = "Comparing marks of 5 subjects",
xlab = "Marks",
ylab = "Subject",
names.arg = c("English", "Science", "Math.", "Hist.", "Marathi"),
col = "darkred",
horiz = FALSE)
```

Output:-



Slip No-7

Write an R program to create a Data frame which contain details of 5 employees and display the details in ascending order.

```
Employees = data.frame(Name=c("Abc","pqr","lmn", "xyz","ams"),
                       Gender=c("M","M","F","F","M"),
                       Age=c(23,22,25,26,21),
Designation=c("Clerk","Manager","Executive","CEO","ASSISTANT
"),
 SSN=c("123-34-2346","123-44-779","556-24-433","123-98-
987","679-77-576"))
print("Details of the employees:")
print(Employees)
print("sort the data in ascending order based on Name ")
print(Employees[order(Employees$Name, decreasing = FALSE),
]  )
```

Output:-

```
[1] "Details of the employees:"
  Name Gender Age Designation         SSN
1  Abc      M  23       Clerk 123-34-2346
2  pqr      M  22     Manager  123-44-779
3  lmn      F  25   Executive  556-24-433
4  xyz      F  26         CEO  123-98-987
5  ams      M  21   ASSISTANT  679-77-576


[1] "sort the data in ascending order based on Name "
  Name Gender Age Designation         SSN
1  Abc      M  23       Clerk 123-34-2346
5  ams      M  21   ASSISTANT  679-77-576
3  lmn      F  25   Executive  556-24-433
2  pqr      M  22     Manager  123-44-779
4  xyz      F  26         CEO  123-98-987
```

Slip No-8

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

```
a = c(10,20,10,10,40,50,20,30)
b = c(10,30,10,20,0,50,30,30)
print("Original data frame:")
ab = data.frame(a,b)
print(ab)
print("Duplicate elements of the said data frame:")
print(duplicated(ab))
print("Unique rows of the said data frame:")
print(unique(ab))
```

Output:-

```
[1] "Original data frame:"
   a  b
1 10 10
2 20 30
3 10 10
4 10 20
5 40  0
6 50 50
7 20 30
8 30 30
[1] "Duplicate elements of the said data frame:"
[1] FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE

[1] "Unique rows of the said data frame:"
   a  b
1 10 10
2 20 30
4 10 20
5 40  0
6 50 50
8 30 30
```

Slip No 9

Write a R program to change the first level of a factor with another level of a given factor.

```
v = c("a", "b", "a", "c", "b")
print("Original vector:")
print(v)
f = factor(v)
print("Factor of the said vector:")
print(f)
levels(f)[1] = "e"
print(f)
```

Output:-

```
[1] "Original vector:"
[1] "a" "b" "a" "c" "b"
[1] "Factor of the said vector:"
[1] a b a c b
Levels: a b c
[1] e b e c b
Levels: e b c
```

Slip No-10

Write a script in R to create a list of cities and perform the following

Q 1) Give names to the elements in the list.

```
list_data=list("Baramati","Phaltan","Indapur","Pune","Satara")
> print(list_data)
[[1]]
[1] "Baramati"
[[2]]
[1] "Phaltan"
[[3]]
[1] "Indapur"
[[4]]
[1] "Pune"
[[5]]
[1] "Satara"

> names(list_data)=c("first city","second city","third city","fourth city","fifth city")
> print(list_data)
$`first city`
[1] "Baramati"
$`second city`
[1] "Phaltan"
$`third city`
[1] "Indapur"
$`fourth city`
[1] "Pune"

$`fifth city`
[1] "Satara"
```

Q 2) Add an element at the end of the list.

```
> new_l=c("Solapur")
> m=append(list_data,new_l)
> print(m)
$`first city`
[1] "Baramati"
```

```
$`second city`
[1] "Phaltan"
$`third city`
[1] "Indapur"
$`fourth city`
[1] "Pune"
$`fifth city`
[1] "Satara"
[[6]]
[1] "Solapur"
```

Q 3) Remove the last element.

```
> h=head(m,-1)
> print(h)
$`first city`
[1] "Baramati"
$`second city`
[1] "Phaltan"
$`third city`
[1] "Indapur"
$`fourth city`
[1] "Pune"
$`fifth city`
[1] "Satara"
```

Q 4) Update the 3rd Element

```
> h[3]="Mumbai"
> print(h)
$`first city`
[1] "Baramati"
$`second city`
[1] "Phaltan"
$`third city`
[1] "Mumbai"
$`fourth city`
[1] "Pune"
$`fifth city`
[1] "Satara"
```

Slip No-11

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.

```
#create 2 vectors
> Vec1=c(5,6,7)
> Vec2=c(5,6,7,8,9,10,11,12)
> arr1=array(c(vec1,vec2),dim=c(3,3,1))
>
> #create another 2 vectors
> vec3=c(2,3,4)
> vec4=c(5,6,7,8,9,10,11,12)
> arr2=array(c(vec3,vec4),dim=c(3,3,2))
> #create matrix
> mat1=arr1[,,1]
> mat2=arr2[,,2]
>
> #addition
> add_res =mat1+mat2
> print(add_res)
     [,1] [,2] [,3]
[1,]   12   13   19
[2,]   15   15   21
[3,]    7   17   23
> #subtraction
> sub_res=mat1-mat2
> print(sub_res)
     [,1] [,2] [,3]
[1,]  -10    7    7
[2,]   -9    7    7
[3,]    3    7    7
```

Slip no-12

Write an R Program to calculate Multiplication Table

```
num = as.integer(readline(prompt = "Enter a number: "))
for(i in 1:10)
{
print(paste(num,'x', i, '=', num*i))
}
```

Output:-

```
Enter a number: 31
[1] "31 x 1 = 31"
[1] "31 x 2 = 62"
[1] "31 x 3 = 93"
[1] "31 x 4 = 124"
[1] "31 x 5 = 155"
[1] "31 x 6 = 186"
[1] "31 x 7 = 217"
[1] "31 x 8 = 248"
[1] "31 x 9 = 279"
[1] "31 x 10 = 310"
```

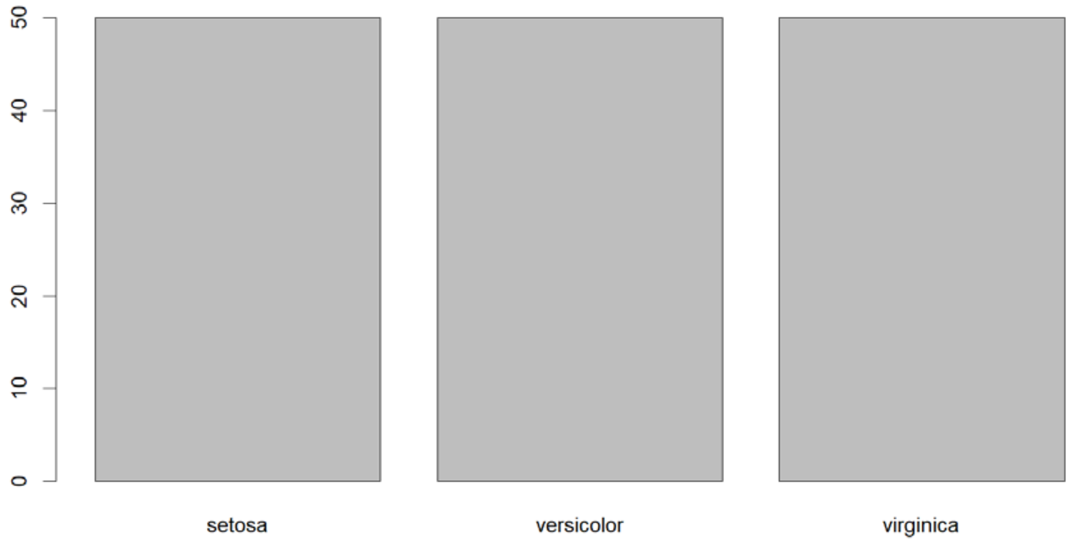
Slip No-13

#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 class value.

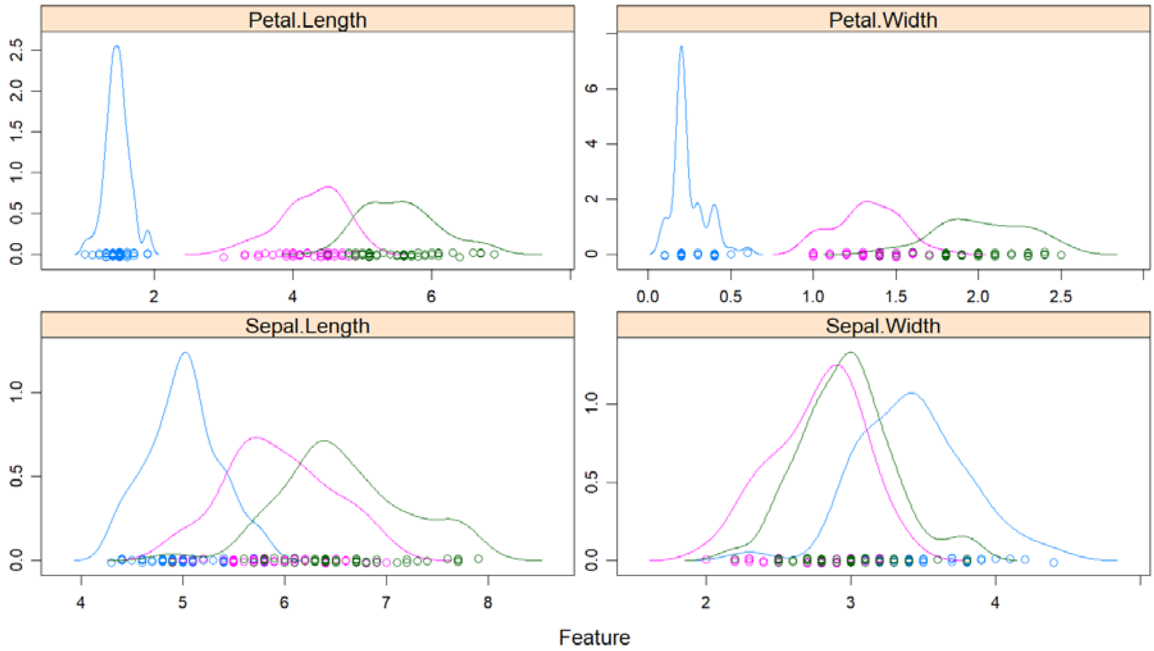
```
install.packages("caret")
library(caret)
data(iris)
dataset<-iris
x <- dataset[,1:4]
y <- dataset[,5]
plot(y)
scales <- list(x=list(relation="free"), y=list(relation="free"))
featurePlot(x=x, y=y, plot="density", scales=scales)
```

Output:-

2)


0
10
20
30
40
50
setosa
versicolor
virginica

3)


Petal.Length
Petal.Width
Sepal.Length
Sepal.Width
0.0
0.5
1.0
1.5
2.0
2.5
2
4
6
0
2
4
6
0.0
0.5
1.0
1.5
2.0
2.5
0.0
0.5
1.0
4
5
6
7
8
0.0
0.5
1.0
2
3
4
Feature

Slip No-14

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

```
f1 <- factor(sample(LETTERS, size=6, replace=TRUE))
f2 <- factor(sample(LETTERS, size=6, replace=TRUE))
print("Original factors:")
print(f1)
print(f2)
f = factor(c(levels(f1)[f1], levels(f2)[f2]))
print("After concatenate factor becomes:")
print(f)
sort(f)
```

Output:-

```
1] "Original factors:"
[1] E C L V Z F
Levels: C E F L V Z
[1] V M N J Z W
Levels: J M N V W Z
[1] "After concatenate factor becomes:"
 [1] E C L V Z F V M N J Z W
Levels: C E F J L M N V W Z
```

Slip No-15

```
L = sample(LETTERS,size=50,replace=TRUE)
print("Original data:")
print(L)
f = factor(L)
print("Original factors:")
print(f)
print("Only five of the levels")
print(table(L[1:5]))
```

Output:-

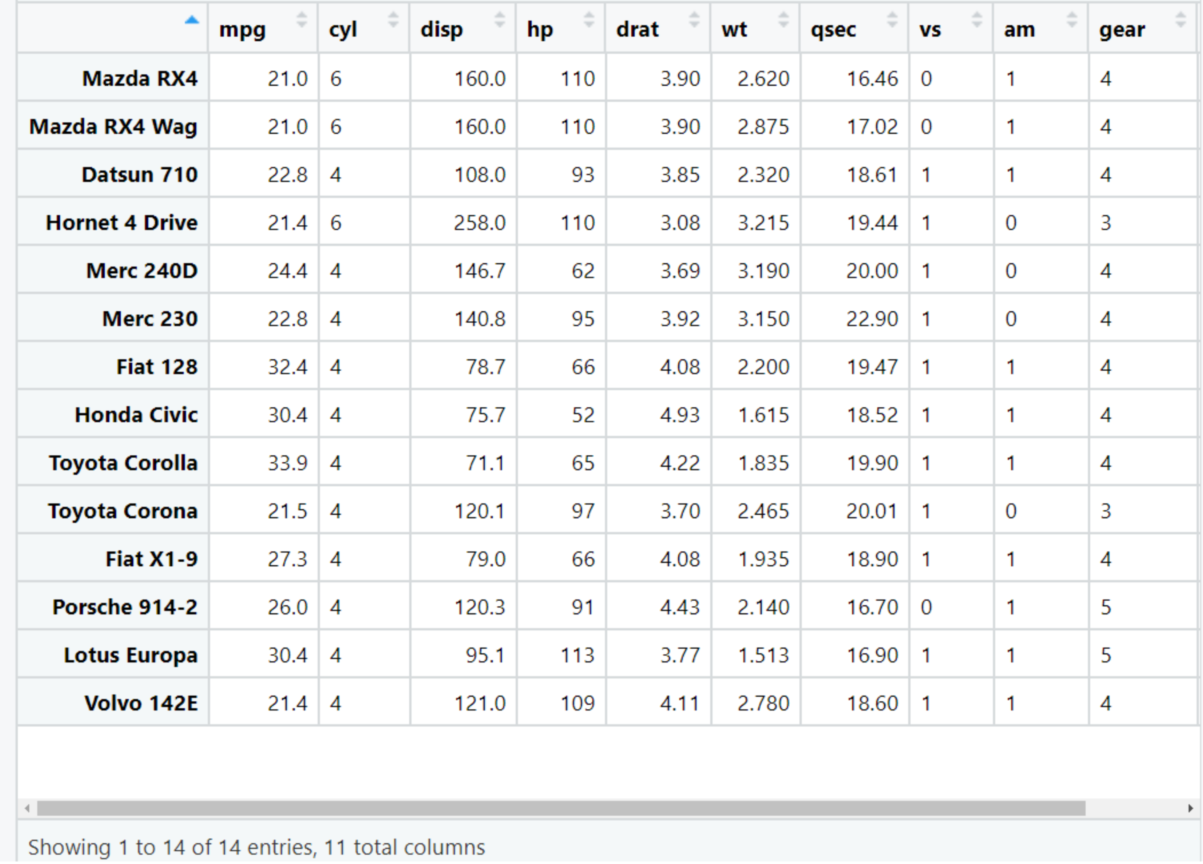
```
[1] "Original data:"
 [1] "Z" "G" "L" "P" "F" "Q" "H" "Y" "Z" "T" "E" "O" "Q" "K" "N"
[16] "M" "S" "Q" "V" "J" "J" "M" "Q" "L" "R" "J" "B" "E" "R" "O"
[31] "W" "R" "S" "D" "C" "L" "T" "K" "L" "J" "U" "J" "X" "A" "B"
[46] "P" "S" "X" "O" "F"
[1] "Original factors:"
 [1] Z G L P F Q H Y Z T E O Q K N M S Q V J J M Q L R J B E R
O W
[32] R S D C L T K L J U J X A B P S X O F
Levels: A B C D E F G H J K L M N O P Q R S T U V W X Y Z
[1] "Only five of the levels"

F G L P Z
1 1 1 1 1
```

Slip No-16

```
#Consider the inbuilt mtcar dataset
#i) Subset the vector, "mtcars[,1]", for values greater than "15.0".
subset(mtcars[,1], mtcars[,1] > 15.0)
[1] 21.0 21.0 22.8 21.4 18.7 18.1 24.4 22.8 19.2 17.8 16.4 17.3
[13] 15.2 32.4 30.4 33.9 21.5 15.5 15.2 19.2 27.3 26.0 30.4 15.8
[25] 19.7 21.4

#ii) Subset the rows of cars that get more than 20 miles per gallon
(mpg) of fuel efficiency
mtcars_mpg=filter(mtcars,mpg>20)
```



	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4

Showing 1 to 14 of 14 entries, 11 total columns

```
#iii) Subset the rows that get less than 16 miles per gallon (mpg) of
fuel efficiency and more than 100 horsepower(hp)
filter(mtcars,mpg<16 & hp>100)
mpg cyl  disp  hp drat    wt  qsec vs am gear
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3
Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3
```

```
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3
Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3
AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3
Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3
Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5
Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5
                    carb
Duster 360             4
Merc 450SLC            3
Cadillac Fleetwood     4
Lincoln Continental    4
Chrysler Imperial      4
Dodge Challenger       2
AMC Javelin            2
Camaro Z28             4
Ford Pantera L         4
Maserati Bora          8
```

Slip No:-17

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

```
convert_to_binary <- function(n) {
if(n > 1) {
convert_to_binary(as.integer(n/2))
}
cat(n %% 2)
}
```

Output:

```
convert_to_binary(15)
>1111
```

OR use this

```
db<-function(n){
+ if(n>1){
+ db(as.integer(n/2))
+ }
+ cat(n%%2)
+ }
> db(15)
1111
```

Slip No 18

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)
m<-cbind(a,b,c)
print("Content of the said matrix:")
print(m)
```

Output:-

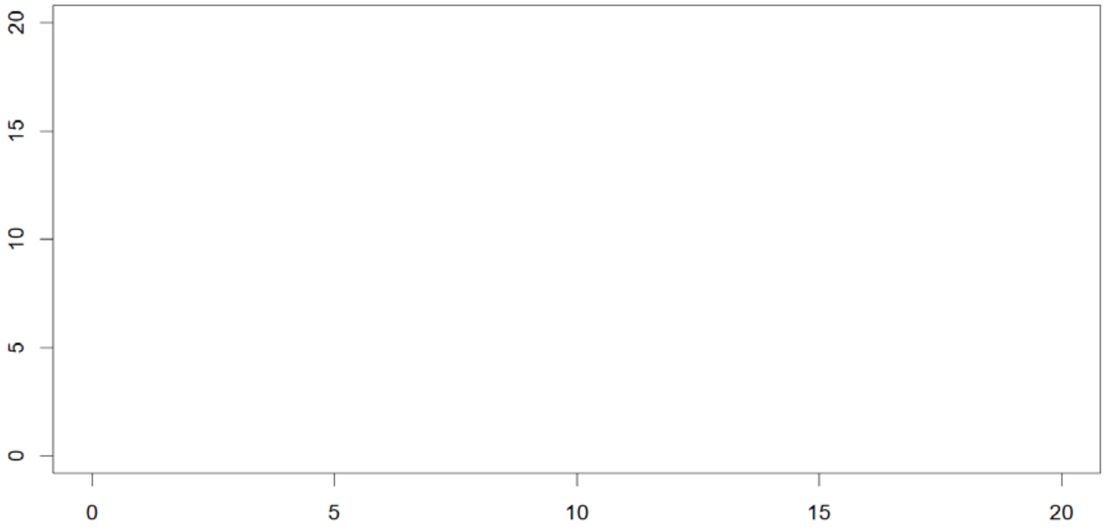
```
[1] "Content of the said matrix:"
> print(m)
     a b c
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9
```

Slip No 19

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

```
#print("Empty plot:")
plot.new()
#print("Empty plot specify the axes limits of the graphic:")
plot(1, type="n", xlab="", ylab="", xlim=c(0, 20), ylim=c(0, 20))
```

Output:-



SlipNo-20

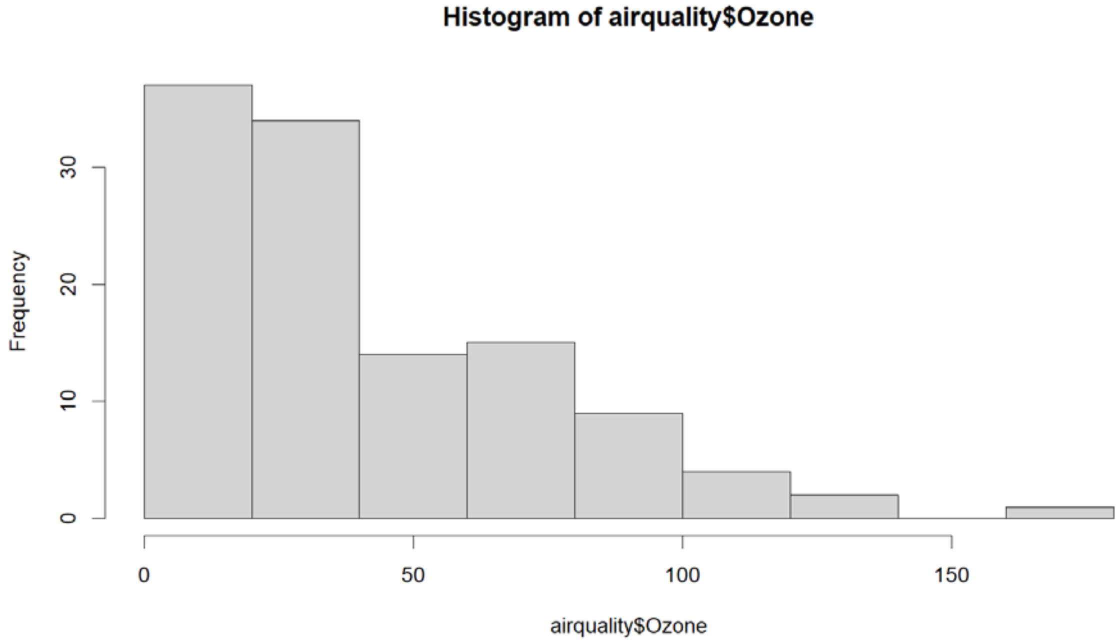
Consider the airquality Dataset

1)Display the details of 10th day

```
airquality[10,]
Ozone Solar.R Wind Temp Month Day
10    NA      194  8.6   69      5  10
```

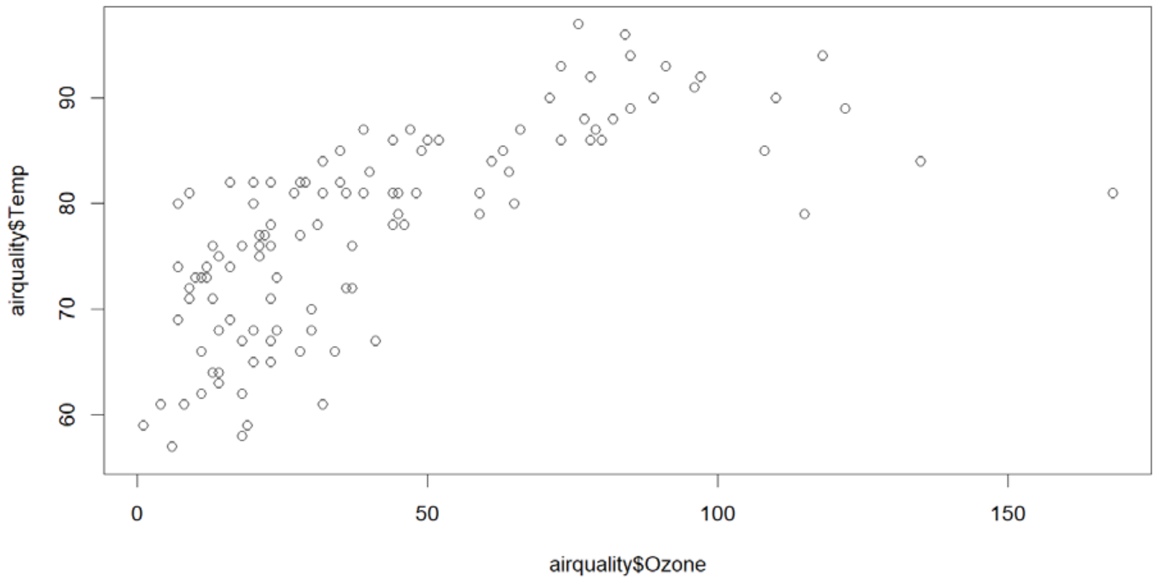
2)Draw histogram for airquality$ozone

```
hist(airquality$Ozone)
```



3)Make a scatterplot to compare ozone and temperature

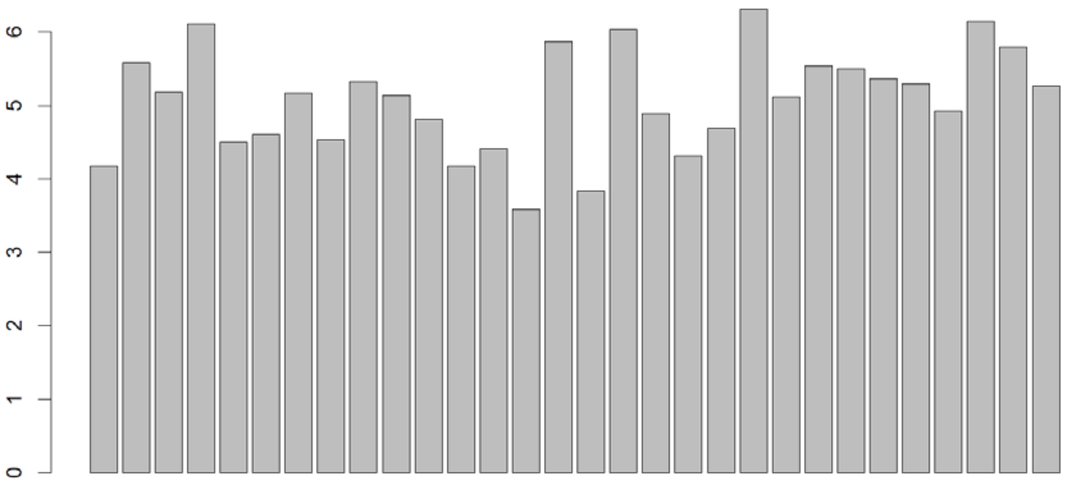
```
plot(airquality$Ozone,airquality$Temp)
```



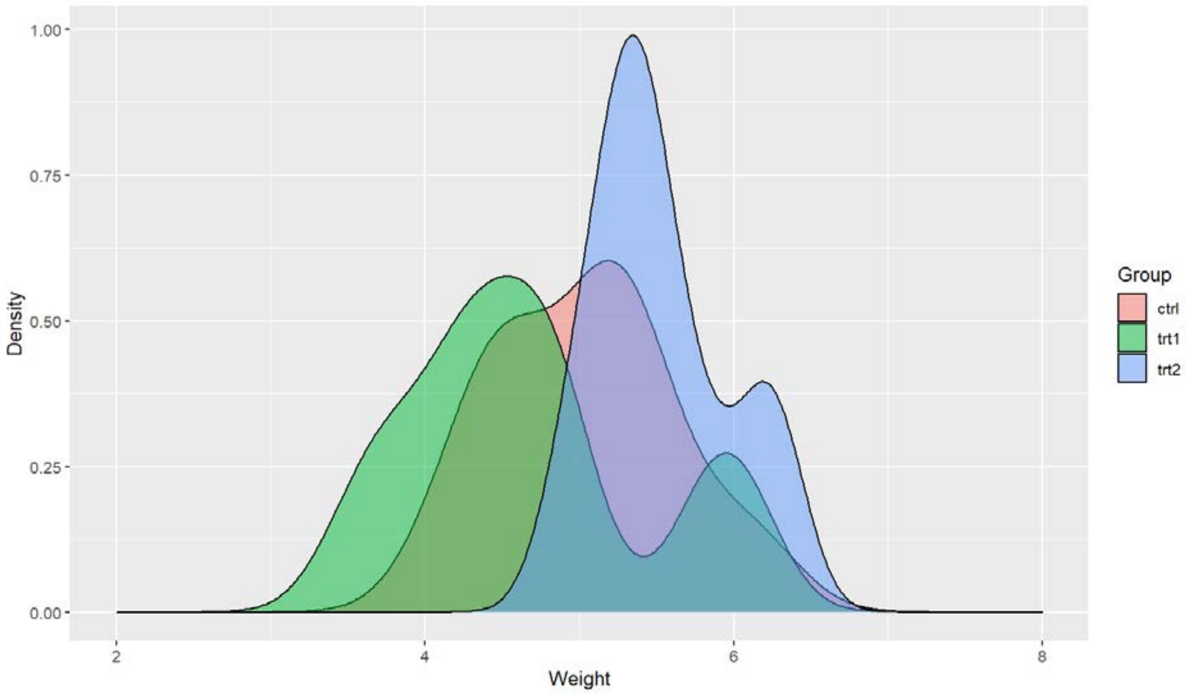
airquality$Temp
60
70
80
90
0
50
100
150
airquality$Ozone

Slip No 21

```
#Consider the plantGrowth inbuilt dataset
#i) Create a variable "y" and attach to it the output attribute of the "plantGrowth" dataset.
data("PlantGrowth")
dataset<-PlantGrowth
#ii) Create a barplot to breakdown your output attribute.
x <- dataset[,1:2]
y <- dataset[,1]
barplot(y)
```



```
#iii) Create a density plot matrix for each attribute by class value
ggplot(PlantGrowth, aes(PlantGrowth$weight)) +
geom_density(aes(data = PlantGrowth$weight, fill =
PlantGrowth$group), position = 'identity', alpha = 0.5) + labs(x =
'Weight', y = 'Density') + scale_fill_discrete(name = 'Group') +
scale_x_continuous(limits = c(2, 8))
```


1.00
0.75
0.50
0.25
0.00
Density
2
4
6
8
Weight
Group
ctrl
trt1
trt2

Slip No-22

Write an R program to print the numbers from 1 to 100 and print "SY" for multiples of 3, print "BBA" for multiples of 5, and print "SYBBA" for multiples of both.

```
for (n in 1:100)
{
if (n %% 3 == 0 & n %% 5 == 0)
{
print("SYBBA")
}
else if (n %% 3 == 0)
{
print("SY")
}
else if (n %% 5 == 0)
{
print("BBA")
}
else print(n)
}
```

Output:-

```
[1] 1
[1] 2
[1] "SY"
[1] 4
[1] "BBA"
[1] "SY"
[1] 7
[1] 8
[1] "SY"
[1] "BBA"
[1] 11
[1] "SY"
[1] 13
[1] 14
```

```
[1] "SYBBA"
[1] 16
[1] 17
[1] "SY"
[1] 19
[1] "BBA"
[1] "SY"
[1] 22
[1] 23
[1] "SY"
[1] "BBA"
[1] 26
[1] "SY"
[1] 28
[1] 29
[1] "SYBBA"
[1] 31
[1] 32
[1] "SY"
[1] 34
[1] "BBA"
[1] "SY"
[1] 37
[1] 38
[1] "SY"
[1] "BBA"
[1] 41
[1] "SY"
[1] 43
[1] 44
[1] "SYBBA"
[1] 46
[1] 47
[1] "SY"
[1] 49
[1] "BBA"
[1] "SY"
```

```
[1] 52
[1] 53
[1] "SY"
[1] "BBA"
[1] 56
[1] "SY"
[1] 58
[1] 59
[1] "SYBBA"
[1] 61
[1] 62
[1] "SY"
[1] 64
[1] "BBA"
[1] "SY"
[1] 67
[1] 68
[1] "SY"
[1] "BBA"
[1] 71
[1] "SY"
[1] 73
[1] 74
[1] "SYBBA"
[1] 76
[1] 77
[1] "SY"
[1] 79
[1] "BBA"
[1] "SY"
[1] 82
[1] 83
[1] "SY"
[1] "BBA"
[1] 86
[1] "SY"
[1] 88
```

```
[1] 89
[1] "SYBBA"
[1] 91
[1] 92
[1] "SY"
[1] 94
[1] "BBA"
[1] "SY"
[1] 97
[1] 98
[1] "SY"
[1] "BBA"
```

Slip No-23

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.

```
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)
print("The second row of the second matrix of the array:")
print(result[2,,2])
```

output:-

```
print("Two vectors of different lengths:")
[1] "Two vectors of different lengths:"
> v1 =  c(1,3,4,5)
> v2 =  c(10,11,12,13,14,15)
> print(v1)
[1] 1 3 4 5
> print(v2)
[1] 10 11 12 13 14 15
> result = array(c(v1,v2),dim = c(3,3,2))
> print("New array:")
[1] "New array:"
> print(result)
, , 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

> print("The second row of the second matrix of the array:")
[1] "The second row of the second matrix of the array:"
> print(result[2,,2])
[1]  1  5 12
```

Slip No-24

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.

```
 v1 = c(1,3,4,5)
 v2 = c(10,11,12,13,14,15)
 result = array(c(v1,v2),dim = c(3,3,1))
 print(result)
, , 1
     [,1] [,2] [,3]
[1,]    1    5   12
[2,]    3   10   13
[3,]    4   11   14
 v1 = c(1,3,4,5)
 v2 = c(10,11,12,13,14,15)
 result2 = array(c(v1,v2),dim = c(3,3,1))
 result3 = result*result2
 print(result3)
, , 1
     [,1] [,2] [,3]
[1,]    1   25  144
[2,]    9  100  169
[3,]   16  121  196
```

Slip No-25

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

```
l = list(
c(1, 2, 2, 5, 7, 12),
month.abb,
matrix(c(3, -8, 1, -3), nrow = 2),
asin
)
print("Content of the list:")
print(l)
```

output:-

```
[1] "Content of the list:"
> print(l)
[[1]]
[1]  1  2  2  5  7 12

[[2]]
 [1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct"
[11] "Nov" "Dec"

[[3]]
     [,1] [,2]
[1,]    3    1
[2,]   -8   -3

[[4]]
function (x)  .Primitive("asin")
```

Slip No-26

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

```
a = array( 6:30,
dim = c(4, 3, 2),
dimnames = list( c("Col1", "Col2", "Col3", "Col4"),c("Row1",
"Row2", "Row3"),
c("Part1", "Part2"))
)
print(a)
```

output:-

```
, , Part1

     Row1 Row2 Row3
Col1    6   10   14
Col2    7   11   15
Col3    8   12   16
Col4    9   13   17

, , Part2

     Row1 Row2 Row3
Col1   18   22   26
Col2   19   23   27
Col3   20   24   28
Col4   21   25   29
```

Slip No-27

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

```
binary<-function(p_number) {
    bsum<-0
   bexp<-1
 while (p_number > 0) {
      digit<-p_number %% 2
      p_number<-floor(p_number / 2)
        bsum<-bsum + digit * bexp
         bexp<-bexp * 10
     }
     return(bsum)
 }

 p_number<-readline("Decimal number?: ")
Decimal number?: 45
 p_number<-as.numeric(p_number)
 bsum<-binary(p_number)
 cat("Binary: ", bsum)
Binary:  101101
```

Slip No-28

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

```
a=matrix(c(10,5,7,11,15,21))
print(a)
a=matrix(c(10,5,7,11,15,21),nrow = 3)
print(a)
l=as.list(a)
print(l)
u=unlist(l)
print(u)
m=list(u)
print(m)
print(lapply(m,sort))
```

```
a=matrix(c(10,5,7,11,15,21))
> print(a)
     [,1]
[1,]   10
[2,]    5
[3,]    7
[4,]   11
[5,]   15
[6,]   21
> a=matrix(c(10,5,7,11,15,21),nrow = 3)
> print(a)
     [,1] [,2]
[1,]   10   11
[2,]    5   15
[3,]    7   21
> l=as.list(a)
> print(l)
[[1]]
[1] 10

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

```
[[3]]
[1] 7

[[4]]
[1] 11

[[5]]
[1] 15

[[6]]
[1] 21

> u=unlist(l)
> print(u)
[1] 10  5  7 11 15 21
> m=list(u)
> print(m)
[[1]]
[1] 10  5  7 11 15 21

> print(lapply(m,sort))
[[1]]
[1]  5  7 10 11 15 21
```

Slip No-29

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.

```
print("1. Give names to the students in the list. ")
list_data <- list(c("21","22","23"),list("saurabh", "radha", "sayali"))
print(list_data)
names(list_data) = c("Rollno", "Student(s)")
print("List with column names:")
print(list_data)
print("2. Add a student at the end of the list. ")
list_data <- list(c("21","22","23"),list("saurabh", "radha", "sayali"))
 print(list_data)
names(list_data) = c("Rollno", "Student(s)")
print("Add a new element at the end of the list:")
 list_data[3] = "Sona"
print("New list:")
print(list_data)
print("3. Remove the firstStudent. ")
list_data <- list(c("21","22","23"),list("saurabh", "radha", "sayali"))
print(list_data)
names(list_data) = c("Rollno", "Student(s)")
print("Remove the first element of the list:")
list_data[[2]][1] = NULL
print("New list:")
print(list_data)
print(" 4. Update the second last student. ")
list_data <- list(c("21","22","23"),list("saurabh", "kisna", "sayali"))
print(list_data)
```

Output:-

```
print("1. Give names to the students in the list. ")
[1] "1. Give names to the students in the list. "
> list_data <- list(c("21","22","23"),list("saurabh", "radha", "sayali"))
> print(list_data)
[[1]]
[1] "21" "22" "23"

[[2]]
[[2]][[1]]
[1] "saurabh"

[[2]][[2]]
[1] "radha"

[[2]][[3]]
[1] "sayali"


> names(list_data) = c("Rollno", "Student(s)")
> print("List with column names:")
[1] "List with column names:"
> print(list_data)
$Rollno
[1] "21" "22" "23"

$`Student(s)`
$`Student(s)`[[1]]
[1] "saurabh"

$`Student(s)`[[2]]
[1] "radha"

$`Student(s)`[[3]]
[1] "sayali"
```

```
> print("2. Add a student at the end of the list. ")
[1] "2. Add a student at the end of the list. "
> list_data <- list(c("21","22","23"),list("saurabh", "radha", "sayali"))
> print(list_data)
[[1]]
[1] "21" "22" "23"

[[2]]
[[2]][[1]]
[1] "saurabh"

[[2]][[2]]
[1] "radha"

[[2]][[3]]
[1] "sayali"


> names(list_data) = c("Rollno", "Student(s)")
> print("Add a new element at the end of the list:")
[1] "Add a new element at the end of the list:"
> list_data[3] = "Sona"
> print("New list:")
[1] "New list:"
> print(list_data)
$Rollno
[1] "21" "22" "23"

$`Student(s)`
$`Student(s)`[[1]]
[1] "saurabh"

$`Student(s)`[[2]]
[1] "radha"

$`Student(s)`[[3]]
[1] "sayali"
```

```
[[3]]
[1] "Sona"

> print("3. Remove the firstStudent. ")
[1] "3. Remove the firstStudent. "
> list_data <- list(c("21","22","23"),list("saurabh", "radha", "sayali"))
> print(list_data)
[[1]]
[1] "21" "22" "23"

[[2]]
[[2]][[1]]
[1] "saurabh"

[[2]][[2]]
[1] "radha"

[[2]][[3]]
[1] "sayali"



> names(list_data) = c("Rollno", "Student(s)")
> print("Remove the first element of the list:")
[1] "Remove the first element of the list:"
> list_data[[2]][1] = NULL
> print("New list:")
[1] "New list:"
> print(list_data)
$Rollno
[1] "21" "22" "23"

$`Student(s)`
$`Student(s)`[[1]]
[1] "radha"
```

```
$`Student(s)`[[2]]
[1] "sayali"


> print(" 4. Update the second last student. ")
[1] " 4. Update the second last student. "
> list_data <- list(c("21","22","23"),list("saurabh", "kisna", "sayali"))
> print(list_data)
[[1]]
[1] "21" "22" "23"

[[2]]
[[2]][[1]]
[1] "saurabh"

[[2]][[2]]
[1] "kisna"

[[2]][[3]]
[1] "sayali"s
```

Slip No-30

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

```
x = c('lmn','pqr','xyz')
> print("Original Vectors:")
[1] "Original Vectors:"
> print(x)
[1] "lmn" "pqr" "xyz"
> print("Sort in ascending order:")
[1] "Sort in ascending order:"
> print(sort(x))
[1] "lmn" "pqr" "xyz"
> print("Sort in descending order:")
[1] "Sort in descending order:"
> print(sort(x, decreasing=TRUE))
[1] "xyz" "pqr" "lmn"
```