Practical no 4

//Concatenate two strings

```
> text <- "hello"

> paste(text, "Hello")
```

//data type of variable

```
>x <- 10.5

class(x)
```

//squareroot of the number

```
>x <- 100

>sqrt(x)
```

//length of the string variable

```
>str <- "Atik Salim Rangnekar"

>nchar(str)
```


//Write a R progra to creat a sequence of numbers from 20 to 50 and find the mean of umbers from 20 to 60 and sum of numbers from 51 to 91

```
>print("sequence of numbers from 20 to 50")

>print(seq(20,50))

>print("Mean of numbers from 20 to 60")

>print(mean(20:60))

>print("Sum of numbers from 51 to 91: ")

>print(sum(51:91))
```

//Write a R program to create three vectors numberic data, character data and logincal data.Display the content of the vectors and their type

```
a = c(1,2,3,4,4,0,-1,-2,-3,-4);

b = c("Red", "Green", "White", "Blue", "Black", "Yellow")

c = c(TRUE, FALSE, TRUE, TRUE, FALSE, TRUE, FALSE)

print(a)

print(class(a))

print(b)

print(class(b))

print(c)

print(class(c))
```

//create data frame of three vectors

```
name <- c("Anastasia", "Dima")

score <-c(12.5,16.5)

attempts <- c(1,2,3)

qualify <-c("yes", "no")

df = data.frame(name, score, attempts, qualify)
```


//Write a R program to extract specific coloumn from a data frame using column name

```
exam_data = data.frame(

name     = c('anstasia', 'dima'),

score = c(12.5,9),

attempts = c(1,2,3)

qualify = c("yes", "no")
```

)

>result <- data.frame(exam_date$name, exam_data$score)

Write a R program to create an ordered factor from data consisting of the names of months

>name_of_mon = c("january, "feb","others")

>fac = factor(name_of_mon)

>print(fac)

>print(table(fac))

practical no 5

line graph

plot(1:10, type = "l", col = "blue", lwd = 2)

line styles

plot(1:10, type = "l", lwd = 5, lty = 3)

0 remove the line

1 solid line

2. dashed line

3. dotted

4. dot dashed

5. long dashed

6. two dashed

subset

info <- subset(data, Salary > 55000)

newdata <- read.csv("output.csv")

dimensions

dim(data)

//subset

new_data <- subset(data, dept == "IT")

//import xlsx

install.packages(xlsx)

library(xlsx)

data <- read.xlsx("input.xlsx", sheetIndex = 1)

//get the people who joined on or after 2014 and write the output in new excel file

data <- read.xlsx("input.xlsx", sheetIndex = 1)

retval <- subset(data, as.Date(start_date) > as.Date("2014-01-01"))

//pie

x <- c(10, 20, 30, 40, 50)

->mylabel <- c("comedy", "action", "drama". "sci-fi", "romance")

-> pie(x, label = mylabel, main = "favourite movie categories", col = rainbow(length(x)))

-> legend("topright", mylabel, cex0.6, fill= rainbow(length(x)))

//scatter()

```
import matplotlib.pyplot as plt


x = [5,7,8,7,2,17,2,9,4,11,12,9,6]

y = [99,86,87,88,111,86,103,87,94,78,77,85,86]


plt.scatter(x, y)

plt.show()
```

Practical 6

Finolex Academy of Management and Technology, Ratnagiri

Department of MCA

Course:- MCAL13 Advance Database Management System Lab

Practical No -05: Graphics and Data importing

I. Import employee.csv file and perform following -

> data <- read.csv("employee.csv")

1. Display the content.

```
> data
   id     Name Age     Designation Salary isLocal
1   1 Michelle  44         Manager  72000      NA
2   2     Ryan  27           Clerk  48000      NA
3   3     Gary  30           Clerk  54000      NA
4   4     Guru  38        Engineer  61000      NA
5   5    Harsh  40           Clerk     NA      NA
6   6     Brad  35        Engineer  58000      NA
7   7    James  NA           Clerk  52000      NA
8   8     Tina  48 Senior_manager  79000      NA
9   9     Mina  50             CEO  83000      NA
10 10     Tara  37        Engineer  67000      NA
```


2. Find the dimensions of the data in the above imported dataset.

```
> dim(data)
[1] 10  6
```

3. Get all the people with designation “clerk”.

```
> new_data <- subset(data,Designation=="Clerk")
> new_data
  id  Name Age Designation Salary isLocal
2  2  Ryan  27       Clerk  48000      NA
3  3  Gary  30       Clerk  54000      NA
5  5 Harsh  40       Clerk     NA      NA
7  7 James  NA       Clerk  52000      NA
```

4. Get the people whose salary is greater than 55,000 and write the output in new excel file.

```
> info <- subset(data, Salary > 55000)
> info
  id     Name Age     Designation Salary isLocal
1  1 Michelle  44         Manager  72000      NA
4  4     Guru  38        Engineer  61000      NA
6  6     Brad  35        Engineer  58000      NA
8  8     Tina  48 Senior_manager  79000      NA
```

```
    9   9      Mina  50           CEO  83000       NA
    10  10     Tara  37      Engineer  67000       NA
```

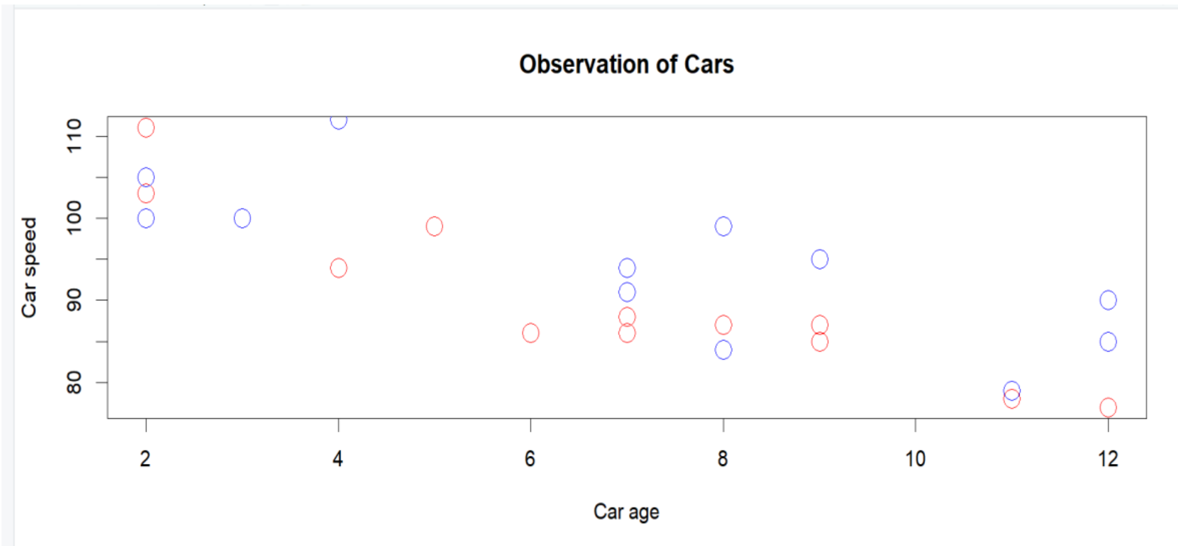
```
> write.csv(info,"Output.csv")
> newdata1 <- read.csv("Output.csv")
> newdata1
   X id     Name Age    Designation Salary isLocal
  1  1 Michelle  44        Manager  72000      NA
  4  4     Guru  38       Engineer  61000      NA
  6  6     Brad  35       Engineer  58000      NA
  8  8     Tina  48 Senior_manager  79000      NA
  9  9     Mina  50            CEO  83000      NA
 10 10     Tara  37       Engineer  67000      NA
```

5. **Summarize the above dataset**

```
> summary(newdata1)
          X               id              Name          
 Min.   : 1.000   Min.   : 1.000   Length:6          
 1st Qu.: 4.500   1st Qu.: 4.500   Class :character  
 Median : 7.000   Median : 7.000   Mode  :character  
 Mean   : 6.333   Mean   : 6.333                     
 3rd Qu.: 8.750   3rd Qu.: 8.750                     
 Max.   :10.000   Max.   :10.000                     
      Age        Designation            Salary     
 Min.   :35.00   Length:6           Min.   :58000  
 1st Qu.:37.25   Class :character   1st Qu.:62500  
 Median :41.00   Mode  :character   Median :69500  
 Mean   :42.00                      Mean   :70000  
 3rd Qu.:47.00                      3rd Qu.:77250  
 Max.   :50.00                      Max.   :83000  
 isLocal       
 Mode:logical  
 NA's:6        
```

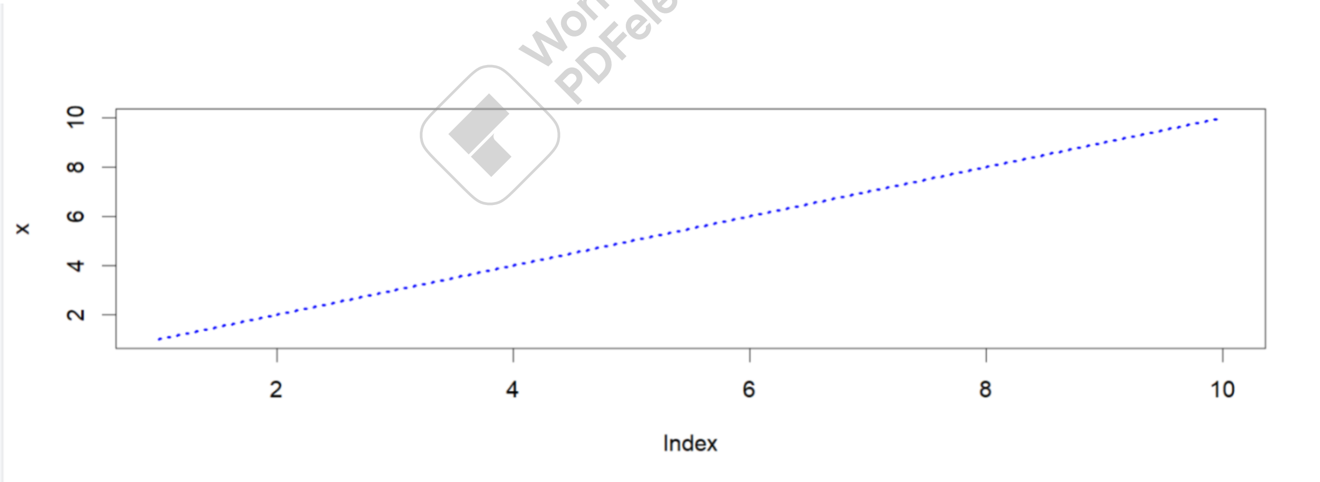
II. The age and speed of 12 cars observed on day 1 are age1(5,7,8,7,2,2,9,4,11,12,9,6), speed1(99,86,87,88,111,103,87,94,78,77,85,86) and on day 2 following values are observed age2(2,2,8,1,15,8,12,9,7,3,11,4,7,14,12), speed2(100,105,84,105,90,99,90,95,94,100,79,112,91,80,85). Write a R program to draw a scatterplot that compares observations of the two days.

```
> x1 <- c(5,7,8,7,2,2,9,4,11,12,9,6)
> y1 <- c(99,86,87,88,111,103,87,94,78,77,85,86)
> x2 <- c(2,2,8,1,15,8,12,9,7,3,11,4,7,14,12)
> y2 <- c(100,105,84,105,90,99,90,95,94,100,79,112,91,80,85)
> plot(x1, y1, main="Observation of Cars", xlab="Car age", yla
b="Car speed", col="red", cex=2)
> points(x2, y2, col="blue", cex=2)
```



iii) **Write a R program to create a vector with numerical values in a sequence from 1 to 10 and draw a blue colored dotted line of width 2 for the above vector.**

```
> x <- 1:10
> plot(x, type="l",col="blue",lty="dotted", lwd=2)
```



4. Write a R program to read the excel file "input.xlsx" and perform following

install.packages(xlsx)

library(xlsx)

```
> data <- read.xlsx("input.xlsx", sheetIndex = 1)
```

1. **Display the content.**

```
> data
  id     name salary start_date       dept
1  1     Rick 623.30 2012-01-01         IT
2  2      Dan 515.20 2013-09-23 Operations
3  3 Michelle 611.00 2014-11-15         IT
4  4     Ryan 729.00 2014-05-11         HR
5  5     Gary 843.25 2015-03-27    Finance
6  6     Nina 578.00 2013-05-21         IT
7  7    Simon 632.80 2013-07-30 Operations
8  8     Guru 722.50 2014-06-17    Finance
```

2. **Find the dimensions of the data in the above imported dataset.**

```
> dim(data)
[1] 8 5
```

3. **Get all the people working in IT department**

```
> new_data <- subset(data, dept=="IT")
> new_data
  id     name salary start_date dept
1  1     Rick  623.3 2012-01-01   IT
3  3 Michelle  611.0 2014-11-15   IT
6  6     Nina  578.0 2013-05-21   IT
```


4. **Get the people who joined on or after 2014 and write the output in new excel file.**

```
> data <- read.xlsx("input.xlsx", sheetIndex=1)
> retval <- subset(data, as.Date(start_date) > as.Date("2
014-01-01"))
> write.xlsx(retval,"output.xlsx")
> new <- read.xlsx("output.xlsx",sheetIndex = 1)
> new
  NA. id     name salary start_date    dept
1   3  3 Michelle 611.00 2014-11-15      IT
2   4  4     Ryan 729.00 2014-05-11      HR
3   5  5     Gary 843.25 2015-03-27 Finance
4   8  8     Guru 722.50 2014-06-17 Finance
```

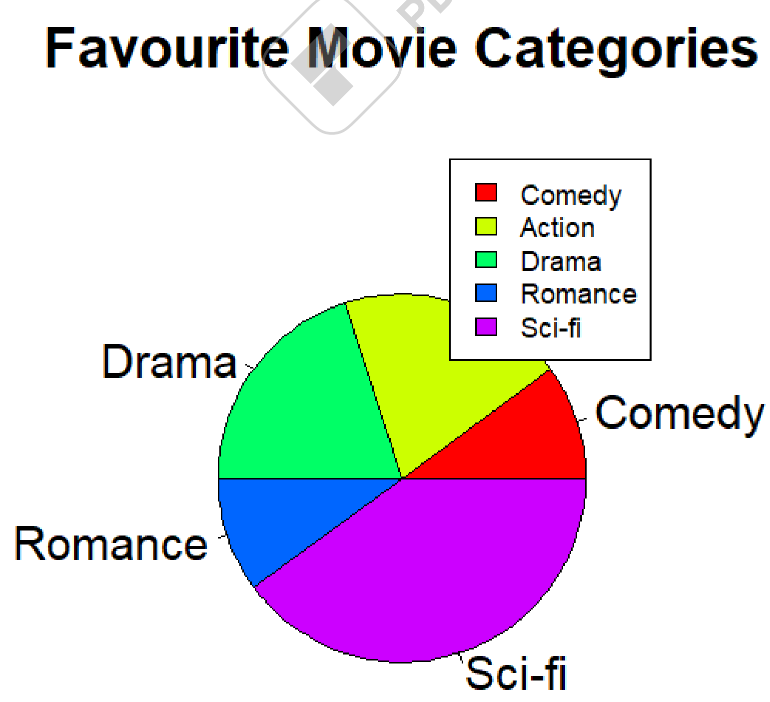
5. **Summarize the above dataset**

```
> summary(new)
     NA.                  id            name          
 Length:4           Min.   :3.00   Length:4          
 Class :character   1st Qu.:3.75   Class :character  
 Mode  :character   Median :4.50   Mode  :character  
                    Mean   :5.00                     
                    3rd Qu.:5.75                     
                    Max.   :8.00                     
     salary        start_date        
 Min.   :611.0   Min.   :2014-05-11  
 1st Qu.:694.6   1st Qu.:2014-06-07  
 Median :725.8   Median :2014-08-31  
```

```
Mean   :726.4   Mean   :2014-09-24
3rd Qu.:757.6   3rd Qu.:2014-12-18
Max.   :843.2   Max.   :2015-03-27
    dept
Length:4
Class :character
Mode  :character
```

6. **Create a pie chart for favourite movie categories (comedy,action,drama,romance,sci-fi). Consider appropriate percentages for creating pies. Add a list of explanation for each pie**

```
> x <- c(10,20,20,10,40)

> mylabel <- c("Comedy","Action","Drama","Sci-fi","Romanc
e")

> pie(x, label=mylabel, main="Favourite Movie Categories"
,col=rainbow(length(x)))

>legend("topright",c("Comedy","Action","Drama","Romance",
"Sci-fi"),cex=0.6,fill=rainbow(length(x)))
```

Finolex Academy of Management and Technology, Ratnagiri

Department of MCA

Course:- MCAL13 Advance Database Management System Lab

Practical No -07

---

1. **Write a program to perform k means clustering on iris dataset. Perform data pre-processing if required.**

```
# load packages-tidyverse,datasets,ggplot2
install.packages(tidyverse)
library(tidyverse)
install.packages(datasets)
library(datasets)
install.packages(ggplot2)
library(ggplot2)

#load dataset iris
>iris
```

```
Sepal.Length Sepal.Width Petal.Length Petal.Width
1           5.1         3.5          1.4         0.2
2           4.9         3.0          1.4         0.2
3           4.7         3.2          1.3         0.2
4           4.6         3.1          1.5         0.2
5           5.0         3.6          1.4         0.2
6           5.4         3.9          1.7         0.4
7           4.6         3.4          1.4         0.3
8           5.0         3.4          1.5         0.2
9           4.4         2.9          1.4         0.2
10          4.9         3.1          1.5         0.1
11          5.4         3.7          1.5         0.2
12          4.8         3.4          1.6         0.2
13          4.8         3.0          1.4         0.1
14          4.3         3.0          1.1         0.1
15          5.8         4.0          1.2         0.2
16          5.7         4.4          1.5         0.4
17          5.4         3.9          1.3         0.4
18          5.1         3.5          1.4         0.3
19          5.7         3.8          1.7         0.3
20          5.1         3.8          1.5         0.3
21          5.4         3.4          1.7         0.2
22          5.1         3.7          1.5         0.4
23          4.6         3.6          1.0         0.2
24          5.1         3.3          1.7         0.5
25          4.8         3.4          1.9         0.2
26          5.0         3.0          1.6         0.2
27          5.0         3.4          1.6         0.4
28          5.2         3.5          1.5         0.2
29          5.2         3.4          1.4         0.2
30          4.7         3.2          1.6         0.2
31          4.8         3.1          1.6         0.2
```

32	5.4	3.4	1.5	0.4
33	5.2	4.1	1.5	0.1
34	5.5	4.2	1.4	0.2
35	4.9	3.1	1.5	0.2
36	5.0	3.2	1.2	0.2
37	5.5	3.5	1.3	0.2
38	4.9	3.6	1.4	0.1
39	4.4	3.0	1.3	0.2
40	5.1	3.4	1.5	0.2
41	5.0	3.5	1.3	0.3
42	4.5	2.3	1.3	0.3
43	4.4	3.2	1.3	0.2
44	5.0	3.5	1.6	0.6
45	5.1	3.8	1.9	0.4
46	4.8	3.0	1.4	0.3
47	5.1	3.8	1.6	0.2
48	4.6	3.2	1.4	0.2
49	5.3	3.7	1.5	0.2
50	5.0	3.3	1.4	0.2
51	7.0	3.2	4.7	1.4
52	6.4	3.2	4.5	1.5
53	6.9	3.1	4.9	1.5
54	5.5	2.3	4.0	1.3
55	6.5	2.8	4.6	1.5
56	5.7	2.8	4.5	1.3
57	6.3	3.3	4.7	1.6
58	4.9	2.4	3.3	1.0
59	6.6	2.9	4.6	1.3
60	5.2	2.7	3.9	1.4
61	5.0	2.0	3.5	1.0
62	5.9	3.0	4.2	1.5
63	6.0	2.2	4.0	1.0
64	6.1	2.9	4.7	1.4
65	5.6	2.9	3.6	1.3
66	6.7	3.1	4.4	1.4
67	5.6	3.0	4.5	1.5
68	5.8	2.7	4.1	1.0
69	6.2	2.2	4.5	1.5
70	5.6	2.5	3.9	1.1
71	5.9	3.2	4.8	1.8
72	6.1	2.8	4.0	1.3
73	6.3	2.5	4.9	1.5
74	6.1	2.8	4.7	1.2
75	6.4	2.9	4.3	1.3
76	6.6	3.0	4.4	1.4
77	6.8	2.8	4.8	1.4
78	6.7	3.0	5.0	1.7
79	6.0	2.9	4.5	1.5
80	5.7	2.6	3.5	1.0
81	5.5	2.4	3.8	1.1
82	5.5	2.4	3.7	1.0
83	5.8	2.7	3.9	1.2
84	6.0	2.7	5.1	1.6
85	5.4	3.0	4.5	1.5
86	6.0	3.4	4.5	1.6
87	6.7	3.1	4.7	1.5
88	6.3	2.3	4.4	1.3
89	5.6	3.0	4.1	1.3

90	5.5	2.5	4.0	1.3
91	5.5	2.6	4.4	1.2
92	6.1	3.0	4.6	1.4
93	5.8	2.6	4.0	1.2
94	5.0	2.3	3.3	1.0
95	5.6	2.7	4.2	1.3
96	5.7	3.0	4.2	1.2
97	5.7	2.9	4.2	1.3
98	6.2	2.9	4.3	1.3
99	5.1	2.5	3.0	1.1
100	5.7	2.8	4.1	1.3
101	6.3	3.3	6.0	2.5
102	5.8	2.7	5.1	1.9
103	7.1	3.0	5.9	2.1
104	6.3	2.9	5.6	1.8
105	6.5	3.0	5.8	2.2
106	7.6	3.0	6.6	2.1
107	4.9	2.5	4.5	1.7
108	7.3	2.9	6.3	1.8
109	6.7	2.5	5.8	1.8
110	7.2	3.6	6.1	2.5
111	6.5	3.2	5.1	2.0
112	6.4	2.7	5.3	1.9
113	6.8	3.0	5.5	2.1
114	5.7	2.5	5.0	2.0
115	5.8	2.8	5.1	2.4
116	6.4	3.2	5.3	2.3
117	6.5	3.0	5.5	1.8
118	7.7	3.8	6.7	2.2
119	7.7	2.6	6.9	2.3
120	6.0	2.2	5.0	1.5
121	6.9	3.2	5.7	2.3
122	5.6	2.8	4.9	2.0
123	7.7	2.8	6.7	2.0
124	6.3	2.7	4.9	1.8
125	6.7	3.3	5.7	2.1
126	7.2	3.2	6.0	1.8
127	6.2	2.8	4.8	1.8
128	6.1	3.0	4.9	1.8
129	6.4	2.8	5.6	2.1
130	7.2	3.0	5.8	1.6
131	7.4	2.8	6.1	1.9
132	7.9	3.8	6.4	2.0
133	6.4	2.8	5.6	2.2
134	6.3	2.8	5.1	1.5
135	6.1	2.6	5.6	1.4
136	7.7	3.0	6.1	2.3
137	6.3	3.4	5.6	2.4
138	6.4	3.1	5.5	1.8
139	6.0	3.0	4.8	1.8
140	6.9	3.1	5.4	2.1
141	6.7	3.1	5.6	2.4
142	6.9	3.1	5.1	2.3
143	5.8	2.7	5.1	1.9
144	6.8	3.2	5.9	2.3
145	6.7	3.3	5.7	2.5
146	6.7	3.0	5.2	2.3
147	6.3	2.5	5.0	1.9

```
148          6.5         3.0          5.2         2.0
149          6.2         3.4          5.4         2.3
150          5.9         3.0          5.1         1.8
       Species
1       setosa
2       setosa
3       setosa
4       setosa
5       setosa
6       setosa
7       setosa
8       setosa
9       setosa
10      setosa
11      setosa
12      setosa
13      setosa
14      setosa
15      setosa
16      setosa
17      setosa
18      setosa
19      setosa
20      setosa
21      setosa
22      setosa
23      setosa
24      setosa
25      setosa
26      setosa
27      setosa
28      setosa
29      setosa
30      setosa
31      setosa
32      setosa
33      setosa
34      setosa
35      setosa
36      setosa
37      setosa
38      setosa
39      setosa
40      setosa
41      setosa
42      setosa
43      setosa
44      setosa
45      setosa
46      setosa
47      setosa
48      setosa
49      setosa
50      setosa
51  versicolor
52  versicolor
53  versicolor
54  versicolor
```

```
55   versicolor
56   versicolor
57   versicolor
58   versicolor
59   versicolor
60   versicolor
61   versicolor
62   versicolor
63   versicolor
64   versicolor
65   versicolor
66   versicolor
67   versicolor
68   versicolor
69   versicolor
70   versicolor
71   versicolor
72   versicolor
73   versicolor
74   versicolor
75   versicolor
76   versicolor
77   versicolor
78   versicolor
79   versicolor
80   versicolor
81   versicolor
82   versicolor
83   versicolor
84   versicolor
85   versicolor
86   versicolor
87   versicolor
88   versicolor
89   versicolor
90   versicolor
91   versicolor
92   versicolor
93   versicolor
94   versicolor
95   versicolor
96   versicolor
97   versicolor
98   versicolor
99   versicolor
100 versicolor
101  virginica
102  virginica
103  virginica
104  virginica
105  virginica
106  virginica
107  virginica
108  virginica
109  virginica
110  virginica
111  virginica
112  virginica
```

```
113  virginica
114  virginica
115  virginica
116  virginica
117  virginica
118  virginica
119  virginica
120  virginica
121  virginica
122  virginica
123  virginica
124  virginica
125  virginica
126  virginica
127  virginica
128  virginica
129  virginica
130  virginica
131  virginica
132  virginica
133  virginica
134  virginica
135  virginica
136  virginica
137  virginica
138  virginica
139  virginica
140  virginica
141  virginica
142  virginica
143  virginica
144  virginica
145  virginica
146  virginica
147  virginica
148  virginica
149  virginica
150  virginica
```


#information about iris dataset

>head(iris,4)

```
  Sepal.Length Sepal.Width
1          5.1         3.5
2          4.9         3.0
3          4.7         3.2
4          4.6         3.1
  Petal.Length Petal.Width Species
1          1.4         0.2  setosa
2          1.4         0.2  setosa
3          1.3         0.2  setosa
4          1.5         0.2  setosa
```

>tail(iris)

```
   Sepal.Length Sepal.Width
```

```
145          6.7         3.3
146          6.7         3.0
147          6.3         2.5
148          6.5         3.0
149          6.2         3.4
150          5.9         3.0
    Petal.Length Petal.Width
145          5.7         2.5
146          5.2         2.3
147          5.0         1.9
148          5.2         2.0
149          5.4         2.3
150          5.1         1.8
      Species
145 virginica
146 virginica
147 virginica
148 virginica
149 virginica
150 virginica


>dim(iris)
  [1] 150   5

>names(iris)
[1] "Sepal.Length" "Sepal.Width"
[3] "Petal.Length" "Petal.Width"
[5] "Species"

>attributes(iris)
$names
[1] "Sepal.Length" "Sepal.Width"
[3] "Petal.Length" "Petal.Width"
[5] "Species"

$class
[1] "data.frame"

$row.names
  [1]   1   2   3   4   5   6   7
  [8]   8   9  10  11  12  13  14
 [15]  15  16  17  18  19  20  21
 [22]  22  23  24  25  26  27  28
 [29]  29  30  31  32  33  34  35
 [36]  36  37  38  39  40  41  42
 [43]  43  44  45  46  47  48  49
 [50]  50  51  52  53  54  55  56
 [57]  57  58  59  60  61  62  63
 [64]  64  65  66  67  68  69  70
 [71]  71  72  73  74  75  76  77
 [78]  78  79  80  81  82  83  84
 [85]  85  86  87  88  89  90  91
 [92]  92  93  94  95  96  97  98
 [99]  99 100 101 102 103 104 105
[106] 106 107 108 109 110 111 112
[113] 113 114 115 116 117 118 119
[120] 120 121 122 123 124 125 126
```

```
[127] 127 128 129 130 131 132 133
[134] 134 135 136 137 138 139 140
[141] 141 142 143 144 145 146 147
[148] 148 149 150
```

```
Summary(iris)

Sepal.Length     Sepal.Width
Min.   :4.300    Min.   :2.000
1st Qu.:5.100    1st Qu.:2.800
Median :5.800    Median :3.000
Mean   :5.843    Mean   :3.057
3rd Qu.:6.400    3rd Qu.:3.300
Max.   :7.900    Max.   :4.400
Petal.Length     Petal.Width
Min.   :1.000    Min.   :0.100
1st Qu.:1.600    1st Qu.:0.300
Median :4.350    Median :1.300
Mean   :3.758    Mean   :1.199
3rd Qu.:5.100    3rd Qu.:1.800
Max.   :6.900    Max.   :2.500
  Species
setosa    :50
versicolor:50
virginica :50
```

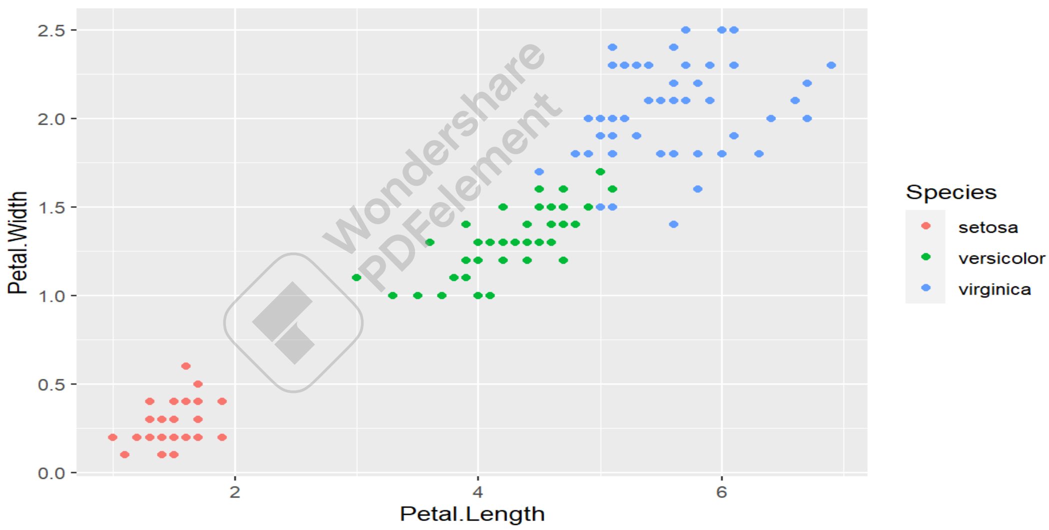
```
 iris[1:5,]
  Sepal.Length Sepal.Width
    1          5.1         3.5
    2          4.9         3.0
    3          4.7         3.2
    4          4.6         3.1
    5          5.0         3.6
      Petal.Length Petal.Width Species
    1          1.4         0.2  setosa
    2          1.4         0.2  setosa
    3          1.3         0.2  setosa
    4          1.5         0.2  setosa
    5          1.4         0.2  setosa
```

```
> iris[,1:1]
    [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6
    [8] 5.0 4.4 4.9 5.4 4.8 4.8 4.3
   [15] 5.8 5.7 5.4 5.1 5.7 5.1 5.4
   [22] 5.1 4.6 5.1 4.8 5.0 5.0 5.2
   [29] 5.2 4.7 4.8 5.4 5.2 5.5 4.9
   [36] 5.0 5.5 4.9 4.4 5.1 5.0 4.5
   [43] 4.4 5.0 5.1 4.8 5.1 4.6 5.3
   [50] 5.0 7.0 6.4 6.9 5.5 6.5 5.7
   [57] 6.3 4.9 6.6 5.2 5.0 5.9 6.0
   [64] 6.1 5.6 6.7 5.6 5.8 6.2 5.6
   [71] 5.9 6.1 6.3 6.1 6.4 6.6 6.8
   [78] 6.7 6.0 5.7 5.5 5.5 5.8 6.0
   [85] 5.4 6.0 6.7 6.3 5.6 5.5 5.5
   [92] 6.1 5.8 5.0 5.6 5.7 5.7 6.2
   [99] 5.1 5.7 6.3 5.8 7.1 6.3 6.5
  [106] 7.6 4.9 7.3 6.7 7.2 6.5 6.4
```

```
[113] 6.8 5.7 5.8 6.4 6.5 7.7 7.7
[120] 6.0 6.9 5.6 7.7 6.3 6.7 7.2
[127] 6.2 6.1 6.4 7.2 7.4 7.9 6.4
[134] 6.3 6.1 7.7 6.3 6.4 6.0 6.9
[141] 6.7 6.9 5.8 6.8 6.7 6.7 6.3
[148] 6.5 6.2 5.9

> iris[1:10,"Sepal.Length"]
 [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6
 [8] 5.0 4.4 4.9


> sum(is.na(iris))
[1] 0
```

#plot data using ggplot() function of ggplot2 library

```
> library(ggplot2)
> ggplot(iris, aes(Petal.Length, Petal.Width, color
= Species)) + geom_point()
```



#clustering

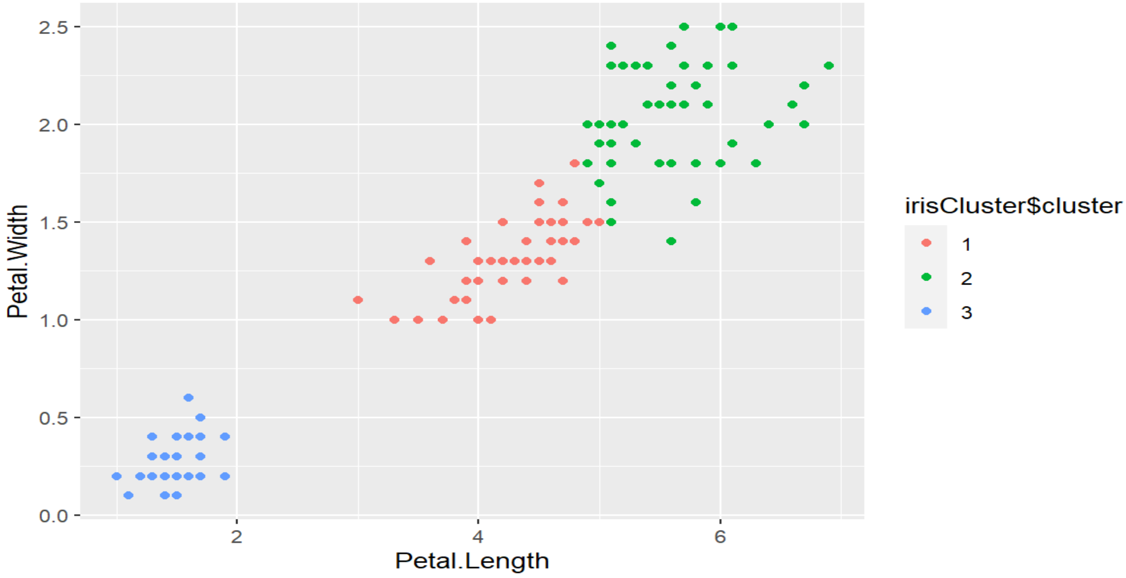
>Set.seed(20)

```
> irisCluster <- kmeans(iris[, 3:4], 3, nstart =
20)
> irisCluster
K-means clustering with 3 clusters of sizes 52, 4
8, 50

Cluster means:
  Petal.Length Petal.Width
1     4.269231    1.342308
2     5.595833    2.037500
```

```
3     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [20] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [39] 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1
 [58] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [77] 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1
 [96] 1 1 1 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2
[115] 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2
[134] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2

Within cluster sum of squares by cluster:
[1] 13.05769 16.29167  2.02200
 (between_SS / total_SS =  94.3 %)

Available components:

[1] "cluster"      "centers"
[3] "totss"        "withinss"
[5] "tot.withinss" "betweenss"
[7] "size"         "iter"
[9] "ifault"

> table(irisCluster$cluster, iris$Species)

    setosa versicolor virginica
  1      0         48         4
  2      0          2        46
  3     50          0         0
```


#plot data to see the clusters

```
> irisCluster$cluster <- as.factor(irisCluster$cl
uster)
> irisCluster$cluster
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [20] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [39] 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1
 [58] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [77] 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1
 [96] 1 1 1 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2
[115] 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2
[134] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2
Levels: 1 2 3


>ggplot(iris, aes(Petal.Length, Petal.W
idth, color = irisCluster$cluster)) + g
eom_point()
```



2. Implement Regression Classification for following example using R

years=(3,8,9,13,3,6,11,21,1,16)

salary=(30,57,64,72,36,43,59,90,20,83)

Predict salary of a person having 10 years of experience in a company.

→

#load packages

library(ggplot2)

library(tidyverse)

#create csv file of years=(3,8,9,13,3,6,11,21,1,16)

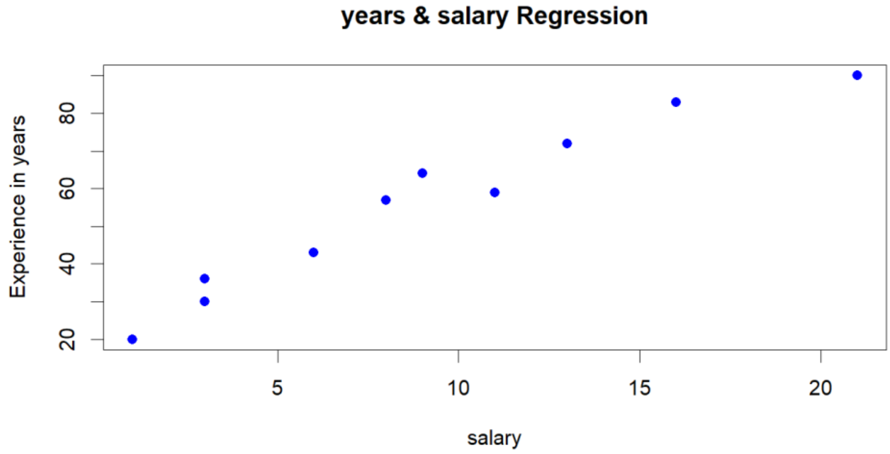
salary=(30,57,64,72,36,43,59,90,20,83) data

#import data from csv file

```
> rldata <- read.csv("linear01.csv")
> rldata
   Years Salary
1      3     30
2      8     57
3      9     64
4     13     72
5      3     36
6      6     43
7     11     59
8     21     90
9      1     20
10    16     83
```

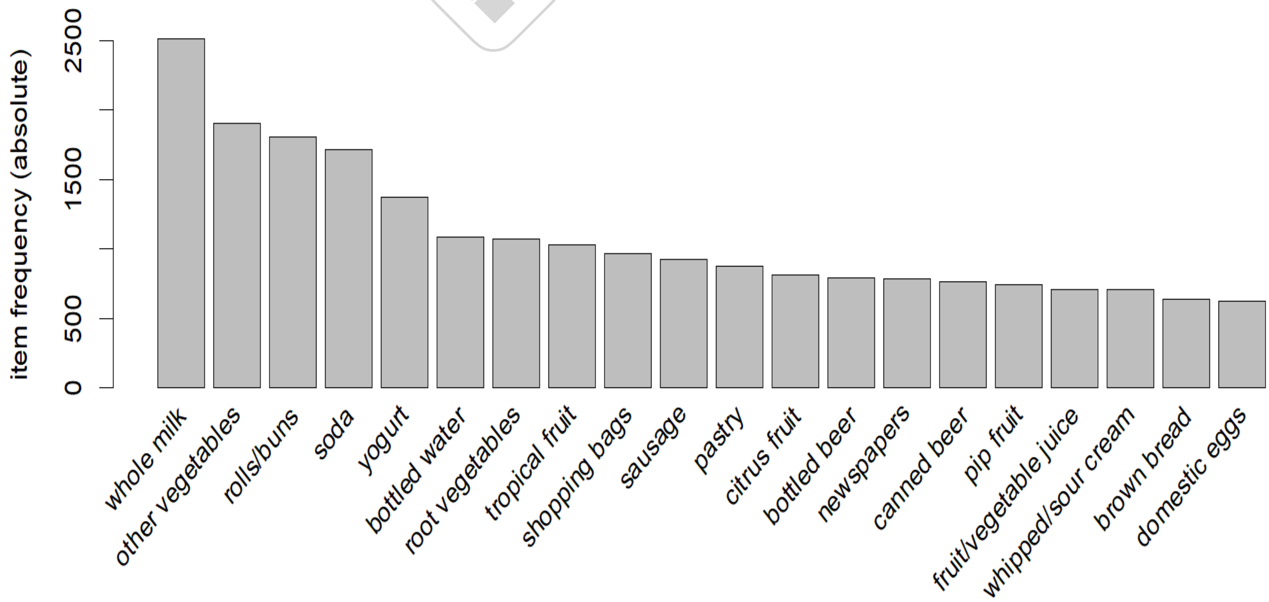
```
> relation <- lm(years~salary,data=rldata)
> relation

Call:
lm(formula = years ~ salary, data = rldata)

Coefficients:
(Intercept)       salary
    -5.7001       0.2671

> summary(relation)

Call:
lm(formula = years ~ salary, data = rldata)

Residuals:
    Min      1Q  Median
-2.3975 -0.8216 -0.1303
     3Q     Max
 0.8751  2.6566

Coefficients:
            Estimate
(Intercept) -5.70007
salary       0.26715
            Std. Error
(Intercept)    1.35614
salary         0.02278
            t value Pr(>|t|)
(Intercept)  -4.203  0.00298
salary       11.728 2.55e-06

(Intercept) **
salary      ***
---
Signif. codes:
  0 ‘***’ 0.001 ‘**’ 0.01
  ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.57 on 8 degrees of freedom
Multiple R-squared:  0.945,	Adjusted R-squared:  0.9382
F-statistic: 137.5 on 1 and 8 DF,  p-value: 2.553e-06

> #predict salary of 10person having 10yrs experience
> a<-data.frame(years=10)
> result <-predict(relation,a)
> result
         1          2
 2.3144096  9.5274388
         3          4
11.3974834 13.5346772
         5          6
 3.9173049  5.7873495
         7          8
10.0617372 18.3433634
         9         10
-0.3570827 16.4733187
```

```
> plot(rldata,col = "blue",pch = 16,main = "years & salary Reg
ression",ylab = "Experience in years",xlab = "salary")
```



3.Write a program to perform market basket analysis on Groceries dataset and display the top 5 important rules after sorting by confidence.

```
> library(arules)
> library(arulesViz)
> data("Groceries")
>#explore the data before making any rules
> itemFrequencyPlot(Groceries,topN=20,type="absolute")
```

> performing apriori algorithm and generating association rules

```
> rules <- apriori(Groceries, parameter = list(supp = 0.001, c
onf = 0.8))
Apriori

Parameter specification:
 confidence minval smax arem
        0.8    0.1    1 none
  aval originalSupport maxtime
 FALSE            TRUE       5
 support minlen maxlen target
   0.001      1     10  rules
  ext
 TRUE

Algorithmic control:
 filter tree heap memopt load
    0.1 TRUE TRUE  FALSE TRUE
 sort verbose
    2    TRUE

Absolute minimum support count: 9

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0
.01s].
sorting and recoding items ... [157 item(s)] done [0.00s].
creating transaction tree ... done [0.01s].
checking subsets of size 1 2 3 4 5 6 done [0.03s].
writing ... [410 rule(s)] done [0.00s].
creating S4 object  ... done [0.04s].


> options(digits=2)
> inspect(rules[1:5])
    lhs                       rhs
support confidence coverage lif
t count
[1] {liquor,
     red/blush wine} => {bottle
d beer}  0.0019       0.90   0.
0021 11.2    19
[2] {curd,
     cereals}        => {whole
milk}    0.0010       0.91   0.
0011  3.6    10
[3] {yogurt,
     cereals}        => {whole
milk}    0.0017       0.81   0.
0021  3.2    17
[4] {butter,
     jam}            => {whole
milk}    0.0010       0.83   0.
0012  3.3    10
[5] {soups,
     bottled beer}   => {whole
```

```
milk}    0.0011       0.92   0.
0012  3.6    11

>
> rules<-sort(rules, by="confidence", decreasing=TRUE)
> rules <- apriori(Groceries, parameter = list(supp = 0.001, c
onf = 0.8,maxlen=3))
Apriori

Parameter specification:
 confidence minval smax arem  aval
        0.8    0.1    1 none FALSE
 originalSupport maxtime support
            TRUE       5   0.001
 minlen maxlen target  ext
      1      3  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort
    0.1 TRUE TRUE  FALSE TRUE    2
 verbose
    TRUE

Absolute minimum support count: 9

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0
.01s].
sorting and recoding items ... [157 item(s)] done [0.00s].
creating transaction tree ... done [0.01s].
checking subsets of size 1 2 3 done [0.01s].
writing ... [29 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

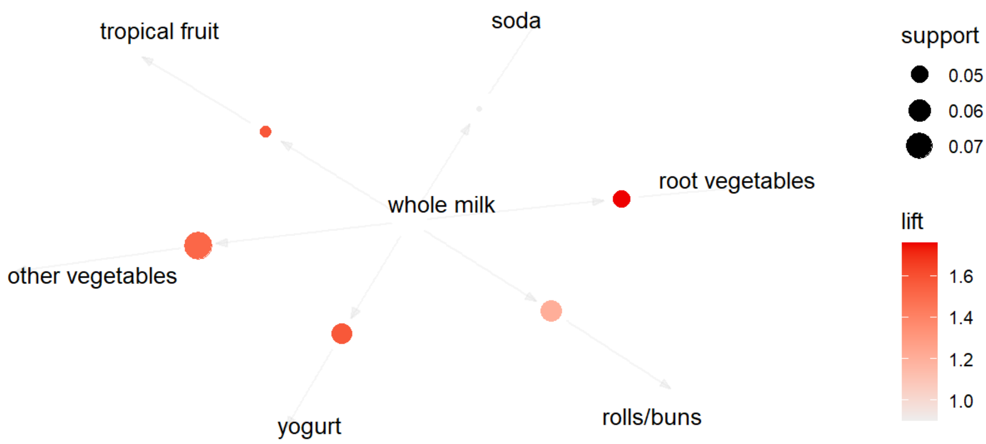

>rules<-apriori(data=Groceries, parameter=list(supp=0.001,conf = 0.15,minlen=2),appearance = list(default="rhs",lhs="whole milk"),control = list(verbose=F))rules<-sort(rules, decreasing=TRUE,by="confidence")

inspect(rules[1:5])

```
  lhs             rhs                  support confidence coverage lift
[1] {whole milk} => {other vegetables} 0.075    0.29        0.26
1.5
[2] {whole milk} => {rolls/buns}       0.057    0.22        0.26
1.2
[3] {whole milk} => {yogurt}           0.056    0.22        0.26
1.6
[4] {whole milk} => {root vegetables}  0.049    0.19        0.26
1.8
[5] {whole milk} => {tropical fruit}   0.042    0.17        0.26
1.6
    count
[1] 736
[2] 557
[3] 551
```

```
[4] 481
[5] 416
```

```
> plot(rules, method="graph")
```



4. Write a Program to perform naïve bayes classification on iris dataset. Perform data pre-processing if required.

Install.packages(caTools)

Library(caTools)

Install.packages(“e1071”)

Library(“e1071”)

Install.packages(“caret”)

#load dataset iris

>iris

```
Sepal.Length Sepal.Width Petal.Length Petal.Width
1           5.1         3.5          1.4         0.2
2           4.9         3.0          1.4         0.2
3           4.7         3.2          1.3         0.2
4           4.6         3.1          1.5         0.2
5           5.0         3.6          1.4         0.2
6           5.4         3.9          1.7         0.4
7           4.6         3.4          1.4         0.3
8           5.0         3.4          1.5         0.2
```

9	4.4	2.9	1.4	0.2
10	4.9	3.1	1.5	0.1
11	5.4	3.7	1.5	0.2
12	4.8	3.4	1.6	0.2
13	4.8	3.0	1.4	0.1
14	4.3	3.0	1.1	0.1
15	5.8	4.0	1.2	0.2
16	5.7	4.4	1.5	0.4
17	5.4	3.9	1.3	0.4
18	5.1	3.5	1.4	0.3
19	5.7	3.8	1.7	0.3
20	5.1	3.8	1.5	0.3
21	5.4	3.4	1.7	0.2
22	5.1	3.7	1.5	0.4
23	4.6	3.6	1.0	0.2
24	5.1	3.3	1.7	0.5
25	4.8	3.4	1.9	0.2
26	5.0	3.0	1.6	0.2
27	5.0	3.4	1.6	0.4
28	5.2	3.5	1.5	0.2
29	5.2	3.4	1.4	0.2
30	4.7	3.2	1.6	0.2
31	4.8	3.1	1.6	0.2
32	5.4	3.4	1.5	0.4
33	5.2	4.1	1.5	0.1
34	5.5	4.2	1.4	0.2
35	4.9	3.1	1.5	0.2
36	5.0	3.2	1.2	0.2
37	5.5	3.5	1.3	0.2
38	4.9	3.6	1.4	0.1
39	4.4	3.0	1.3	0.2
40	5.1	3.4	1.5	0.2
41	5.0	3.5	1.3	0.3
42	4.5	2.3	1.3	0.3
43	4.4	3.2	1.3	0.2
44	5.0	3.5	1.6	0.6
45	5.1	3.8	1.9	0.4
46	4.8	3.0	1.4	0.3
47	5.1	3.8	1.6	0.2
48	4.6	3.2	1.4	0.2
49	5.3	3.7	1.5	0.2
50	5.0	3.3	1.4	0.2
51	7.0	3.2	4.7	1.4
52	6.4	3.2	4.5	1.5
53	6.9	3.1	4.9	1.5
54	5.5	2.3	4.0	1.3
55	6.5	2.8	4.6	1.5
56	5.7	2.8	4.5	1.3
57	6.3	3.3	4.7	1.6
58	4.9	2.4	3.3	1.0
59	6.6	2.9	4.6	1.3
60	5.2	2.7	3.9	1.4
61	5.0	2.0	3.5	1.0
62	5.9	3.0	4.2	1.5
63	6.0	2.2	4.0	1.0
64	6.1	2.9	4.7	1.4
65	5.6	2.9	3.6	1.3
66	6.7	3.1	4.4	1.4

67	5.6	3.0	4.5	1.5
68	5.8	2.7	4.1	1.0
69	6.2	2.2	4.5	1.5
70	5.6	2.5	3.9	1.1
71	5.9	3.2	4.8	1.8
72	6.1	2.8	4.0	1.3
73	6.3	2.5	4.9	1.5
74	6.1	2.8	4.7	1.2
75	6.4	2.9	4.3	1.3
76	6.6	3.0	4.4	1.4
77	6.8	2.8	4.8	1.4
78	6.7	3.0	5.0	1.7
79	6.0	2.9	4.5	1.5
80	5.7	2.6	3.5	1.0
81	5.5	2.4	3.8	1.1
82	5.5	2.4	3.7	1.0
83	5.8	2.7	3.9	1.2
84	6.0	2.7	5.1	1.6
85	5.4	3.0	4.5	1.5
86	6.0	3.4	4.5	1.6
87	6.7	3.1	4.7	1.5
88	6.3	2.3	4.4	1.3
89	5.6	3.0	4.1	1.3
90	5.5	2.5	4.0	1.3
91	5.5	2.6	4.4	1.2
92	6.1	3.0	4.6	1.4
93	5.8	2.6	4.0	1.2
94	5.0	2.3	3.3	1.0
95	5.6	2.7	4.2	1.3
96	5.7	3.0	4.2	1.2
97	5.7	2.9	4.2	1.3
98	6.2	2.9	4.3	1.3
99	5.1	2.5	3.0	1.1
100	5.7	2.8	4.1	1.3
101	6.3	3.3	6.0	2.5
102	5.8	2.7	5.1	1.9
103	7.1	3.0	5.9	2.1
104	6.3	2.9	5.6	1.8
105	6.5	3.0	5.8	2.2
106	7.6	3.0	6.6	2.1
107	4.9	2.5	4.5	1.7
108	7.3	2.9	6.3	1.8
109	6.7	2.5	5.8	1.8
110	7.2	3.6	6.1	2.5
111	6.5	3.2	5.1	2.0
112	6.4	2.7	5.3	1.9
113	6.8	3.0	5.5	2.1
114	5.7	2.5	5.0	2.0
115	5.8	2.8	5.1	2.4
116	6.4	3.2	5.3	2.3
117	6.5	3.0	5.5	1.8
118	7.7	3.8	6.7	2.2
119	7.7	2.6	6.9	2.3
120	6.0	2.2	5.0	1.5
121	6.9	3.2	5.7	2.3
122	5.6	2.8	4.9	2.0
123	7.7	2.8	6.7	2.0
124	6.3	2.7	4.9	1.8

125	6.7	3.3	5.7	2.1
126	7.2	3.2	6.0	1.8
127	6.2	2.8	4.8	1.8
128	6.1	3.0	4.9	1.8
129	6.4	2.8	5.6	2.1
130	7.2	3.0	5.8	1.6
131	7.4	2.8	6.1	1.9
132	7.9	3.8	6.4	2.0
133	6.4	2.8	5.6	2.2
134	6.3	2.8	5.1	1.5
135	6.1	2.6	5.6	1.4
136	7.7	3.0	6.1	2.3
137	6.3	3.4	5.6	2.4
138	6.4	3.1	5.5	1.8
139	6.0	3.0	4.8	1.8
140	6.9	3.1	5.4	2.1
141	6.7	3.1	5.6	2.4
142	6.9	3.1	5.1	2.3
143	5.8	2.7	5.1	1.9
144	6.8	3.2	5.9	2.3
145	6.7	3.3	5.7	2.5
146	6.7	3.0	5.2	2.3
147	6.3	2.5	5.0	1.9
148	6.5	3.0	5.2	2.0
149	6.2	3.4	5.4	2.3
150	5.9	3.0	5.1	1.8

	Species
1	setosa
2	setosa
3	setosa
4	setosa
5	setosa
6	setosa
7	setosa
8	setosa
9	setosa
10	setosa
11	setosa
12	setosa
13	setosa
14	setosa
15	setosa
16	setosa
17	setosa
18	setosa
19	setosa
20	setosa
21	setosa
22	setosa
23	setosa
24	setosa
25	setosa
26	setosa
27	setosa
28	setosa
29	setosa
30	setosa
31	setosa

```
32      setosa
33      setosa
34      setosa
35      setosa
36      setosa
37      setosa
38      setosa
39      setosa
40      setosa
41      setosa
42      setosa
43      setosa
44      setosa
45      setosa
46      setosa
47      setosa
48      setosa
49      setosa
50      setosa
51  versicolor
52  versicolor
53  versicolor
54  versicolor
55  versicolor
56  versicolor
57  versicolor
58  versicolor
59  versicolor
60  versicolor
61  versicolor
62  versicolor
63  versicolor
64  versicolor
65  versicolor
66  versicolor
67  versicolor
68  versicolor
69  versicolor
70  versicolor
71  versicolor
72  versicolor
73  versicolor
74  versicolor
75  versicolor
76  versicolor
77  versicolor
78  versicolor
79  versicolor
80  versicolor
81  versicolor
82  versicolor
83  versicolor
84  versicolor
85  versicolor
86  versicolor
87  versicolor
88  versicolor
89  versicolor
```

```
90   versicolor
91   versicolor
92   versicolor
93   versicolor
94   versicolor
95   versicolor
96   versicolor
97   versicolor
98   versicolor
99   versicolor
100 versicolor
101   virginica
102   virginica
103   virginica
104   virginica
105   virginica
106   virginica
107   virginica
108   virginica
109   virginica
110   virginica
111   virginica
112   virginica
113   virginica
114   virginica
115   virginica
116   virginica
117   virginica
118   virginica
119   virginica
120   virginica
121   virginica
122   virginica
123   virginica
124   virginica
125   virginica
126   virginica
127   virginica
128   virginica
129   virginica
130   virginica
131   virginica
132   virginica
133   virginica
134   virginica
135   virginica
136   virginica
137   virginica
138   virginica
139   virginica
140   virginica
141   virginica
142   virginica
143   virginica
144   virginica
145   virginica
146   virginica
147   virginica
```

```
148  virginica
149  virginica
150  virginica


> dim(iris)
[1] 150   5


> table(iris$Species)

    setosa versicolor  virginica
        50         50         50
> set.seed(123)
> split = sample.split(iris$Species, SplitRatio = 0.7)#
> split
  [1]  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE
  [9]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
 [17]  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE
 [25]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
 [33]  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
 [41]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [49]  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
 [57]  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
 [65] FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE
 [73] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [81]  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE
 [89] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [97] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
[105]  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
[113]  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE
[121]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
[129]  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE
[137] FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
[145] FALSE  TRUE  TRUE  TRUE  TRUE FALSE


> #Creating the training set and test set separately
> training_set = subset(iris, split == TRUE)
> test_set = subset(iris, split == FALSE)
> training_set
    Sepal.Length Sepal.Width Petal.Length Petal.Width
1            5.1         3.5          1.4         0.2
3            4.7         3.2          1.3         0.2
6            5.4         3.9          1.7         0.4
7            4.6         3.4          1.4         0.3
9            4.4         2.9          1.4         0.2
10           4.9         3.1          1.5         0.1
12           4.8         3.4          1.6         0.2
13           4.8         3.0          1.4         0.1
14           4.3         3.0          1.1         0.1
15           5.8         4.0          1.2         0.2
17           5.4         3.9          1.3         0.4
18           5.1         3.5          1.4         0.3
19           5.7         3.8          1.7         0.3
22           5.1         3.7          1.5         0.4
23           4.6         3.6          1.0         0.2
```

25	4.8	3.4	1.9	0.2
27	5.0	3.4	1.6	0.4
28	5.2	3.5	1.5	0.2
29	5.2	3.4	1.4	0.2
30	4.7	3.2	1.6	0.2
33	5.2	4.1	1.5	0.1
35	4.9	3.1	1.5	0.2
36	5.0	3.2	1.2	0.2
38	4.9	3.6	1.4	0.1
39	4.4	3.0	1.3	0.2
40	5.1	3.4	1.5	0.2
41	5.0	3.5	1.3	0.3
42	4.5	2.3	1.3	0.3
43	4.4	3.2	1.3	0.2
44	5.0	3.5	1.6	0.6
45	5.1	3.8	1.9	0.4
46	4.8	3.0	1.4	0.3
47	5.1	3.8	1.6	0.2
48	4.6	3.2	1.4	0.2
49	5.3	3.7	1.5	0.2
51	7.0	3.2	4.7	1.4
52	6.4	3.2	4.5	1.5
54	5.5	2.3	4.0	1.3
55	6.5	2.8	4.6	1.5
56	5.7	2.8	4.5	1.3
57	6.3	3.3	4.7	1.6
60	5.2	2.7	3.9	1.4
61	5.0	2.0	3.5	1.0
62	5.9	3.0	4.2	1.5
63	6.0	2.2	4.0	1.0
64	6.1	2.9	4.7	1.4
66	6.7	3.1	4.4	1.4
70	5.6	2.5	3.9	1.1
72	6.1	2.8	4.0	1.3
74	6.1	2.8	4.7	1.2
75	6.4	2.9	4.3	1.3
76	6.6	3.0	4.4	1.4
77	6.8	2.8	4.8	1.4
78	6.7	3.0	5.0	1.7
79	6.0	2.9	4.5	1.5
80	5.7	2.6	3.5	1.0
81	5.5	2.4	3.8	1.1
83	5.8	2.7	3.9	1.2
85	5.4	3.0	4.5	1.5
86	6.0	3.4	4.5	1.6
90	5.5	2.5	4.0	1.3
91	5.5	2.6	4.4	1.2
92	6.1	3.0	4.6	1.4
93	5.8	2.6	4.0	1.2
94	5.0	2.3	3.3	1.0
95	5.6	2.7	4.2	1.3
96	5.7	3.0	4.2	1.2
98	6.2	2.9	4.3	1.3
99	5.1	2.5	3.0	1.1
100	5.7	2.8	4.1	1.3
101	6.3	3.3	6.0	2.5
102	5.8	2.7	5.1	1.9
103	7.1	3.0	5.9	2.1

105	6.5	3.0	5.8	2.2
108	7.3	2.9	6.3	1.8
109	6.7	2.5	5.8	1.8
110	7.2	3.6	6.1	2.5
112	6.4	2.7	5.3	1.9
113	6.8	3.0	5.5	2.1
116	6.4	3.2	5.3	2.3
117	6.5	3.0	5.5	1.8
119	7.7	2.6	6.9	2.3
120	6.0	2.2	5.0	1.5
121	6.9	3.2	5.7	2.3
122	5.6	2.8	4.9	2.0
123	7.7	2.8	6.7	2.0
124	6.3	2.7	4.9	1.8
125	6.7	3.3	5.7	2.1
127	6.2	2.8	4.8	1.8
128	6.1	3.0	4.9	1.8
129	6.4	2.8	5.6	2.1
130	7.2	3.0	5.8	1.6
131	7.4	2.8	6.1	1.9
133	6.4	2.8	5.6	2.2
135	6.1	2.6	5.6	1.4
136	7.7	3.0	6.1	2.3
140	6.9	3.1	5.4	2.1
141	6.7	3.1	5.6	2.4
142	6.9	3.1	5.1	2.3
143	5.8	2.7	5.1	1.9
144	6.8	3.2	5.9	2.3
146	6.7	3.0	5.2	2.3
147	6.3	2.5	5.0	1.9
148	6.5	3.0	5.2	2.0
149	6.2	3.4	5.4	2.3

	Species
1	setosa
3	setosa
6	setosa
7	setosa
9	setosa
10	setosa
12	setosa
13	setosa
14	setosa
15	setosa
17	setosa
18	setosa
19	setosa
22	setosa
23	setosa
25	setosa
27	setosa
28	setosa
29	setosa
30	setosa
33	setosa
35	setosa
36	setosa
38	setosa
39	setosa

40	setosa
41	setosa
42	setosa
43	setosa
44	setosa
45	setosa
46	setosa
47	setosa
48	setosa
49	setosa
51	versicolor
52	versicolor
54	versicolor
55	versicolor
56	versicolor
57	versicolor
60	versicolor
61	versicolor
62	versicolor
63	versicolor
64	versicolor
66	versicolor
70	versicolor
72	versicolor
74	versicolor
75	versicolor
76	versicolor
77	versicolor
78	versicolor
79	versicolor
80	versicolor
81	versicolor
83	versicolor
85	versicolor
86	versicolor
90	versicolor
91	versicolor
92	versicolor
93	versicolor
94	versicolor
95	versicolor
96	versicolor
98	versicolor
99	versicolor
100	versicolor
101	virginica
102	virginica
103	virginica
105	virginica
108	virginica
109	virginica
110	virginica
112	virginica
113	virginica
116	virginica
117	virginica
119	virginica
120	virginica

```
121  virginica
122  virginica
123  virginica
124  virginica
125  virginica
127  virginica
128  virginica
129  virginica
130  virginica
131  virginica
133  virginica
135  virginica
136  virginica
140  virginica
141  virginica
142  virginica
143  virginica
144  virginica
146  virginica
147  virginica
148  virginica
149  virginica
> test_set
    Sepal.Length Sepal.Width Petal.Length Petal.Width
2            4.9         3.0          1.4         0.2
4            4.6         3.1          1.5         0.2
5            5.0         3.6          1.4         0.2
8            5.0         3.4          1.5         0.2
11           5.4         3.7          1.5         0.2
16           5.7         4.4          1.5         0.4
20           5.1         3.8          1.5         0.3
21           5.4         3.4          1.7         0.2
24           5.1         3.3          1.7         0.5
26           5.0         3.0          1.6         0.2
31           4.8         3.1          1.6         0.2
32           5.4         3.4          1.5         0.4
34           5.5         4.2          1.4         0.2
37           5.5         3.5          1.3         0.2
50           5.0         3.3          1.4         0.2
53           6.9         3.1          4.9         1.5
58           4.9         2.4          3.3         1.0
59           6.6         2.9          4.6         1.3
65           5.6         2.9          3.6         1.3
67           5.6         3.0          4.5         1.5
68           5.8         2.7          4.1         1.0
69           6.2         2.2          4.5         1.5
71           5.9         3.2          4.8         1.8
73           6.3         2.5          4.9         1.5
82           5.5         2.4          3.7         1.0
84           6.0         2.7          5.1         1.6
87           6.7         3.1          4.7         1.5
88           6.3         2.3          4.4         1.3
89           5.6         3.0          4.1         1.3
97           5.7         2.9          4.2         1.3
104          6.3         2.9          5.6         1.8
106          7.6         3.0          6.6         2.1
107          4.9         2.5          4.5         1.7
111          6.5         3.2          5.1         2.0
```

```
114          5.7         2.5          5.0         2.0
115          5.8         2.8          5.1         2.4
118          7.7         3.8          6.7         2.2
126          7.2         3.2          6.0         1.8
132          7.9         3.8          6.4         2.0
134          6.3         2.8          5.1         1.5
137          6.3         3.4          5.6         2.4
138          6.4         3.1          5.5         1.8
139          6.0         3.0          4.8         1.8
145          6.7         3.3          5.7         2.5
150          5.9         3.0          5.1         1.8
       Species
2        setosa
4        setosa
5        setosa
8        setosa
11       setosa
16       setosa
20       setosa
21       setosa
24       setosa
26       setosa
31       setosa
32       setosa
34       setosa
37       setosa
50       setosa
53   versicolor
58   versicolor
59   versicolor
65   versicolor
67   versicolor
68   versicolor
69   versicolor
71   versicolor
73   versicolor
82   versicolor
84   versicolor
87   versicolor
88   versicolor
89   versicolor
97   versicolor
104   virginica
106   virginica
107   virginica
111   virginica
114   virginica
115   virginica
118   virginica
126   virginica
132   virginica
134   virginica
137   virginica
138   virginica
139   virginica
145   virginica
150   virginica
> table(test_set$Species)
```

```
    setosa versicolor  virginica 
        15         15         15 
> iris_classifier=naiveBayes(Species ~ ., data = training_set)
> iris_classifier

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
    setosa versicolor  virginica 
 0.3333333  0.3333333  0.3333333 

Conditional probabilities:
            Sepal.Length
Y                [,1]      [,2]
  setosa     4.940000 0.3541352
  versicolor 5.920000 0.5166635
  virginica  6.634286 0.5422952

            Sepal.Width
Y                [,1]      [,2]
  setosa     3.405714 0.3685766
  versicolor 2.777143 0.3144423
  virginica  2.925714 0.2831990

            Petal.Length
Y                [,1]      [,2]
  setosa     1.445714 0.1930298
  versicolor 4.217143 0.4462166
  virginica  5.565714 0.5075563

            Petal.Width
Y                 [,1]      [,2]
  setosa     0.2428571 0.1092372
  versicolor 1.3114286 0.1827429
  virginica  2.0428571 0.2714728

> iris_test_pred=predict(iris_classifier,test_set)
> iris_test_pred
 [1] setosa     setosa    
 [3] setosa     setosa    
 [5] setosa     setosa    
 [7] setosa     setosa    
 [9] setosa     setosa    
[11] setosa     setosa    
[13] setosa     setosa    
[15] setosa     virginica 
[17] versicolor versicolor
[19] versicolor versicolor
[21] versicolor versicolor
[23] virginica  versicolor
[25] versicolor virginica 
[27] versicolor versicolor
[29] versicolor versicolor
```

```
[31] virginica  virginica
[33] versicolor virginica
[35] virginica  virginica
[37] virginica  virginica
[39] virginica  versicolor
[41] virginica  virginica
[43] virginica  virginica
[45] virginica
3 Levels: setosa ... virginica


> table(test_set$Species)

    setosa versicolor  virginica
        15         15         15



> table(iris_test_pred)
iris_test_pred
    setosa versicolor  virginica
        15         14         16
> table(iris_test_pred,test_set$Species,dnn=c("Prediction","Ac
tual"))
            Actual
Prediction   setosa versicolor virginica
  setosa         15          0         0
  versicolor      0         12         2
  virginica       0          3        13
> iris_classifier_lap=naiveBayes(Species ~ ., data = training_
set,laplace=1)
> iris_classifier_lap

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
    setosa versicolor  virginica
 0.3333333  0.3333333  0.3333333

Conditional probabilities:
            Sepal.Length
Y                [,1]      [,2]
  setosa     4.940000 0.3541352
  versicolor 5.920000 0.5166635
  virginica  6.634286 0.5422952

            Sepal.Width
Y                [,1]      [,2]
  setosa     3.405714 0.3685766
  versicolor 2.777143 0.3144423
  virginica  2.925714 0.2831990

            Petal.Length
```

```
Y                [,1]      [,2]
  setosa     1.445714 0.1930298
  versicolor 4.217143 0.4462166
  virginica  5.565714 0.5075563

            Petal.Width
Y                 [,1]      [,2]
  setosa     0.2428571 0.1092372
  versicolor 1.3114286 0.1827429
  virginica  2.0428571 0.2714728
> table(iris_test_pred_lab)
iris_test_pred_lab
    setosa versicolor  virginica 
        15         14         16 


> table(iris_test_pred,test_set$Species,dnn=c("Prediction","Ac
tual"))
            Actual
Prediction   setosa versicolor virginica
  setosa         15          0         0
  versicolor      0         12         2
  virginica       0          3        13
cm=confusionMatrix(test_set$Species,iris_test_pred)
> print(cm)
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         15          0         0
  versicolor      0         12         3
  virginica       0          2        13

Overall Statistics
                                          
               Accuracy : 0.8889          
                 95% CI : (0.7595, 0.9629)
    No Information Rate : 0.3556          
    P-Value [Acc > NIR] : 1.581e-13       
                                          
                  Kappa : 0.8333          
                                          
 Mcnemar's Test P-Value : NA              

Statistics by Class:

                     Class: setosa Class: versicolor
Sensitivity                 1.0000            0.8571
Specificity                 1.0000            0.9032
Pos Pred Value              1.0000            0.8000
Neg Pred Value              1.0000            0.9333
Prevalence                  0.3333            0.3111
Detection Rate              0.3333            0.2667
Detection Prevalence        0.3333            0.3333
Balanced Accuracy           1.0000            0.8802
                     Class: virginica
Sensitivity                    0.8125
```

```
Specificity                    0.9310
Pos Pred Value                 0.8667
Neg Pred Value                 0.9000
Prevalence                     0.3556
Detection Rate                 0.2889
Detection Prevalence           0.3333
Balanced Accuracy              0.8718
```

5. Write a Program to perform naïve bayes classification on Titanic dataset. Perform data pre-processing if required.

```
> Titanic
, , Age = Child, Survived = No

      Sex
Class  Male Female
  1st     0      0
  2nd     0      0
  3rd    35     17
  Crew    0      0

, , Age = Adult, Survived = No

      Sex
Class  Male Female
  1st   118      4
  2nd   154     13
  3rd   387     89
  Crew  670      3

, , Age = Child, Survived = Yes

      Sex
Class  Male Female
  1st     5      1
  2nd    11     13
  3rd    13     14
  Crew    0      0

, , Age = Adult, Survived = Yes

      Sex
Class  Male Female
  1st    57    140
  2nd    14     80
  3rd    75     76
  Crew  192     20

> class(Titanic)
[1] "table"


> head(Titanic)
, , Age = Child, Survived = No

      Sex
Class  Male Female
  1st     0      0
```

```
  2nd     0      0
  3rd    35     17
  Crew    0      0

, , Age = Adult, Survived = No

      Sex
Class  Male Female
  1st   118      4
  2nd   154     13
  3rd   387     89
  Crew  670      3

, , Age = Child, Survived = Yes

      Sex
Class  Male Female
  1st     5      1
  2nd    11     13
  3rd    13     14
  Crew    0      0

, , Age = Adult, Survived = Yes

      Sex
Class  Male Female
  1st    57    140
  2nd    14     80
  3rd    75     76
  Crew  192     20


> str(Titanic)
 'table' num [1:4, 1:2, 1:2, 1:2] 0 0 35 0 0 0 17 0 118 154 ..
.
 - attr(*, "dimnames")=List of 4
  ..$ Class   : chr [1:4] "1st" "2nd" "3rd" "Crew"
  ..$ Sex     : chr [1:2] "Male" "Female"
  ..$ Age     : chr [1:2] "Child" "Adult"
  ..$ Survived: chr [1:2] "No" "Yes"
> dfdata <- as.data.frame(Titanic)
> dfdata
Class    Sex   Age Survived Freq
1    1st   Male Child       No    0
2    2nd   Male Child       No    0
3    3rd   Male Child       No   35
4   Crew   Male Child       No    0
5    1st Female Child       No    0
6    2nd Female Child       No    0
7    3rd Female Child       No   17
8   Crew Female Child       No    0
9    1st   Male Adult       No  118
10   2nd   Male Adult       No  154
11   3rd   Male Adult       No  387
12  Crew   Male Adult       No  670
13   1st Female Adult       No    4
14   2nd Female Adult       No   13
15   3rd Female Adult       No   89
16  Crew Female Adult       No    3
17   1st   Male Child      Yes    5
18   2nd   Male Child      Yes   11
```

```
19   3rd   Male Child      Yes   13
20  Crew   Male Child      Yes    0
21   1st Female Child      Yes    1
22   2nd Female Child      Yes   13
23   3rd Female Child      Yes   14
24  Crew Female Child      Yes    0
25   1st   Male Adult      Yes   57
26   2nd   Male Adult      Yes   14
27   3rd   Male Adult      Yes   75
28  Crew   Male Adult      Yes  192
29   1st Female Adult      Yes  140
30   2nd Female Adult      Yes   80
31   3rd Female Adult      Yes   76
32  Crew Female Adult      Yes   20
```

```
> names(dfdata)
[1] "Class"     "Sex"
[3] "Age"       "Survived"
[5] "Freq"
```

```
> dim(dfdata)
[1] 32  5
```

```
> set.seed(123)
> split=sample.split(df_data$Survived,SplitRatio = 0.7)
> split
 [1]  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TR
UE
[11] FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FAL
SE
[21] FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE
[31] FALSE FALSE
```

```
> training_set1=subset(dfdata,split==TRUE)
> training_set1
   Class    Sex   Age Survived Freq
1    1st   Male Child       No    0
2    2nd   Male Child       No    0
3    3rd   Male Child       No   35
6    2nd Female Child       No    0
7    3rd Female Child       No   17
9    1st   Male Adult       No  118
10   2nd   Male Adult       No  154
12  Crew   Male Adult       No  670
13   1st Female Adult       No    4
14   2nd Female Adult       No   13
15   3rd Female Adult       No   89
17   1st   Male Child      Yes    5
18   2nd   Male Child      Yes   11
19   3rd   Male Child      Yes   13
22   2nd Female Child      Yes   13
23   3rd Female Child      Yes   14
25   1st   Male Adult      Yes   57
26   2nd   Male Adult      Yes   14
27   3rd   Male Adult      Yes   75
```

```
28   Crew   Male Adult      Yes   192
29    1st Female Adult      Yes   140
30    2nd Female Adult      Yes    80
> nrow(training_set1)
[1] 22
> ncol(training_set1)
[1] 5

> test_set1 = subset(dfdata, t_split == FALSE)
> test_set1
   Class    Sex   Age
5    1st Female Child
11   3rd   Male Adult
16  Crew Female Adult
20  Crew   Male Child
24  Crew Female Child
31   3rd Female Adult
   Survived Freq
5        No    0
11       No  387
16       No    3
20      Yes    0
24      Yes    0
31      Yes   76


>table(test_set1$Survived)

 No Yes
  3   3
> titanic_classifier=naiveBayes(Survived ~ ., data = training_
set1)
> titanic_classifier

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
 No Yes
0.5 0.5

Conditional probabilities:
     Class
Y           1st       2nd
  No  0.2307692 0.3076923
  Yes 0.3076923 0.3076923
     Class
Y           3rd      Crew
  No  0.2307692 0.2307692
  Yes 0.2307692 0.1538462

     Sex
Y          Male    Female
  No  0.5384615 0.4615385
```

```
  Yes 0.5384615 0.4615385

     Age
Y         Child     Adult
  No  0.5384615 0.4615385
  Yes 0.4615385 0.5384615

     Freq
Y         [,1]      [,2]
  No  84.61538 183.27645
  Yes 48.84615  59.15917

> titanic_test_pred=predict(titanic_classifier,test_set1)
> titanic_test_pred
[1] Yes No  Yes Yes Yes Yes
Levels: No Yes

> table(titanic_test_pred)
titanic_test_pred
 No Yes
  1   5

> table(titanic_test_pred, test_set1$Survived,dnn=c("Predictio
n","Actual"))
          Actual
Prediction No Yes
       No   1   0
       Yes  2   3

table(titanic_test_pred, test_set1$Survived,dnn=c("Prediction"
,"Actual"))
          Actual
Prediction No Yes
       No   1   0
       Yes  2   3
>
> cm_titanic = confusionMatrix(test_set1$Survived, titanic_tes
t_pred)
>
> cm_titanic
Confusion Matrix and Statistics

          Reference
Prediction No Yes
       No   1   2
       Yes  0   3

               Accuracy : 0.6667
                 95% CI : (0.2228, 0.9567)
    No Information Rate : 0.8333
    P-Value [Acc > NIR] : 0.9377

                  Kappa : 0.3333

 Mcnemar's Test P-Value : 0.4795

            Sensitivity : 1.0000
            Specificity : 0.6000
```

```
         Pos Pred Value : 0.3333
         Neg Pred Value : 1.0000
             Prevalence : 0.1667
         Detection Rate : 0.1667
   Detection Prevalence : 0.5000
      Balanced Accuracy : 0.8000

       'Positive' Class : No
```