EXPERIMENT -1: Introduction:

QUESTION : 1.1 Data frames

AIM:-

To create a data frame for raw data and observe the subset of data after selecting particular rows and columns based on conditions.

MATHEMATICAL DEFINITION:-

Data frames are two-dimensional, heterogeneous data structures. They are lists of vectors of equal lengths. Data frames have the following constraints placed upon them:

1. A data-frame must have column-names and each row should have a unique name.
2. Each column should have the same number of items.
3. Each item in a single column should be of the same type.
4. Different columns can have different data types.

SYNTAX EXPLANATION:-

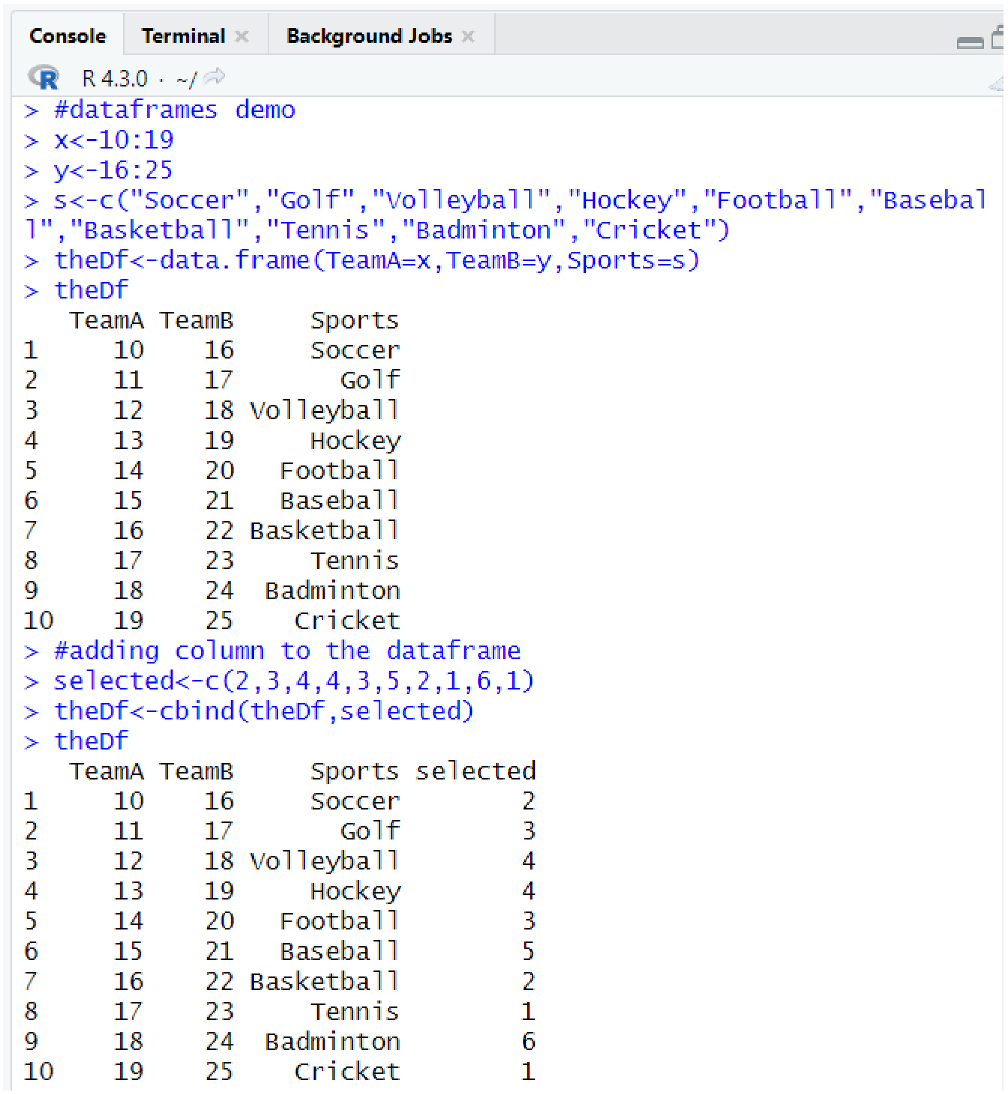
c()- combines data to form an array/list
data.frame(c1, c2…..cn)
rbind(nameofdf, rto be added)
cbind(nameofdf, cto be added)
subset(nameofdf, subset=condition)

PROGRAM:-

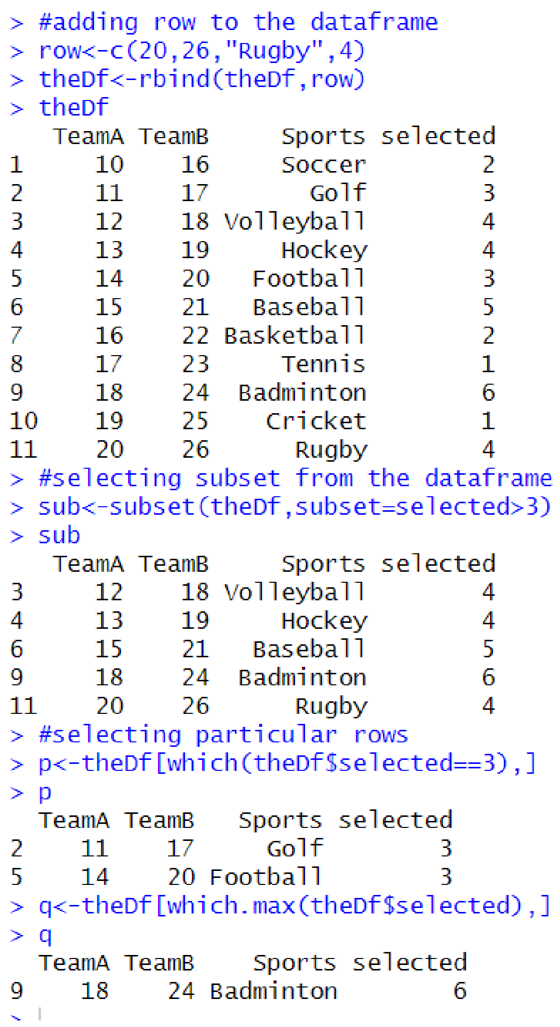
```
#dataframes demo
x<-10:19
y<-16:25
s<-c("Soccer","Golf","Volleyball","Hockey","Football","Baseball","Basketball","Tennis","Badmi
nton","Cricket")
theDf<-data.frame(TeamA=x,TeamB=y,Sports=s)
theDf
#adding column to the dataframe
selected<-c(2,3,4,4,3,5,2,1,6,1)
theDf<-cbind(theDf,selected)
theDf
#adding row to the dataframe
row<-c(20,26,"Rugby",4)
theDf<-rbind(theDf,row)
theDf
#selecting subset from the dataframe
sub<-subset(theDf,subset=selected>3)
sub
#selecting particular rows
```

```
p<-theDf[which(theDf$selected==3),]
p
q<-theDf[which.max(theDf$selected),]
q
```

OUTPUT:-



```
> #dataframes demo
> x<-10:19
> y<-16:25
> s<-c("Soccer","Golf","Volleyball","Hockey","Football","Basebal
l","Basketball","Tennis","Badminton","Cricket")
> theDf<-data.frame(TeamA=x,TeamB=y,Sports=s)
> theDf
   TeamA TeamB     Sports
1     10    16     Soccer
2     11    17       Golf
3     12    18 Volleyball
4     13    19     Hockey
5     14    20   Football
6     15    21   Baseball
7     16    22 Basketball
8     17    23     Tennis
9     18    24  Badminton
10    19    25    Cricket
> #adding column to the dataframe
> selected<-c(2,3,4,4,3,5,2,1,6,1)
> theDf<-cbind(theDf,selected)
> theDf
   TeamA TeamB     Sports selected
1     10    16     Soccer        2
2     11    17       Golf        3
3     12    18 Volleyball        4
4     13    19     Hockey        4
5     14    20   Football        3
6     15    21   Baseball        5
7     16    22 Basketball        2
8     17    23     Tennis        1
9     18    24  Badminton        6
10    19    25    Cricket        1
```



```
> #adding row to the dataframe
> row<-c(20,26,"Rugby",4)
> theDf<-rbind(theDf,row)
> theDf
   TeamA TeamB     Sports selected
1     10    16     Soccer        2
2     11    17       Golf        3
3     12    18 Volleyball        4
4     13    19     Hockey        4
5     14    20   Football        3
6     15    21   Baseball        5
7     16    22 Basketball        2
8     17    23     Tennis        1
9     18    24  Badminton        6
10    19    25    Cricket        1
11    20    26      Rugby        4
> #selecting subset from the dataframe
> sub<-subset(theDf,subset=selected>3)
> sub
   TeamA TeamB     Sports selected
3     12    18 Volleyball        4
4     13    19     Hockey        4
6     15    21   Baseball        5
9     18    24  Badminton        6
11    20    26      Rugby        4
> #selecting particular rows
> p<-theDf[which(theDf$selected==3),]
> p
  TeamA TeamB   Sports selected
2    11    17     Golf        3
5    14    20 Football        3
> q<-theDf[which.max(theDf$selected),]
> q
  TeamA TeamB    Sports selected
9    18    24 Badminton        6
>
```

CONCLUSION:-

->data frames were created using raw data and vectors for sports teams.

->particular rows and column were selected based on conditions.

QUESTION : 1.2 Vectors

AIM:-

To create a vector for raw data and apply basic arithmetic operations on it.

MATHEMATICAL DEFINITION:-

Vectors are single-dimensional, homogeneous data structures.

SYNTAX EXPLANATION:-

To create a vector, use the c() function.

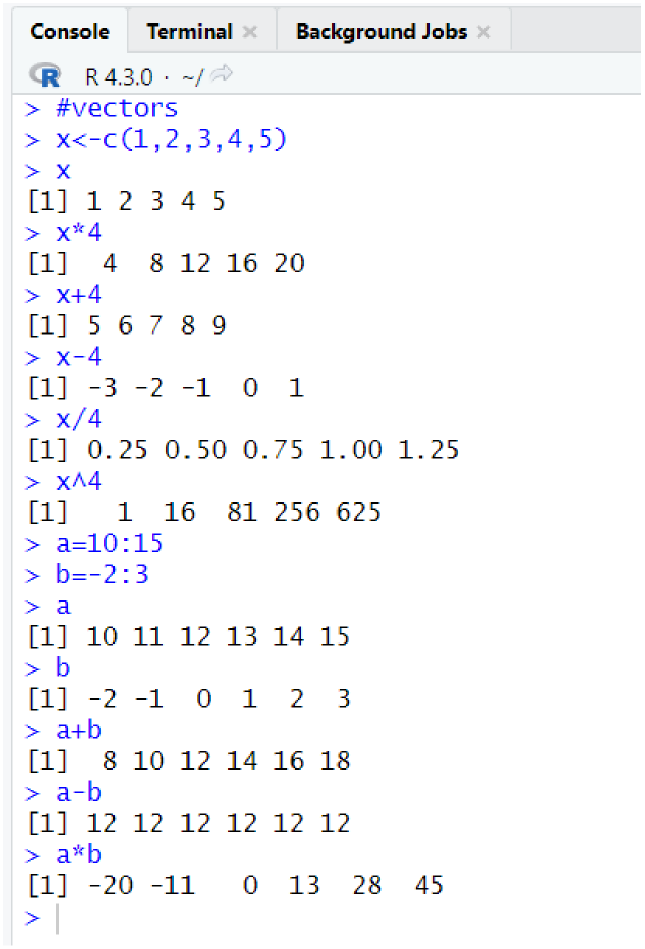
c()- combines data to form an array/list

PROGRAM:-

#vectors

```
x<-c(1,2,3,4,5)
x
x*4
x+4
x-4
x/4
x^4
a=10:15
b=-2:3
a
b
a+b
a-b
a*b
```

OUTPUT:-



```
> #vectors
> x<-c(1,2,3,4,5)
> x
[1] 1 2 3 4 5
> x*4
[1]  4  8 12 16 20
> x+4
[1] 5 6 7 8 9
> x-4
[1] -3 -2 -1  0  1
> x/4
[1] 0.25 0.50 0.75 1.00 1.25
> x^4
[1]   1  16  81 256 625
> a=10:15
> b=-2:3
> a
[1] 10 11 12 13 14 15
> b
[1] -2 -1  0  1  2  3
> a+b
[1]  8 10 12 14 16 18
> a-b
[1] 12 12 12 12 12 12
> a*b
[1] -20 -11   0  13  28  45
> 
```

CONCLUSION:-

-> Constructed a vector and performed basic +,-,*,^ ,/ arithmetic operations on it.

QUESTION : 1.3 Matrices

AIM:-

To create a matrix using the inbuilt function and check the number of rows, columns and dimensions of the matrix created.

MATHEMATICAL DEFINITION:-

Matrices are two-dimensional, homogeneous data structures. This means that all values in a matrix have to be of the same type.

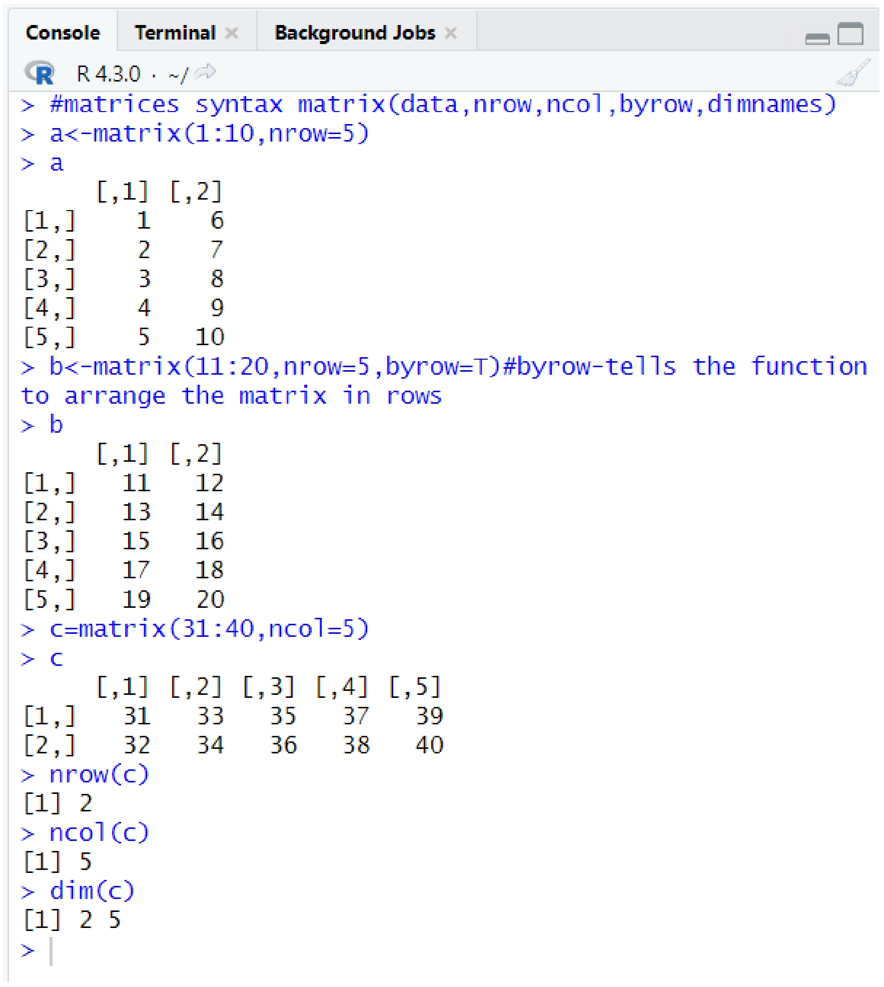
SYNTAX EXPLANATION:-

The basic syntax to create a matrix is: >matrix(data, nrow, ncol, byrow, dimnames)
nrow is the number of rows,
ncol is the number of columns,
byrow is a logical which tells the function to arrange the matrix row-wise, by default it is set to FALSE,
dimnames is a list of the names of the rows/columns created

PROGRAM:-

```
#matrices syntax matrix(data,nrow,ncol,byrow,dimnames)
a<-matrix(1:10,nrow=5)
a
b<-matrix(11:20,nrow=5,byrow=T)#byrow-tells the function to arrange the matrix in rows
b
c=matrix(31:40,ncol=5)
c
nrow(c)
ncol(c)
dim(c)
```

OUTPUT:-



```
> #matrices syntax matrix(data,nrow,ncol,byrow,dimnames)
> a<-matrix(1:10,nrow=5)
> a
     [,1] [,2]
[1,]    1    6
[2,]    2    7
[3,]    3    8
[4,]    4    9
[5,]    5   10
> b<-matrix(11:20,nrow=5,byrow=T)#byrow-tells the function
to arrange the matrix in rows
> b
     [,1] [,2]
[1,]   11   12
[2,]   13   14
[3,]   15   16
[4,]   17   18
[5,]   19   20
> c=matrix(31:40,ncol=5)
> c
     [,1] [,2] [,3] [,4] [,5]
[1,]   31   33   35   37   39
[2,]   32   34   36   38   40
> nrow(c)
[1] 2
> ncol(c)
[1] 5
> dim(c)
[1] 2 5
> 
```

CONCLUSION:-

-> Matrix for some raw data was created and the no. of rows, columns and dimensions were found.

EXPERIMENT -2. Graphs:

QUESTION : 2.1 Histogram

AIM:-

Draw a histogram and interpret intervals with maximum students.

MATHEMATICAL DEFINITION:-

Histogram is a graphical representation of continuous/discrete frequency distribution by means of rectangles called bars, whose width represents class interval and whose areas are proportional to the corresponding frequencies.

SYNTAX EXPLANATION:-

The basic syntax for creating a histogram using R is –
hist(v,main,xlab,xlim,ylim,breaks,col,border)
where,

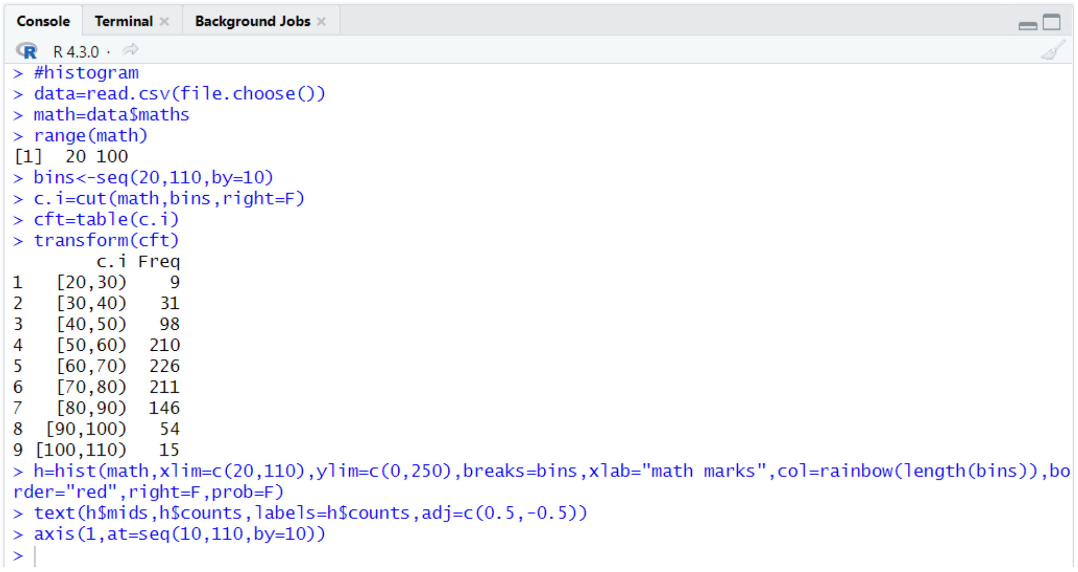
• v is a vector containing numeric values used in histogram.

- main indicates title of the chart.
- col is used to set color of the bars.
- border is used to set border color of each bar.
- xlab is used to give description of x-axis.
- xlim is used to specify the range of values on the x-axis.
- ylim is used to specify the range of values on the y-axis.
- breaks is used to mention the width of each bar.

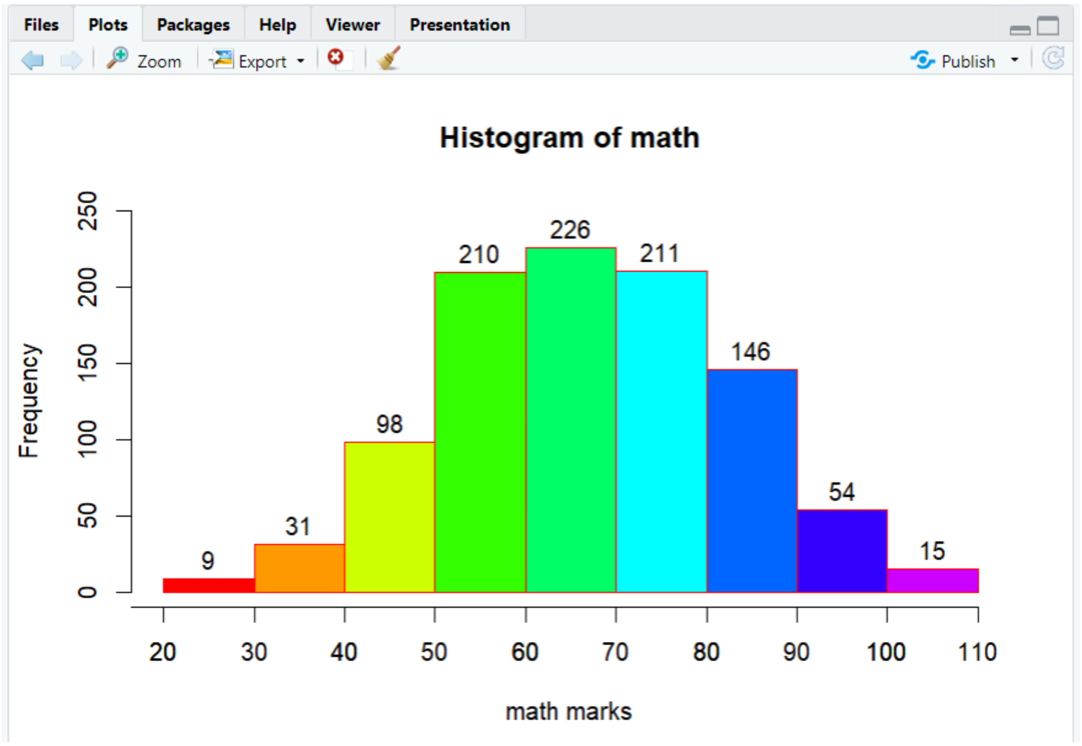
PROGRAM:-

```
#histogram
data=read.csv(file.choose())
math=data$maths
range(math)
bins<-seq(20,110,by=10)
c.i=cut(math,bins,right=F)
cft=table(c.i)
transform(cft)
h=hist(math,xlim=c(20,110),ylim=c(0,250),breaks=bins,xlab="math
marks",col=rainbow(length(bins)),border="red",right=F,prob=F)
text(h$mids,h$counts,labels=h$counts,adj=c(0.5,-0.5))
axis(1,at=seq(10,110,by=10))
```

OUTPUT:-



```
> #histogram
> data=read.csv(file.choose())
> math=data$maths
> range(math)
[1]  20 100
> bins<-seq(20,110,by=10)
> c.i=cut(math,bins,right=F)
> cft=table(c.i)
> transform(cft)
        c.i Freq
1   [20,30)    9
2   [30,40)   31
3   [40,50)   98
4   [50,60)  210
5   [60,70)  226
6   [70,80)  211
7   [80,90)  146
8  [90,100)   54
9 [100,110)   15
> h=hist(math,xlim=c(20,110),ylim=c(0,250),breaks=bins,xlab="math marks",col=rainbow(length(bins)),bo
rder="red",right=F,prob=F)
> text(h$mids,h$counts,labels=h$counts,adj=c(0.5,-0.5))
> axis(1,at=seq(10,110,by=10))
> 
```



CONCLUSION:-

->Hsitogram of students-marks is drawn.

-> On observing the histogram, it is concluded that maximum students got <insert interval>

QUESTION : 2.2 Piechart

AIM:-

To create a pie chart with imported data and interpret the grading.

MATHEMATICAL DEFINITION:-

A **pie chart** is a type of graph that represents the data in the circular graph. The slices of pie show the relative size of the data, and it is a type of pictorial representation of data. A pie chart requires a list of categorical variables and numerical variables. Here, the term "pie" represents the whole, and the "slices" represent the parts of the whole.

SYNTAX EXPLANATION:-

The basic syntax for creating a pie-chart using the R is – pie(x, labels, radius, main, col, clockwise) where,

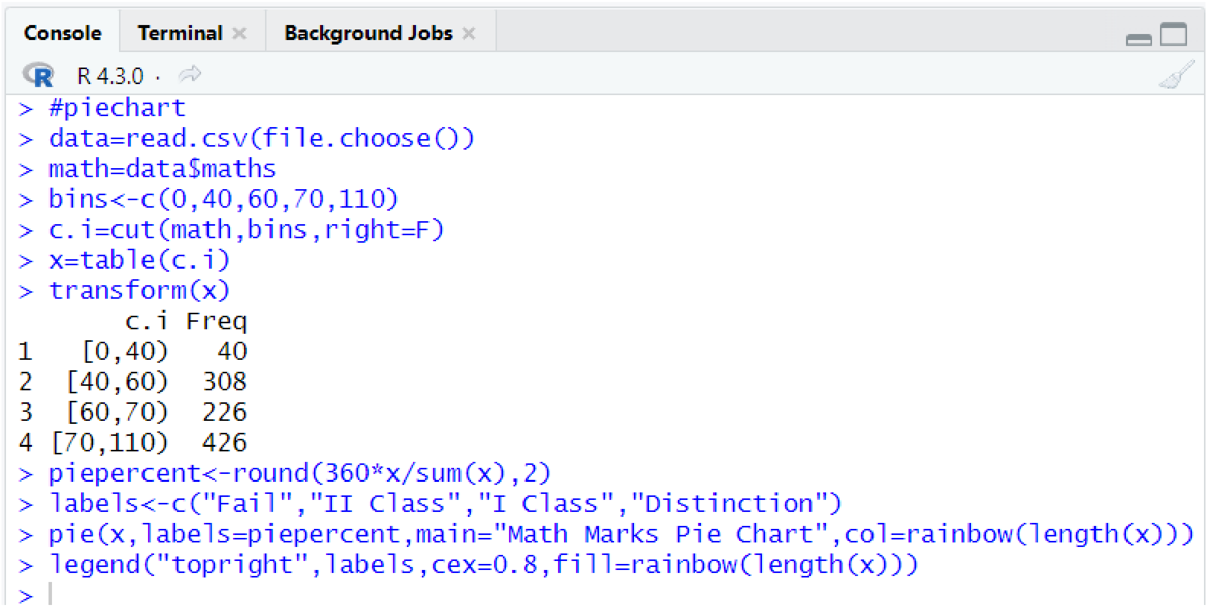
• x is a vector containing the numeric values used in the pie chart.
• labels is used to give description to the slices.
• radius indicates the radius of the circle of the pie chart.(value between −1 and +1).
• main indicates the title of the chart.
• col indicates the color palette.
• clockwise is a logical value indicating if the slices are drawn clockwise or anti clockwise.

PROGRAM:-

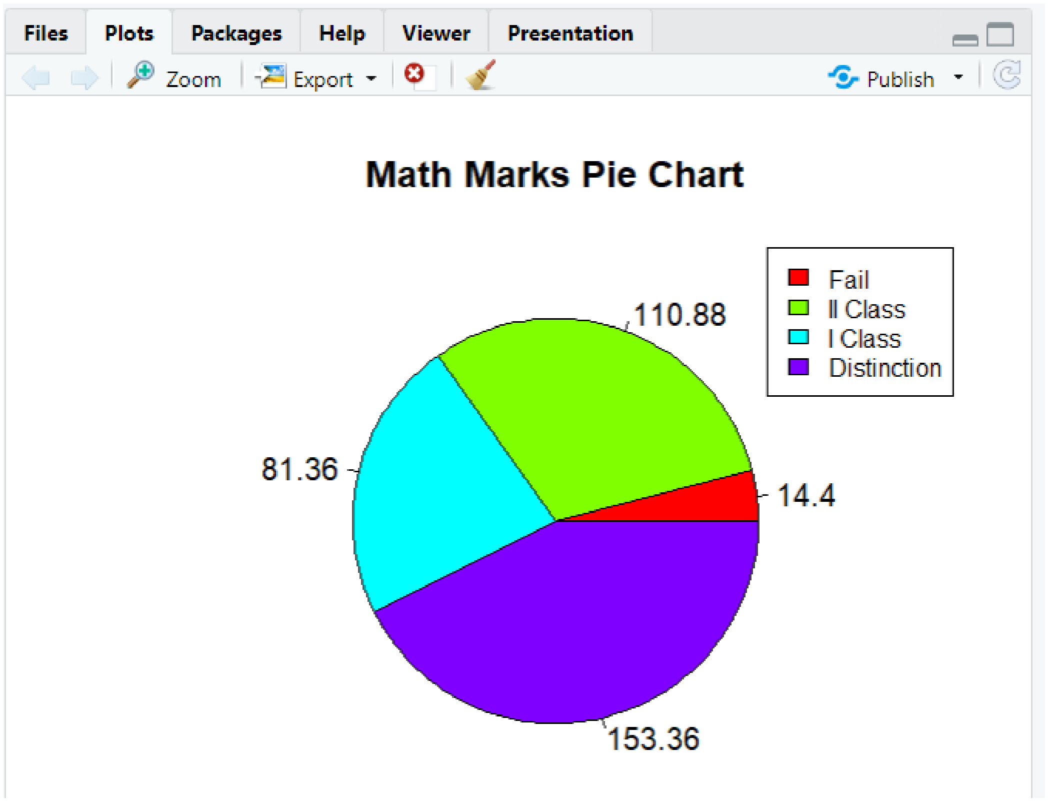
#piechart

```
data=read.csv(file.choose())
math=data$maths
bins<-c(0,40,60,70,110)
c.i=cut(math,bins,right=F)
x=table(c.i)
transform(x)
piepercent<-round(360*x/sum(x),2)
labels<-c("Fail","II Class","I Class","Distinction")
pie(x,labels=piepercent,main="Math Marks Pie Chart",col=rainbow(length(x)))
legend("topright",labels,cex=0.8,fill=rainbow(length(x)))
```

OUTPUT:-



```
> #piechart
> data=read.csv(file.choose())
> math=data$maths
> bins<-c(0,40,60,70,110)
> c.i=cut(math,bins,right=F)
> x=table(c.i)
> transform(x)
       c.i Freq
1    [0,40)   40
2   [40,60)  308
3   [60,70)  226
4 [70,110)  426
> piepercent<-round(360*x/sum(x),2)
> labels<-c("Fail","II Class","I Class","Distinction")
> pie(x,labels=piepercent,main="Math Marks Pie Chart",col=rainbow(length(x)))
> legend("topright",labels,cex=0.8,fill=rainbow(length(x)))
> 
```



CONCLUSION:-

->Pie diagram is plotted

->Grading for the students data is performed

QUESTION : **2.3 3d piechart**

AIM:-

Draw 3D pie chart for students-marks data and interpret the grading.

MATHEMATICAL DEFINITION:-

A pie chart is a graphical representation of data ina circular-shaped graph. A pie chart shows how a total amount is divided between levels of a categorical variable as a circle divided into radial slices. 3D pie chart is one type of pie char visualised in three dimensions.

SYNTAX EXPLANATION:-

pie3D(x, labels, radius, main, col, clockwise) where,

• x is a vector containing the numeric values used in the pie chart.

• labels is used to give description to the slices.

• radius indicates the radius of the circle of the pie chart.(value between −1 and +1).

• main indicates the title of the chart.

• col indicates the color palette.

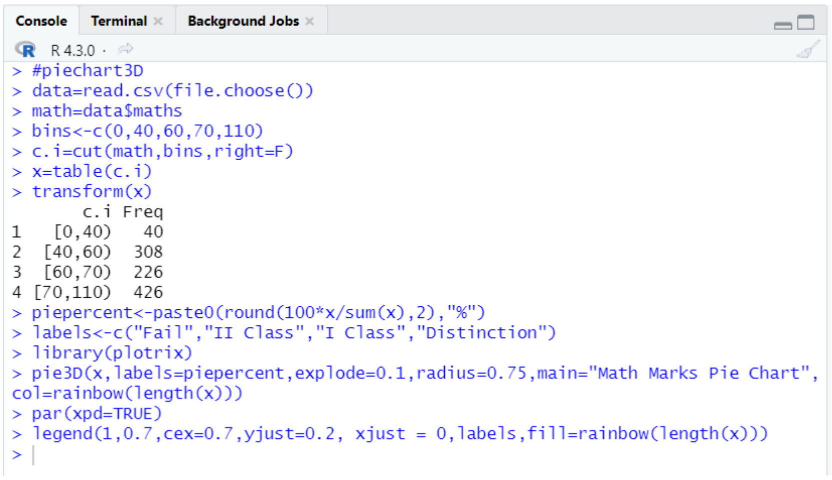
• clockwise is a logical value indicating if the slices are drawn clockwise or anti clockwise.
• explode is used to separate the slices.

PROGRAM:-

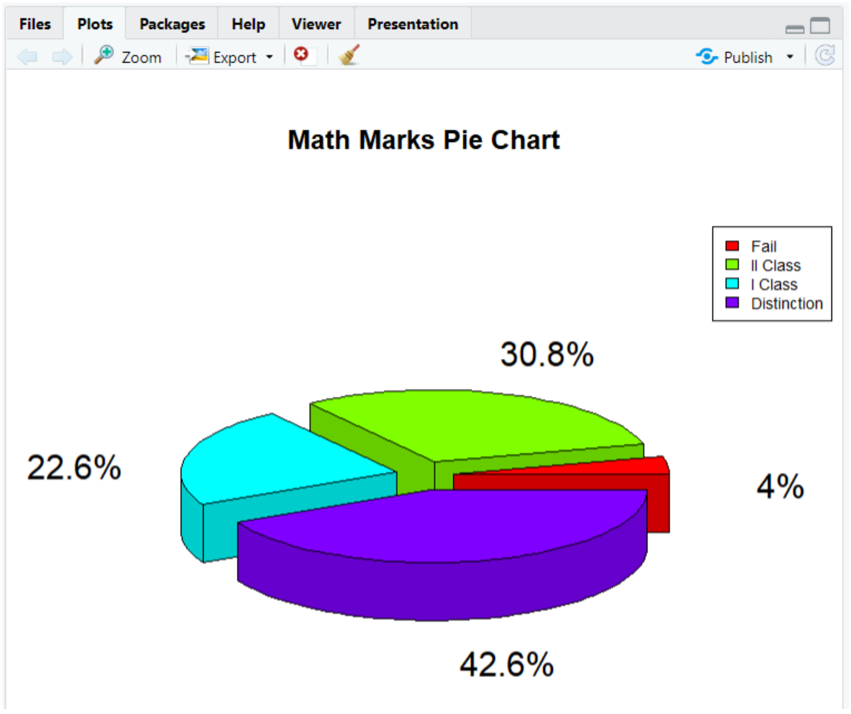
#piechart3D

```
data=read.csv(file.choose())
math=data$maths
bins<-c(0,40,60,70,110)
c.i=cut(math,bins,right=F)
x=table(c.i)
transform(x)
piepercent<-paste0(round(100*x/sum(x),2),"%")
labels<-c("Fail","II Class","I Class","Distinction")
install.packages("plotrix")
library(plotrix)
pie3D(x,labels=piepercent,explode=0.1,radius=0.75,main="Math Marks Pie
Chart",col=rainbow(length(x)))
par(xpd=TRUE)
legend(1,0.7,cex=0.7,yjust=0.2, xjust = -0.1,labels,fill=rainbow(length(x)))
```

OUTPUT:-



```
> #piechart3D
> data=read.csv(file.choose())
> math=data$maths
> bins<-c(0,40,60,70,110)
> c.i=cut(math,bins,right=F)
> x=table(c.i)
> transform(x)
       c.i Freq
1   [0,40)   40
2  [40,60)  308
3  [60,70)  226
4 [70,110)  426
> piepercent<-paste0(round(100*x/sum(x),2),"%")
> labels<-c("Fail","II Class","I Class","Distinction")
> library(plotrix)
> pie3D(x,labels=piepercent,explode=0.1,radius=0.75,main="Math Marks Pie Chart",
col=rainbow(length(x)))
> par(xpd=TRUE)
> legend(1,0.7,cex=0.7,yjust=0.2, xjust = 0,labels,fill=rainbow(length(x)))
> |
```



CONCLUSION:-
->3D pie diagram is plotted
->Grading for the students data is performed

QUESTION : **2.4 bar plot**

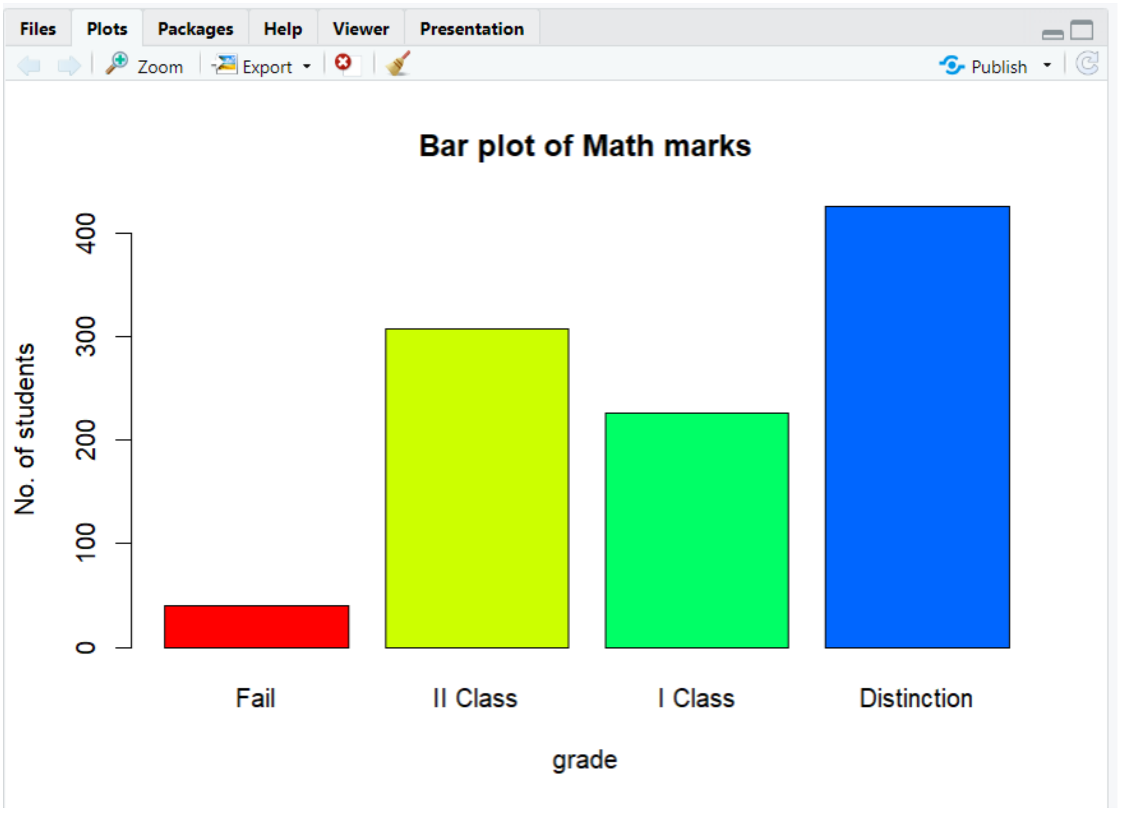
AIM:-
To draw a grouped bar chart for marks of students.

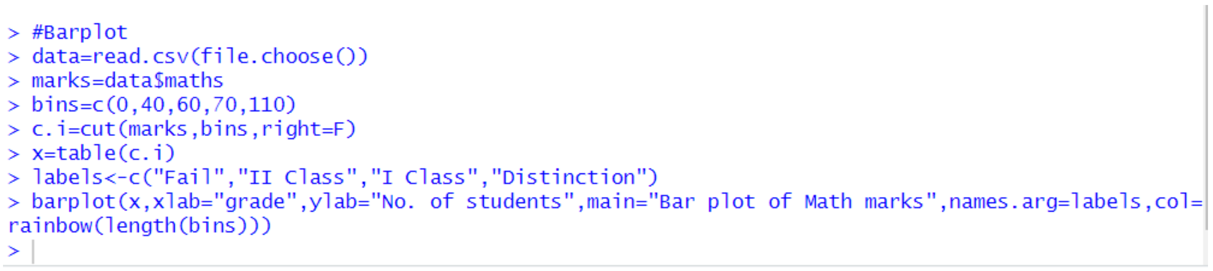
MATHEMATICAL DEFINITION:-
A bar chart or bar graph presents data with rectangular bars with heights or lengths proportional to the values they represent. The bars can be vertical or horizontal. Grouped and stacked bar charts are used to present more complex comparisons of data. In grouped bar charts, there are two or more bars for each categorical group, with the bars being color-coded to depict a particular segment. In stacked bar cjart, bars representing different groups or segments stack on top of each other, with the height of the resulting bar showing the aggregate result of these groups
SYNTAX EXPLANATION:-
The basic syntax to create a bar-chart in R is – barplot(H, xlab, ylab, main, names.arg, col) where
• H is a vector or matrix containing numeric values used in bar chart.
• xlab is the label for x axis.
• ylab is the label for y axis.
• main is the title of the bar chart.
• names.arg is a vector of names appearing under each bar.
• col is used to give colors to the bars in the graph.

PROGRAM:-

#Barplot
data=read.csv(file.choose())
marks=data$maths
bins=c(0,40,60,70,110)
c.i=cut(marks,bins,right=F)
x=table(c.i)
labels<-c("Fail","II Class","I Class","Distinction")
barplot(x,xlab="grade",ylab="No. of students",main="Bar plot of Math marks",names.arg=labels,col=rainbow(length(bins)))

OUTPUT:-





```
> #Barplot
> data=read.csv(file.choose())
> marks=data$maths
> bins=c(0,40,60,70,110)
> c.i=cut(marks,bins,right=F)
> x=table(c.i)
> labels<-c("Fail","II Class","I Class","Distinction")
> barplot(x,xlab="grade",ylab="No. of students",main="Bar plot of Math marks",names.arg=labels,col=
rainbow(length(bins)))
> |
```

CONCLUSION:-
->Bar plot is plotted for the imported data

->Distribution of marks for the students data is performed

QUESTION : <u>2.5 box plot</u>

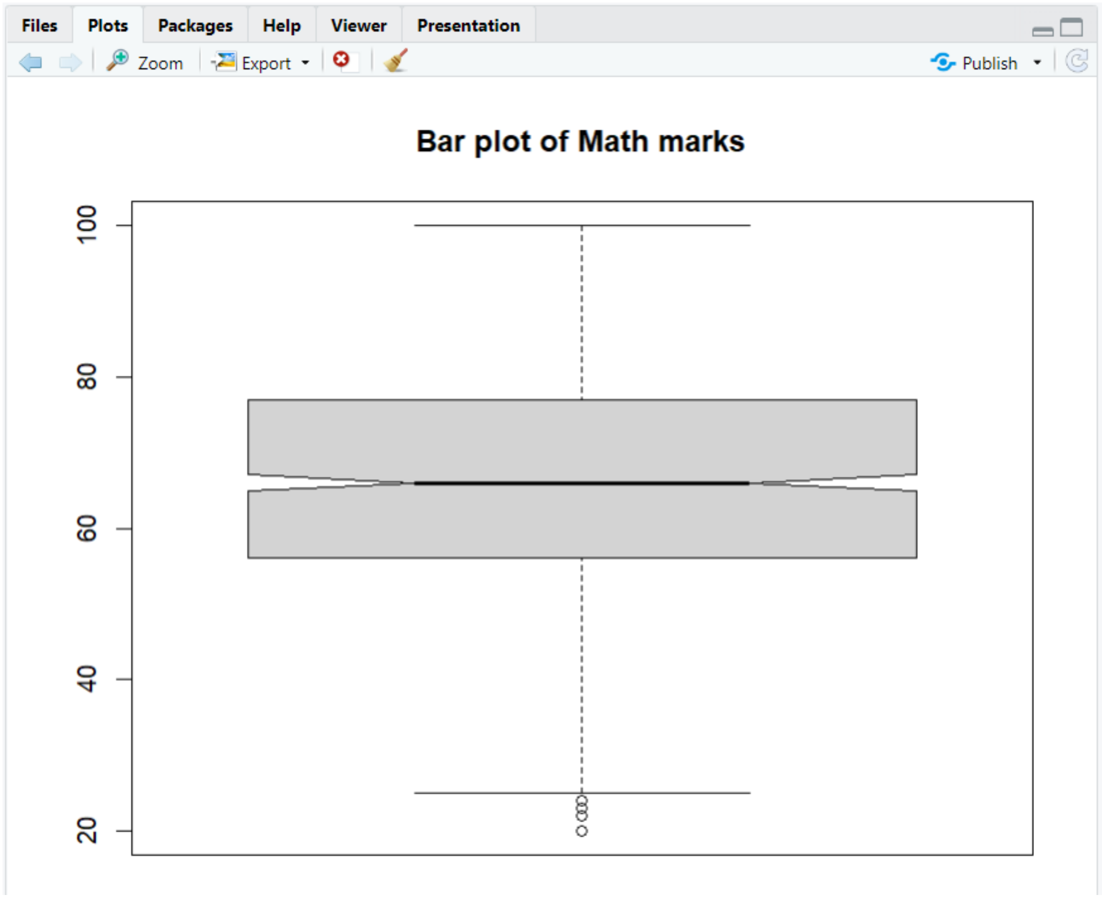
AIM:-
To create a box plot for the imported marks and find out the summary of the marks.

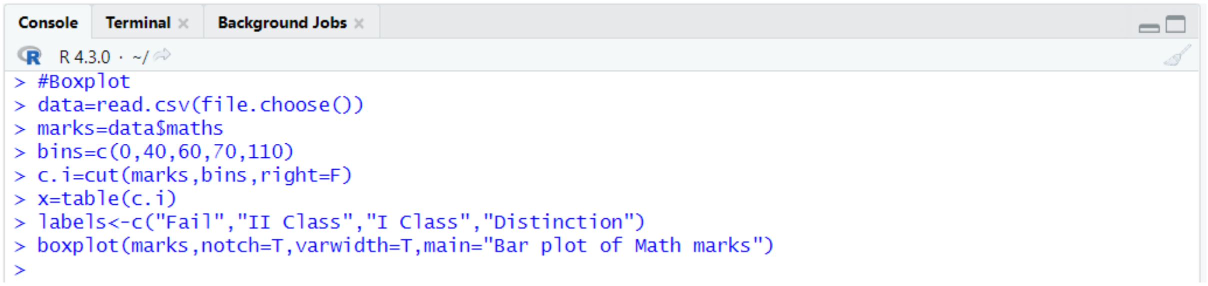
MATHEMATICAL DEFINITION:-
A box and whisker plot is a graph that exhibits data from a five-number summary, including one of the measures of central tendency. It does not display the distribution as accurately as a stem and leaf plot or histogram does. But, it is principally used to show whether a distribution is skewed or not and if there are potential unusual observations present in the data set, which are also called outliers. Boxplots are also very useful when huge numbers of data collections are involved or compared.

SYNTAX EXPLANATION:-
The basic syntax to create a boxplot in R is – boxplot(x, data, notch, varwidth, names, main)
where
• x is a vector or a formula.
• data is the data frame.
• notch is a logical value. Set as TRUE to draw a notch.
• varwidth is a logical value. Set as true to draw width of the box proportionate to the sample size.
• names are the group labels which will be printed under each boxplot.
• main is used to give a title to the graph.

PROGRAM:-
```
#Boxplot
data=read.csv(file.choose())
marks=data$maths
bins=c(0,40,60,70,110)
c.i=cut(marks,bins,right=F)
x=table(c.i)
labels<-c("Fail","II Class","I Class","Distinction")
boxplot(marks,notch=T,varwidth=T,main="Bar plot of Math marks")
```

OUTPUT:-





```
> #Boxplot
> data=read.csv(file.choose())
> marks=data$maths
> bins=c(0,40,60,70,110)
> c.i=cut(marks,bins,right=F)
> x=table(c.i)
> labels<-c("Fail","II Class","I Class","Distinction")
> boxplot(marks,notch=T,varwidth=T,main="Bar plot of Math marks")
>
```

<u>CONCLUSION:-</u>

->Box plot for the imported marks was drawn

->Summary of the marks i.e. the mean, first quartile, median, mean, third quartile, and maximum marks was found and displayed.

<u>EXPERIMENT -3. Measures of central tendency</u>

QUESTION : <u>3.1 Mean,median,mode - raw data</u>

<u>AIM:-</u>

To find mean median mode for raw data.

<u>MATHEMATICAL DEFINITION:-</u>

The mean (average) of a data set is found by adding all numbers in the data set and then dividing by the number of values in the set. The median is the middle value when a data set is ordered from least to greatest. The mode is the number that occurs most often in a data set.

(i) mean = $\bar{x} = \sum xi/N$, where N is total no. of observations.

SYNTAX EXPLANATION:-

mean(x, trim = 0, na.rm = FALSE) where,

• x is the input vector. • trim is used to drop some observations from both end of the sorted vector.

• na.rm is used to remove the missing values from the input vector.

median(x, na.rm = FALSE) where,

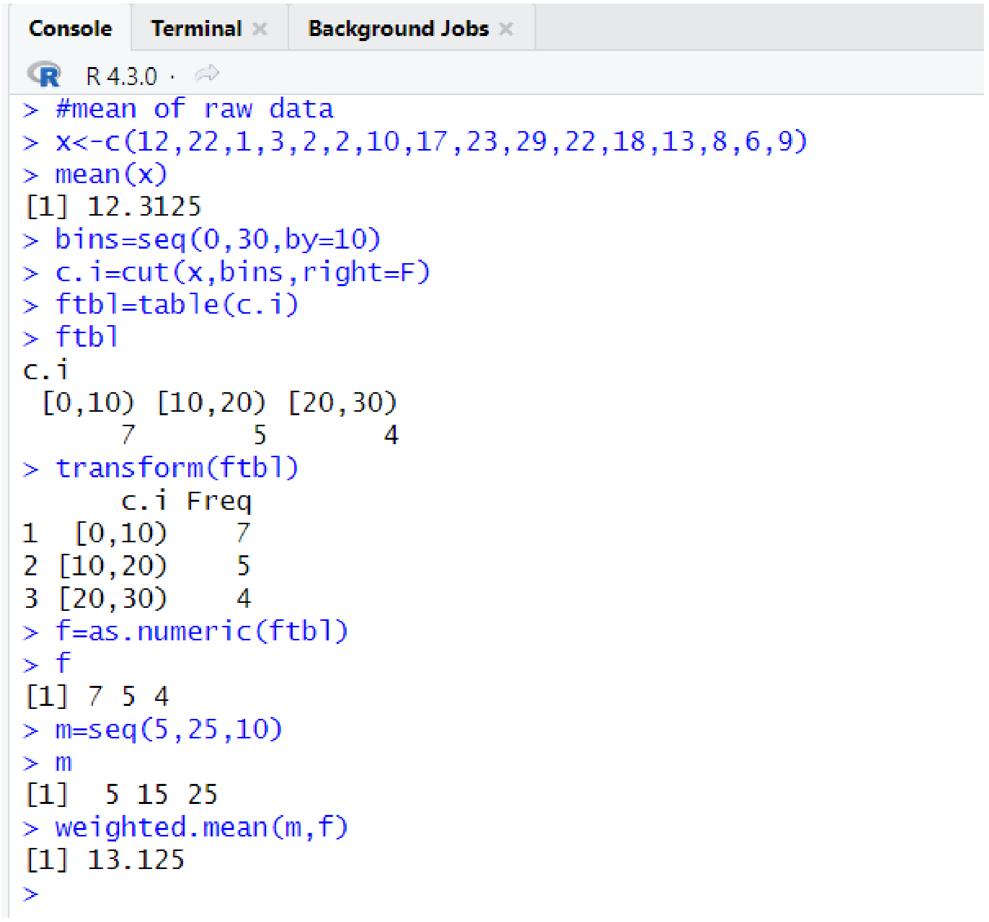
• x is the input vector.

• na.rm is used to remove the missing values from the input vector.

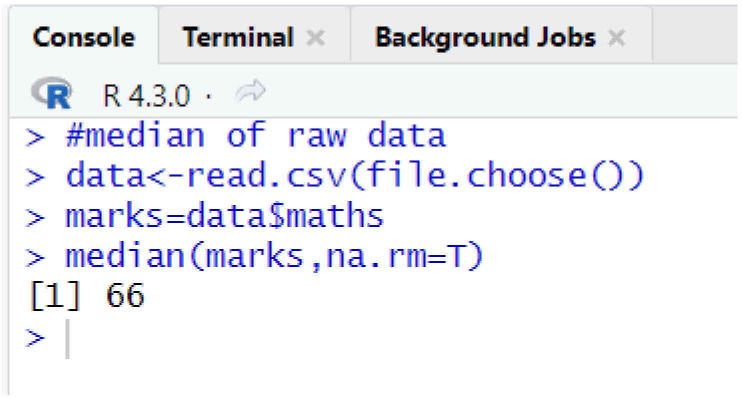
PROGRAM:-

```
#mean of raw data
x<-c(12,22,1,3,2,2,10,17,23,29,22,18,13,8,6,9)
mean(x)
bins=seq(0,30,by=10)
c.i=cut(x,bins,right=F)
ftbl=table(c.i)
ftbl
transform(ftbl)
f=as.numeric(ftbl)
f
m=seq(5,25,10)
m
weighted.mean(m,f)
#median of raw data(DOUBT!!!!)
data<-read.csv(file.choose())
marks=data$maths
median(marks,na.rm=T)
#mode of raw data
data=read.csv(file.choose())
marks=data$maths
y=table(marks)
y
max(y)
names(y)
mode=names(y)[y==max(y)]
mode
```

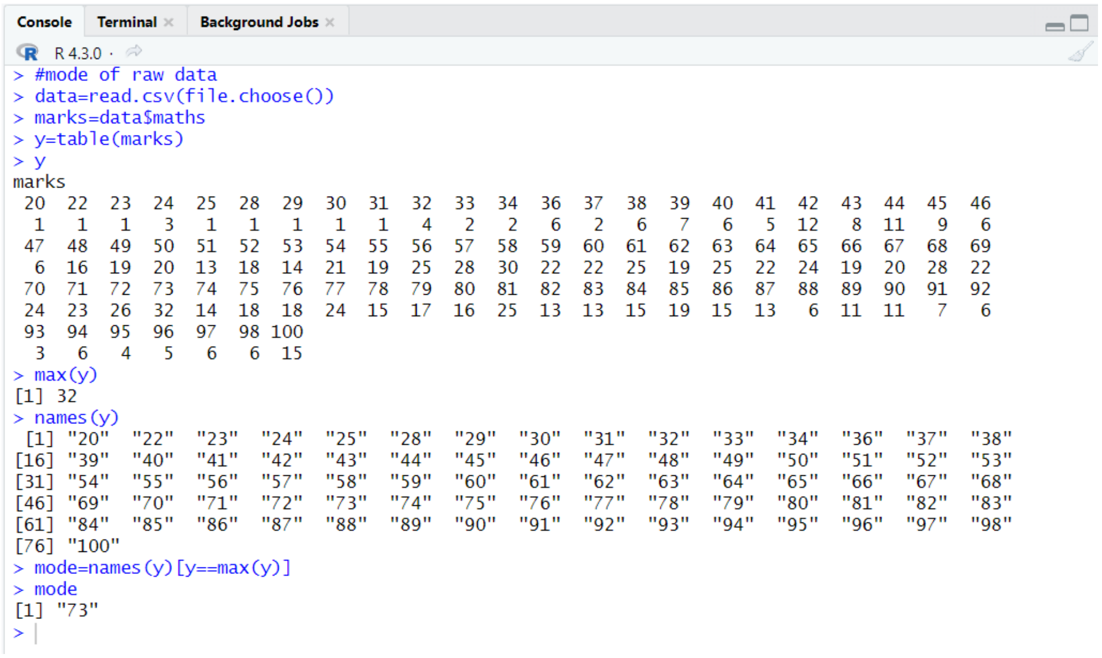
OUTPUT:-



```
> #mean of raw data
> x<-c(12,22,1,3,2,2,10,17,23,29,22,18,13,8,6,9)
> mean(x)
[1] 12.3125
> bins=seq(0,30,by=10)
> c.i=cut(x,bins,right=F)
> ftbl=table(c.i)
> ftbl
c.i
 [0,10) [10,20) [20,30) 
      7       5       4 
> transform(ftbl)
      c.i Freq
1  [0,10)    7
2 [10,20)    5
3 [20,30)    4
> f=as.numeric(ftbl)
> f
[1] 7 5 4
> m=seq(5,25,10)
> m
[1]  5 15 25
> weighted.mean(m,f)
[1] 13.125
> 
```



```
> #median of raw data
> data<-read.csv(file.choose())
> marks=data$maths
> median(marks,na.rm=T)
[1] 66
> 
```



```
> #mode of raw data
> data=read.csv(file.choose())
> marks=data$maths
> y=table(marks)
> y
marks
 20  22  23  24  25  28  29  30  31  32  33  34  36  37  38  39  40  41  42  43  44  45  46 
  1   1   1   3   1   1   1   1   1   4   2   2   6   2   6   7   6   5  12   8  11   9   6 
 47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69 
  6  16  19  20  13  18  14  21  19  25  28  30  22  22  25  19  25  22  24  19  20  28  22 
 70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92 
 24  23  26  32  14  18  18  24  15  17  16  25  13  13  15  19  15  13   6  11  11   7   6 
 93  94  95  96  97  98 100 
  3   6   4   5   6   6  15 
> max(y)
[1] 32
> names(y)
 [1] "20"  "22"  "23"  "24"  "25"  "28"  "29"  "30"  "31"  "32"  "33"  "34"  "36"  "37"  "38" 
[16] "39"  "40"  "41"  "42"  "43"  "44"  "45"  "46"  "47"  "48"  "49"  "50"  "51"  "52"  "53" 
[31] "54"  "55"  "56"  "57"  "58"  "59"  "60"  "61"  "62"  "63"  "64"  "65"  "66"  "67"  "68" 
[46] "69"  "70"  "71"  "72"  "73"  "74"  "75"  "76"  "77"  "78"  "79"  "80"  "81"  "82"  "83" 
[61] "84"  "85"  "86"  "87"  "88"  "89"  "90"  "91"  "92"  "93"  "94"  "95"  "96"  "97"  "98" 
[76] "100"
> mode=names(y)[y==max(y)]
> mode
[1] "73"
> 
```

CONCLUSION:-

-> mean median mode of raw data was found using inbuilt functions.

QUESTION : **3.2 Mean,median,mode - grouped data**

AIM:-

To find mean median mode of grouped data, using grouped data.

MATHEMATICAL DEFINITION:-

(ii) mean of grouped data = $\bar{x} = \sum f_i x_i / \sum f_i$, where *fi* is frequency and *xi* is mid value of class interval.

Median of grouped data is median=l+(N/2-c)*(h/f)

Mode of grouped data is mode=l+h*((f1-f0)/(2*f1-f0-f2))

SYNTAX EXPLANATION:-

read.csv(file.choose()

range()

seq()

cut()

table()

transform()

as.numeric()

as.data.frame()

cumsum()

which()

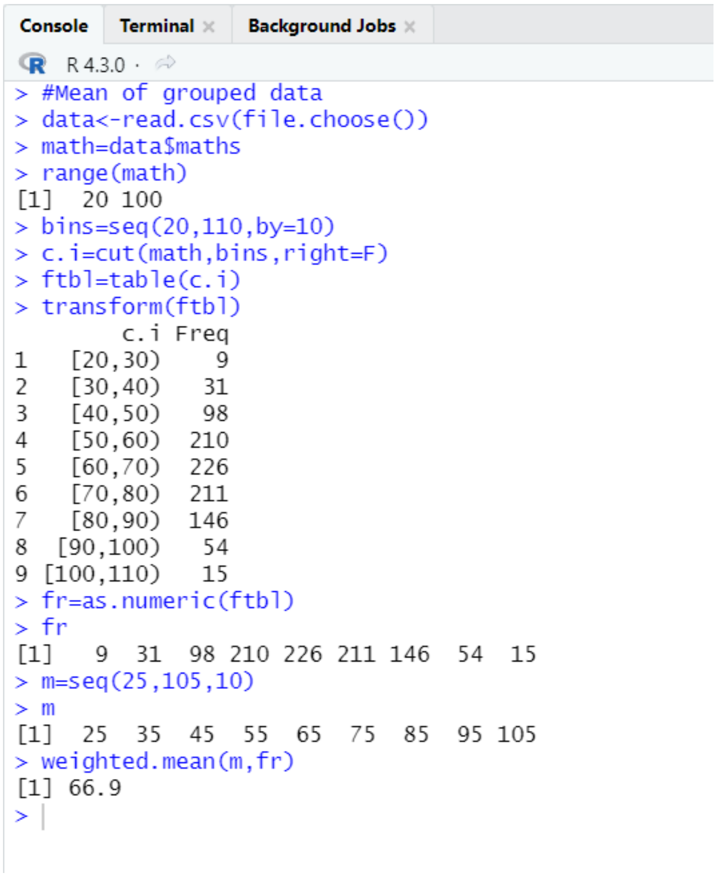
max()

PROGRAM:-

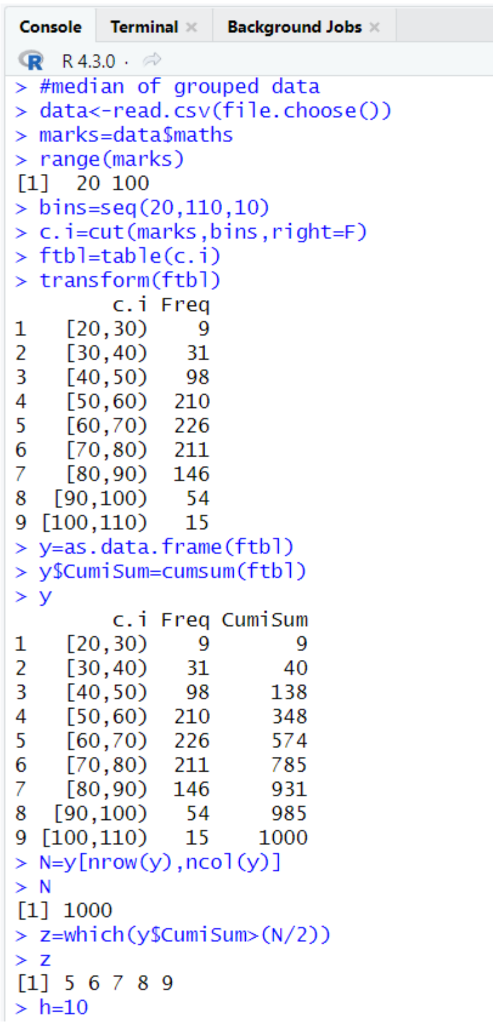
```
#Mean of grouped data
data<-read.csv(file.choose())
math=data$maths
range(math)
bins=seq(20,110,by=10)
c.i=cut(math,bins,right=F)
ftbl=table(c.i)
transform(ftbl)
fr=as.numeric(ftbl)
fr
m=seq(25,105,10)
m
weighted.mean(m,fr)
#median of grouped data
data<-read.csv(file.choose())
marks=data$maths
range(marks)
bins=seq(20,110,10)
c.i=cut(marks,bins,right=F)
ftbl=table(c.i)
ftbl
transform(ftbl)
y=as.data.frame(ftbl)
y$CumiSum=cumsum(ftbl)
y
N=y[nrow(y),ncol(y)]
N
z=which(y$CumiSum>(N/2))
z
h=10
l=(z[1]+1)*h
l
f=y[z[1],2]
f
c=y[z[1]-1,3]
c
median=l+(N/2-c)*(h/f)
median
#mode of grouped data
data=read.csv(file.choose())
marks=data$maths
range(marks,na.rm=T)
bins=seq(20,110,10)
```

```
c.i=cut(marks,bins,right=F)
ftbl=table(c.i)
y=as.data.frame(ftbl)
y
z=which.max(y$Freq)
z
h=10
l=(z+1)*h
l
f1=y[z,2]
f1
f0=y[z-1,2]
f0
f2=y[z+1,2]
f2
mode=l+h*((f1-f0)/(2*f1-f0-f2))
mode
```

OUTPUT:-



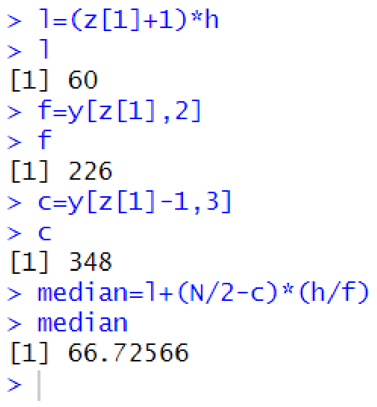
```
> #Mean of grouped data
> data<-read.csv(file.choose())
> math=data$maths
> range(math)
[1]  20 100
> bins=seq(20,110,by=10)
> c.i=cut(math,bins,right=F)
> ftbl=table(c.i)
> transform(ftbl)
        c.i Freq
1   [20,30)    9
2   [30,40)   31
3   [40,50)   98
4   [50,60)  210
5   [60,70)  226
6   [70,80)  211
7   [80,90)  146
8  [90,100)   54
9 [100,110)   15
> fr=as.numeric(ftbl)
> fr
[1]   9  31  98 210 226 211 146  54  15
> m=seq(25,105,10)
> m
[1]  25  35  45  55  65  75  85  95 105
> weighted.mean(m,fr)
[1] 66.9
> 
```



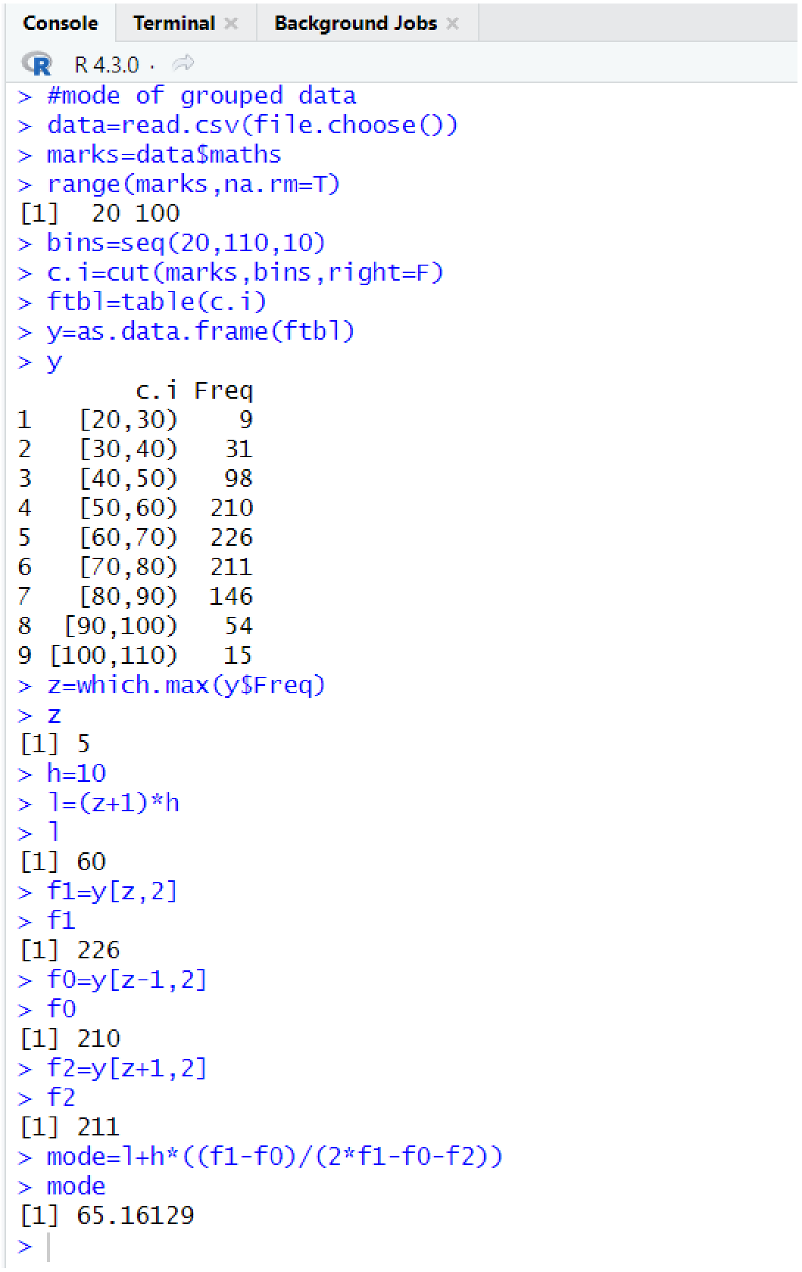
Console Terminal Background Jobs

R 4.3.0 ·

```
> #median of grouped data
> data<-read.csv(file.choose())
> marks=data$maths
> range(marks)
[1]  20 100
> bins=seq(20,110,10)
> c.i=cut(marks,bins,right=F)
> ftbl=table(c.i)
> transform(ftbl)
        c.i Freq
1   [20,30)    9
2   [30,40)   31
3   [40,50)   98
4   [50,60)  210
5   [60,70)  226
6   [70,80)  211
7   [80,90)  146
8  [90,100)   54
9 [100,110)   15
> y=as.data.frame(ftbl)
> y$CumiSum=cumsum(ftbl)
> y
        c.i Freq CumiSum
1   [20,30)    9       9
2   [30,40)   31      40
3   [40,50)   98     138
4   [50,60)  210     348
5   [60,70)  226     574
6   [70,80)  211     785
7   [80,90)  146     931
8  [90,100)   54     985
9 [100,110)   15    1000
> N=y[nrow(y),ncol(y)]
> N
[1] 1000
> z=which(y$CumiSum>(N/2))
> z
[1] 5 6 7 8 9
> h=10
```



```
> l=(z[1]+1)*h
> l
[1] 60
> f=y[z[1],2]
> f
[1] 226
> c=y[z[1]-1,3]
> c
[1] 348
> median=l+(N/2-c)*(h/f)
> median
[1] 66.72566
> 
```



```
> #mode of grouped data
> data=read.csv(file.choose())
> marks=data$maths
> range(marks,na.rm=T)
[1]  20 100
> bins=seq(20,110,10)
> c.i=cut(marks,bins,right=F)
> ftbl=table(c.i)
> y=as.data.frame(ftbl)
> y
        c.i Freq
1   [20,30)    9
2   [30,40)   31
3   [40,50)   98
4   [50,60)  210
5   [60,70)  226
6   [70,80)  211
7   [80,90)  146
8  [90,100)   54
9 [100,110)   15
> z=which.max(y$Freq)
> z
[1] 5
> h=10
> l=(z+1)*h
> l
[1] 60
> f1=y[z,2]
> f1
[1] 226
> f0=y[z-1,2]
> f0
[1] 210
> f2=y[z+1,2]
> f2
[1] 211
> mode=l+h*((f1-f0)/(2*f1-f0-f2))
> mode
[1] 65.16129
> 
```

CONCLUSION:-

-> mean median mode of raw data was found using formulas.

EXPERIMENT 4: MEASURES OF DISPERSION

EXPERIMENT-4.1:

AIM: To find the variance and standard deviation of raw data.

MATHEMATICAL DEFINITION: Variance and Standard deviation are the two important topics in Statistics. It is the measure of the dispersion of statistical data. Dispersion is the extent to which values in a distribution differ from the average of the distribution.**Standard Deviation** is a measure which shows how much variation (such as spread, dispersion) from the mean exists.**Variance** is the measure of how notably a collection of data is spread out. If all the data values are identical, then it indicates the variance is zero.

-standard deviation of raw data = (FORMULA and term explanation)

- Variance of raw data =(standard deviation)^2

SYNTAX EXPLANATION:

The syntax for finding the variance and standard deviation is given by:

Standard deviation-sd(x)

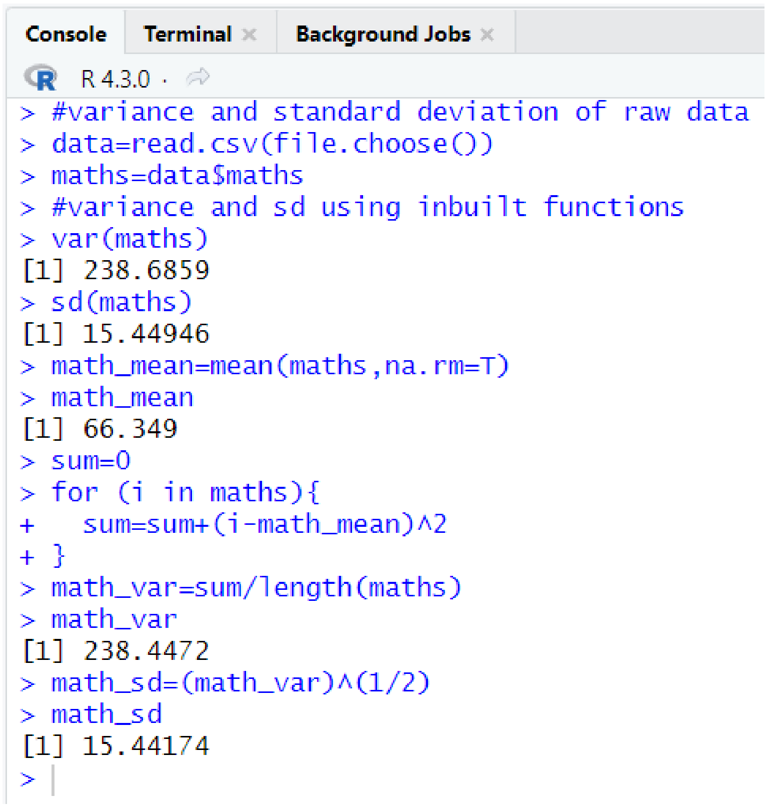
Variance - var(x)

Where , x is the numeric vector

PROGRAM:

```
#variance and standard deviation of raw data
data=read.csv(file.choose())
maths=data$maths
#variance and sd using inbuilt functions
var(maths)
sd(maths)
math_mean=mean(maths,na.rm=T)
math_mean
sum=0
for (i in maths){
  sum=sum+(i-math_mean)^2
}
math_var=sum/length(maths)
math_var
math_sd=(math_var)^(1/2)
math_sd
```

OUTPUT:



```
> #variance and standard deviation of raw data
> data=read.csv(file.choose())
> maths=data$maths
> #variance and sd using inbuilt functions
> var(maths)
[1] 238.6859
> sd(maths)
[1] 15.44946
> math_mean=mean(maths,na.rm=T)
> math_mean
[1] 66.349
> sum=0
> for (i in maths){
+   sum=sum+(i-math_mean)^2
+ }
> math_var=sum/length(maths)
> math_var
[1] 238.4472
> math_sd=(math_var)^(1/2)
> math_sd
[1] 15.44174
> 
```

CONCLUSION:

- The standard deviation and variance for the given data is found using inbuilt formulas ……..!!!(more info to be added)

EXPERIMENT-4.2:

AIM: To find the variance and standard deviation of grouped data.

MATHEMATICAL DEFINITION:

The standard deviation is calculated as the square root of variance. Standard deviation helps us understand the degree of deviation of value from the mean value of the data set.
Standard deviation of grouped data – (FORMULA and explain terms)
Variance = (standard deviation)^2

SYNTAX EXPLANATION:

- read.csv(file.choose()) - The function read.csv() is used to import data from a csv file. file.choose() method to select a csv file to load in R.
- range()-The range() function in R is used to return a vector with two elements: The first element represents the minimum value of the input vector. The second element is the maximum value of the input vector.
- seq()-seq () is an operator in R for creating a sequence of numbers. syntax — seq(from, to, by,length.out) where, from — the starting number in the sequence to — the last number in the sequence, by — difference between the numbers in the sequence (optional), length.out — maximum length of the vector (optional)
- cut()-The cut function is used in R for cutting a numeric value into bins of continuous values and is specified with cut labels. Syntax: cut (x, breaks, labels, right) where, x: numeric input value

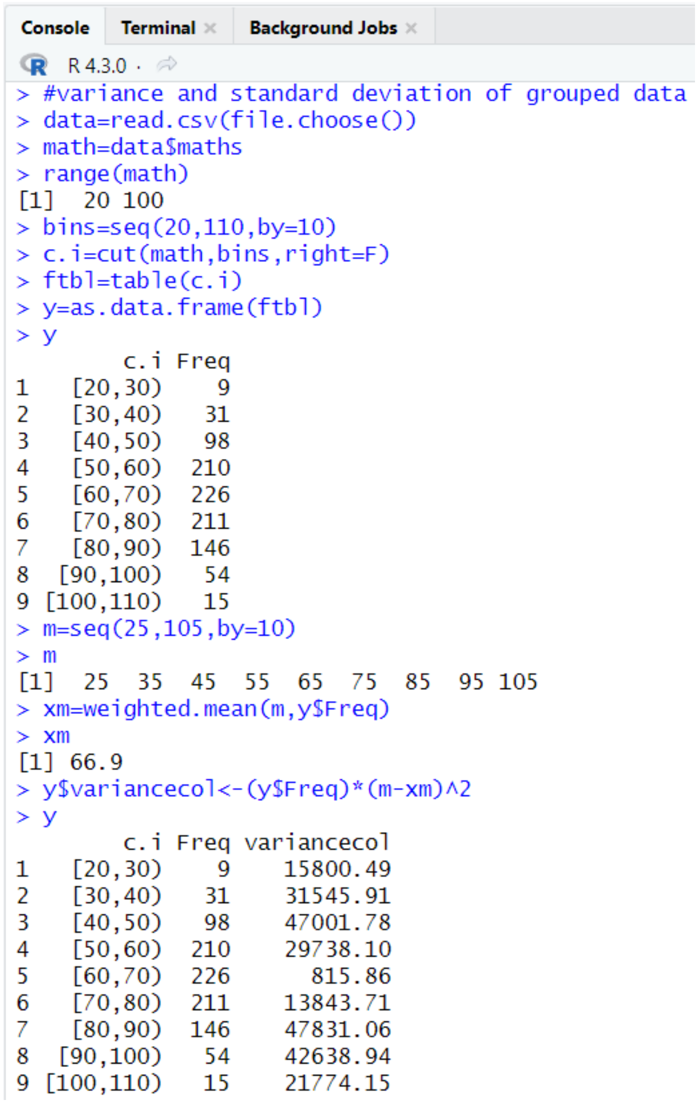
(vector) Break : No. of breaks points Labels: labels to each group/bin. Right: interval should be closed on the right and open on the left or vice versa.

- table()-Table function (table())in R performs a tabulation of categorical variable and gives its frequency as output
- as.data.frame()-it converts a table into dataframe
- weighted.mean()-is used to compute the weighted arithmetic mean of input vector values.

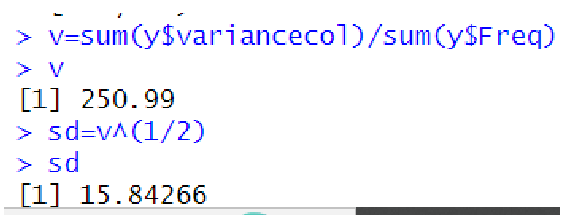
PROGRAM:

```
#variance and standard deviation of grouped data
data=read.csv(file.choose())
math=data$maths
range(math)
bins=seq(20,110,by=10)
c.i=cut(math,bins,right=F)
ftbl=table(c.i)
y=as.data.frame(ftbl)
y
m=seq(25,105,by=10)
m
xm=weighted.mean(m,y$Freq)
xm
y$variancecol<-(y$Freq)*(m-xm)^2
y
v=sum(y$variancecol)/sum(y$Freq)
v
sd=v^(1/2)
sd
```

OUTPUT:



```
> #variance and standard deviation of grouped data
> data=read.csv(file.choose())
> math=data$maths
> range(math)
[1]  20 100
> bins=seq(20,110,by=10)
> c.i=cut(math,bins,right=F)
> ftbl=table(c.i)
> y=as.data.frame(ftbl)
> y
        c.i Freq
1   [20,30)    9
2   [30,40)   31
3   [40,50)   98
4   [50,60)  210
5   [60,70)  226
6   [70,80)  211
7   [80,90)  146
8  [90,100)   54
9 [100,110)   15
> m=seq(25,105,by=10)
> m
[1]  25  35  45  55  65  75  85  95 105
> xm=weighted.mean(m,y$Freq)
> xm
[1] 66.9
> y$variancecol<-(y$Freq)*(m-xm)^2
> y
        c.i Freq variancecol
1   [20,30)    9    15800.49
2   [30,40)   31    31545.91
3   [40,50)   98    47001.78
4   [50,60)  210    29738.10
5   [60,70)  226      815.86
6   [70,80)  211    13843.71
7   [80,90)  146    47831.06
8  [90,100)   54    42638.94
9 [100,110)   15    21774.15
```



```
> v=sum(y$variancecol)/sum(y$Freq)
> v
[1] 250.99
> sd=v^(1/2)
> sd
[1] 15.84266
```

<u>CONCLUSION:</u> Standard deviation and variance of math marks was found using formulas

EXPERIMENT-4.3:

<u>AIM:</u> To find the quartile deviation of raw data

<u>MATHEMATICAL DEFINITION:</u> Quartile deviation is also referred to as the semi interquartile range and is half of the difference between the third quartile and the first quartile value. The formula for quartile deviation of the raw data is Q.D = (Q3 - Q1)/2.

Where , Q3 - third quartile - (3N/4)th observation

Q1 - first quartile - (N/4)th observation

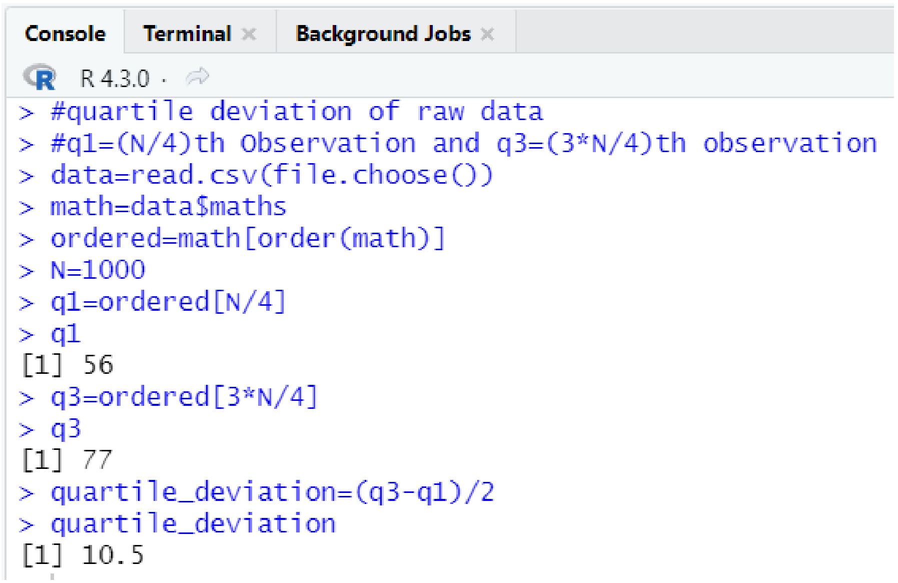
<u>SYNTAX EXPLANATION:</u>

Order function allows the user to sort the data frames, matrices or vectors in ascending order or descending order.
Syntax - order(x, decreasing= TRUE or FALSE) where,
x- dataframe, vector or matrix
decreasing- if true then sorts in descending order else in ascending order

PROGRAM:

```
#quartile deviation of raw data
#q1=(N/4)th Observation and q3=(3*N/4)th observation
data=read.csv(file.choose())
math=data$maths
ordered=math[order(math)]
N=1000
q1=ordered[N/4]
q1
q3=ordered[3*N/4]
q3
quartile_deviation=(q3-q1)/2
quartile_deviation
```

OUTPUT:



```
> #quartile deviation of raw data
> #q1=(N/4)th Observation and q3=(3*N/4)th observation
> data=read.csv(file.choose())
> math=data$maths
> ordered=math[order(math)]
> N=1000
> q1=ordered[N/4]
> q1
[1] 56
> q3=ordered[3*N/4]
> q3
[1] 77
> quartile_deviation=(q3-q1)/2
> quartile_deviation
[1] 10.5
```

CONCLUSION: Thus quartile deviation for raw data is found

EXPERIMENT-4.4:

AIM: To find the quartile deviation of grouped data.

MATHEMATICAL DEFINITION: The Quartile Deviation can be defined mathematically as half of the difference between the upper and lower quartile.
QD=(Q3-Q1)/2
Where Q3 - third quartile = l3+h*(3N/4-cf3)/f3
l3= lower limit of the third quartile class
cf3=cumulative frequency of the third quartile class

f3=frequency of the third quartile class
Q1 - first quartile = l1+h*(3N/4-cf1)/f1
l1= lower limit of the first quartile class
N=Number of observations
cf1=cumulative frequency of the first quartile class
h=height of the quartile class
f1=frequency of the first quartile class

SYNTAX EXPLANATION:

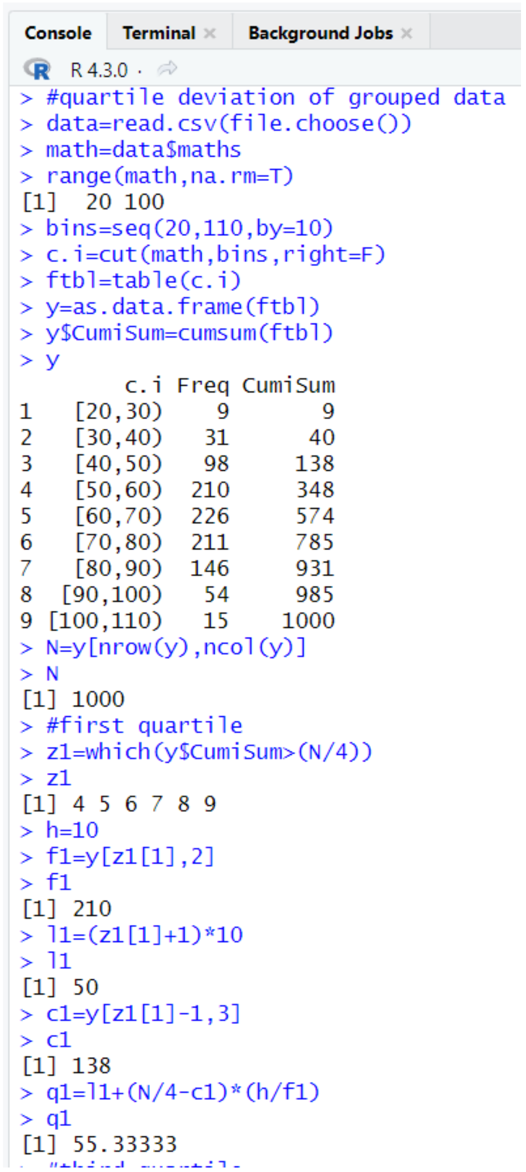
- cumsum()-The cumsum() function in R computes the cumulative sum of elements in a vector object.
- y[nrow(y),ncol(y)] - nrow(y) returns number of rows in y and ncol(y) returns number of columns in y. And y[nrow(y),ncol(y)] returns the data present in the last row and last column.
- which()-The which() function in R returns the position or the index of the value which satisfies the given condition.

PROGRAM:

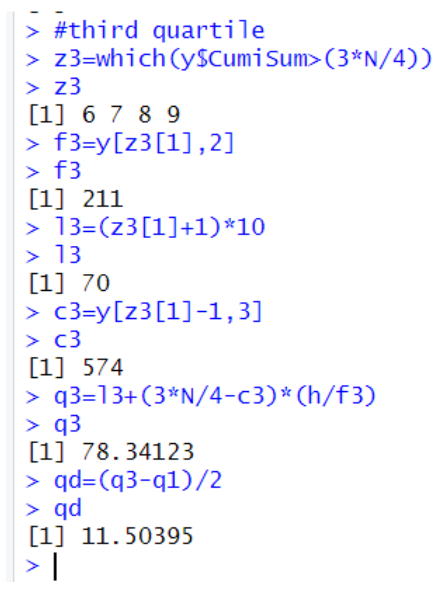
```
#quartile deviation of grouped data
data=read.csv(file.choose())
math=data$maths
range(math,na.rm=T)
bins=seq(20,110,by=10)
c.i=cut(math,bins,right=F)
ftbl=table(c.i)
y=as.data.frame(ftbl)
y$CumiSum=cumsum(ftbl)
y
N=y[nrow(y),ncol(y)]
N
#first quartile
z1=which(y$CumiSum>(N/4))
z1
h=10
f1=y[z1[1],2]
f1
l1=(z1[1]+1)*10
l1
c1=y[z1[1]-1,3]
c1
q1=l1+(N/4-c1)*(h/f1)
q1
#third quartile
z3=which(y$CumiSum>(3*N/4))
z3
f3=y[z3[1],2]
f3
```

l3=(z3[1]+1)*10
l3
c3=y[z3[1]-1,3]
c3
q3=l3+(3*N/4-c3)*(h/f3)
q3
qd=(q3-q1)/2
qd

OUTPUT:



```
> #quartile deviation of grouped data
> data=read.csv(file.choose())
> math=data$maths
> range(math,na.rm=T)
[1]  20 100
> bins=seq(20,110,by=10)
> c.i=cut(math,bins,right=F)
> ftbl=table(c.i)
> y=as.data.frame(ftbl)
> y$CumiSum=cumsum(ftbl)
> y
        c.i Freq CumiSum
1   [20,30)    9       9
2   [30,40)   31      40
3   [40,50)   98     138
4   [50,60)  210     348
5   [60,70)  226     574
6   [70,80)  211     785
7   [80,90)  146     931
8  [90,100)   54     985
9 [100,110)   15    1000
> N=y[nrow(y),ncol(y)]
> N
[1] 1000
> #first quartile
> z1=which(y$CumiSum>(N/4))
> z1
[1] 4 5 6 7 8 9
> h=10
> f1=y[z1[1],2]
> f1
[1] 210
> l1=(z1[1]+1)*10
> l1
[1] 50
> c1=y[z1[1]-1,3]
> c1
[1] 138
> q1=l1+(N/4-c1)*(h/f1)
> q1
[1] 55.33333
```



```
> #third quartile
> z3=which(y$CumiSum>(3*N/4))
> z3
[1] 6 7 8 9
> f3=y[z3[1],2]
> f3
[1] 211
> l3=(z3[1]+1)*10
> l3
[1] 70
> c3=y[z3[1]-1,3]
> c3
[1] 574
> q3=l3+(3*N/4-c3)*(h/f3)
> q3
[1] 78.34123
> qd=(q3-q1)/2
> qd
[1] 11.50395
> |
```

CONCLUSION: Thus the quartile deviation of math marks was found using formulas

EXPERIMENT-5:

AIM: To calculate the covariance and correlation coefficient of the given data.

MATHEMATICAL DEFINITION: Covariance is an indicator of the extent to which 2 random variables are dependent on each other. A higher number denotes higher dependency.Correlation is a statistical measure that indicates how strongly two variables are related.Correlation is limited to values between the range -1 and +1.

cov(x,y) = ((xi-x_mean)(yi-y_mean))/n —-(explain terms)

r(x,y) = cov(x,y)/(sd(x)*sd(y)) —-(explain terms)

SYNTAX EXPLANATION:

The basic syntax of covariance

cov(x, y, method) **where,**

- **x** and **y** represents the data vectors
- **method** defines the type of method to be used to compute covariance. Default is "pearson".

The syntax for correlation is

cor(x, y, method)

where,

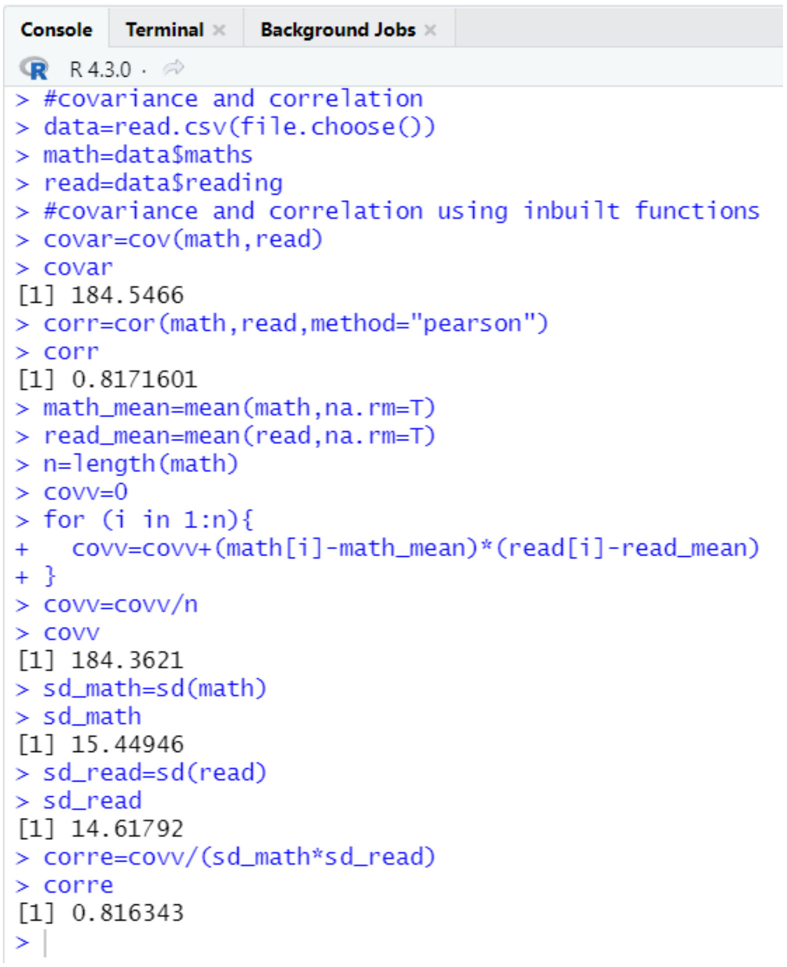
- **x** and **y** represents the data vectors
- **method** defines the type of method to be used to compute covariance. Default is "pearson".

PROGRAM:

```
#covariance and correlation
data=read.csv(file.choose())
math=data$maths
read=data$reading
#covariance and correlation using inbuilt functions
covar=cov(math,read)
```

```
covar
corr=cor(math,read,method="pearson")
corr
math_mean=mean(math,na.rm=T)
read_mean=mean(read,na.rm=T)
n=length(math)
covv=0
for (i in 1:n){
 covv=covv+(math[i]-math_mean)*(read[i]-read_mean)
}
covv=covv/n
covv
sd_math=sd(math)
sd_math
sd_read=sd(read)
sd_read
corre=covv/(sd_math*sd_read)
corre
```

OUTPUT:



```
> #covariance and correlation
> data=read.csv(file.choose())
> math=data$maths
> read=data$reading
> #covariance and correlation using inbuilt functions
> covar=cov(math,read)
> covar
[1] 184.5466
> corr=cor(math,read,method="pearson")
> corr
[1] 0.8171601
> math_mean=mean(math,na.rm=T)
> read_mean=mean(read,na.rm=T)
> n=length(math)
> covv=0
> for (i in 1:n){
+   covv=covv+(math[i]-math_mean)*(read[i]-read_mean)
+ }
> covv=covv/n
> covv
[1] 184.3621
> sd_math=sd(math)
> sd_math
[1] 15.44946
> sd_read=sd(read)
> sd_read
[1] 14.61792
> corre=covv/(sd_math*sd_read)
> corre
[1] 0.816343
> 
```

CONCLUSION: Thus covariance and correlation of math marks and reading marks was found. 81.63% of math marks varies with reading marks.