Noida Institute of Engineering and Technology
B-Tech in Computer Science & Engineering
(Data Science) - Programme (Autonomous)
Regulation 2020, Greater Noida
SEMESTER III

Data Analysis Lab

Session (2021-2022) ODD Semester
2nd YEAR

Submitted By:

NAME:-Amritanshu Sharma
STUDENT ID:-0201CSDS087@NIET.CO.IN

Noida Institute of Engineering Technology
19, Knowledge Park- II, Institutional Area,
Greater Noida (UP) - 201306

Noida Institute of Engineering and Technology, Greater Noida
B-Tech in CSE (Data Science) (Autonomous) Syllabus
Regulation 2020
SEMESTER III

Lab Course Outcome: After completion of this course students will be able to

CO 1 Develop basic R programs. K3
CO 2 Implement statistical techniques on variety of data. K3
CO 3 Explore different types of data and file formats. K2
CO 4 Perform exploratory data analysis on different data types. K3
CO 5 Apply visualization techniques on various data sets. K3

S.No	Experiment Name	Date of Experiment	Sign of Faculty
1	**Write a R program to create a Data frame which contain details of 5 employees and display the details.**		
2	**Write a R program to get the first 10 Fibonacci numbers.**		
3	**Write a R program to get all prime numbers up to a given number.**		
4	**Write a R program to find the maximum and the minimum value of a given vector.**		
5	**Create an array, passing in a vector of values and a vector of dimensions, also provide names for each dimension.**		
6	**Write a R program to create a list containing a vector, a matrix and a list and give names to the elements in the list. Access the first and second element of the list.**		
7	**Write a R program to create a list containing a vector, a matrix and a list and add element at the end of the list.**		
8	**Read the following file formats in Python/R:** **• Comma-separated values** **• XLSX** **• ZIP** **• Plain Text (txt)** **• JSON** **• XML** **• HTML** **• Images** **• Hierarchical Data Format** **• PDF** **• DOCX** **• MP3**		

9	**Load the Iris dataset as a list of lists** **• Compute and print the mean and the standard deviation for each of the 4 measurement columns (i.e., sepal length and width, petal length and width** **Compute and print the mean and the standard deviation for each of the 4 measurement columns, separately for each of the three Iris species.**		
10	**a. Find the data distributions using box and scatter plot.** **b. Find the outliers using box plot** **c. Plot the histogram, bar chart and pie chart on sample data** **d. Plot Pie Chart, Histogram (3D) [including colourful ones]**		
11	**Import a sample dataset and perform Regression techniques to find out relation between variables.**		
12	**Find the correlation matrix.** **a. Plot the correlation plot on dataset and visualize giving an overview of relationships among variables on data set.** **b. Analysis of covariance: variance (ANOVA)if data have categorical variables on data set.**		
13	**Write a program to create 3D plot, to add title, change viewing direction, add color and shade to the plot.**		
14	**a. Create a data frame from the sample data set.** **b. Create a table with the needed variables** **c. Perform the Chi-Square test.**		
15	**Perform complete steps of exploratory data analysis on standard data sets (iris flowers, Wine Quality Dataset etc.)**		

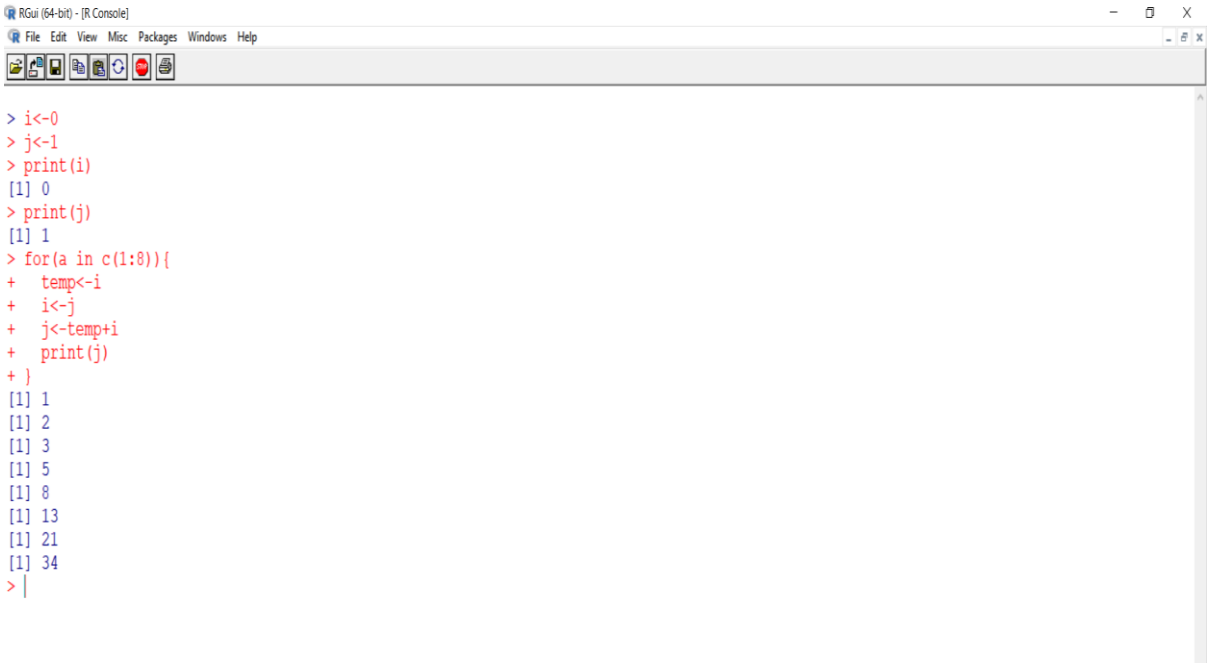
PROGRAM 1

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

```
RGui (64-bit) - [R Console]
File  Edit  View  Misc  Packages  Windows  Help

> Employees<-data.frame(
+ Name=c("Rahul","Aadya","Namit","Sharon","Arjun"),
+ Age=c(29,34,23,28,41),
+ Marital_Status=c("Married","Married","Unmarried","Unmarried","Married"),
+ Contact_Number=c(7364873468,9784678463,8473284667,9487326231,7003576372),
+ Aadhar_Number=c("8348 3847 3392","3763 6473 7846","7342 4663 3347","9794 3786 2378","9743 3467 8947"),
+ Blood_Group=c("B+","A-","O+","B+","A+"),
+ Desgination=c("Manager","CEO","Intern","Receptionist","Project Head"),
+ Salary=c(600000,1000000,25000,300000,800000))
> Employees
    Name Age Marital_Status Contact_Number  Aadhar_Number Blood_Group  Desgination  Salary
1  Rahul  29        Married     7364873468 8348 3847 3392          B+      Manager  600000
2  Aadya  34        Married     9784678463 3763 6473 7846          A-          CEO 1000000
3  Namit  23      Unmarried     8473284667 7342 4663 3347          O+       Intern   25000
4 Sharon  28      Unmarried     9487326231 9794 3786 2378          B+ Receptionist  300000
5  Arjun  41        Married     7003576372 9743 3467 8947          A+ Project Head  800000
>
```

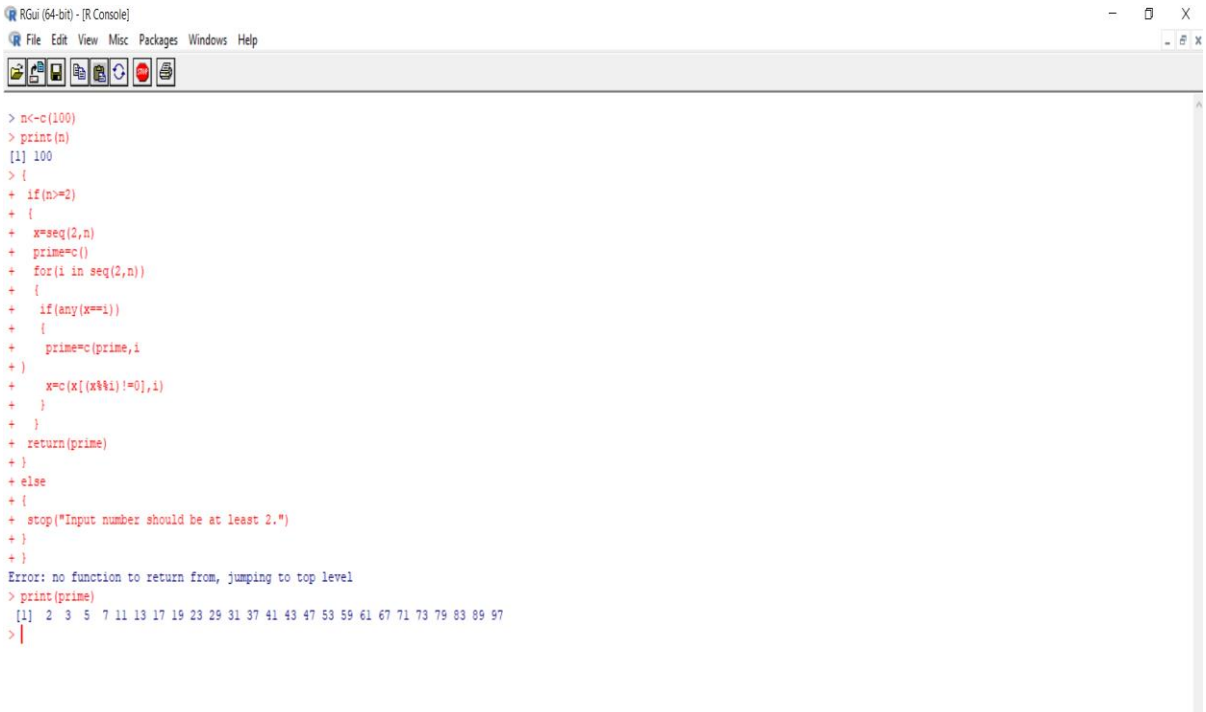
PROGRAM 2

Write a R program to get the first 10 Fibonacci numbers.



PROGRAM 3

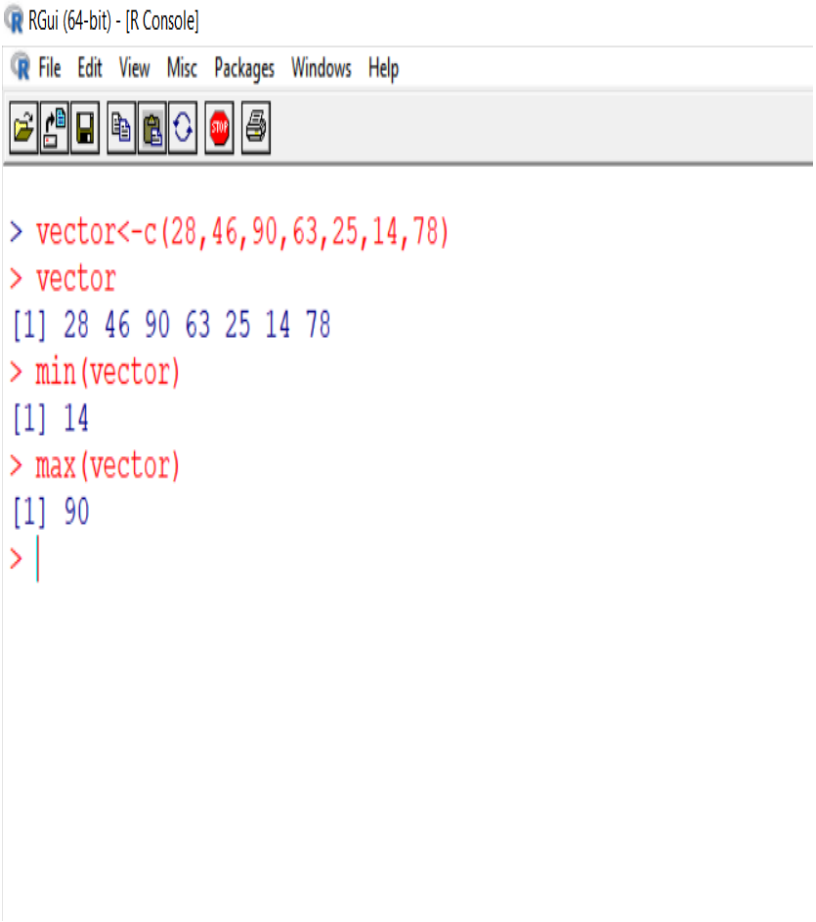
Write a R program to get all prime numbers up to a given number.



```
RGui (64-bit) - [R Console]
File  Edit  View  Misc  Packages  Windows  Help
> n<-c(100)
> print(n)
[1] 100
> {
+  if(n>=2)
+  {
+   x=seq(2,n)
+   prime=c()
+   for(i in seq(2,n))
+   {
+    if(any(x==i))
+    {
+     prime=c(prime,i
+ )
+     x=c(x[(x%%i)!=0],i)
+    }
+   }
+  return(prime)
+ }
+ else
+ {
+  stop("Input number should be at least 2.")
+ }
+ }
Error: no function to return from, jumping to top level
> print(prime)
 [1]  2  3  5  7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
>
```

PROGRAM 4

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



PROGRAM 5

Create an array, passing in a vector of values and a vector of dimensions, also provide names for each dimension.

RGui (64-bit) - [R Console]

File Edit View Misc Packages Windows Help

```
>  v<-c(54,89,12,34,90,12,7,45,32,72,2,82)
> d<-c(2,3,2)
> arr<-array(v,dim=d,dimnames=list(c("Row 1","Row 2"),c("Col 1","Col 2","Col 3"),c("Array 1","Array 2")))
> arr
, , Array 1

      Col 1 Col 2 Col 3
Row 1    54    12    90
Row 2    89    34    12

, , Array 2

      Col 1 Col 2 Col 3
Row 1     7    32     2
Row 2    45    72    82

>
```

PROGRAM 6

Write a R program to create a list containing a vector, a matrix and a list and give names to the elements in the list. Access the first and second element of the list.

RGui (64-bit) - [R Console]

File Edit View Misc Packages Windows Help

```
> v<-c(1,2,3)
> m<-matrix(c(4,5,6,7),nrow=2,ncol=2)
> l<-list(8,9,10,11,12)
> Combine<-list(v,m,l)
> names(Combine)<-c("Vector","Matrix","List")
> print(Combine)
$Vector
[1] 1 2 3

$Matrix
     [,1] [,2]
[1,]    4    6
[2,]    5    7

$List
$List[[1]]
[1] 8

$List[[2]]
[1] 9

$List[[3]]
[1] 10

$List[[4]]
[1] 11

$List[[5]]
[1] 12


> Combine[1]
$Vector
[1] 1 2 3

> Combine[2]
$Matrix
     [,1] [,2]
[1,]    4    6
[2,]    5    7
```

PROGRAM 7

Write a R program to create a list containing a vector, a matrix and a list and add element at the end of the list.

RGui (64-bit) - [R Console]

File Edit View Misc Packages Windows Help

```
> v<-c(45,87)
> m<-matrix(c(23,94,12,49),nrow=2,ncol=2)
> l<-list(34,65)
> Combine<-list(c(v,m,l))
> Combine
[[1]]
[[1]][[1]]
[1] 45

[[1]][[2]]
[1] 87

[[1]][[3]]
[1] 23

[[1]][[4]]
[1] 94

[[1]][[5]]
[1] 12

[[1]][[6]]
[1] 49

[[1]][[7]]
[1] 34

[[1]][[8]]
[1] 65


> Combine<-append(Combine,2,after=65)
> Combine
[[1]]
[[1]][[1]]
[1] 45

[[1]][[2]]
[1] 87

[[1]][[3]]
[1] 23

[[1]][[4]]
```

RGui (64-bit) - [R Console]

File Edit View Misc Packages Windows Help

```
[1] 12

[[1]][[6]]
[1] 49

[[1]][[7]]
[1] 34

[[1]][[8]]
[1] 65


> Combine<-append(Combine,2,after=65)
> Combine
[[1]]
[[1]][[1]]
[1] 45

[[1]][[2]]
[1] 87

[[1]][[3]]
[1] 23

[[1]][[4]]
[1] 94

[[1]][[5]]
[1] 12

[[1]][[6]]
[1] 49

[[1]][[7]]
[1] 34

[[1]][[8]]
[1] 65


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

Program No 08

Read csv file:

```
#read different file formate

data1<-read.csv("iris.csv")
head(data1)
tail(data1)
```

39:1 (Top Level)

Console ~/my_R_workSpace/

```
In addition: Warning message:
In file(file, "rt") : cannot open file 'iris': No such file or directory
> head(data1)
Error in head(data1) : object 'data1' not found
> tail(data1)
Error in tail(data1) : object 'data1' not found
> data1<-read.csv("iris.csv")
> head(data1)
  sepal.length sepal.width petal.length petal.width variety
1          5.1         3.5          1.4         0.2  Setosa
2          4.9         3.0          1.4         0.2  Setosa
3          4.7         3.2          1.3         0.2  Setosa
4          4.6         3.1          1.5         0.2  Setosa
5          5.0         3.6          1.4         0.2  Setosa
6          5.4         3.9          1.7         0.4  Setosa
> tail(data1)
    sepal.length sepal.width petal.length petal.width   variety
145          6.7         3.3          5.7         2.5 Virginica
146          6.7         3.0          5.2         2.3 Virginica
147          6.3         2.5          5.0         1.9 Virginica
148          6.5         3.0          5.2         2.0 Virginica
149          6.2         3.4          5.4         2.3 Virginica
150          5.9         3.0          5.1         1.8 Virginica
>
```

Read xlsx file

```
#read xlsx
#install.packages("xlsx")
library("xlsx")
data5<- read.xlsx("Demo.xlsx", sheetIndex = 1)
head(data5)
tail(data5)
```

9:1 (Top Level)

Console ~/my_R_workSpace/

```
> data6<-read.xlsx("Demo.xlsx",sheetIndex = 1)
Error in read.xlsx("Demo.xlsx", sheetIndex = 1) :
  could not find function "read.xlsx"
> library("xlsx")
Warning message:
package 'xlsx' was built under R version 4.1.2
> data5<- read.xlsx("Demo.xlsx", sheetIndex = 1)
> head(data5)
  Year   Category    Product Sales Rating
1 2017 Components     Chains 20000   0.75
2 2015   Clothing      Socks  3700   0.22
3 2017   Clothing Bib-Shorts  4000   0.22
4 2015   Clothing     Shorts 13300   0.56
5 2017   Clothing     Tights 36000   1.00
6 2015 Components Handlebars  2300   0.35
> tail(data5)
   Year   Category       Product Sales Rating
70 2016   Clothing         Vests  1300   0.25
71 2016   Clothing        Tights 22100   0.99
72 2017 Components       Saddles  3100   0.42
73 2015   Clothing          Caps   500   0.50
74 2017      Bikes Touring Bikes  3100   0.22
75 2015 Components        Chains  8700   0.92
>
```

Read Zip file: -

```
> library(plyr)
Warning message:
package 'plyr' was built under R version 4.1.2
> my_dir<-"/Users/Amritanshu/OneDrive - Noida Institute of Engineering and Technology/Documents/my_R_workSpace/temp"
> zip_file<-list.files(path=my_dir,pattern="*.zip",full.names = TRUE)
> ldply(.data = zip_file,.fun = unzip,exdir=my_dir)
```

1 /Users/Amritanshu/OneDrive - Noida Institute of Engineering and Technology/Documents/my_R_workSpace/temp/Sales-Data-Analysis-Excel-master/README.md
V2
1 /Users/Amritanshu/OneDrive - Noida Institute of Engineering and Technology/Documents/my_R_workSpace/temp/Sales-Data-Analysis-Excel-master/Sales-Data-Analysis-Workbook.xlsx

Read text file:

```
#read Txt file
my_data <- read.delim("http://www.sthda.com/upload/boxplot_format.txt")
head(my_data)
```

31:1 (Top Level)

Console ~/my_R_workSpace/

```
Error in ldply(.data = zip_file, .fun = unzip, exdir = my_dir) :
  could not find function "ldply"
> my_data <- read.delim("http://www.sthda.com/upload/boxplot_format.txt")
> head(my_data)
   Nom variable Group
1 IND1       10     A
2 IND2        7     A
3 IND3       20     A
4 IND4       14     A
5 IND5       14     A
6 IND6       12     A
>
>
>
```

Read rjson file:

```r
#read Rjson file
library("rjson")
reading<-fromJSON(file = "example_1.json")
reading

```

34:17 (Top Level)

Console ~/my_R_workSpace/

```
>
> #read Rjson file
> library("rjson")
> reading<-fromJSON(file = "example_1.json")
> reading
$fruit
[1] "Apple"

$size
[1] "Large"

$color
[1] "Red"

>
```

Read xml file: -

```r
#Read xml file
library("XML")
library("methods")
res<-xmlParse(file = "Demo_xml.xml")
print(res)

```

43:1 (Top Level)

Console ~/my_R_workSpace/

```
> library("XML")
> library("methods")
> res<-xmlParse(file = "Demo_xml.xml")
> print(res)
<?xml version="1.0"?>
<catalog>
  <book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description>An in-depth look at creating applications
      with XML.</description>
  </book>
  <book id="bk102">
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-12-16</publish_date>
    <description>A former architect battles corporate zombies,
      an evil sorceress, and her own childhood to become queen
      of the world.</description>
```

Read image file: -

```r
# Read html file
#install.packages("magick")
#install.packages("rsvg")
library(magick)
tiger <- image_read_svg('http://jeroen.github.io/images/tiger.svg', width = 350)

print(tiger)
```

251:13 (Untitled)

Console ~/my_R_workSpace/

```
package 'magick' was built under R version 4.1.2
> tiger <- image_read_svg('http://jeroen.github.io/images/tiger.svg', width = 350)
>
> print(tiger)
  format width height colorspace matte filesize density
1    PNG   350    350       sRGB  TRUE        0   72x72
>
>
>
>
```

Microsoft Excel
97-2003 Worksheet

Read pdf file: -

```
#read pdf file
#install.packages("pdftools")
pdftools::pdf_text(pdf = "http://arxiv.org/pdf/1403.2805.pdf")
```

46:30 (Top Level) R Script

Console ~/my_R_workSpace/

```
> pdftools::pdf_text(pdf = "http://arxiv.org/pdf/1403.2805.pdf")
 [1] "                                          The jsonlite Package: A Practical and Consistent
 Mapping\n                                                                  Between JSON Data and R
 Objects\n\n                                                                                     Jero
en Ooms\narXiv:1403.2805v1 [stat.CO] 12 Mar 2014\n\n\n\n\n
                                   UCLA Department of Statistics\n\n\n
                                                                Abstract\n
                       A naive realization of JSON data in R maps JSON arrays to an unnamed list, a
nd JSON objects to a\n                                                   named list. However, in practic
e a list is an awkward, inefficient type to store and manipulate data.\n
                   Most statistical applications work with (homogeneous) vectors, matrices or data f
rames. Therefore JSON\n                                                   packages in R typically define
 certain special cases of JSON structures which map to simpler R types.\n
                    Currently there exist no formal guidelines, or even consensus between implementa
tions on how R data\n                                                  should be represented in JSON. F
urthermore, upon closer inspection, even the most basic data structures\n
                    in R actually do not perfectly map to their JSON counterparts and leave some amb
iguity for edge cases.\n                                                  These problems have resulted
in different behavior between implementations and can lead to unexpected\n
                     output. This paper explicitly describes a mapping between R classes and JSON da
ta, highlights potential\n                                                   problems, and proposes conv
entions that generalize the mapping to cover all common structures. We\n
                   emphasize the importance of type consistency when using JSON to exchange dynamic
```

Read docx file: -

read_docx("Users/Amritanshu/OneDrive - Noida Institute of Engineering and Technology/Documents/my_R_workSpace.docx")

Program No 09

9.Load the Iris dataset as a list of lists • Compute and print the mean and the standard deviation for each of the 4 measurement columns (i.e., sepal length and width, petal length and width Compute and print the mean and the standard deviation for each of the 4 measurement columns, separately for each of the three Iris species.

> Code:- install.packages("dplyr")

```
library(dplyr)
data("iris")
iris_grouped <- group_by(iris,Species)
head(iris_grouped)
summarise(iris_grouped,mean_Sepal_length=mean(Sepal.Length),sd_sepal_length=sd(Sepal.Length))
summarise(iris_grouped,mean_Sepal_width=mean(Sepal.Width),sd_sepal_width=sd(Sepal.Width))
summarise(iris_grouped,mean_Petal_length=mean(Petal.Length),sd_petal_length=sd(Petal.Length))
summarise(iris_grouped,mean_Petal_width=mean(Petal.Width),sd_petal_width=sd(Petal.Width))
summarise(iris_grouped,count = n())
summarise(
  iris_grouped,
  percent = sum(Sepal.Length > 5.5)/n()
)
summarise(
  iris_grouped,
  percent = sum(Sepal.Width > 5.5)/n()
)
summarise(
  iris_grouped,
  percent = sum(Petal.Length > 5.5)/n()
)
summarise(
  iris_grouped,
  percent = sum(Petal.Width > 5.5)/n()
)
```

```
> library(dplyr)
> data("iris")
> iris_grouped <- group_by(iris,Species)
> head(iris_grouped)
# A tibble: 6 x 5
# Groups:   Species [1]
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
         <dbl>       <dbl>        <dbl>       <dbl> <fct>
1          5.1         3.5          1.4         0.2 setosa
2          4.9         3            1.4         0.2 setosa
3          4.7         3.2          1.3         0.2 setosa
4          4.6         3.1          1.5         0.2 setosa
5          5           3.6          1.4         0.2 setosa
6          5.4         3.9          1.7         0.4 setosa
> summarise(iris_grouped,mean_Sepal_length=mean(Sepal.Length),sd_sepal_length=sd(Sepal.Length))
# A tibble: 3 x 3
  Species    mean_Sepal_length sd_sepal_length
  <fct>                  <dbl>           <dbl>
1 setosa                  5.01           0.352
2 versicolor              5.94           0.516
3 virginica               6.59           0.636
> summarise(iris_grouped,mean_Sepal_width=mean(Sepal.Width),sd_sepal_width=sd(Sepal.Width))
# A tibble: 3 x 3
  Species    mean_Sepal_width sd_sepal_width
  <fct>                 <dbl>          <dbl>
1 setosa                 3.43          0.379
2 versicolor             2.77          0.314
3 virginica              2.97          0.322
> summarise(iris_grouped,mean_Petal_length=mean(Petal.Length),sd_petal_length=sd(Petal.Length))
# A tibble: 3 x 3
  Species    mean_Petal_length sd_petal_length
  <fct>                  <dbl>           <dbl>
1 setosa                  1.46           0.174
2 versicolor              4.26           0.470
3 virginica               5.55           0.552
> summarise(iris_grouped,mean_Petal_width=mean(Petal.Width),sd_petal_width=sd(Petal.Width))
# A tibble: 3 x 3
  Species    mean_Petal_width sd_petal_width
  <fct>                 <dbl>          <dbl>
1 setosa                0.246          0.105
2 versicolor            1.33           0.198
3 virginica             2.03           0.275
```

```
Console   Jobs
R 4.1.1 · ~/
3 virginica              2.03              0.273
> summarise(iris_grouped,count = n())
# A tibble: 3 x 2
  Species    count
  <fct>      <int>
1 setosa        50
2 versicolor    50
3 virginica     50
> summarise(
+   iris_grouped,
+   percent = sum(Sepal.Length > 5.5)/n()
+ )
# A tibble: 3 x 2
  Species    percent
  <fct>        <dbl>
1 setosa        0.06
2 versicolor    0.78
3 virginica     0.98
> summarise(
+   iris_grouped,
+   percent = sum(Sepal.Width > 5.5)/n()
+ )
# A tibble: 3 x 2
  Species    percent
  <fct>        <dbl>
1 setosa           0
2 versicolor       0
3 virginica        0
> summarise(
+   iris_grouped,
+   percent = sum(Petal.Length > 5.5)/n()
+ )
# A tibble: 3 x 2
  Species    percent
  <fct>        <dbl>
1 setosa         0
2 versicolor     0
3 virginica      0.5
> summarise(
+   iris_grouped,
+   percent = sum(Petal.Width > 5.5)/n()
+ )
# A tibble: 3 x 2
  Species    percent
  <fct>        <dbl>
1 setosa           0
2 versicolor       0
3 virginica        0
>
```

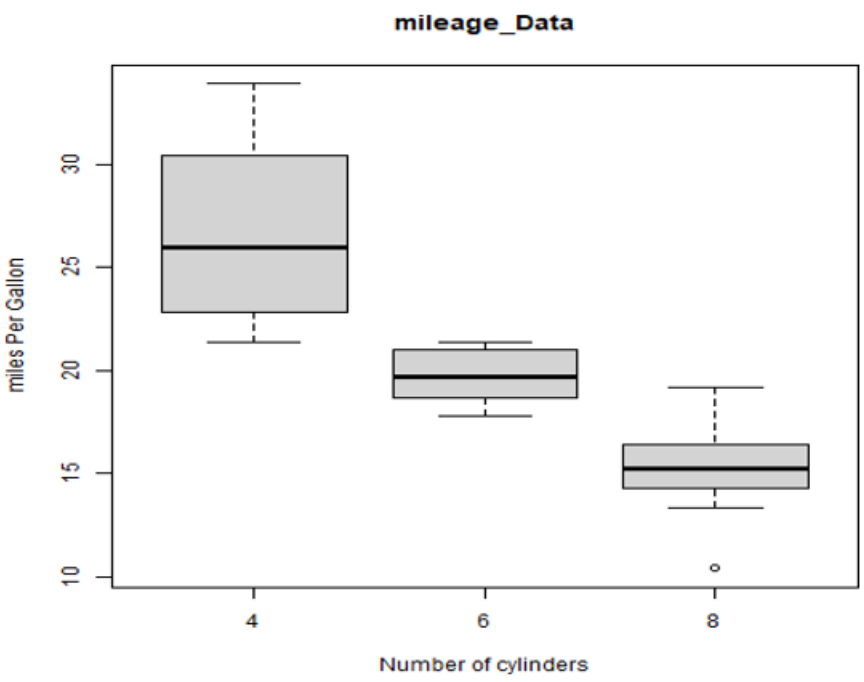
Program No: -10

A.Find the data distributions using box and scatter plot. b. Find the outliers using box plot c. Plot the histogram, bar chart and pie chart on sample data d. Plot Pie Chart, Histogram (3D) [including colorful ones]

Code:-

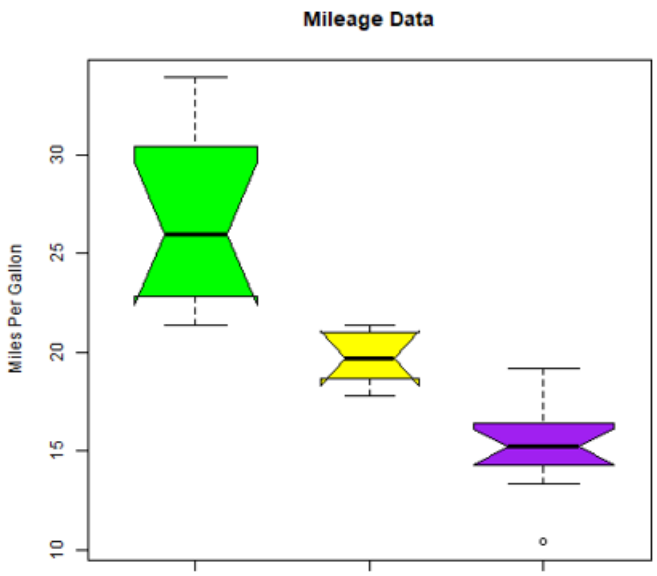
```
#10 no.program
input <- mtcars[,c('mpg','cyl')]
print(head(input))
png(file = "boxplot.png")
boxplot(mpg ~ cyl, data = mtcars, xlab = "Number of Cylinders", ylab = "Miles Per Gallon", main = "Mileage Data")
dev.off()
```

```
Console ~/my_R_workSpace/
  could not find function "read_docx"
> input <- mtcars[,c('mpg','cyl')]
> print(head(input))
                   mpg cyl
Mazda RX4         21.0   6
Mazda RX4 Wag     21.0   6
Datsun 710        22.8   4
Hornet 4 Drive    21.4   6
Hornet Sportabout 18.7   8
Valiant           18.1   6
> png(file = "boxplot.png")
> boxplot(mpg ~ cyl, data = mtcars, xlab = "Number of Cylinders", ylab = "Miles Per Gallon", main = "Mileage Data")
> dev.off()
null device
          1
>
>
>
```



```r


# chart file a name.
png(file = "boxplot_with_notch.png")
# Plot the chart.
boxplot(mpg ~ cyl, data = mtcars, xlab = "Number of Cylinders",



        ylab = "Miles Per Gallon", main = "Mileage Data", notch = TRUE,
        varwidth = TRUE,
        col = c("green","yellow","red"),
        names = c("High","Medium","Low")
)
dev.off()
```

```
>
> png(file = "boxplot_with_notch.png")
> # Plot the chart.
> boxplot(mpg ~ cyl, data = mtcars, xlab = "Number of Cylinders",
+
+
+
+         ylab = "Miles Per Gallon", main = "Mileage Data", notch = TRUE,
+         varwidth = TRUE,
+         col = c("green","yellow","red"),
+         names = c("High","Medium","Low")
+ )
Warning message:
In (function (z, notch = FALSE, width = NULL, varwidth = FALSE,  :
  some notches went outside hinges ('box'): maybe set notch=FALSE
> dev.off()
null device
          1
```

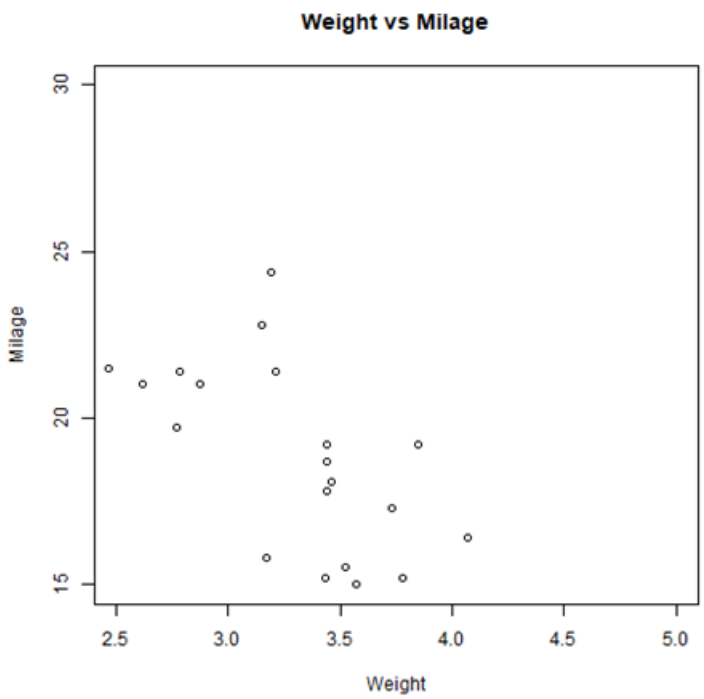


```r
input <- mtcars[,c('wt','mpg')]
print(head(input))
png(file = "scatterplot.png")
plot(x = input$wt,y = input$mpg,
     xlab = "Weight",
     ylab = "Milage",
     xlim = c(2.5,5),
     ylim = c(15,30),
     main = "Weight vs Milage"
)
dev.off()
```

88:1 (Untitled)

Console ~/my_R_workSpace/

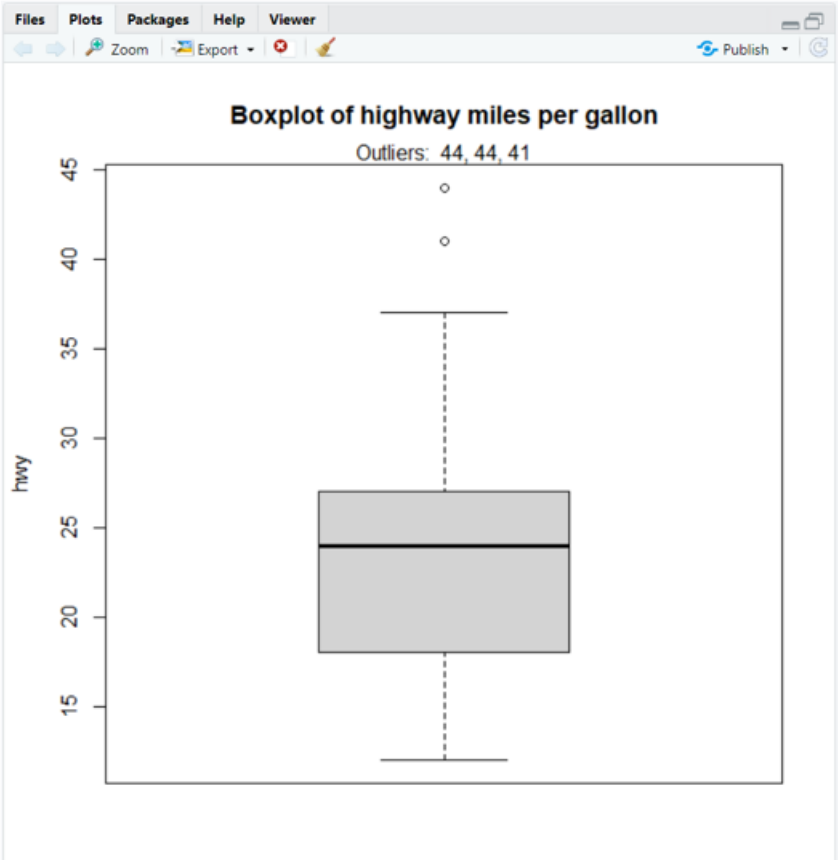
```
> input <- mtcars[,c('wt','mpg')]
> print(head(input))
                   wt  mpg
Mazda RX4         2.620 21.0
Mazda RX4 Wag     2.875 21.0
Datsun 710        2.320 22.8
Hornet 4 Drive    3.215 21.4
Hornet Sportabout 3.440 18.7
Valiant           3.460 18.1
> png(file = "scatterplot.png")
> plot(x = input$wt,y = input$mpg,
+      xlab = "Weight",
+      ylab = "Milage",
+      xlim = c(2.5,5),
+      ylim = c(15,30),
+      main = "Weight vs Milage"
+ )
> dev.off()
png
  2
```



B. find outliers using boxplot: -

```r
boxplot(dat$hwy,
        ylab = "hwy",
        main = "Boxplot of highway miles per gallon"
)
mtext(paste("Outliers: ", paste(out, collapse = ", ")))
```

```
> boxplot(dat$hwy,
+         ylab = "hwy",
+         main = "Boxplot of highway miles per gallon"
+ )
> mtext(paste("Outliers: ", paste(out, collapse = ", ")))
>
```



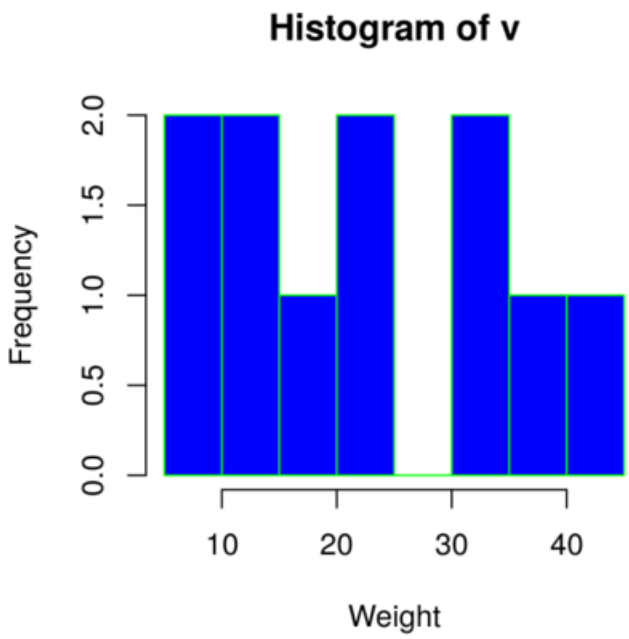
C. Plot the histogram, bar chat and pie chart on sample data

```r

#c plot the histogram
#install.packages(graphics)
library(graphics)
v<-c(9,13,21,8,36,22,12,41,31,43,19)
hist(v,xlab = "weight",col = "blue",border = "green")
dev.off()
```

96:28 (Untitled)

Console ~/my_R_workSpace/

```
>
> library(graphics)
> v<-c(9,13,21,8,36,22,12,41,31,43,19)
> hist(v,xlab = "weight",col = "blue",border = "green")
> dev.off()
null device
          1
```

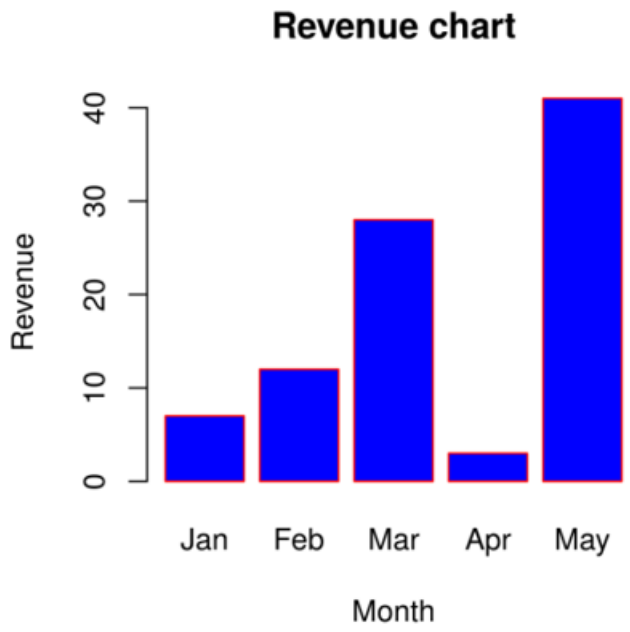


```r
library(graphics)
H<-c(7,12,28,3,41)
M<-c("jan","Feb","Mar","Apr","May")
barplot(H,names.arg = M,xlab = "Month",ylab = "Revenue",
        col = "blue",main = "revenue chart",border = "red")
dev.off()
```

107:37 (Untitled)

Console ~/my_R_workSpace/

```
> library(graphics)
> H<-c(7,12,28,3,41)
> M<-c("jan","Feb","Mar","Apr","May")
> barplot(H,names.arg = M,xlab = "Month",ylab = "Revenue",
+         col = "blue",main = "revenue chart",border = "red")
> dev.off()
null device
          1
>
```



Pie chart

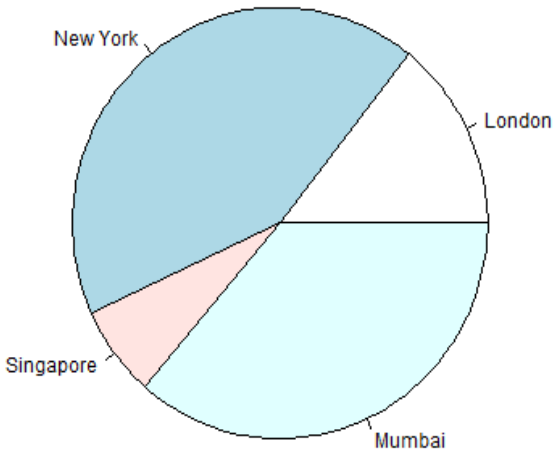
```
x <- c(21, 62, 10, 53)
labels <- c("London", "New York", "Singapore", "Mumbai")
png(file = "city.png")
pie(x,labels)
dev.off()
```

5:10 (Top Level)

Console Jobs

R 4.1.1 · ~/

```
> x <- c(21, 62, 10, 53)
> labels <- c("London", "New York", "Singapore", "Mumbai")
> png(file = "city.png")
> pie(x,labels)
> dev.off()
null device
          1
>
```



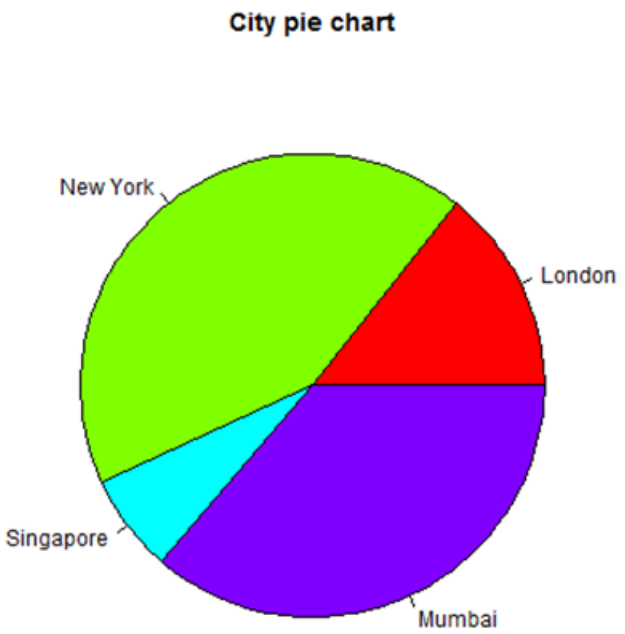
D.lot Pie Chart, Histogram (3D) [including colorful ones]

```
#lot Pie Chart, Histogram (3D) [including colorful ones]
x <- c(21, 62, 10, 53)
labels <- c("London", "New York", "Singapore", "Mumbai")
png(file = "city_title_colours.jpg")
pie(x, labels, main = "City pie chart", col = rainbow(length(x)))
dev.off()
```

115:63 (Untitled)

Console ~/my_R_workSpace/

```
>
> x <- c(21, 62, 10, 53)
> labels <- c("London", "New York", "Singapore", "Mumbai")
> png(file = "city_title_colours.jpg")
> pie(x, labels, main = "City pie chart", col = rainbow(length(x)))
> dev.off()
null device
          1
>
>
```



3D pie chart

```r
#3d pie chart
#install.packages("plotrix")
library(plotrix)
x <- c(21, 62, 10,53)
lbl <- c("London","New York","Singapore","Mumbai")

png(file = "3d_pie_chart.jpg")

# Plot the chart.
pie3D(x,labels = lbl,explode = 0.1, main = "Pie Chart of Countries ")

dev.off()
```

131:1 (Untitled)

Console ~/my_R_workSpace/

```
> library(plotrix)
> x <- c(21, 62, 10,53)
> lbl <- c("London","New York","Singapore","Mumbai")
>
> png(file = "3d_pie_chart.jpg")
>
> # Plot the chart.
> pie3D(x,labels = lbl,explode = 0.1, main = "Pie Chart of Countries ")
>
> dev.off()
null device
          1
>
```

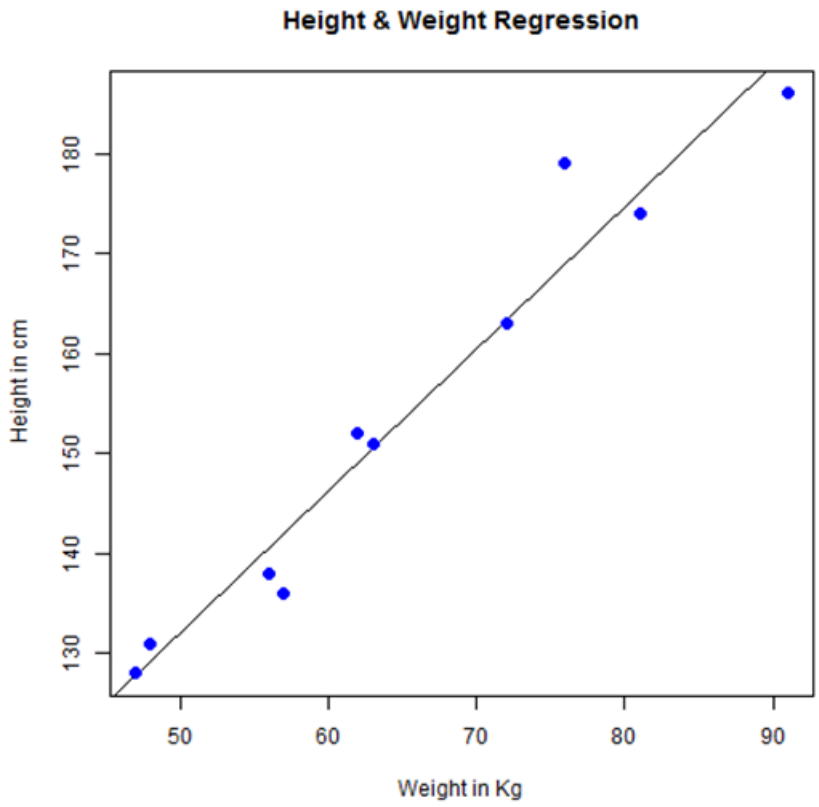
Program No.11

11.Import a sample dataset and perform Regression techniques to find out relation between variables.

```r
#11 no program

x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
relation <- lm(y~x)

print(relation)
print(summary(relation))
a <- data.frame(x = 170)
result <- predict(relation,a) print(result)
png(file = "linearregression.png")
plot(y,x,col = "blue",main = "Height & Weight Regression", abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Weight in Kg",
     ylab = "Height in cm")
dev.off()
```

```
Console ~/my_R_workSpace/
> x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
> y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
> relation <- lm(y~x)
>
> print(relation)

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
   -38.4551       0.6746

> print(summary(relation))
```

```
> print(summary(relation))

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-6.3002 -1.6629  0.0412  1.8944  3.9775

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -38.45509    8.04901  -4.778  0.00139 **
x             0.67461    0.05191  12.997 1.16e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.253 on 8 degrees of freedom
Multiple R-squared:  0.9548,    Adjusted R-squared:  0.9491
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.164e-06

> a <- data.frame(x = 170)
> result <- predict(relation,a) print(result)
Error: unexpected symbol in "result <- predict(relation,a) print"
> png(file = "linearregression.png")
> plot(y,x,col = "blue",main = "Height & Weight Regression", abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Weight in Kg",
+      ylab = "Height in cm")
> dev.off()
null device
          1
>
```


Height & Weight Regression
Height in cm
130
140
150
160
170
180
50
60
70
80
90
Weight in Kg

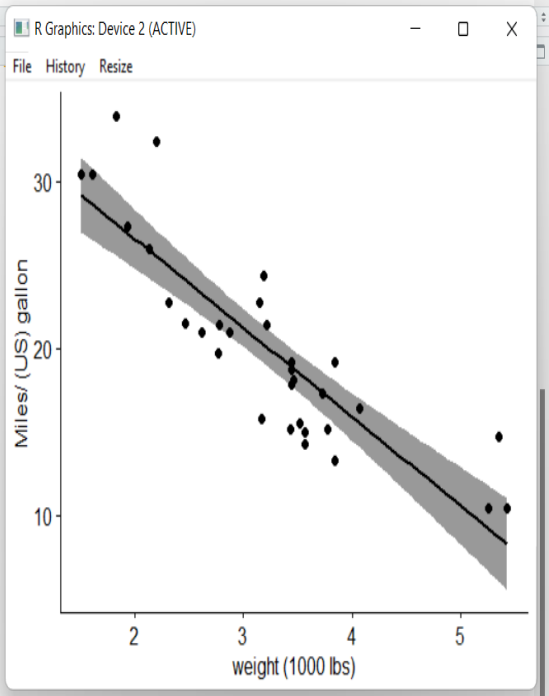
Program No: -12

12.Find the correlation matrix. a. Plot the correlation plot on dataset and visualize giving an overview of relationships among variables on data set. b. Analysis of covariance: variance (ANOVA)if data have categorical variables on data set.

Code: -

```
library(ggpubr)
head(mtcars)
my_data <- mtcars
my_data$cyl <- factor(my_data$cyl)
str(my_data)

#Pearson Correlation Test
ggscatter(my_data, x = "wt", y ="mpg", add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method = "Pearson",
          xlab = "weight (1000 lbs)", ylab = "Miles/ (US) gallon")
```

```
RStudio is installed.
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
> my_data <- mtcars
> my_data$cyl <- factor(my_data$cyl)
> str(my_data)
'data.frame':   32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : Factor w/ 3 levels "4","6","8": 2 2 1 2 3 2 3 1 1 2 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
>
> #Pearson Correlation Test
> ggscatter(my_data, x = "wt", y ="mpg", add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method = "Pearson", xlab = "weig
ht (1000 lbs)", ylab = "Miles/ (US) gallon")
`geom_smooth()` using formula 'y ~ x'
Warning message:
Computation failed in `stat_cor()`:
```



Program No. 13

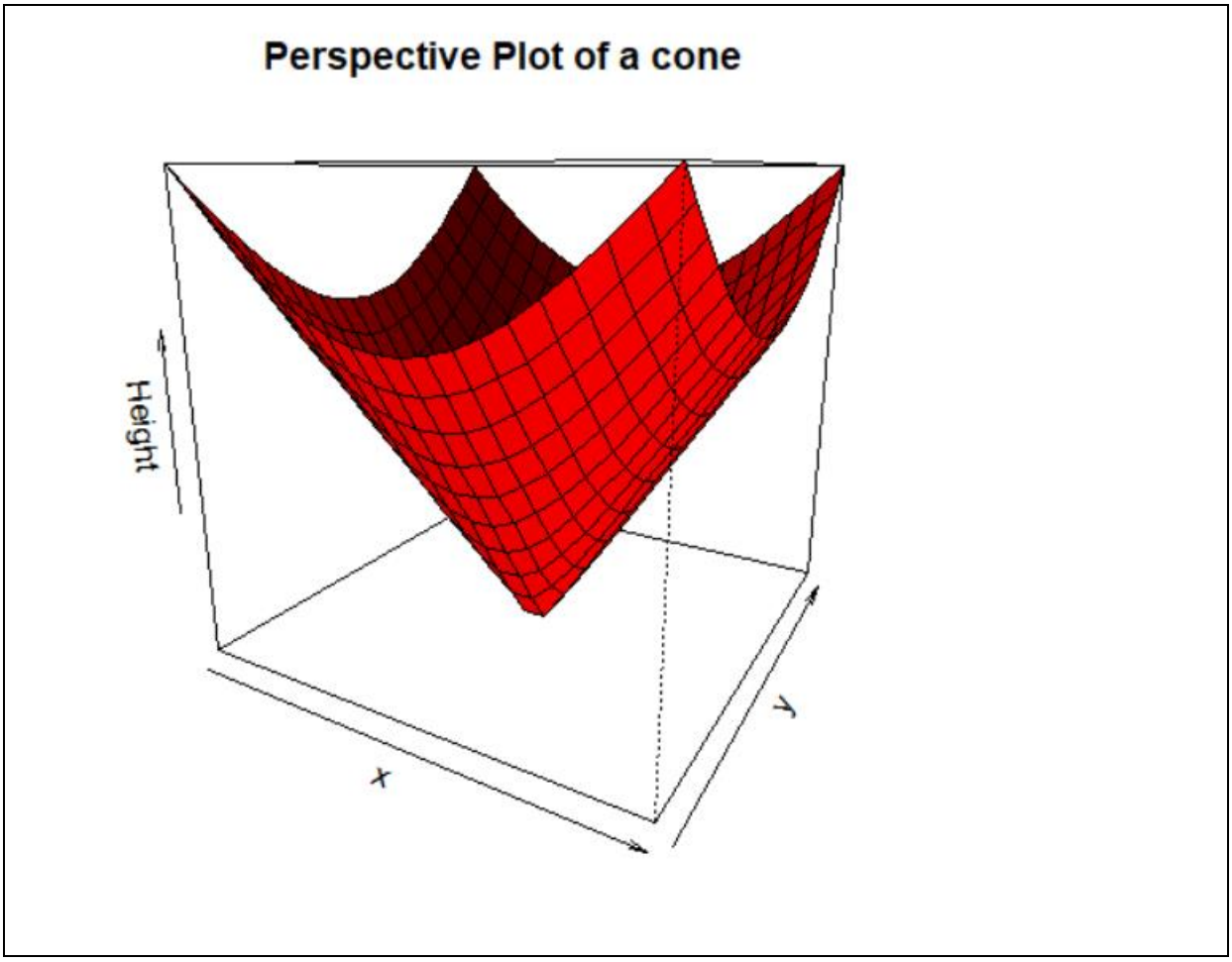
13. Write a program to create 3D plot, to add title, change viewing direction, add colour and shade to the plot.

```
#13.Write a program to create 3D plot, to add title, change
#viewing direction, add color and shade to the plot.

cone <- function(x, y){
  sqrt(x^2+y^2)
}
x <- y <-seq(-1, 1, length=20)
z <- outer(x,  y, cone)
persp(x, y, z)

persp(x, y ,z,
      main = "Perspective Plot of a cone",
      zlab = "Height",
      theta = 30, phi = 15,
      col = "springgreen", shade = 0.5)
```

23:15 viewing direction, add color and shade to the plot.

Console ~/my_R_workSpace/

```
> cone <- function(x, y){
+   sqrt(x^2+y^2)
+ }
> x <- y <-seq(-1, 1, length=20)
> z <- outer(x,  y, cone)
> persp(x, y, z)
>
> persp(x, y ,z,
+       main = "Perspective Plot of a cone",
+       zlab = "Height",
+       theta = 30, phi = 15,
+       col = "red", shade = 0.5)
>
>
>
>
>
>
>
>
>
```


Perspective Plot of a cone
Height
x
y

Program No.14

14. a. Create a data frame from the sample data set. b. Create a table with the needed variables c. Perform the Chi-Square test.

```r
#4 a. Create a data frame from the sample data set.
#b. Create a table with the needed variables
#c. Perform the Chi-Square test.

new_data <- read.csv("https://raw.githubusercontent.com/selva86/datasets/master/treatment.csv")
#new_data
head(new_data)

table(new_data$treatment,new_data$improvement)

chisq.test(new_data$treatment,new_data$improvement,correct = FALSE)
```

42:68 new_data

Console ~/my_R_workSpace/

```
> new_data <- read.csv("https://raw.githubusercontent.com/selva86/datasets/master/treatment.csv")
> #new_data
> head(new_data)
  id   treatment  improvement
1  1     treated     improved
2  2     treated     improved
3  3 not-treated     improved
4  4     treated     improved
5  5     treated not-improved
6  6     treated not-improved
>
> table(new_data$treatment,new_data$improvement)

              improved not-improved
  not-treated       26           29
  treated           35           15
>
> chisq.test(new_data$treatment,new_data$improvement,correct = FALSE)

        Pearson's Chi-squared test

data:  new_data$treatment and new_data$improvement
X-squared = 5.5569, df = 1, p-value = 0.01841

>
```

Program No. 15

15. Perform complete steps of exploratory data analysis on standard data sets (iris flowers, Wine Quality Dataset etc.)

```r
data1<-data(iris)
head(iris)
tail(iris)
dim(iris)
names(iris)
str(iris)
summary(iris)
aggregate(.~Species, iris, mean)
aggregate(.~Species, iris, sd)
data<- iris
table(data$Species)
```

11:20 (Top Level)

Console ~/my_R_workSpace/

```
> data1<-data(iris)
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> tail(iris)
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
145          6.7         3.3          5.7         2.5 virginica
146          6.7         3.0          5.2         2.3 virginica
147          6.3         2.5          5.0         1.9 virginica
148          6.5         3.0          5.2         2.0 virginica
149          6.2         3.4          5.4         2.3 virginica
150          5.9         3.0          5.1         1.8 virginica
> dim(iris)
[1] 150   5
> names(iris)
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
> summary(iris)
```

```
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species  
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199                  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800                  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500                  
> aggregate(.~Species, iris, mean)
     Species Sepal.Length Sepal.Width Petal.Length Petal.Width
1     setosa        5.006       3.428        1.462       0.246
2 versicolor        5.936       2.770        4.260       1.326
3  virginica        6.588       2.974        5.552       2.026
> aggregate(.~Species, iris, sd)
     Species Sepal.Length Sepal.Width Petal.Length Petal.Width
1     setosa    0.3524897   0.3790644    0.1736640   0.1053856
2 versicolor    0.5161711   0.3137983    0.4699110   0.1977527
3  virginica    0.6358796   0.3224966    0.5518947   0.2746501
> data<- iris
> table(data$Species)

    setosa versicolor  virginica 
        50         50         50 
>
```