PROGRAM 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**

Execution code: -

```
# Comma-separated values

import pandas as pd
df = pd.read_csv(“/home/Loan_Prediction/train.csv”)

# XLSX
import pandas as pd
df = pd.read_excel(“/home/Loan_Prediction/world_city.xlsx”, sheetname = “Invoice”)

# ZIP

import zipfile
archive = zipfile.ZipFile('T.zip', 'r')
df = archive.read('train.csv')

# Plain Text (txt)

text_file = open("text.txt", "r")
lines = text_file.read()

# JSON

import pandas as pd
df = pd.read_json(“/home/kunal/Downloads/Loan_Prediction/train.json”)

# XML

import xml.etree.ElementTree as ET
tree = ET.parse('/home/sunilray/Desktop/2 sigma/train.xml')
root = tree.getroot()
print root.tag
```

HTML

```
url="https://en.wikipedia.org/wiki/Delhi"
resp = requests.get(url)
text = resp.text
print(text)
```

Images

```
from scipy import misc
f = misc.face()
misc.imsave('face.png', f) # uses the Image module (PIL)
import matplotlib.pyplot as plt
plt.imshow(f)
plt.show()
```

Hierarchical Data Format

```
Import pandas as pd
t = pd.read_hdf(‘train.h5’)
```

PDF

```
pdf2txt.py<pdf_file>.pdf
```

DOCX

```
pip install docx2txt
import docx2txt
text =docx2txt.process("file.docx")
```

MP3

```
from moviepy.editor import VideoFileClip
clip=VideoFileClip(‘<video_file>.mp3’)
ipython_display(clip)
```

Output: -

	Id	Name	Job
0	1	Raju	Full Stack
1	2	Shyam	Frontend
2	3	Ghanshyam	Backend
3	4	Radheshyam	Data Science

	Id	City	Country
0	1	Delhi	India
1	2	Tokyo	Japan
2	3	Thimpu	Bhutan
3	4	Kathmandu	Nepal

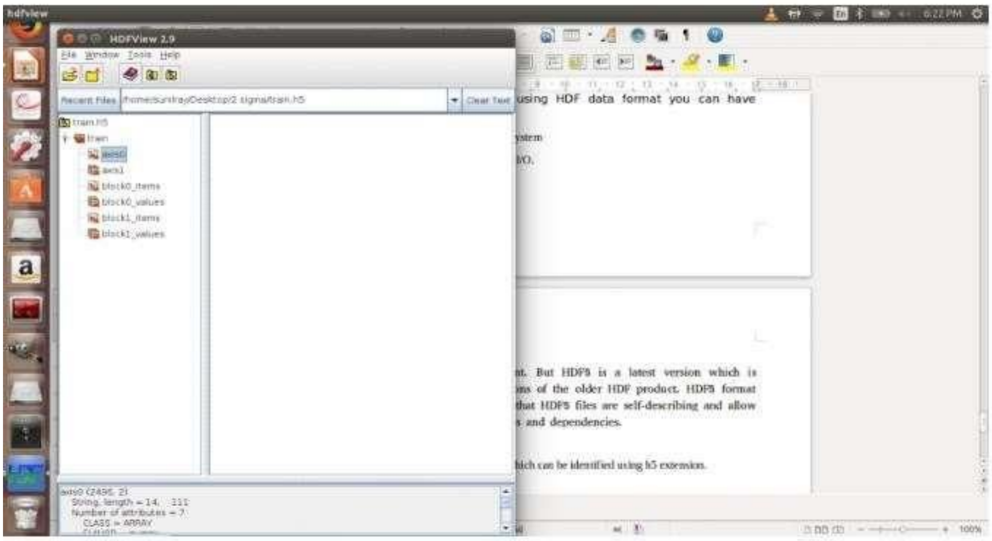
.ipynb_checkpoints	19-Mar-20 3:59 PM	File folder	
Images	21-Mar-20 1:53 AM	File folder	
Importing files	21-Mar-20 1:12 AM	File folder	
AV-Importing data files.ipynb	21-Mar-20 1:46 AM	IPYNB File	2,823 KB
Importing files	21-Mar-20 1:12 AM	Compressed (zipp...	146 KB

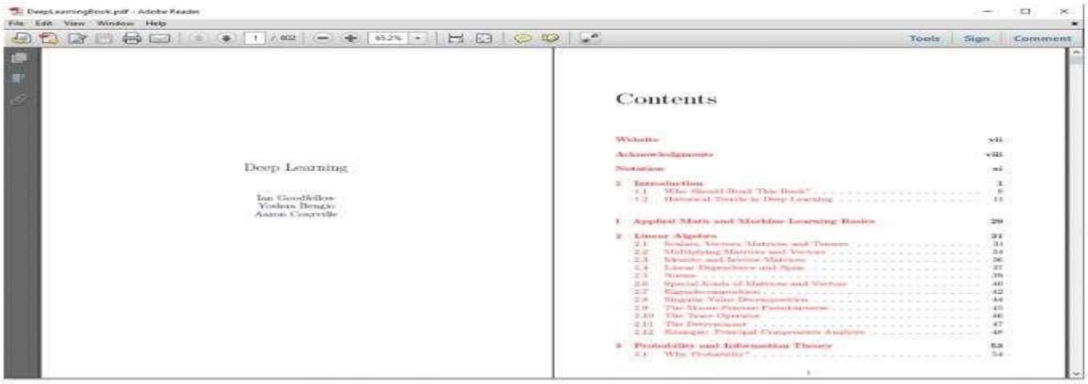
```
"In my previous article, I introduced you to the basics of
(RDD / DataFrame / Dataset) and basics of operations (Trans
learning problem from one of our past hackathons. In this a
my previous article. I will focus on manipulating RDD in Py
(Transformation and Actions)."
```

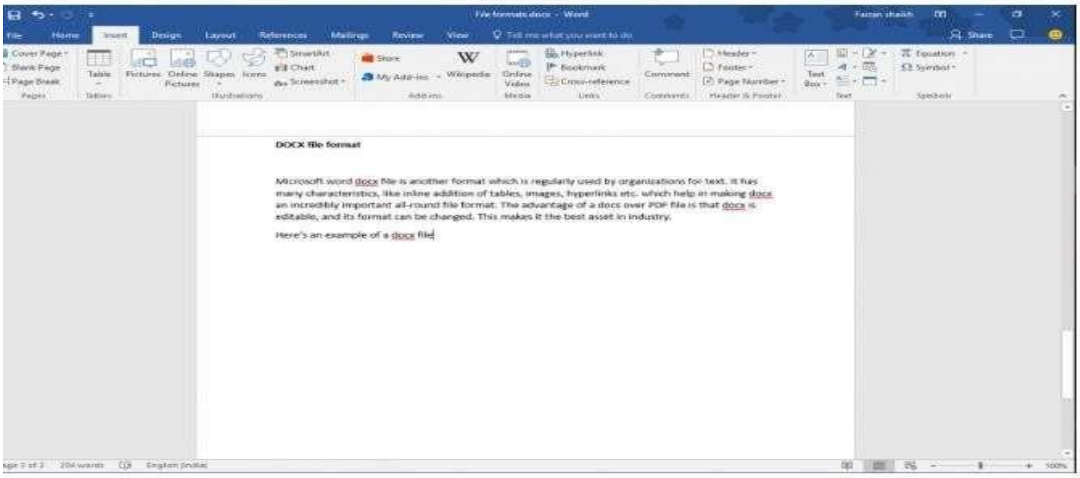
```
<class 'dict'>
```

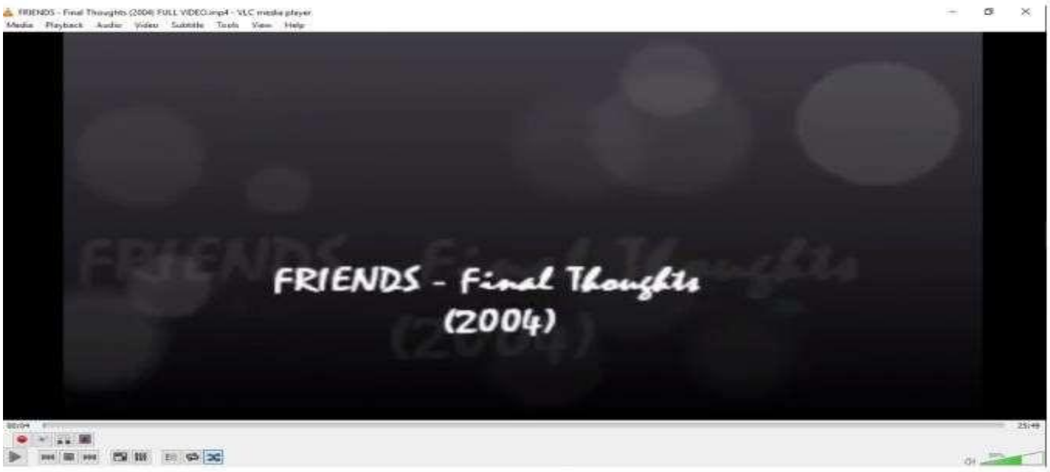
	Name	Company	Job
0	Aniruddha	Analytics Vidhya	Intern
1	Jill	Google	Full time

```
<!DOCTYPE html>
<html class="client-nojs" lang="en" dir="ltr">
<head>
<meta charset="UTF-8"/>
<title>Delhi - Wikipedia</title>
<script>document.documentElement.className="client-js";RLCONF={"wgBreakFrames":!1,"wgSeparatorTransformTable":["",""],"wg
DigitTransformTable":["",""],"wgDefaultDateFormat":"dmy","wgMonthNames":["","January","February","March","April","May","J
une","July","August","September","October","November","December"],"wgRequestId":"XnT6cQpAICsAAGoWFuwAAADB","wgCSPNonce":!
1,"wgCanonicalNamespace":"","wgCanonicalSpecialPageName":!1,"wgNamespaceNumber":0,"wgPageName":"Delhi","wgTitle":"Delh
i","wgCurRevisionId":946313369,"wgRevisionId":946313369,"wgArticleId":37756,"wgIsArticle":!0,"wgIsRedirect":!1,"wgActio
n":"view","wgUserName":null,"wgUserGroups":["*"],"wgCategories":["CS1 errors: missing periodical","Webarchive template wa
yback links","CS1 maint: archived copy as title","Articles with short description","Wikipedia indefinitely semi-protected
pages","Wikipedia indefinitely move-protected pages","Use Indian English from October 2019","All Wikipedia articles writt
en in Indian English",
"Use dmy dates from March 2020","Coordinates on Wikidata","Articles containing potentially dated statements from 2016","A
ll articles containing potentially dated statements","Articles containing Persian-language text","Articles containing pot
entially dated statements from 2013","All articles with unsourced statements","Articles with unsourced statements from Ap
ril 2018","Articles containing potentially dated statements from 2015","Pages using multiple image with auto scaled image
```




Contents
Deep Learning


DOCX file format
Microsoft word docx file is another format which is regularly used by organizations for text. It has many characteristics, like inline addition of tables, images, hyperlinks etc. which help in making docx an incredibly important all-round file format. The advantage of a docs over PDF file is that docx is editable, and its format can be changed. This makes it the best asset in industry.
Here's an example of a docx file


FRIENDS - Final Thoughts
(2004)

PROGRAM 9

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

Execution code: -

```
library(tidyverse)
View(iris)
df <- data.frame(iris)
grp_spc <- group_by(iris,Species)
summarise(iris,cnt=n())
summarise(grp_spc,cnt=n())
summarize(iris,mn = mean(Sepal.Length),sd = sd(Sepal.Length))
summarize(iris,mn = mean(Sepal.Width),sd = sd(Sepal.Width))
summarize(iris,mn = mean(Petal.Length),sd = sd(Petal.Length))
summarize(iris,mn = mean(Petal.Width),sd = sd(Petal.Width))
summarize(grp_spc,mn = mean(Sepal.Length),sd = sd(Sepal.Length))
summarize(grp_spc,mn = mean(Sepal.Width),sd = sd(Sepal.Width))
summarize(grp_spc,mn = mean(Petal.Length),sd = sd(Petal.Length))
summarize(grp_spc,mn = mean(Petal.Width),sd = sd(Petal.Width))
```

Output: -

```
> library(tidyverse)
> View(iris)
> df <- data.frame(iris)
> grp_spc <- group_by(iris,Species)
> summarise(iris,cnt=n())
  cnt
1 150
> summarise(grp_spc,cnt=n())
# A tibble: 3 x 2
  Species      cnt
  <fct>      <int>
1 setosa        50
2 versicolor    50
3 virginica     50
> summarize(iris,mn = mean(Sepal.Length),sd = sd(Sepal.Length))
        mn        sd
1 5.843333 0.8280661
> summarize(iris,mn = mean(Sepal.Width),sd = sd(Sepal.Width))
        mn        sd
1 3.057333 0.4358663
> summarize(iris,mn = mean(Petal.Length),sd = sd(Petal.Length))
     mn       sd
1 3.758 1.765298
> summarize(iris,mn = mean(Petal.Width),sd = sd(Petal.Width))
        mn        sd
1 1.199333 0.7622377
```

```
> summarize(grp_spc,mn = mean(Sepal.Length),sd = sd(Sepal.Length))
# A tibble: 3 × 3
  Species       mn    sd
  <fct>      <dbl> <dbl>
1 setosa      5.01 0.352
2 versicolor  5.94 0.516
3 virginica   6.59 0.636
> summarize(grp_spc,mn = mean(Sepal.Width),sd = sd(Sepal.Width))
# A tibble: 3 × 3
  Species       mn    sd
  <fct>      <dbl> <dbl>
1 setosa      3.43 0.379
2 versicolor  2.77 0.314
3 virginica   2.97 0.322
> summarize(grp_spc,mn = mean(Petal.Length),sd = sd(Petal.Length))
# A tibble: 3 × 3
  Species       mn    sd
  <fct>      <dbl> <dbl>
1 setosa      1.46 0.174
2 versicolor  4.26 0.470
3 virginica   5.55 0.552
> summarize(grp_spc,mn = mean(Petal.Width),sd = sd(Petal.Width))
# A tibble: 3 × 3
  Species       mn    sd
  <fct>      <dbl> <dbl>
1 setosa     0.246 0.105
2 versicolor 1.33  0.198
3 virginica  2.03  0.275
> 
```

Program 12

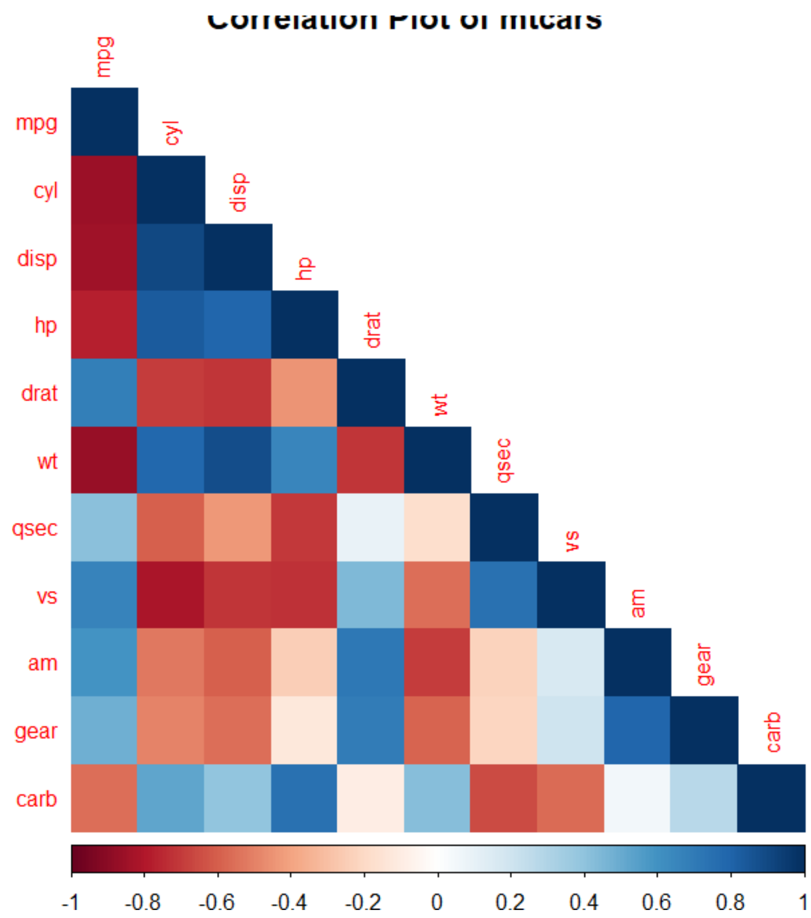
Find the correlation matrix:

A:-Plot the correlation plot on dataset and visualize giving an overview of relationship among variable on dataset.

Input: -

```
library(corrplot)

data(mtcars)

correlation_matrix <- cor(mtcars)

corrplot(correlation_matrix, method = "color", type = "lower", tl.cex = 0.8, title = "Correlation Plot of mtcars")
```

Output: -



B:-Analysis of covariance :variance (ANOVA) if data have categorical variables on data set.

Input:

data(mtcars)

model <- aov(mpg ~ am + hp, data = mtcars)

summary(model)

Output:

```
> summary(model)
            Df Sum Sq Mean Sq F value   Pr(>F)    
am           1  405.2   405.2   47.87 1.33e-07 ***
hp           1  475.5   475.5   56.18 2.92e-08 ***
Residuals   29  245.4     8.5                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> 
```

Program 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

Input: -

```
df <- mtcars
df
name <- c("Alice", "Bob", "Charlie", "David", "Eva")
age <- c(25, 30, 22, 35, 28)
score <- c(85, 92, 78, 88, 95)
df <- data.frame(Name = name, Age = age, Score = score)
df
name <- c("Alice", "Bob", "Charlie", "David", "Eva")
age <- c(25, 30, 22, 35, 28)
score <- c(85, 92, 78, 88, 95)
df <- data.frame(Name = name, Age = age, Score = score)
library(knitr)
kable(df, caption = "Sample Data Table")
name <- c("Alice", "Bob", "Charlie", "David", "Eva")
age <- c(25, 30, 22, 35, 28)
age_group <- cut(age, breaks = c(20, 25, 30, 35, 40), labels = c("20-25", "26-30", "31-35", "36-
40"))
contingency_table <- table(name, age_group)
chisq_test <- chisq.test(contingency_table)
print(chisq_test)
```

Output: -

Console Terminal × Background Jobs ×

R 4.3.2 · C:/Users/Harsh Raj/AppData/Local/Temp/354232b0-9aeb-4565-b7aa-7ccc920c8701_programR[1].zip.701/

```
> df <- mtcars
> df
                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
> name <- c("Alice", "Bob", "Charlie", "David", "Eva")
> age <- c(25, 30, 22, 35, 28)
> score <- c(85, 92, 78, 88, 95)
> df <- data.frame(Name = name, Age = age, Score = score)
> df
```

```
> df
     Name Age Score
1   Alice  25    85
2     Bob  30    92
3 Charlie  22    78
4   David  35    88
5     Eva  28    95
> name <- c("Alice", "Bob", "Charlie", "David", "Eva")
> age <- c(25, 30, 22, 35, 28)
> score <- c(85, 92, 78, 88, 95)
> df <- data.frame(Name = name, Age = age, Score = score)
> library(knitr)
> kable(df, caption = "Sample Data Table")


Table: Sample Data Table

|Name    | Age| Score|
|:-------|---:|-----:|
|Alice   |  25|    85|
|Bob     |  30|    92|
|Charlie |  22|    78|
|David   |  35|    88|
|Eva     |  28|    95|
> name <- c("Alice", "Bob", "Charlie", "David", "Eva")
> age <- c(25, 30, 22, 35, 28)
> age_group <- cut(age, breaks = c(20, 25, 30, 35, 40), labels = c("20-25", "26-30", "31-35", "36-4
0"))
> contingency_table <- table(name, age_group)
> chisq_test <- chisq.test(contingency_table)
Warning message:
In chisq.test(contingency_table) :
  Chi-squared approximation may be incorrect
> print(chisq_test)

        Pearson's Chi-squared test

data:  contingency_table
X-squared = NaN, df = 12, p-value = NA

>
```

Program-15

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

Input:

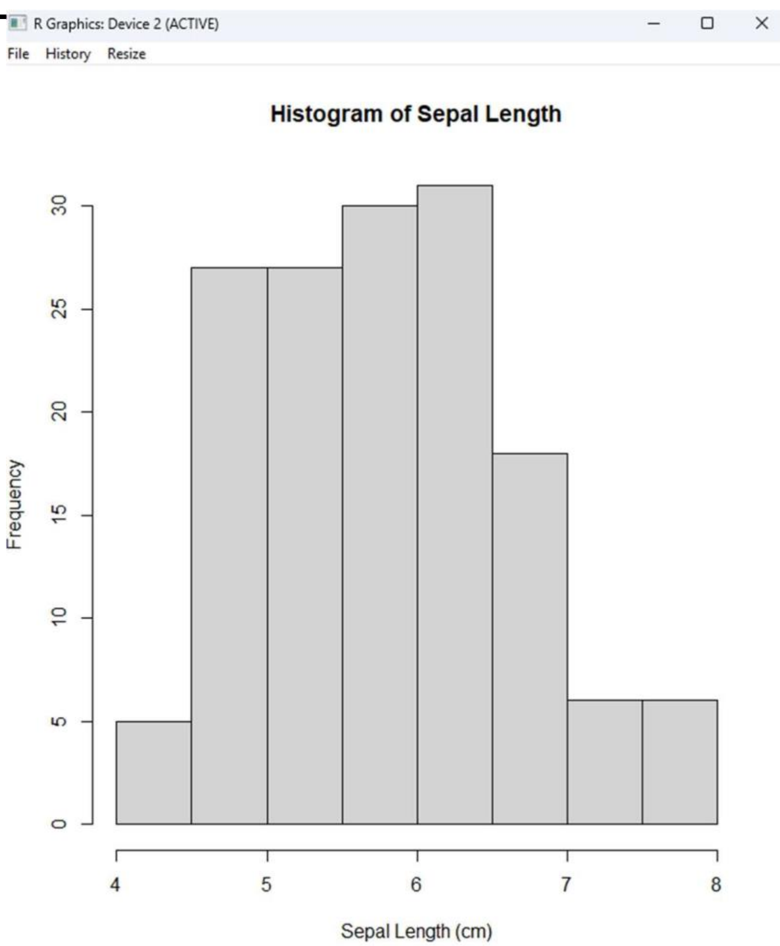
```
data(iris)
head(iris)
summary(iris)
hist(iris$Sepal.Length, main = "Histogram of Sepal Length", xlab = "Sepal Length (cm)")
hist(iris$Sepal.Width, main = "Histogram of Sepal Width", xlab = "Sepal Width (cm)")
hist(iris$Petal.Length, main = "Histogram of Petal Length", xlab = "Petal Length (cm)")
hist(iris$Petal.Width, main = "Histogram of Petal Width", xlab = "Petal Width (cm)")

boxplot(iris$Sepal.Length, main = "Boxplot of Sepal Length")
boxplot(iris$Sepal.Width, main = "Boxplot of Sepal Width")
boxplot(iris$Petal.Length, main = "Boxplot of Petal Length")
boxplot(iris$Petal.Width, main = "Boxplot of Petal Width")

pairs(iris[, 1:4], pch = 19, col = iris$Species)

cor(iris[, 1:4])
```

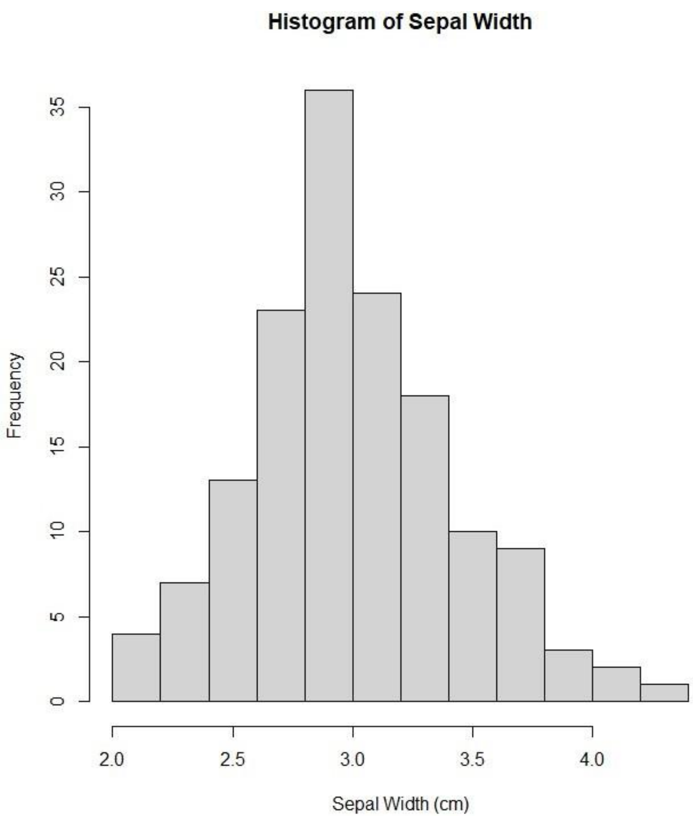
Output:

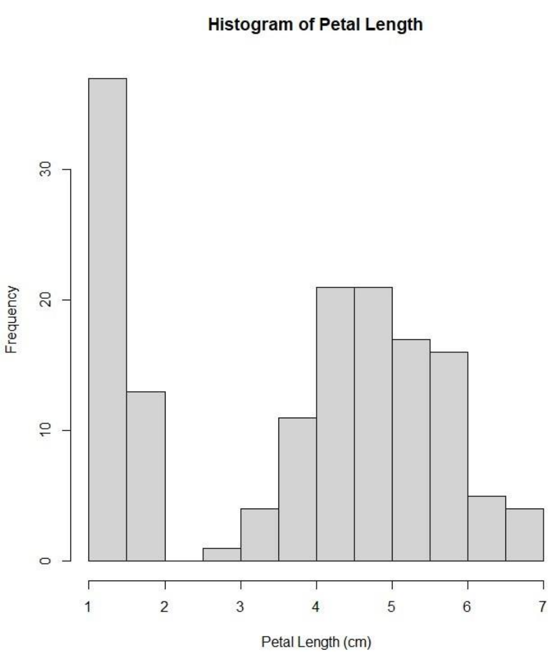
Console Terminal Jobs

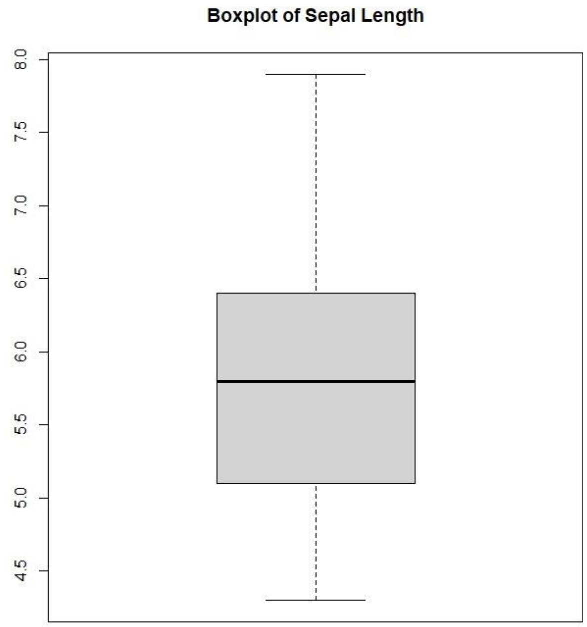
R 4.3.1 · ~/

```
> 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
> 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
> hist(iris$Sepal.Length, main = "Histogram of Sepal Length", xlab = "Sepal Length (cm)")
> hist(iris$Sepal.Width, main = "Histogram of Sepal Width", xlab = "Sepal Width (cm)")
> hist(iris$Petal.Length, main = "Histogram of Petal Length", xlab = "Petal Length (cm)")
> hist(iris$Petal.Width, main = "Histogram of Petal Width", xlab = "Petal Width (cm)")
> boxplot(iris$Sepal.Length, main = "Boxplot of Sepal Length")
> boxplot(iris$Sepal.Width, main = "Boxplot of Sepal Width")
> boxplot(iris$Petal.Length, main = "Boxplot of Petal Length")
> boxplot(iris$Petal.Width, main = "Boxplot of Petal Width")
> pairs(iris[, 1:4], pch = 19, col = iris$Species)
> cor(iris[, 1:4])
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
> 
```

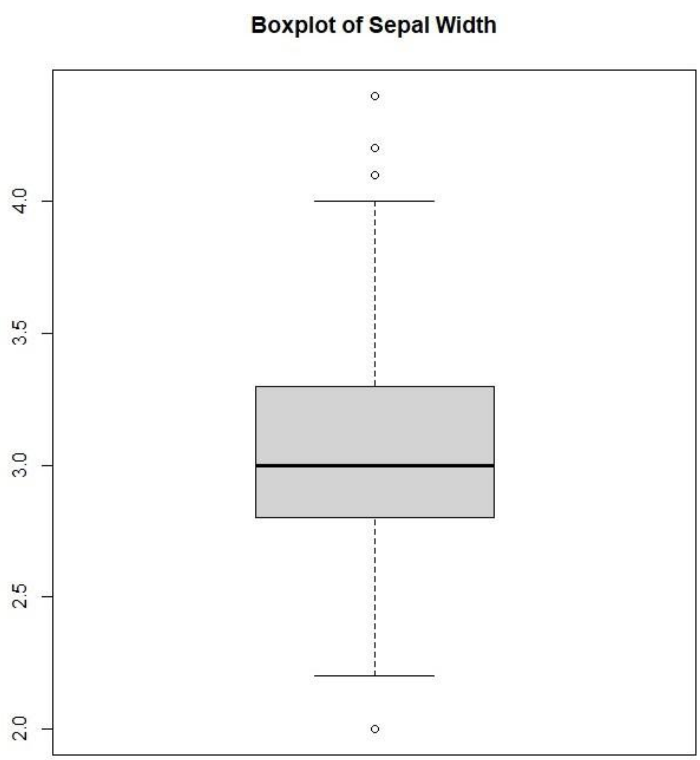



Histogram of Sepal Width
Frequency
0
5
10
15
20
25
30
35
2.0
2.5
3.0
3.5
4.0
Sepal Width (cm)

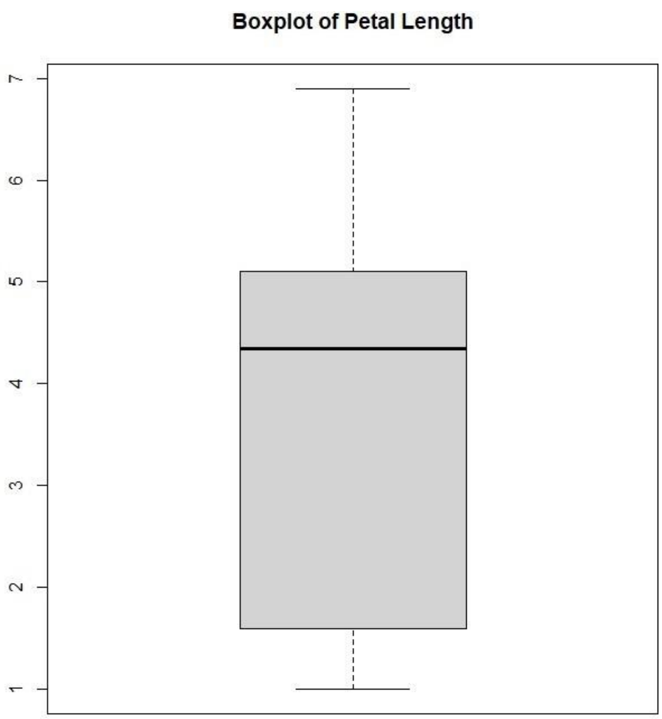

Histogram of Petal Length
Frequency
0
10
20
30
1
2
3
4
5
6
7
Petal Length (cm)

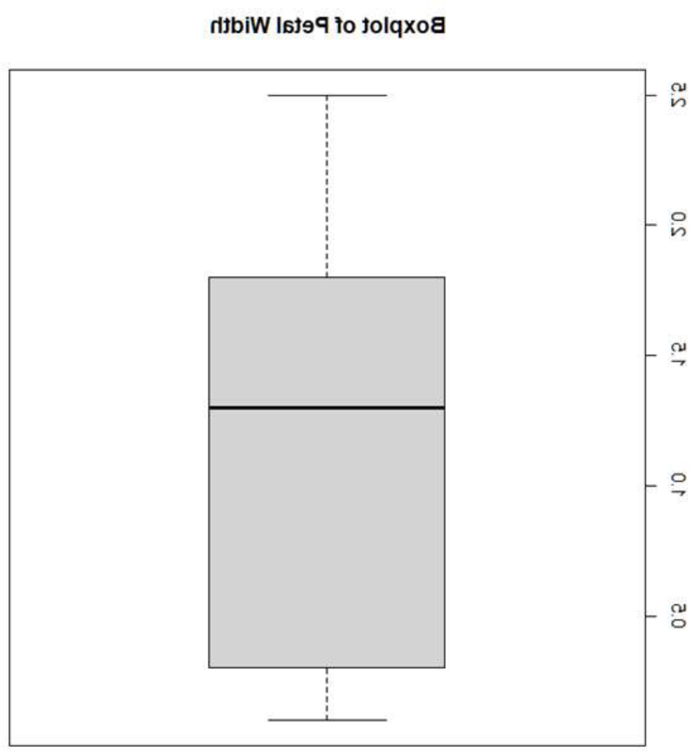


Boxplot of Sepal Length
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5



Boxplot of Sepal Width
4.0
3.5
3.0
2.5
2.0


Boxplot of Petal Length
1
2
3
4
5
6
7


Boxplot of Petal Width
0.5
1.0
1.5
2.0
2.5

<u>Program :- 10</u>

1. 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]

Input Code:

a.

```
 # Sample data generation
set.seed(123)  # Set a seed for reproducibility
data <- data.frame(
  Group = rep(1:3, each = 50),  # Three groups
  Value = c(rnorm(50), rnorm(50, mean = 2), rnorm(50, mean = 4))  # Three sets of
random values
)

# Create a boxplot
boxplot(data$Value ~ data$Group,
        main = "Boxplot of Data Distributions",
        xlab = "Group",
        ylab = "Value",
        col = "lightblue",
        border = "blue")

# Create a scatter plot
plot(data$Group, data$Value,
     col = data$Group,
     pch = 19,
     main = "Scatter Plot of Data Distributions",
     xlab = "Group",
     ylab = "Value")

legend("topright", legend = unique(data$Group), col = 1:3, pch = 19)
```

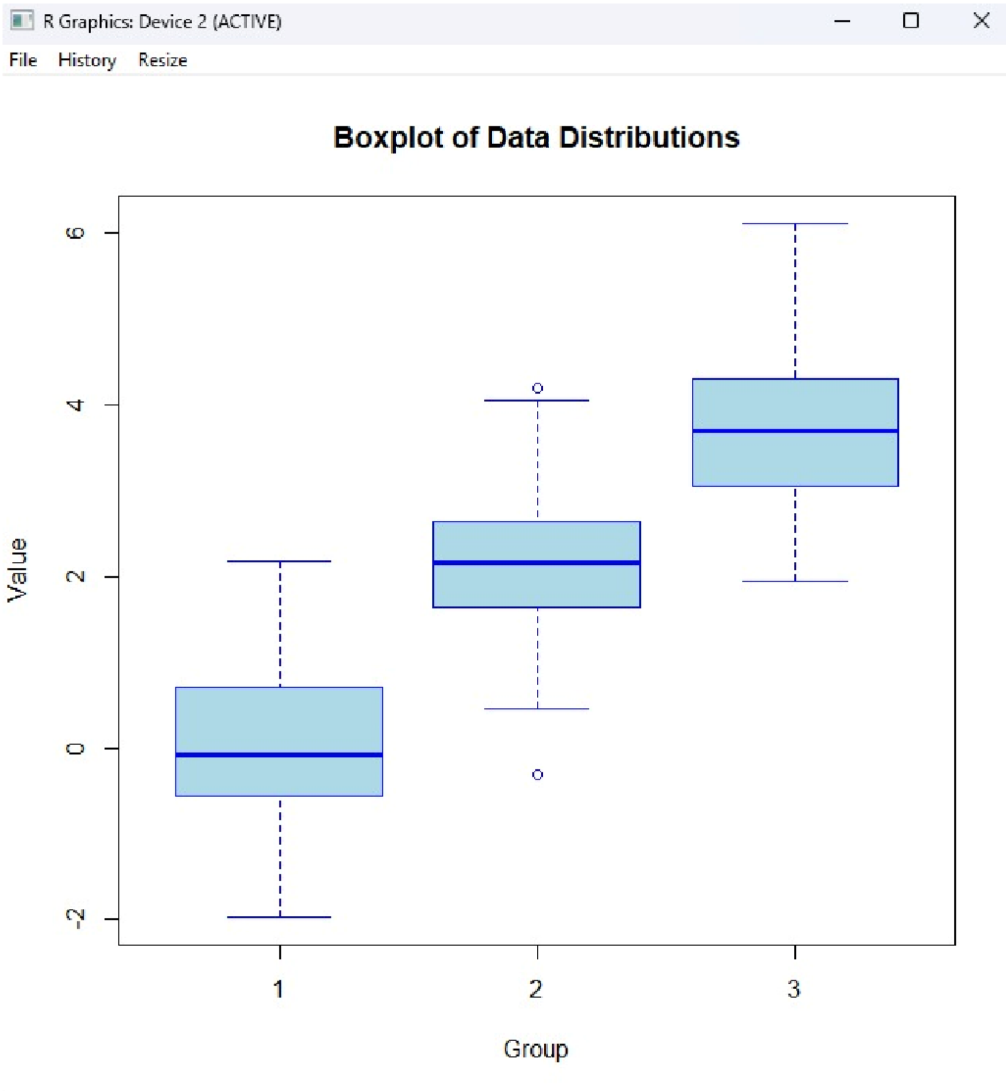
b.

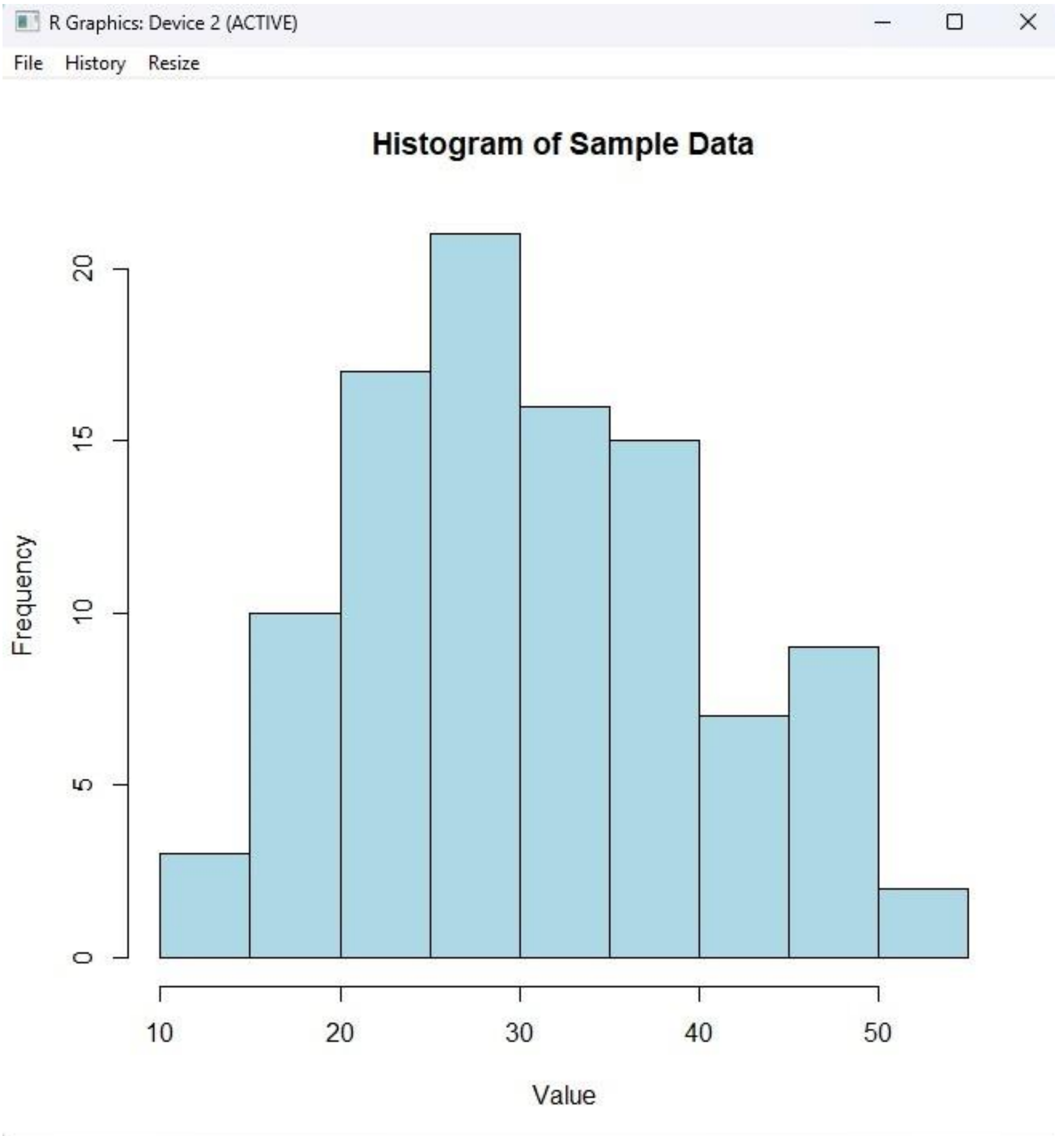
```
# Sample data generation
set.seed(123)  # Set a seed for reproducibility
data <- data.frame(
  Value = c(rnorm(100), 5, 6, 15, 18)  # 100 normal values and 4 outliers
)

# Create a boxplot
boxplot(data$Value,
        main = "Boxplot with Outliers")

# Identify potential outliers
outliers <- boxplot(data$Value)$out

# Print the potential outliers
cat("Potential Outliers:", outliers, "\n")
```

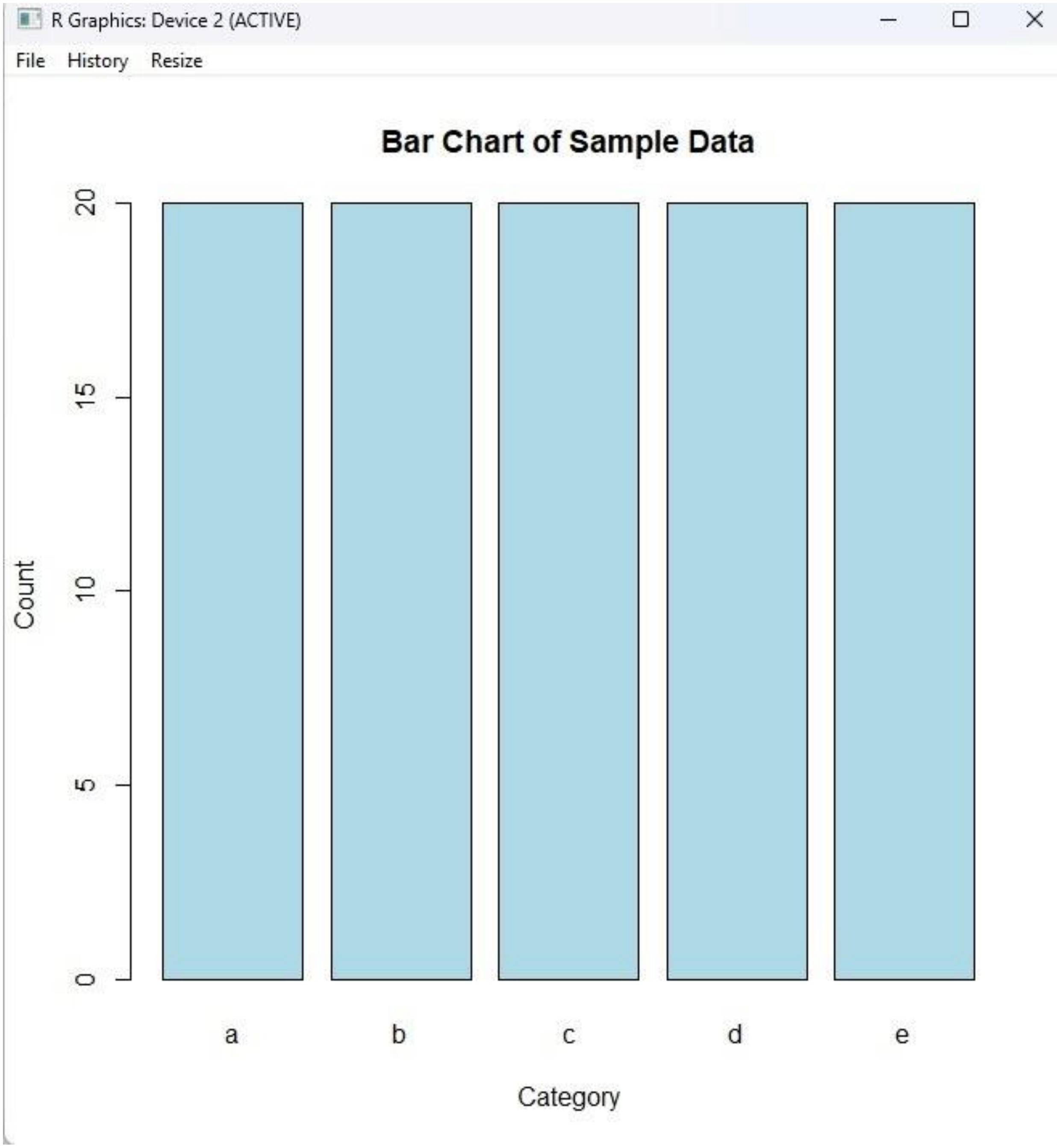
c.

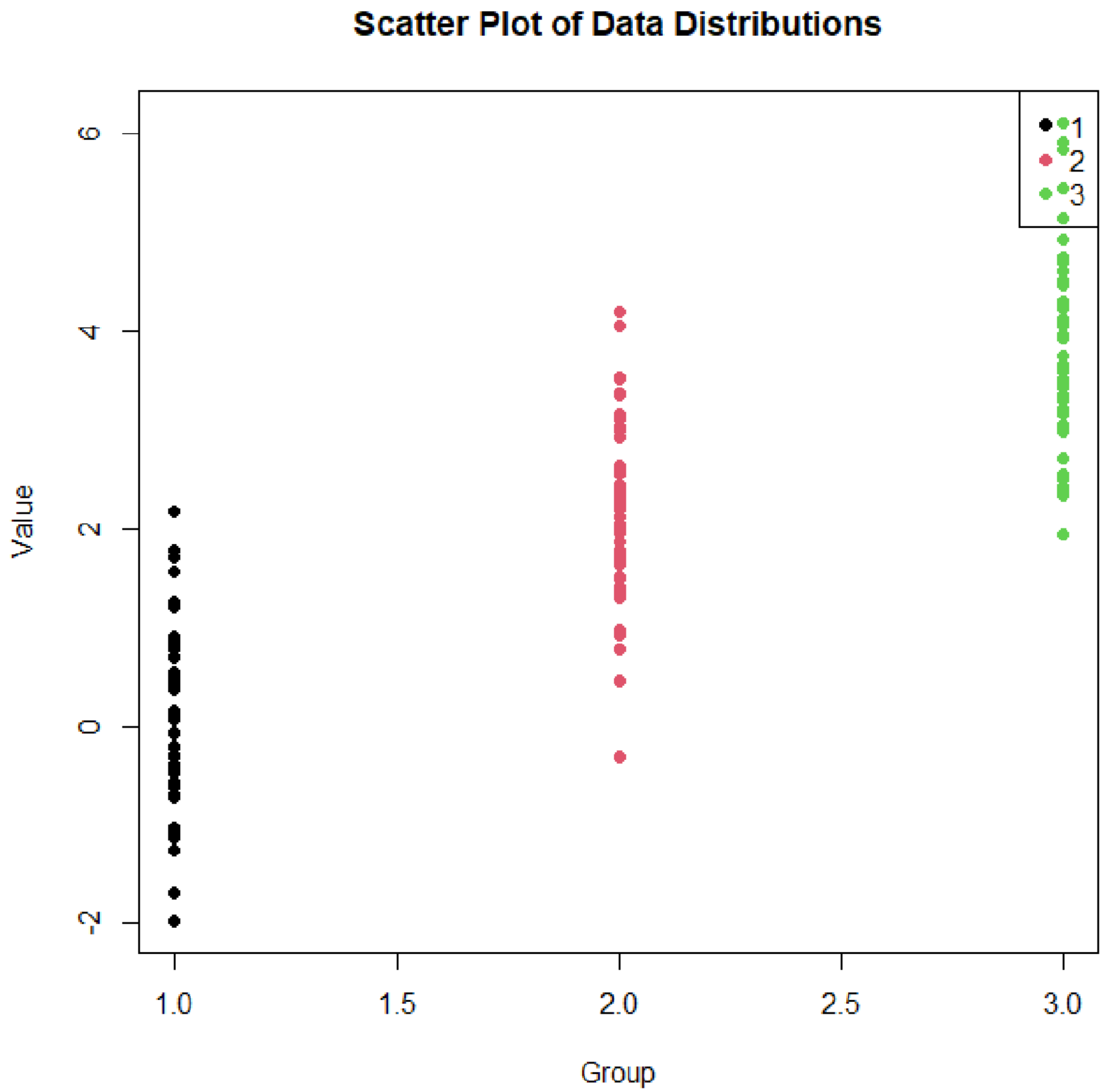
d.

```
# Sample data generation
set.seed(123) # Set a seed for reproducibility
sample_data <- data.frame(
  Category = rep(letters[1:5], each = 20),  # Five categories
  Value = c(rnorm(20, mean = 20, sd = 5), rnorm(20, mean = 40, sd = 8),
            rnorm(20, mean = 30, sd = 7), rnorm(20, mean = 25, sd = 6),
            rnorm(20, mean = 35, sd = 9))
)

# Create a histogram
hist(sample_data$Value, breaks = 10, main = "Histogram of Sample Data", xlab =
"Value", col = "lightblue")


# Create a Histogram
hist(sample_data$Value,
     breaks = 10,
     main = "Histogram of Sample Data",
     xlab = "Value",
     col = "lightblue")

# Create a Bar Chart
category_counts <- table(sample_data$Category)
barplot(category_counts,
        main = "Bar Chart of Sample Data",
        xlab = "Category",
        ylab = "Count",
        col = "lightblue")

# Create a Pie Chart
category_proportions <- prop.table(category_counts)
colors <- rainbow(length(category_proportions))

pie(category_proportions,
    labels = category_counts,
    main = "Pie Chart of Sample Data",
    col = colors)
```

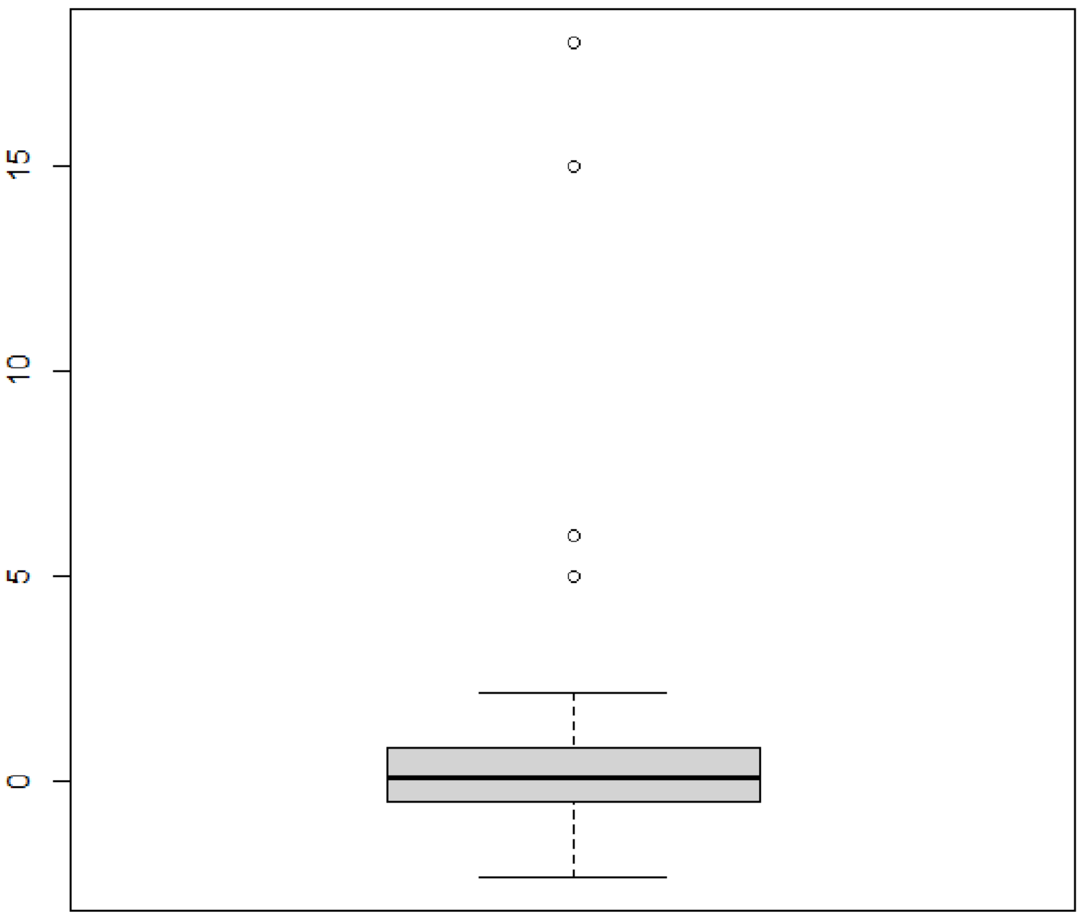

R Graphics: Device 2 (ACTIVE)
File History Resize
Boxplot of Data Distributions
Value
-2
0
2
4
6
1
2
3
Group

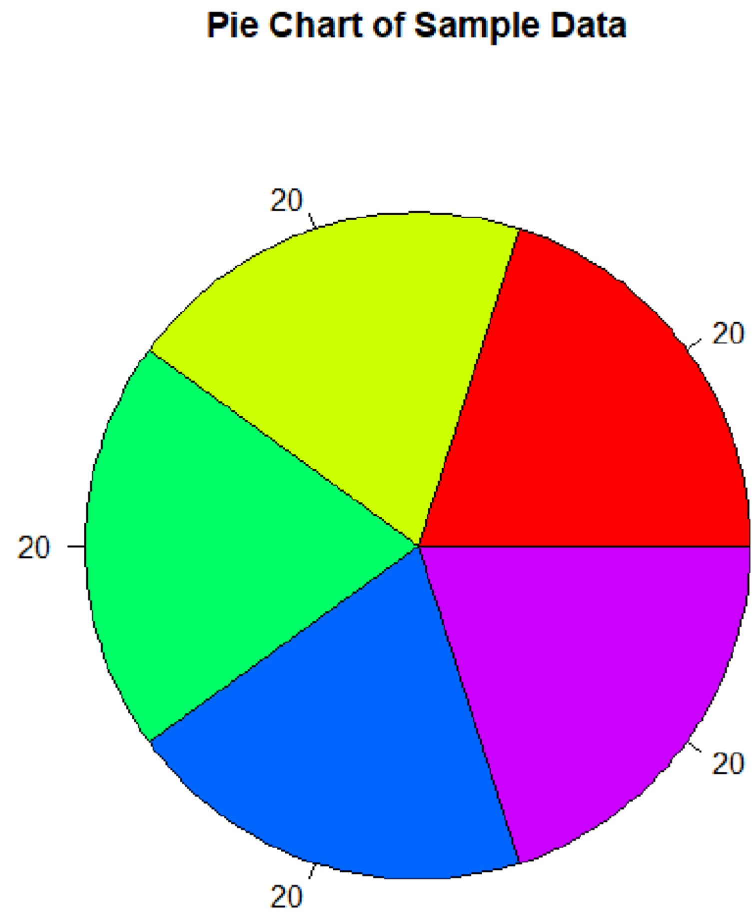

R Graphics: Device 2 (ACTIVE)
File History Resize
Histogram of Sample Data
Frequency
0
5
10
15
20
10
20
30
40
50
Value


R Graphics: Device 2 (ACTIVE)
File History Resize
Bar Chart of Sample Data
Count
0
5
10
15
20
a
b
c
d
e
Category



Scatter Plot of Data Distributions
Value
Group
1
2
3
-2
0
2
4
6
1.0
1.5
2.0
2.5
3.0


15
10
5
0


Pie Chart of Sample Data
20
20
20
20
20

Program :- 11

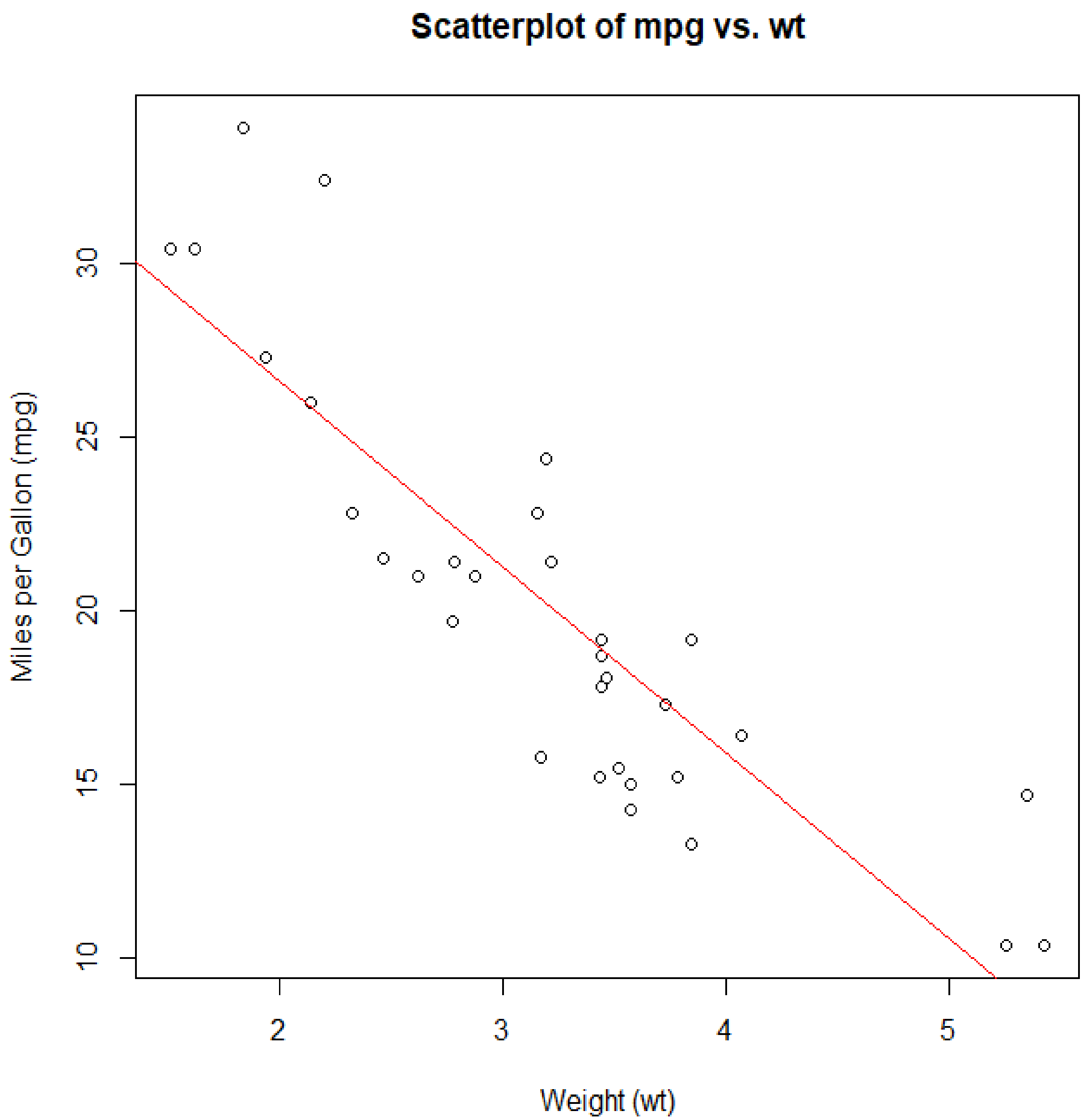
1. Import a sample dataset and perform Regression techniques to find out relation between variables.

Input Code:

```
# Load the dataset
data(mtcars)
# View the first few rows of the dataset
head(mtcars)
# Perform a linear regression
linear_model <- lm(mpg ~ wt, data = mtcars)
# Summary of the linear regression
summary(linear_model)
# Plot the regression line
plot(mtcars$wt, mtcars$mpg, main = "Scatterplot of mpg vs. wt", xlab = "Weight (wt)", ylab = "Miles per Gallon (mpg)")
abline(linear_model, col = "red")
# Perform a multiple linear regression
multi_linear_model <- lm(mpg ~ wt + hp + qsec, data = mtcars)
# Summary of the multiple linear regression
summary(multi_linear_model)
```

Output:

```
> # Load the dataset
> data(mtcars)
> # View the first few rows of the dataset
> 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
> # Perform a linear regression
> linear_model <- lm(mpg ~ wt, data = mtcars)
> # Summary of the linear regression
> summary(linear_model)

Call:
lm(formula = mpg ~ wt, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.5432 -2.3647 -0.1252  1.4096  6.8727 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
wt           -5.3445     0.5591  -9.559 1.29e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.046 on 30 degrees of freedom
Multiple R-squared:  0.7528,	Adjusted R-squared:  0.7446 
F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10

> # Plot the regression line
> plot(mtcars$wt, mtcars$mpg, main = "Scatterplot of mpg vs. wt", xlab = "Weight (wt)", ylab = "Miles per Gallon (mpg)")
> abline(linear_model, col = "red")
> # Perform a multiple linear regression
> multi_linear_model <- lm(mpg ~ wt + hp + qsec, data = mtcars)
> # Summary of the multiple linear regression
> summary(multi_linear_model)

Call:
lm(formula = mpg ~ wt + hp + qsec, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.8591 -1.6418 -0.4636  1.1940  5.6092 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 27.61053    8.41993   3.279  0.00278 ** 
wt          -4.35880    0.75270  -5.791 3.22e-06 ***
hp          -0.01782    0.01498  -1.190  0.24418    
qsec         0.51083    0.43922   1.163  0.25463    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.578 on 28 degrees of freedom
Multiple R-squared:  0.8348,	Adjusted R-squared:  0.8171 
F-statistic: 47.15 on 3 and 28 DF,  p-value: 4.506e-11
```


Scatterplot of mpg vs. wt
Miles per Gallon (mpg)
10
15
20
25
30
2
3
4
5
Weight (wt)

Program :- 13

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

Input Code:

```
# Create sample data
x <- seq(-10, 10, length.out = 50)
y <- seq(-10, 10, length.out = 50)
z <- outer(x, y, function(x, y) sin(sqrt(x^2 + y^2)) / (sqrt(x^2 + y^2)))

# Create the 3D plot
persp(x, y, z,
      main = "3D Plot with Title",
      theta = 30, phi = 20, # Change viewing angles
      col = "skyblue",      # Set color
      shade = 0.5)          # Add shading
```

Output: