```
> cat("Original vector:",x,"\n")
Original vector: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
> #subsetting vector:
> cat("First 5 values of vector:",x[1:5],"\n")
First 5 values of vector: 1 2 3 4 5
> cat("Without values present at index 1,2and 3:",x[-c(1,2,3)])
Without values present at index 1,2and 3: 4 5 6 7 8 9 10 11 12 13 14 15> #Subsetting in R
using [[]]operator:
> #create list:
> ls<-list(a=1,b=2,c=10,d=20)
> cat("Original List:\n")
Original List:
> print(ls)
$a
[1] 1
$b
[1] 2
$c
[1] 10
$d
[1] 20

> #select first element of list:
> cat("Element of list:",ls[[3]],"\n")
Element of list: 10
> #Subsetting using c() function:
> ls2<-list(a=list(x=1,y="students").b=1:10)
> ls2
$a
$a$x
[1] 1
```

```
$a$y
[1] "students"


$b
[1]  1  2  3  4  5  6  7  8  9 10

> cat("Using c() function:\n")
Using c() function:
> print(ls2[[c(1,2)]])
[1] "students"
> print(ls2[[1]][[2]])
[1] "students"
> #Subsetting Using $ operator:
> ls3<-list(a="Roshani",b=1,c="Hello")
> ls3
$a
[1] "Roshani"
$b
[1] 1
$c
[1] "Hello"
> cat("Using $ operator:\n")
Using $ operator:
> print(ls3$a)
[1] "Roshani"
> #-----------------------------------------------------------------------------------
> #4)Merging:-
> #Merge DataFrames by Row Names:-
> data_frame1<-data.frame(No=c(1:5),
+                  Name=letters[1:5],
+                  Salary=c(200,200,300,NA,300)
```

```
                     )
> data_frame1
  No Name Salary
1  1    a    200
2  2    b    200
3  3    c    300
4  4    d     NA
5  5    e    300
>
> data_frame2<-data.frame(No=c(6:8),
+                      Name=letters[8:10],
+                      Salary=c(400,350,NA)
+                      )
> data_frame2
  No Name Salary
1  6    h    400
2  7    i    350
3  8    j     NA
>
> data_frame_merge<-merge(data_frame1,data_frame2,by='row.names',all=TRUE)
> print("Merged DataFrame")
[1] "Merged DataFrame"
> print(data_frame_merge)
  Row.names No.x Name.x Salary.x No.y Name.y Salary.y
1         1    1      a      200    6      h      400
2         2    2      b      200    7      i      350
3         3    3      c      300    8      j       NA
4         4    4      d       NA   NA   <NA>       NA
5         5    5      e      300   NA   <NA>       NA
> #------------------------------------------------------------------------
> #5)Joining:-
> #Using Inner join:-
> data1<-data.frame(ID=c(1:5))

> data2<-data.frame(ID=c(4:8))

> inner_join(data1,data2,by="ID")

 ID

1  4

2  5

>

> #Using Left join:-

> data1<-data.frame(ID=c(1:5),

+                  Name=c("Rutuja","Lokesh","Ram","Purvi","Nita"))

> data2<-data.frame(ID=c(4:8),

+                  Marks=c(70,85,80,90,75))

> left_join(data1,data2,by="ID")

  ID   Name Marks

1  1 Rutuja    NA

2  2 Lokesh    NA

3  3    Ram    NA

4  4  Purvi    70

5  5   Nita    85
```

#Validating data:-

```
data(cars)

head(cars, 3)

library(validate)

rules <- validator(speed >= 0,

                   dist >= 0,

                   speed/dist <= 1.5,

                   cor(speed, dist)>=0.2)

out <- confront(cars, rules)

summary(out)
```

Output: -

```
data(cars)

> head(cars, 3)

  speed dist

1     4    2

2     4   10

3     7    4

>

> library(validate)

> rules <- validator(speed >= 0,

+                   dist >= 0,

+                   speed/dist <= 1.5,

+                   cor(speed, dist)>=0.2)

> out <- confront(cars, rules)

> summary(out)

  name items passes fails nNA error warning          expression

1   V1    50     50     0   0 FALSE   FALSE  speed - 0 >= -1e-08

2   V2    50     50     0   0 FALSE   FALSE   dist - 0 >= -1e-08

3   V3    50     48     2   0 FALSE   FALSE     speed/dist <= 1.5

4   V4     1      1     0   0 FALSE   FALSE cor(speed, dist) >= 0.2
```

Experiment No: 6
Experiment Name: Write program to implement the following analysis techniques using R.
1.Statistical hypothesis generation and testing
2. Chi-Square test
3. t-Test
4. Correlation analysis
Name:
Roll No:-

1)Stastical hypothesis testing:-

```
#One-sample T-testing:

x<-rnorm(100)#sample vector

t.test(x,mu=5)#one-sample t-test
```

```
#two sample T-testing:

x<-rnorm(100)

y<-rnorm(100)

t.test(x,y)

#Directional Hypothesis:-

t.test(x,mu=2,alternative='greater')

#one sample u-test:-

wilcox.test(y,exact = FALSE)

#Two sample u-test:-

wilcox.test(x,y)
```

2)Correlation Test:-

```
cor.test(matcars$mpg,matcars$hp)
```

3)Chi-Square Test:-

```
library(MASS)

#create DataFrame:

print(str(survey))

# Create a data frame from the main data set.

stu_data = data.frame(survey$Smoke,survey$Exer)


# Create a contingency table with the needed variables.

stu_data = table(survey$Smoke,survey$Exer)


print(stu_data)
```

OUTPUT:-

```
#1)Stastical hypothesis testing:-

> #One-sample T-testing:

> x<-rnorm(100)#sample vector

> t.test(x,mu=5)#one-sample t-test


        One Sample t-test
```

```
data:  x
t = -52.314, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 5
95 percent confidence interval:
 -0.2298852  0.1523448
sample estimates:
  mean of x
-0.03877023

> #two-sample T-testing:
> x<-rnorm(100)
> y<-rnorm(100)
> t.test(x,y)

	Welch Two Sample t-test

data:  x and y
t = -0.062003, df = 197.96, p-value = 0.9506
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.2842159  0.2668885
sample estimates:
 mean of x  mean of y
0.04941380 0.05807748

> #Directional Hypothesis:-
> t.test(x,mu=2,alternative = 'greater')

	One Sample t-test

data:  x
t = -19.884, df = 99, p-value = 1
```

```
alternative hypothesis: true mean is greater than 2

95 percent confidence interval:

-0.1134708      Inf

sample estimates:

mean of x

0.0494138


> #one sample u-test:-

> wilcox.test(y,exact = FALSE)


        Wilcoxon signed rank test with continuity correction


data:  y

V = 2589, p-value = 0.8272

alternative hypothesis: true location is not equal to 0


> #Two sample u-test:-

> wilcox.test(x,y)

        Wilcoxon rank sum test with continuity correction

data:  x and y

W = 5039, p-value = 0.9251

alternative hypothesis: true location shift is not equal to 0

> #2)Correlation Test:-

> cor.test(mtcars$mpg,mtcars$hp)

        Pearson's product-moment correlation

data:  mtcars$mpg and mtcars$hp

t = -6.7424, df = 30, p-value = 1.788e-07

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

 -0.8852686 -0.5860994

sample estimates:

      cor
```

```
-0.7761684

> #3)Chi-Square Test:-

> library(MASS)

> #create DataFrame:

> print(str(survey))

'data.frame':    237 obs. of  12 variables:

 $ Sex    : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 1 2 1 2 2 ...

 $ Wr.Hnd: num  18.5 19.5 18 18.8 20 18 17.7 17 20 18.5 ...

 $ NW.Hnd: num  18 20.5 13.3 18.9 20 17.7 17.7 17.3 19.5 18.5 ...

 $ W.Hnd : Factor w/ 2 levels "Left","Right": 2 1 2 2 2 2 2 2 2 2 ...

 $ Fold   : Factor w/ 3 levels "L on R","Neither",..: 3 3 1 3 2 1 1 3 3 3 ...

 $ Pulse : int  92 104 87 NA 35 64 83 74 72 90 ...

 $ Clap   : Factor w/ 3 levels "Left","Neither",..: 1 1 2 2 3 3 3 3 3 3 ...

 $ Exer   : Factor w/ 3 levels "Freq","None",..: 3 2 2 2 3 3 1 1 3 3 ...

 $ Smoke : Factor w/ 4 levels "Heavy","Never",..: 2 4 3 2 2 2 2 2 2 2 ...

 $ Height: num  173 178 NA 160 165 ...

 $ M.I    : Factor w/ 2 levels "Imperial","Metric": 2 1 NA 2 2 1 1 2 2 2 ...

 $ Age    : num  18.2 17.6 16.9 20.3 23.7 ...

NULL

> # Create a data frame from the main data set.

> stu_data = data.frame(survey$Smoke,survey$Exer)

> # Create a contingency table with the needed variables.

> stu_data = table(survey$Smoke,survey$Exer)

> print(stu_data)

   Freq None Some

  Heavy    7    1    3

  Never   87   18   84

  Occas   12    3    4

  Regul    9    1    7
```

Experiment Name: Write program to implement the following analysis techniques using R.

4. Analysis of variance (ANOVA)

Name:

Roll No:-

Analysis of variance test

ANOVA also known as Analysis of variance is used to investigate relations between categorical variables and continuous variable in **R Programming.** It is a type of hypothesis testing for population variance.

R – ANOVA Test

ANOVA test involves setting up:

- **Null Hypothesis:** All population means are equal.
- **Alternate Hypothesis:** At least one population mean is different from other.

ANOVA tests are of two types:

- **One-way ANOVA:** It takes one categorical group into consideration.
- **Two-way ANOVA:** It takes two categorical group into consideration.

The Dataset we used for Analysis of Variance test

The mtcars (motor trend car road test) dataset is used which consist of 32 car brands and 11 attributes. The dataset comes preinstalled in **dplyr** package in R.

To get started with ANOVA, we need to install and load the **dplyr** package.

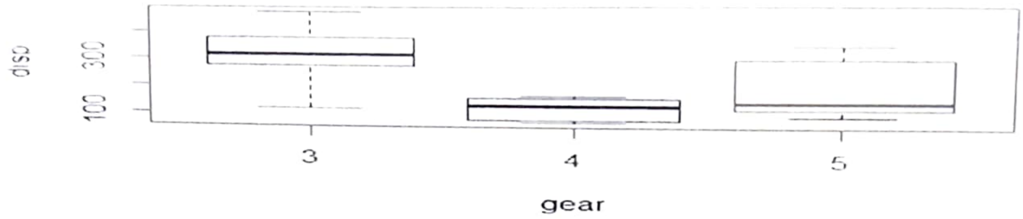
Performing One Way ANOVA test in R language

One-way ANOVA test is performed using mtcars dataset which comes preinstalled with dplyr package between disp attribute, a continuous attribute and gear attribute, a categorical attribute.

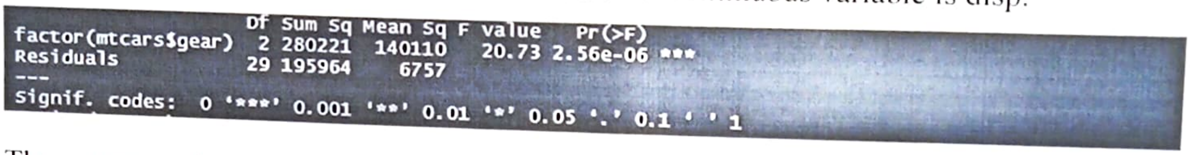
Program: -

```
# Installing the package
install.packages("dplyr")

# Loading the package
library(dplyr)

# Variance in mean within group and between group
boxplot(mtcars$disp~factor(mtcars$gear) xlab = "gear", ylab = "disp")

# Step 1: Setup Null Hypothesis and Alternate Hypothesis
# H0 = mu = mu01 = mu02(There is no difference
# between average displacement for different gear)
# H1 = Not all means are equal
# Step 2: Calculate test statistics using aov function
mtcars_aov <- aov(mtcars$disp~factor(mtcars$gear))
summary(mtcars_aov)

# Step 3: Calculate F-Critical Value
# For 0.05 Significant value, critical value = alpha = 0.05
# Step 4: Compare test statistics with F-Critical value
# and conclude test p < alpha, Reject Null Hypothesis
```

Output:



The box plot shows the mean values of gear with respect of displacement. Hear categorical variable is gear on which factor function is used and continuous variable is disp.



```
                   Df Sum Sq Mean Sq F value   Pr(>F)    
factor(mtcars$gear)  2 280221  140110   20.73 2.56e-06 ***
Residuals           29 195964    6757                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The summary shows that the gear attribute is very significant to displacement (Three stars denoting it). Also, the P value is less than 0.05, so proves that gear is significant to displacement i.e related to each other and we reject the Null Hypothesis.

Performing Two Way ANOVA test in R

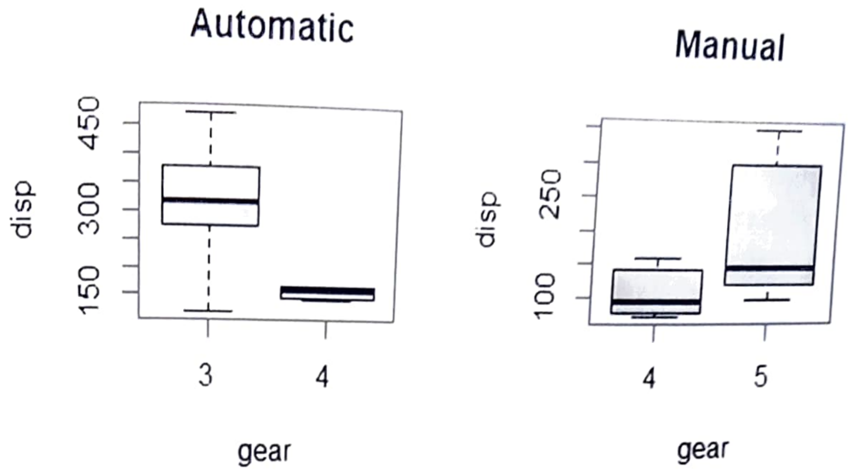
Two-way ANOVA test is performed using mtcars dataset which comes preinstalled with dplyr package between disp attribute, a continuous attribute and gear attribute, a categorical attribute, am attribute, a categorical attribute.

Program: -

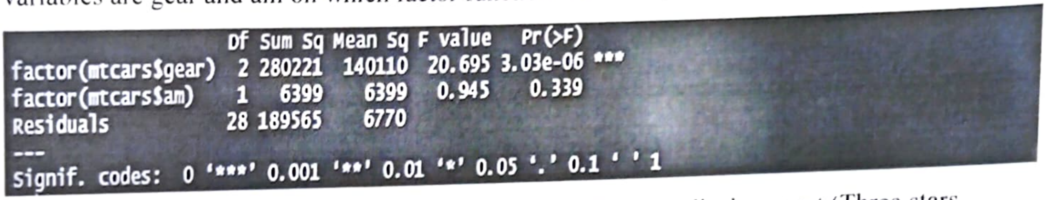
```
# Installing the package
install.packages("dplyr")

# Loading the package
library(dplyr)

# Variance in mean within group and between group
boxplot(mtcars$disp~mtcars$gear, subset = (mtcars$am == 0),
        xlab = "gear", ylab = "disp", main = "Automatic")
boxplot(mtcars$disp~mtcars$gear, subset = (mtcars$am == 1),
          xlab = "gear", ylab = "disp", main = "Manual")

# Step 1: Setup Null Hypothesis and Alternate Hypothesis
# H0 = mu0 = mu01 = mu02(There is no difference between
# average displacement for different gear)
# H1 = Not all means are equal

# Step 2: Calculate test statistics using aov function

mtcars_aov2 <- aov(mtcars$disp~factor(mtcars$gear) *factor(mtcars$am))
summary(mtcars_aov2)

# Step 3: Calculate F-Critical Value
# For 0.05 Significant value, critical value = alpha = 0.05
# Step 4: Compare test statistics with F-Critical value
```

and conclude test p < alpha, Reject Null Hypothesis

Output:



The box plot shows the mean values of gear with respect to displacement. Hear categorical variables are gear and am on which factor function is used and continuous variable is disp.



```
                   Df Sum Sq Mean Sq F value   Pr(>F)    
factor(mtcars$gear)  2 280221  140110  20.695 3.03e-06 ***
factor(mtcars$am)    1   6399    6399   0.945    0.339    
Residuals           28 189565    6770                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The summary shows that the gear attribute is very significant to displacement (Three stars denoting it) and am attribute is not much significant to displacement. P-value of gear is less than 0.05. so it proves that gear is significant to displacement i.e related to each other. P-value of am is greater than 0.05, am is not significant to displacement i.e not related to each other.

Results

We see significant results from boxplots and summaries.

- Displacement is strongly related to Gears in cars i.e displacement is dependent on gears with p < 0.05.
- Displacement is strongly related to Gears but not related to transmission mode in cars with p 0.05 with am.

Experiment No: 8

Experiment Name: Write program to implement the following analysis techniques using R. Regression analysis

Name:

Roll No:-

Regression analysis test

Regression analysis is a statistical tool to estimate the relationship between two or more variables. There is always one response variable and one or more predictor variables. Regression analysis is widely used to fit the data accordingly and further, predicting the data for forecasting. It helps businesses and organizations to learn about the behavior of their product in the market using the dependent/response variable and independent/predictor variable.

Types of Regression in R

There are mainly three types of Regression in R programming that is widely used. They are:

- Linear Regression
- Multiple Regression
- Logistic Regression

Linear Regression

The Linear Regression model is one of the widely used among three of the regression types. In linear regression. the relationship is estimated between two variables i.e.. one response variable and one predictor variable. Linear regression produces a straight line on the graph. Mathematically

where,

- *x indicates predictor or independent variable*
- *y indicates response or dependent variable*
- ***a** and **b** are coefficients*

Implementation in R

In R programming. **<u>lm()</u>** function is used to create linear regression model.

Syntax: *lm(formula)*

Parameter:

formula: *represents the formula on which data has to be fitted To know about more optional parameters, use below command in console: help("lm")*

Example: In this example. let us plot the linear regression line on the graph and predict the weight-based using height.

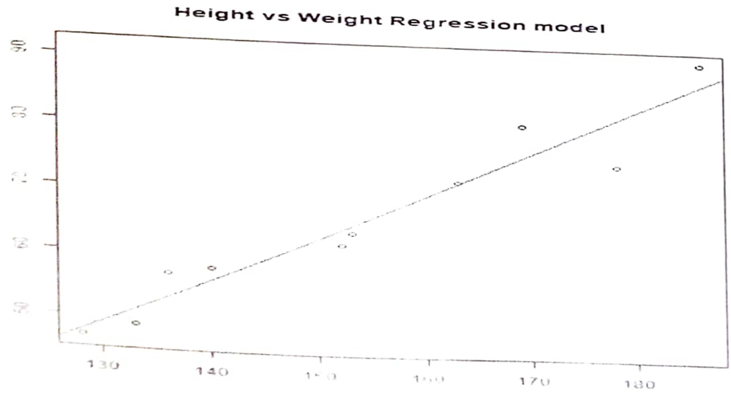
Program: -

```
# R program to illustrate
# Linear Regression
# Height vector
x <- c(153, 169, 140, 186, 128,
      136, 178, 163, 152, 133)

# Weight vector
```

```
y <- c(64. 81. 58. 91. 47. 57.
    75. 72. 62. 49)

# Create a linear regression model
model <- lm(y~x)

# Print regression model
print(model)

# Find the weight of a person With height 182
df <- data.frame(x = 182)
res <-  predict(model, df)
cat("\nPredicted value of a person
                with height = 182")
print(res)

# Output to be present as PNG file
png(file = "linearRegGFG.png")

# Plot
plot(x, y, main = "Height vs Weight Regression model")
abline(lm(y~x))

# Save the file.
dev.off()
```

Output:

```
Call:

lm(formula = y ~ x)

Coefficients:

(Intercept)          x

  -39.7137      0.6847

Predicted value of a person with height = 182

     1

84.9098
```



Multiple Regression

Multiple regression is another type of regression analysis technique that is an extension of the linear regression model as it uses more than one predictor variables to create the model.

Mathematically,

Implementation in R

Multiple regression in R programming uses the same **lm()** function to create the model.
Syntax: *lm(formula, data)*
Parameters:

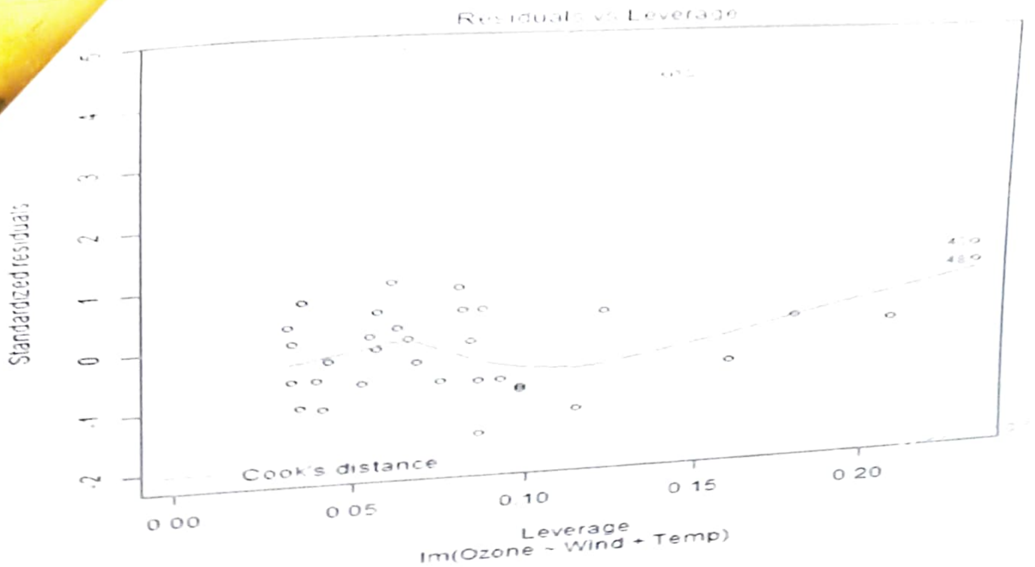
- ***formula:*** *represents the formula on which data has to be fitted*
- ***data:*** *represents dataframe on which formula has to be applied*

Example: Let us create a multiple regression model of air quality dataset present in R base package and plot the model on the graph.
Program: -

```
# R program to illustrate
# Multiple Linear Regression
# Using airquality dataset
input <- airquality[1:50,c("Ozone", "Wind", "Temp")]

# Create regression model
model <- lm(Ozone~Wind + Temp,data = input)

# Print the regression model
cat("Regression model:\n")
print(model)

# Output to be present as PNG file
png(file = "multipleRegGFG.png")

# Plot
plot(model)

# Save the file.
dev.off()
```

Output:

```
Regression model:

Call:

lm(formula = Ozone ~ Wind + Temp, data = input)

Coefficients:

(Intercept)      Wind      Temp

   -58.239     -0.739     1.329
```



Logistic Regression

Logistic Regression is another widely used regression analysis technique and predicts the value with a range. Moreover, it is used for predicting the values for categorical data. For example, Email is either spam or non-spam, winner or loser, male or female, etc. Mathematically,

where,

- *y represents response variable*
- *z represents equation of independent variables or features*

Implementation in R

In R programming, **glm()** function is used to create a logistic regression model.

Syntax: *glm(formula, data, family)*

Parameters:

- ***formula:*** *represents a formula on the basis of which model has to be fitted*
- ***data:*** *represents dataframe on which formula has to be applied*
- ***family:*** *represents the type of function to be used. "binomial" for logistic regression*

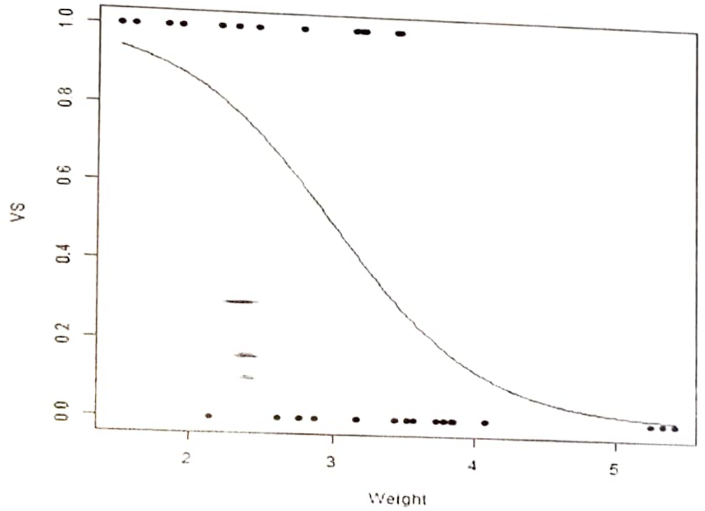
Example:

```
# R program to illustrate
# Logistic Regression
# Using mtcars dataset
# To create the logistic model
model <- glm(formula = vs ~ wt,family = binomial,data = mtcars)

# Creating a range of wt values
x <- seq(min(mtcars$wt),max(mtcars$wt),0.01)

# Predict using weight
y <- predict(model, list(wt = x), type = "response")

# Print model
print(model)
```

```
# Output to be present as PNG file
png(file = "LogRegGFG.png")

# Plot
plot(mtcars$wt, mtcars$vs, pch = 16, xlab = "Weight", ylab = "VS")
lines(x, y)

# Saving the file
dev.off()
```

Output:

```
Call:  glm(formula = vs ~ wt, family = binomial, data = mtcars)

Coefficients:

(Intercept)          wt

    5.715        -1.911


Degrees of Freedom: 31 Total (i.e. Null);  30 Residual

Null Deviance:      43.86

Residual Deviance: 31.37        AIC: 35.37
```


VS
Weight