NAME: Hammad Ansari **ROLL NO: 2018450002**

Experiment No. 5

Title: To Implement Logistic Regression in R

Tools Required: RStudio

Concept :

Logistic regression is used for binary classification.

It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables. To represent the binary/categorical outcome, we use dummy variables

Example Problem

For this analysis, we will use the mtcars dataset that comes with R by default. mtcars is a standard built-in dataset. Here we need to predict the type of engine if weight and displacement is given

1. **Import the data**

   ```
   #Loading the data

   library(caTools)

   write.csv(mtcars, file = "mtcars.csv")

   myData <- read.csv("mtcars.csv", header = T)
   ```

2. **Build a Model on entire dataset: use glm() function**

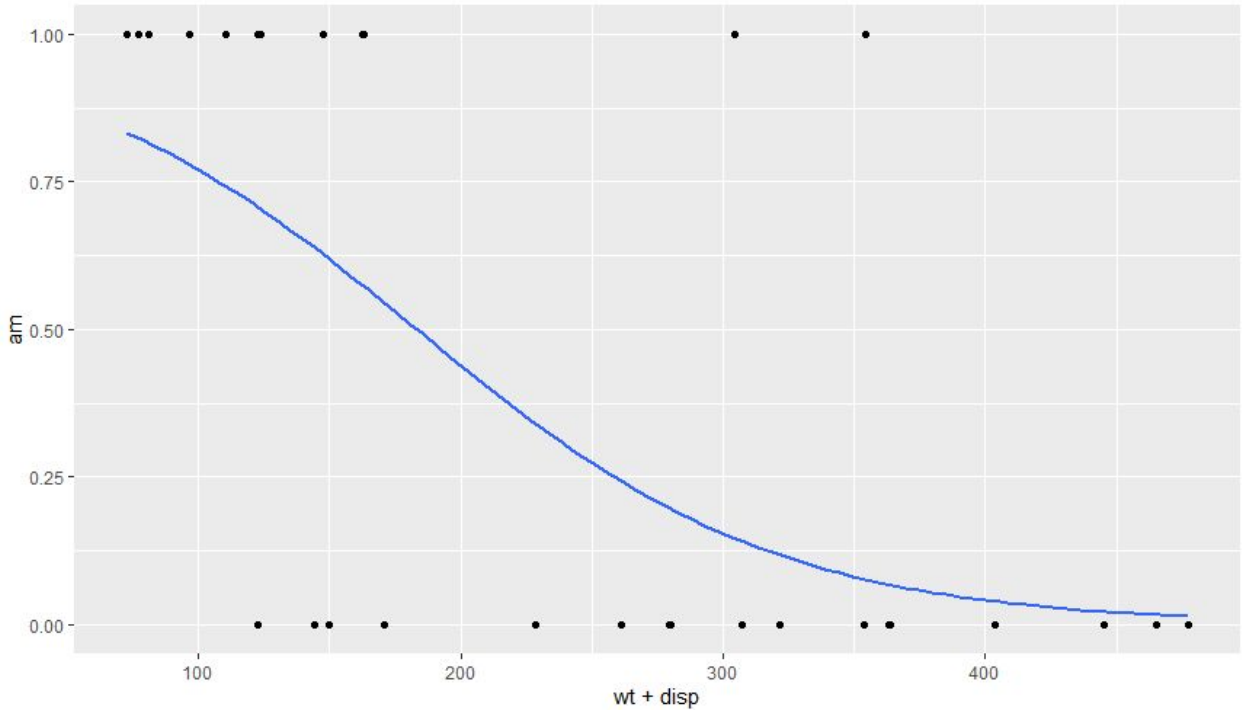
   ```
   #Train the model using entire data

   myModel <- glm(am ~ wt+disp, data = myData, family="binomial")

   summary(myModel)

   ggplot(myData, aes(x=wt+disp, y=am)) + geom_point() +
   ```

stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)



3. **Do Logistic Regression Diagnostics**

newData = data.frame(disp=120, wt=2.8)

predict(myModel, newData, type="response")

```
> newData = data.frame(disp=120, wt=2.8)

> predict(myModel, newData, type="response")
        1
0.5649092
```

4. **Predicting Logistic Models:**
   a. **Create the training (development) and test (validation) data samples from original data.**

```
#Splitting the data into training and testing

split <- sample.split(myData, SplitRatio = 0.8)

split

train <- subset(myData, split=="TRUE")

test <- subset(myData, split=="FALSE")
```

b. Develop the model on the training data and use it to predict the type of engine on test data.

```
#Train the model using training data

myModel <- glm(am ~ wt+disp, data = train, family="binomial")

summary(myModel)
```

```
> summary(myModel)

Call:
glm(formula = am ~ wt + disp, family = "binomial", data = train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.69731  -0.49489  -0.24292   0.08672   2.19755

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 12.99832    6.32822   2.054   0.0400 *
wt          -5.49294    2.79325  -1.967   0.0492 *
disp         0.01425    0.01267   1.125   0.2605
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 28.267  on 22  degrees of freedom
Residual deviance: 14.340  on 20  degrees of freedom
AIC: 20.34

Number of Fisher Scoring iterations: 6
```

c. Review diagnostic measures.

#Running the test data through the model

```
res <- predict(myModel, test, type="response")

res

res <- predict(myModel, train, type="response")

res
```

```
> #Running the test data through the model
>
> res <- predict(myModel, test, type="response")

> res
          6           8          12          18          20          24          30          32
0.069231243 0.100271101 0.005593008 0.899121924 0.982607350 0.048716248 0.505981035 0.414112210

> res <- predict(myModel, train, type="response")

> res
           1            2            3            4            5            7            9           10
7.354850e-01 4.163553e-01 8.738043e-01 2.984133e-01 3.305849e-01 1.979636e-01 1.131571e-01 3.727737e-02
          11           13           14           15           16           17           19           21
3.727737e-02 3.334991e-02 2.574272e-02 1.391764e-04 4.693610e-05 5.490431e-05 9.948761e-01 7.894439e-01
          22           23           25           26           27           28           29           31
1.560294e-01 1.946837e-01 8.811479e-02 9.735376e-01 9.551287e-01 9.976931e-01 6.492581e-01 1.015829e-01
>
```