Experiment 3.2

Student Name: Kanishk Soni **UID: 20BCS9398**
Branch: BE-CSE **Section/Group: 20BCS-DM_708B**
Semester: 6th **Subject Name: Data Mining Lab**
Subject Code: 20CSP-376

1. Aim:

To study of Regression Analysis using R Programming

2. Objective:

- To create a curve based on prediction using the regression model.

3. Code and Output:

PROGRAM

```
# Generate random IQ values with mean = 30 and sd =2

IQ <- rnorm(40, 30, 2)

# Sorting IQ level in ascending order

IQ <- sort(IQ)

IQ

# Generate vector with pass and fail values of 40 students

result <- c(0, 0, 0, 1, 0, 0, 0, 0, 0, 1,

            1, 0, 0, 0, 1, 1, 0, 0, 1, 0,

            0, 0, 1, 0, 0, 1, 1, 0, 1, 1,

            1, 1, 1, 0, 1, 1, 1, 1, 0, 1)

# Data Frame

df <- as.data.frame(cbind(IQ, result))

# Print data frame
```

```
print(df)

# Plotting IQ on x-axis and result on y-axis

plot(IQ, result, xlab = "IQ Level",ylab = "Probability of Passing")


#Linear regression

lrm <- lm(result ~ IQ)

summary(lrm)

#find the result of a person with IQ 35

a<-data.frame(IQ=35)

predRes<-predict(lrm,a)

print(predRes)


# Create a logistic model

lgm = glm(result~IQ, family=binomial, df)

# Summary of the regression model

summary(lgm)

# Create a curve based on prediction using the regression model

curve(predict(lgm, data.frame(IQ=x), type="resp"), add=TRUE)
```

CONSOLE

```
> # Generate random IQ values with mean = 30 and sd =2
> IQ <- rnorm(40, 30, 2)
> # Sorting IQ level in ascending order
> IQ <- sort(IQ)
```

```
> IQ
 [1] 25.71663 26.91365 27.01924 27.57227 27.89350 27.91015 28.34477
 [8] 28.36160 28.48058 28.55516 28.55747 28.59613 28.66078 29.01703
[15] 29.36088 29.37515 29.44792 29.67874 29.68006 29.77399 29.83156
[22] 29.90915 29.96073 30.49449 30.52550 30.69951 30.85703 30.98425
[29] 31.02837 31.10218 31.34112 31.37475 31.50119 31.56083 31.73027
[36] 32.45873 33.23908 34.05884 34.53273 36.13736
> # Generate vector with pass and fail values of 40 students
> result <- c(0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
+             1, 0, 0, 0, 1, 1, 0, 0, 1, 0,
+             0, 0, 1, 0, 0, 1, 1, 0, 1, 1,
+             1, 1, 1, 0, 1, 1, 1, 1, 0, 1)
> # Data Frame
> df <- as.data.frame(cbind(IQ, result))
> # Print data frame
> print(df)
          IQ result
1  25.71663      0
2  26.91365      0
3  27.01924      0
4  27.57227      1
5  27.89350      0
6  27.91015      0
7  28.34477      0
8  28.36160      0
9  28.48058      0
10 28.55516      1
```

11	28.55747	1
12	28.59613	0
13	28.66078	0
14	29.01703	0
15	29.36088	1
16	29.37515	1
17	29.44792	0
18	29.67874	0
19	29.68006	1
20	29.77399	0
21	29.83156	0
22	29.90915	0
23	29.96073	1
24	30.49449	0
25	30.52550	0
26	30.69951	1
27	30.85703	1
28	30.98425	0
29	31.02837	1
30	31.10218	1
31	31.34112	1
32	31.37475	1
33	31.50119	1
34	31.56083	0
35	31.73027	1
36	32.45873	1
37	33.23908	1

```
38 34.05884       1
39 34.53273       0
40 36.13736       1
> # Plotting IQ on x-axis and result on y-axis
> plot(IQ, result, xlab = "IQ Level",ylab = "Probability of Passing")
> #Linear regression
> lrm <- lm(result ~ IQ)
> summary(lrm)

Call:
lm(formula = result ~ IQ)

Residuals:
    Min      1Q  Median      3Q     Max
-0.9184 -0.3828 -0.1205  0.4232  0.7710

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.50178    1.05471  -2.372  0.02286 *
IQ           0.09904    0.03501   2.829  0.00741 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4656 on 38 degrees of freedom
Multiple R-squared:  0.174,      Adjusted R-squared:  0.1523
F-statistic: 8.005 on 1 and 38 DF,  p-value: 0.007411
```

```
> #find the result of a person with IQ 35
> a<-data.frame(IQ=35)
> predRes<-predict(lrm,a)
> print(predRes)
        1
0.9646496
> # Create a logistic model
> lgm = glm(result~IQ, family=binomial, df)
> # Summary of the regression model
> summary(lgm)

Call:
glm(formula = result ~ IQ, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1296  -0.9546  -0.5183   1.0008   1.7682

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -15.1398     6.3381  -2.389   0.0169 *
IQ            0.5009     0.2112   2.372   0.0177 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

Null deviance: 55.352 on 39 degrees of freedom

Residual deviance: 47.584 on 38 degrees of freedom

AIC: 51.584

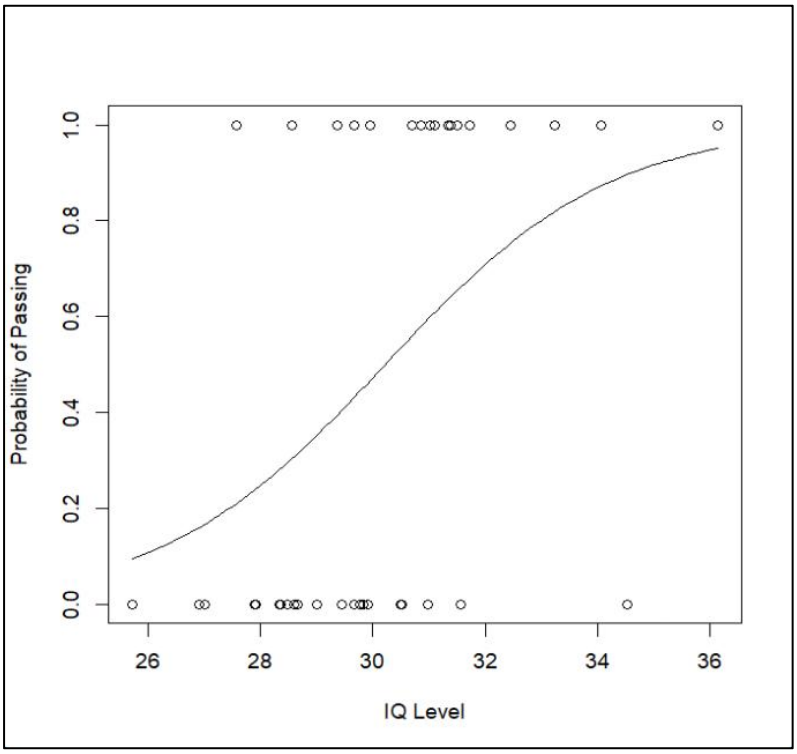
Number of Fisher Scoring iterations: 4

```
> # Create a curve based on prediction using the regression model
> curve(predict(lgm, data.frame(IQ=x), type="resp"), add=TRUE)


>
```

4. Output:

Curve based on prediction using the regression model:



Learning Outcomes:

- Linear Regression is a commonly used type of predictive analysis. It is a statistical approach for modelling the relationship between a dependent variable and a given set of independent variables.
- It is a statistical method that allows us to summarize and study relationships between two continuous (quantitative) variables. One variable denoted x is regarded as an independent variable and the other one denoted y is regarded as a dependent variable.
- It is assumed that the two variables are linearly related. Hence, we try to find a linear function that predicts the response values as accurately as possible as a function of the feature or independent variable(x).