DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Experiment - 9

Student Name: Rajdeep Jaiswal	**UID:** 20BCS2761
Branch: CSE	**Section/Group:** 20BCS-DM-902/(B)
Semester: 6th	**Subject Code:** 20CSP-376
Subject Name: Data mining lab	

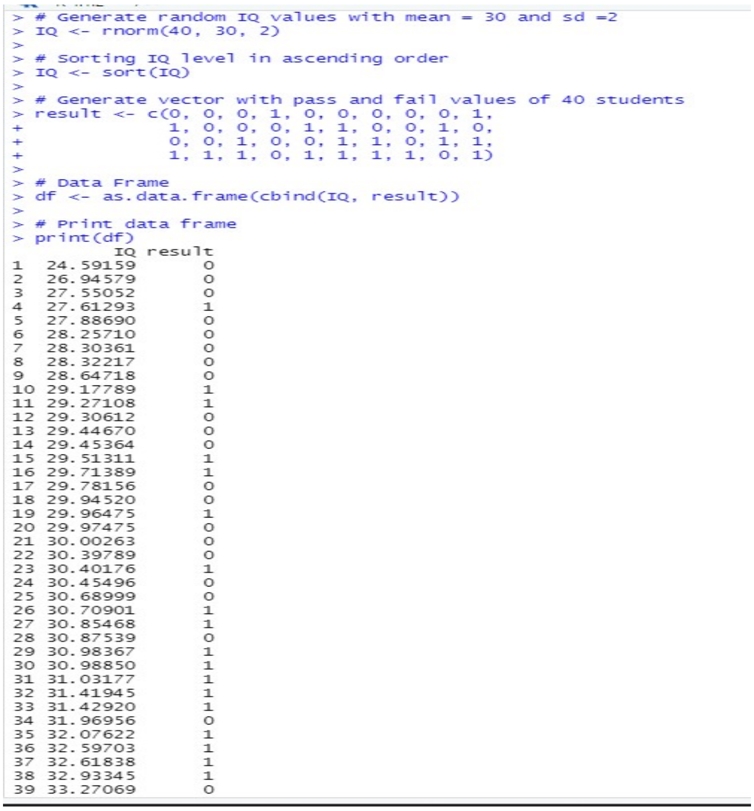
1.Aim: Study of Regression Analysis using R Programming.

2. CODE:

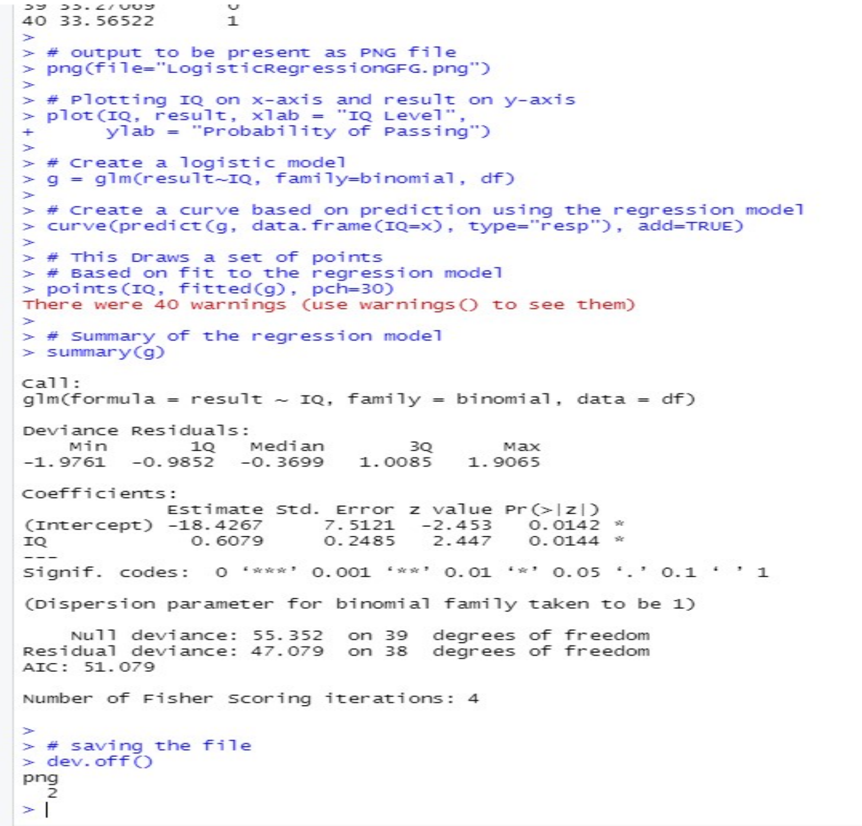
```
# Generate random IQ values with mean = 30 and sd =2
IQ <- rnorm(40, 30, 2)

# Sorting IQ level in ascending order
IQ <- sort(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)

# output to be present as PNG file
png(file="LogisticRegressionGFG.png")
```

```
# Plotting IQ on x-axis and result on y-axis
plot(IQ, result, xlab = "IQ Level",
    ylab = "Probability of Passing")

# Create a logistic model
g = glm(result~IQ, family=binomial, df)

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

# This Draws a set of points # Based
on fit to the regression model
points(IQ, fitted(g), pch=30)

# Summary of the regression model
summary(g)

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

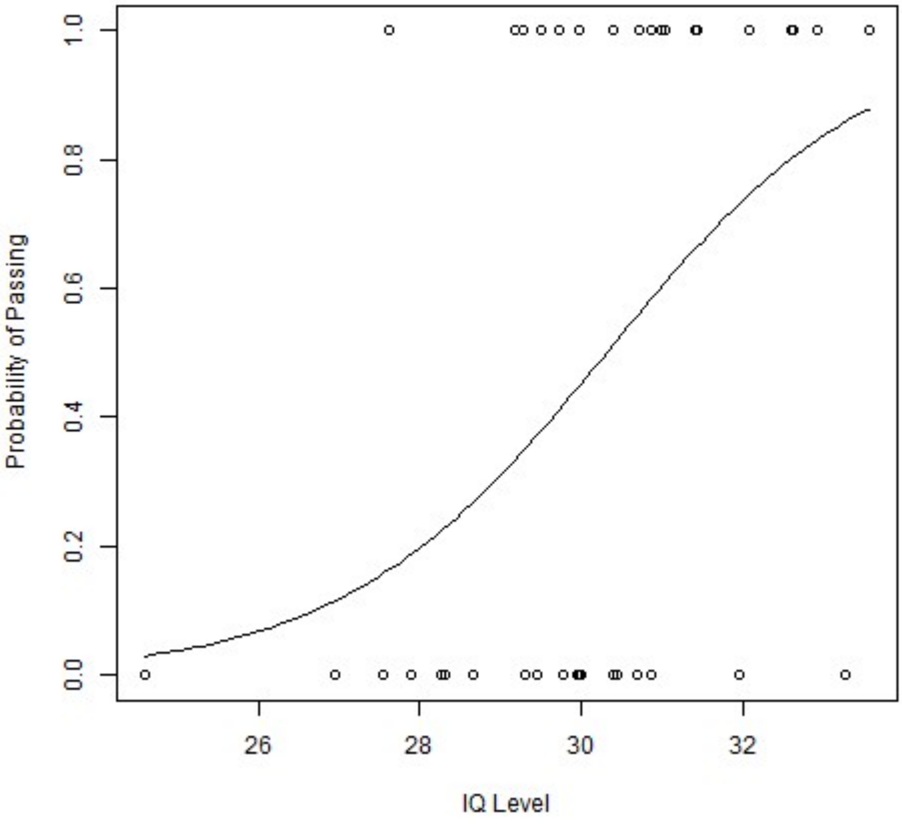
OUTPUT:



```
> # Generate random IQ values with mean = 30 and sd =2
> IQ <- rnorm(40, 30, 2)
>
> # Sorting IQ level in ascending order
> IQ <- sort(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)
         IQ result
1  24.59159      0
2  26.94579      0
3  27.55052      0
4  27.61293      1
5  27.88690      0
6  28.25710      0
7  28.30361      0
8  28.32217      0
9  28.64718      0
10 29.17789      1
11 29.27108      1
12 29.30612      0
13 29.44670      0
14 29.45364      0
15 29.51311      1
16 29.71389      1
17 29.78156      0
18 29.94520      0
19 29.96475      1
20 29.97475      0
21 30.00263      0
22 30.39789      0
23 30.40176      1
24 30.45496      0
25 30.68999      0
26 30.70901      1
27 30.85468      1
28 30.87539      0
29 30.98367      1
30 30.98850      1
31 31.03177      1
32 31.41945      1
33 31.42920      1
34 31.96956      0
35 32.07622      1
36 32.59703      1
37 32.61838      1
38 32.93345      1
39 33.27069      0
```



```
39 33.27069       0
40 33.56522       1
>
> # output to be present as PNG file
> png(file="LogisticRegressionGFG.png")
>
> # Plotting IQ on x-axis and result on y-axis
> plot(IQ, result, xlab = "IQ Level",
+      ylab = "Probability of Passing")
>
> # Create a logistic model
> g = glm(result~IQ, family=binomial, df)
>
> # Create a curve based on prediction using the regression model
> curve(predict(g, data.frame(IQ=x), type="resp"), add=TRUE)
>
> # This Draws a set of points
> # Based on fit to the regression model
> points(IQ, fitted(g), pch=30)
There were 40 warnings (use warnings() to see them)
>
> # Summary of the regression model
> summary(g)

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9761  -0.9852  -0.3699   1.0085   1.9065

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -18.4267     7.5121  -2.453   0.0142 *
IQ            0.6079     0.2485   2.447   0.0144 *
---
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: 47.079  on 38  degrees of freedom
AIC: 51.079

Number of Fisher Scoring iterations: 4

>
> # saving the file
> dev.off()
png
  2
> |
```


Probability of Passing
IQ Level