Experiment 3.2

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

2. Objective:

- To Explore the relationship between two or more variables quantitatively.
- To Develop predictive models to estimate or forecast future outcomes.

3. Script:

Linear Regression: It is a commonly used type of predictive analysis. It is a statistical approach for modeling the relationship between a dependent variable and a given set of independent variables.

There are two types of linear regression:

- Simple Linear Regression
- Multiple Linear Regression

- Use Case: We will be predicting students' success in an exam using their IQ levels.
- Let's generate some random IQ numbers to come up with our dataset.

4. 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)
```

Output:

Using the rnorm(), we have created a list of 40 IQ values that have a mean of 30 and a STD of 2.

```
> IQ
 [1] 25.67690 25.80425 25.94778 26.46862 26.75969 26.92143 27.17286 27.69117
 [9] 28.38959 28.39843 28.53443 28.78154 28.85173 28.91572 29.12570 29.31386
[17] 29.51318 29.62658 29.73714 30.09602 30.35026 30.36554 30.41757 30.44856
[25] 30.59577 30.90778 30.91678 30.91952 31.15490 31.19633 31.45548 31.92624
[33] 32.27773 32.57897 32.61706 32.69305 33.21784 33.35132 33.63331 33.79548
>
```

```
# 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)
```

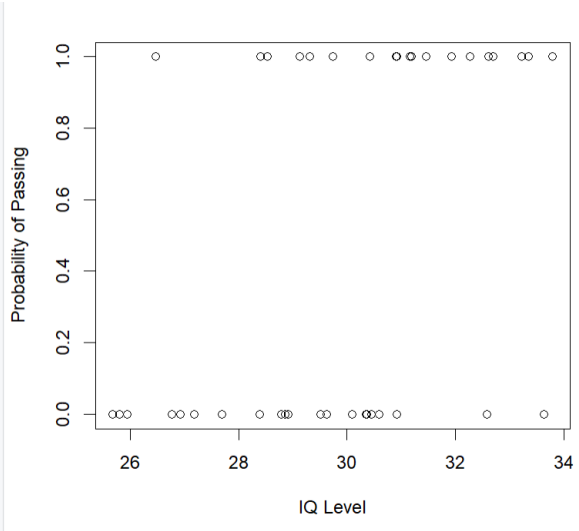
Output:

Now, we randomly created pass/fail values as 0/1 for 40 students and put them in a dataframe. Also, we will associate each value we create with an IQ so our dataframe is complete.

```
> # Print data frame
> print(df)
         IQ result
1  25.67690      0
2  25.80425      0
3  25.94778      0
4  26.46862      1
5  26.75969      0
6  26.92143      0
7  27.17286      0
8  27.69117      0
9  28.38959      0
10 28.39843      1
11 28.53443      1
12 28.78154      0
13 28.85173      0
14 28.91572      0
15 29.12570      1
16 29.31386      1
17 29.51318      0
18 29.62658      0
19 29.73714      1
20 30.09602      0
21 30.35026      0
22 30.36554      0
23 30.41757      1
24 30.44856      0
25 30.59577      0
26 30.90778      1
27 30.91678      1
28 30.91952      0
29 31.15490      1
30 31.19633      1
31 31.45548      1
32 31.92624      1
33 32.27773      1
34 32.57897      0
35 32.61706      1
36 32.69305      1
37 33.21784      1
38 33.35132      1
39 33.63331      0
40 33.79548      1
>
```

Now, let's create a regression model based on our dataset and create a curve to see how the regression model performs on it. We can use the glm() function to create and train a regression model and the curve() method to plot the curve based on prediction.

```
# 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)
```

Output:

```
> print(predRes)
        1
0.9658862
>
```

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

# Summary of the regression model
summary(lgm)
```

Output:

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8737  -0.9656  -0.4940   0.9945   1.9369

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -13.8081     5.4959  -2.512   0.0120 *
IQ            0.4571     0.1824   2.506   0.0122 *
---
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: 47.323  on 38  degrees of freedom
AIC: 51.323

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)
```

Output: