

Experiment - 9

Student Name: Rajdeep Jaiswal

Branch: CSE

UID: 20BCS2761

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)</pre>

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, 0, 1)

Data Frame
df <- as.data.frame(cbind(IQ, result))</pre>

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:



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHANDIGARH UNIVERSITY Discover. Learn. Empower.

<pre>> # 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, 1, + 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, + 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</pre>	
<pre>> 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, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, + 0, 0, 1, 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, 0, 1] > # Data Frame > df <- as.data.frame(cbind(IQ, result)) > # print(df) 1 24.59159 2 26.94579 0 3 27.55052 0 4 27.61293 1 5 27.88690 0 6 28.5710 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 13 29.44670 0 13 29.44670 0 13 29.44670 0 13 29.44670 0 13 29.94575 1 20.99475 1 10 29.97475 0 3 30.4076 1 24.304364 0 15 30.87399 0 23 30.4076 1 24.304546 0 23 30.4076 1 24.3045486 1 23 30.8458 1 23 30.8458 1 23 30.8458 1 23 30.9880 1 13 31.41345 1 33 31.42920 1 33 32.7069 0</pre>	> # Generate random IO values with mean = 30 and sd =2
<pre>/ # Generate vector with pass and fail values of 40 students / [q <- sort(IQ) // Generate vector with pass and fail values of 40 students // I, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</pre>	> TO <- ppopm(40, 30, 3)
<pre>> # 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, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</pre>	2 10 (- 110/11(40, 30, 2)
<pre>> # 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, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</pre>	>
<pre>> IQ <- sort(IQ) > # Generate vector with pass and fail values of 40 students > result <- c(0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</pre>	> # Sorting IQ level in ascending order
<pre> # Generate vector with pass and fail values of 40 students result <- c(0, 0, 0, 1, 0, 0, 0, 0, 0, 1,</pre>	> IO <- sort(IO)
<pre> # Generate vector with pass and fail values of 40 students result <- c(0, 0, 0, 1, 0, 0, 0, 0, 1,</pre>	
<pre>> # Generate vector with pass and tail values of 40 students > result <- c(0, 0, 0, 1, 0, 0, 0, 0, 1, 1, +</pre>	
<pre>> result <- c(0, 0, 0, 1, 0, 0, 0, 0, 0, 1, +</pre>	> # Generate Vector with pass and fail values of 40 students
<pre>+ 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 3 27.55052 0 4 27.61293 1 5 27.88690 0 6 28.52710 0 7 28.30361 0 8 28.32217 0 9 28.64718 0 11 29.27108 1 11 29.27108 1 12 29.30612 0 13 29.44670 0 14 29.45364 0 15 29.51311 1 17 29.78156 0 18 29.94520 0 19 29.96475 1 10 29.7455 0 21 30.00263 0 23 30.39789 0 23 30.40176 1 24 30.45496 1 25 30.68999 0 26 30.79901 1 27 30.85468 1 31 31.03177 1 33 31.42920 1 31 31.03177 1 33 31.42920 1 37 32.61838 1 38 32.7069 0</pre>	> result <- c(0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
<pre>+ 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, + 1, 1, 1, 0, 1, 1, 1, 0, 1) * # Data Frame > df <- as.data.frame(cbind(IQ, result)) * # Print data frame > print(df) 1 24.59159 0 2 26.94579 0 3 27.55052 0 4 27.61293 1 5 27.86690 0 6 28.25710 0 7 28.30361 0 7 28.30361 0 7 28.30612 0 13 29.44670 0 13 29.44670 0 13 29.44670 0 13 29.44670 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 23 30.39789 0 23 30.40176 1 24 30.45496 0 25 30.68999 0 6 30.70901 1 27 30.85468 1 33 31.42920 1 34 31.03177 1 33 31.42920 1 37 32.61838 1 38 32.7069 0</pre>	+ 1, 0, 0, 0, 1, 1, 0, 0, 1, 0,
<pre>+ 1, 1, 1, 0, 1, 1, 1, 0, 1) * # Data Frame > df <- as.data.frame(cbind(IQ, result)) * # Print data frame > print(df) 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 11 29.27108 1 11 29.27108 1 11 29.4564 0 15 29.51311 1 16 29.71389 1 17 29.78156 0 18 29.94520 0 19 29.96475 1 17 29.78156 0 18 29.94520 0 19 29.96475 1 17 29.78156 0 21 30.00263 0 23 30.40176 1 24 30.45496 1 25 30.68999 0 25 30.68999 0 25 30.68999 0 25 30.68999 0 26 30.70901 1 27 30.85468 1 28 30.87539 0 29 30.98567 1 30 30.98567 1 31 31.03177 1 32 31.41945 1 33 31.42920 1 31 31.03177 1 32 31.41945 1 33 31.42920 1 37 32.61838 1 39 33.27069 0 </pre>	+ 0.0.1.0.0.1.1.0.1.1.
<pre></pre>	
<pre>> # Data Frame > df <- as.data.frame(cbind(IQ, result)) > # Print data frame > print(df) I 24.9359 0 2 26.94579 0 2 27.55052 0 4 27.61293 1 5 27.88690 0 6 28.25710 0 7 28.302617 0 8 28.32217 0 9 28.64718 0 10 29.17788 1 11 20.3012 0 10 29.17788 1 11 20.3454 0 11 20.3454 0 11 20.3454 0 11 20.3454 0 11 20.3454 0 11 20.94520 0 12 29.78156 0 13 29.94520 0 13 29.94575 1 10 29.97475 0 21 30.00263 0 22 30.39789 0 23 30.40176 1 24 30.45496 0 25 30.68999 0 25 30.68999 0 26 30.79001 1 27 30.85468 1 28 30.87539 0 29 30.9850 1 31 31.03177 1 33 31.42920 1 31 31.03177 1 33 31.42920 1 33 32.07622 1 34 32.5769 0</pre>	- 1, 1, 1, 0, 1, 1, 1, 0, 1)
<pre>> # 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 17 29.78156 0 19 29.94475 1 17 29.78156 0 19 29.94575 1 20 29.97475 0 21 30.00263 0 23 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 30.9850 1 31 31.03177 1 32 31.41945 1 33 31.42920 1 34 31.96956 0 35 32.07622 1 36 32.9769 0</pre>	>
<pre>> df <- as.data.frame(cbind(IQ, result)) > # Print data frame > print(df)</pre>	> # Data Frame
<pre>> # 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 9 28.64718 0 10 29.17789 1 11 29.20612 0 13 29.44670 0 13 29.44670 0 13 29.44670 0 15 29.51311 1 17 29.78156 0 19 29.96475 1 17 29.78156 0 19 29.94575 0 21 30.00263 0 23 30.9789 0 23 30.40176 1 4 30.45496 0 25 30.68999 0 25 30.68999 0 25 30.68999 0 26 30.7901 1 27 30.85468 1 31 31.03177 1 32 31.41945 1 33 31.42920 1 4 31.96956 0 35 32.07622 1 36 32.9769 0</pre>	<pre>> df <- as.data.frame(cbind(IQ, result))</pre>
<pre>> # Print data frame > print(df) 1 24.59159 0 3 27.55052 0 4 27.61293 1 5 27.88690 0 6 28.25710 0 7 28.30361 0 7 28.30361 0 9 28.64718 0 10 29.17789 1 11 29.27108 1 12 29.30612 0 13 29.44670 0 13 29.44670 0 13 29.44670 0 15 29.51311 1 16 29.71389 1 17 29.78156 0 19 29.96475 1 20 29.97475 0 21 30.00263 0 23 30.40176 1 24 30.45496 0 25 30.68999 0 25 30.68999 0 25 30.68999 0 26 30.70901 1 27 30.85468 1 31 31.03177 1 32 31.41945 1 33 31.42920 1 31 31.03177 1 32 31.41945 1 33 31.42920 1 34 31.96956 0 35 32.07622 1 36 32.07622 1 37 32.61838 1 39 33.27069 0</pre>	
<pre>> print(df) 1 24.9139 result 1 24.9139 0 2 26.94579 0 3 27.515052 0 4 27.61293 1 5 27.85690 0 6 28.55710 0 6 7 28.20361 0 8 28.2217 0 9 28.64718 0 10 29.17789 1 11 29.27108 1 11 29.27108 1 11 29.27108 1 11 29.4564 0 15 29.51311 1 16 29.71389 1 17 29.78156 0 18 29.94520 0 19 29.96475 1 10 29.94520 0 19 29.96475 1 10 29.97475 0 21 30.00263 0 22 30.39789 0 23 30.40176 1 24 30.45496 0 25 30.68999 0 25 30.68999 0 25 30.68999 0 26 30.70901 1 27 30.85468 1 28 30.87539 0 29 30.98567 1 31 31.03177 1 32 31.41945 1 33 31.42920 1 34 31.96956 0 35 32.07622 1 36 32.9769 0 </pre>	# Brint data framo
<pre>> print(dr)</pre>	
To result 2 46.99159 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 12 29.30612 0 13 29.4670 0 14 29.45364 0 15 29.51311 1 16 29.71389 1 17 29.94520 0 19 29.96475 0 19 29.96475 0 21 30.00263 0 22 30.39789 0 23 30.40176 1 24 30.45496 0 25 30.68999 0 25 30.68999 0 25 30.68999 0 25 30.85468 1 27 30.85468 1 27 30.85468 1 31 31.03177 1 33 31.42920 1 31 31.03177 1 33 31.42920 1 31 31.03177 1 33 31.42920 1 34 31.96956 0 35 32.07622 1 36 32.07622 1 37 32.61838 1 38 32.59703 1 37 32.61838 1 39 33.27069 0	> princ(ar)
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	IQ result
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 29.29.30612 0 13 29.44670 0 14 29.45364 0 15 25.51311 1 16 29.71389 1 17 29.44670 0 18 29.44520 0 19 29.4520 0 19 29.96475 1 20 29.97475 0 21 30.00263 0 22 30.39789 0 23 30.39789 0 23 30.45496 0 25 30.68999 0 26 30.70901 1 27 30.85468 1	1 24.59159 0
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	2 26.94579 0
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	2 27 55052 0
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	3 27.33032 U
$\begin{array}{llllllllllllllllllllllllllllllllllll$	4 27.01293 1
	5 27.88690 0
$\begin{array}{rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr$	6 28,25710 0
8 228 32217 0 9 28 64718 0 10 29 17789 1 11 29 27108 1 12 29 30612 0 13 29 44670 0 14 29 27108 1 17 29 30617 0 13 29 44670 0 15 29 51311 1 16 29 78156 0 18 29 94520 0 19 29 96475 1 20 29 97475 0 21 30 00263 0 23 30 39789 0 23 30 9789 0 23 30 40176 1 24 30 45496 0 25 30 689999 0 26 </td <td>7 28 30361 0</td>	7 28 30361 0
$\begin{array}{llllllllllllllllllllllllllllllllllll$	8 28.32217 0
$\begin{array}{llllllllllllllllllllllllllllllllllll$	9 28.64718 0
$\begin{array}{llllllllllllllllllllllllllllllllllll$	10 29.17789 1
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	11 29.27108 1
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	12 20 20612 0
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	
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 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, 98850 1 31 31, 03177 1 32 31, 41945 1 33 31, 42920 1 34 31, 96956 0 35 32, 07622 1 37 32, 61838 1 38 32, 9345 1 39 33, 27069 0	13 29.44670 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 23 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, 98850 1 31 31, 03177 1 32 31, 41945 1 33 31, 42920 1 34 31, 96956 0 35 32, 07622 1 37 32, 61838 1 38 32, 9345 1 39 33, 27069 0	14 29.45364 0
$\begin{array}{llllllllllllllllllllllllllllllllllll$	15 29.51311 1
17 29. 78156 0 18 29. 94520 0 19 29. 96475 1 20 29. 97475 0 21 30. 00263 0 23 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. 98850 1 31 31. 03177 1 32 31. 41945 1 33 31. 42920 1 431. 96956 0 35 32. 07622 1 36 32. 59703 1 37 32. 61838 1 38 32. 9345 1 39 33. 27069 0	16 29,71389 1
17 25 94520 0 19 25 96475 1 19 25 96475 1 21 20 97475 0 21 30 00263 0 23 30,39789 0 23 24 30.45496 0 25 25 30.68999 0 25 26 30.70901 1 27 27 30.85468 1 28 28 30.87539 0 29 29 30.9850 1 31 31 31.03177 1 32 32 31.41945 1 33 33 31.42920 1 34 34 31.96956 0 35 35 32.07622 1 36 37 32.61838 1 38 39 33.27069 0 0	17 20 78156 0
18 29, 94520 0 19 29, 96475 1 20 29, 97475 0 130.00263 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 30.98850 1 31 31.03177 1 33 31.41945 1 33 31.42920 1 43 31.96956 0 35 32.07622 1 37 32.61838 1 38 32.9345 1 39 33.27069 0	1/ 29./6130 0
19 29, 96475 1 0 29, 97475 0 21 30, 00263 0 23 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, 98567 1 30 30, 98850 1 31 31, 03177 1 32 31, 41945 1 33 31, 42920 1 33 31, 69656 0 35 32, 07622 1 37 32, 61838 1 38 32, 9345 1 39 33, 27069 0	18 29.94520 0
20 29.97475 0 1 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 30.98850 1 31 31.03177 1 32 31.41945 1 33 31.42920 1 43 31.96956 0 35 32.07622 1 36 32.59703 1 37 32.61838 1 39 33.27069 0	19 29.96475 1
21 30.00263 0 23 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 30.98567 1 30 30.98850 1 31 31.03177 1 23 31.41945 1 33 31.42920 1 431.96956 0 35 32.07622 1 37 32.61838 1 38 32.99345 1 39 33.27069 0	20 29,97475 0
22 30. 30780 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. 9850 1 31 31. 03177 1 23 31. 41945 1 33 31. 42920 1 43 31. 96956 0 35 32. 07622 1 36 32. 59703 1 37 32. 61838 1 38 32. 9345 1 39 33. 27069 0	21 30,00263 0
22 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. 98867 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 37 32. 61838 1 38 32. 9345 1 39 33. 27069 0	22 30 30780 0
23 30.401/6 1 24 30.45496 0 25 30.68999 0 26 30.70901 1 27 30.85468 1 28 30.87539 0 29 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 32.9345 1 39 33.27069 0	22 30.39789 0
24 30.45496 0 25 30.688999 0 26 30.70901 1 27 30.85468 1 28 30.87539 0 29 30.98850 1 30 3850 1 31 31.03177 1 32 31.41945 1 33 31.42920 1 34 31.96956 0 35 32.07622 1 37 32.61838 1 38 32.9345 1 39 33.27069 0	23 30.401/6 1
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 31.77 1 23 31.41945 1 33 31.42920 1 431.96956 0 35 32.07622 1 37 32.61838 1 38 32.9345 1 39 33.27069 0	24 30.45496 0
26 30.70901 1 27 30.85468 1 28 30.87539 0 29 30.98850 1 31 31.03177 1 23 31.41945 1 33 31.42920 1 34 31.96956 0 35 32.07622 1 36 32.59703 1 37 32.61838 1 39 33.27069 0	25 30,68999 0
27 30. 85468 1 28 30. 87539 0 29 30. 98367 1 30 30. 98850 1 31 31. 77 1 23 31. 41945 1 33 31. 42920 1 431. 96956 0 35 32. 07622 1 37 37 32. 61838 1 33. 27069 0 0	26 30 70901 1
27 30.83408 1 28 30.87539 0 29 30.9850 1 31 31.03177 1 32 31.41945 1 33 31.42920 1 431.96956 0 35 32.07622 1 36 32.59703 1 37 32.61838 1 38 32.9345 1 39 33.27069 0	
28 30.87539 0 29 30.988367 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 39 33.27069 0	27 30.83408 1
29 30.98367 1 30 30.98850 1 31.31.03177 1 32 31.41945 33 31.42920 4 31.96956 0 35 32.07622 1 36 32.59703 37 32.61838 1 33.27069 0 33.27069	28 30.87539 0
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 32.9345 1 39 33.27069 0	29 30.98367 1
31 03177 1 23 31.41945 1 33 31.42920 1 431 96956 0 35 32.07622 1 36 32.59703 1 37 32.61838 1 38 32.9345 1 39 33.27069 0	30 30, 98850 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 32.9345 1 39 33.27069 0	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	
33 31.42920 1 34 31.96956 0 35 32.07622 1 36 32.59703 1 37 32.61838 1 38 32.9345 1 39 33.27069 0	52 51.41945 1
34 31.96956 0 35 32.07622 1 36 32.59703 1 37 32.61838 1 38 32.9345 1 39 33.27069 0	33 31.42920 1
35 32.07622 1 36 32.59703 1 37 32.61838 1 38 32.9345 1 39 33.27069 0	34 31.96956 0
36 32,59703 1 37 32,61838 1 38 32,9345 1 39 33,27069 0	35 32,07622 1
37 32.61838 1 38 32.93345 1 39 33.27069 0	36 32 59703 1
3/ 32.01030 1 38 32.9345 1 39 33.27069 0	27 22 (1020 L
38 32.93345 1 39 33.27069 0	3/ 32.01838 1
39 33.27069 0	38 32.93345 1
	39 33.27069 0



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GARH In Discover. Learn. Empower.

40 33.56522 1
<pre>> # output to be present as PNG file > png(file="LogisticRegressionGFG.png")</pre>
<pre>> # Plotting IQ on x-axis and result on y-axis > plot(IQ, result, xlab = "IQ Level", + ylab = "Probability of Passing") ></pre>
<pre>> # Create a logistic model > g = glm(result~IQ, family=binomial, df) ></pre>
<pre>> # Create a curve based on prediction using the regression model > curve(predict(g, data.frame(IQ=x), type="resp"), add=TRUE)</pre>
<pre>> # 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)</pre>
<pre>> # Summary of the regression model > summary(g)</pre>
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
<pre>> # saving the file > dev.off() png 2 2</pre>

