

Implement Linear and Logistic Regression**AIM:**

To implement Linear and Logistic Regression using R programming in R Studio.

a)Linear regression

```
# Sample data
heights <- c(150, 160, 165, 170, 175, 180, 185)
weights <- c(55, 60, 62, 68, 70, 75, 80)

# Create a data frame data
<- data.frame(heights,
weights)

# Fit a linear regression model
linear_model <- lm(weights ~ heights,
data = data)

# Print the summary of the model
print(summary(linear_model))

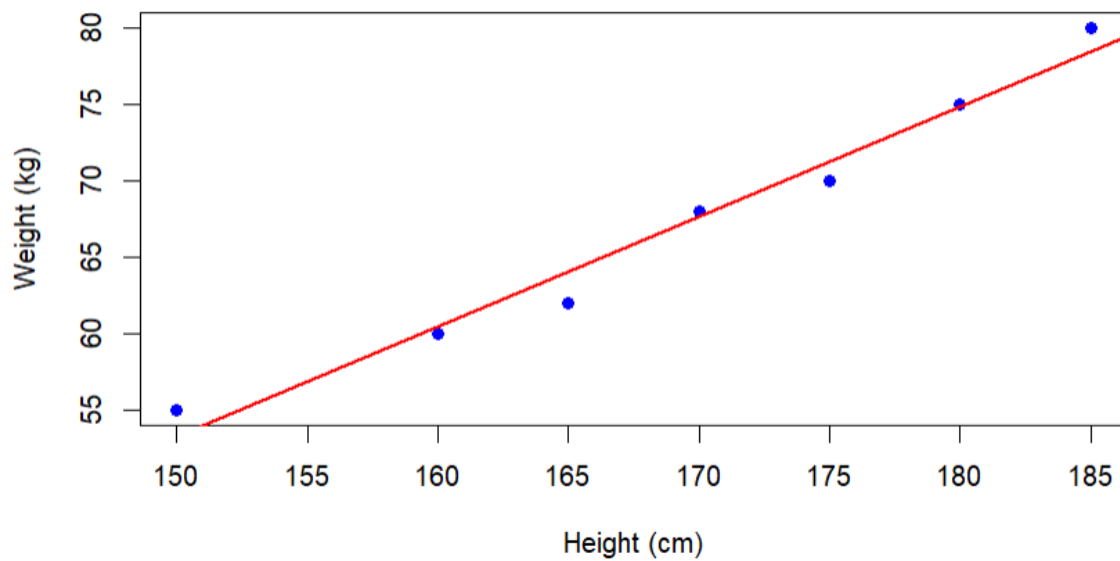
# Plotting the data and regression line
plot(data$heights, data$weights,
main = "Linear Regression: Weight vs.
Height", xlab = "Height (cm)", ylab
= "Weight (kg)",
pch = 19, col = "blue")

# Add regression line
abline(linear_model, col = "red", lwd = 2)
```

OUTPUT:

```
linear regression - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Untitled1* x
1 # Sample data
2 heights <- c(150, 160, 165, 170, 175, 180, 185)
3 weights <- c(55, 60, 62, 68, 70, 75, 80)
4 # Create a data frame
5 data <- data.frame(heights, weights)
6 # Fit a linear regression model
7 linear_model <- lm(weights ~ heights, data = data)
8 # Print the summary of the model
9 print(summary(linear_model))
10 # Plotting the data and regression line
11 plot(data$heights, data$weights,
12      main = "Linear Regression: Weight vs. Height",
13      xlab = "Height (cm)",
14      ylab = "Weight (kg)",
15      pch = 19, col = "blue")
16 # Add regression line
17 abline(linear_model, col = "red", lwd = 2)
18
19
```

Linear Regression: Weight vs. Height



b) Logistic regression

```
# Load the dataset
data(mtcars)

# Convert 'am' to a factor (categorical variable) mtcars$am <-
factor(mtcars$am, levels = c(0, 1), labels = c("Automatic", "Manual"))

# Fit a logistic regression model logistic_model <- glm(am
~ mpg, data = mtcars, family = binomial)

# Print the summary of the model
print(summary(logistic_model))

# Predict probabilities for the logistic model
predicted_probs <- predict(logistic_model, type =
"response")

# Display the predicted probabilities
print(predicted_probs)

# Plotting the data and logistic regression curve
plot(mtcars$mpg, as.numeric(mtcars$am) - 1,
main = "Logistic Regression: Transmission vs.
MPG", xlab = "Miles Per Gallon (mpg)",
ylab = "Probability of Manual Transmission",
pch = 19, col = "blue")

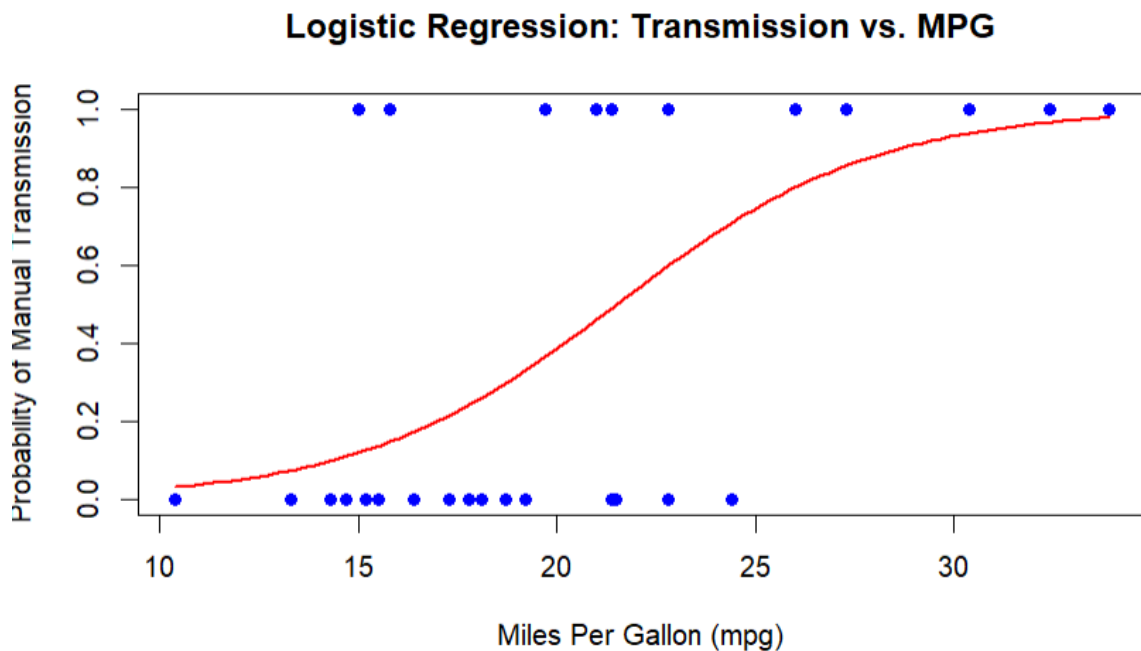
# Add the logistic regression curve
curve(predict(logistic_model, data.frame(mpg = x), type =
"response"), add = TRUE, col = "red", lwd = 2)
```

OUTPUT:

```

1 # Load the dataset
2 data(mtcars)
3 # Convert 'am' to a factor (categorical variable)
4 mtcars$am <- factor(mtcars$am, levels = c(0, 1), labels = c("Automatic", "Manual"))
5 # Fit a logistic regression model
6 logistic_model <- glm(am ~ mpg, data = mtcars, family = binomial)
7 # Print the summary of the model
8 print(summary(logistic_model))
9 # Predict probabilities for the logistic model
10 predicted_probs <- predict(logistic_model, type = "response")
11 # Display the predicted probabilities
12 print(predicted_probs)
13 # Plotting the data and logistic regression curve
14 plot(mtcars$mpg, as.numeric(mtcars$am) - 1,
15      main = "Logistic Regression: Transmission vs. MPG",
16      xlab = "Miles Per Gallon (mpg)",
17      ylab = "Probability of Manual Transmission",
18      pch = 19, col = "blue")
19 # Add the logistic regression curve
20 curve(predict(logistic_model, data.frame(mpg = x), type = "response"),
21       add = TRUE, col = "red", lwd = 2)

```

**RESULT:**

Thus the implementation Linear and Logistic Regression using R programming in R Studio have been successfully executed.