Exp.No: 7

<div align="center">

IMPLEMENT LINEAR AND LOGISTIC REGRESSION

</div>

AIM:

To write an R code to implement linear and logistic regression.

PROCEDURE:

1. Create sample data for heights and weights, fit a linear regression model, and plot the data with the regression line.

2. Use the sample data to create a data frame for the regression model.

3. Fit the linear regression model using the `lm()` function and display the summary.

4. Plot the data points and add the regression line using the `plot()` and `abline()` functions.

5. Load the `mtcars` dataset, convert the 'am' variable to a factor, fit a logistic regression model using the `glm()` function, and plot the probabilities.

PROGRAM CODE:

a)Linear regression

```
# Linear Regression heights <- c(150, 160, 165,
170, 175, 180, 185) weights <- c(55, 60, 62, 68, 70,
75, 80) data <- data.frame(heights, weights)
linear_model <- lm(weights ~ heights, data = data)
print(summary(linear_model))


# Plotting Linear Regression plot(data$heights,
data$weights, main = "Linear Regression:
Weight vs. Height", xlab = "Height (cm)", ylab
= "Weight (kg)", pch = 19, col = "blue")
abline(linear_model, col = "red", lwd = 2)
```

OUTPUT:
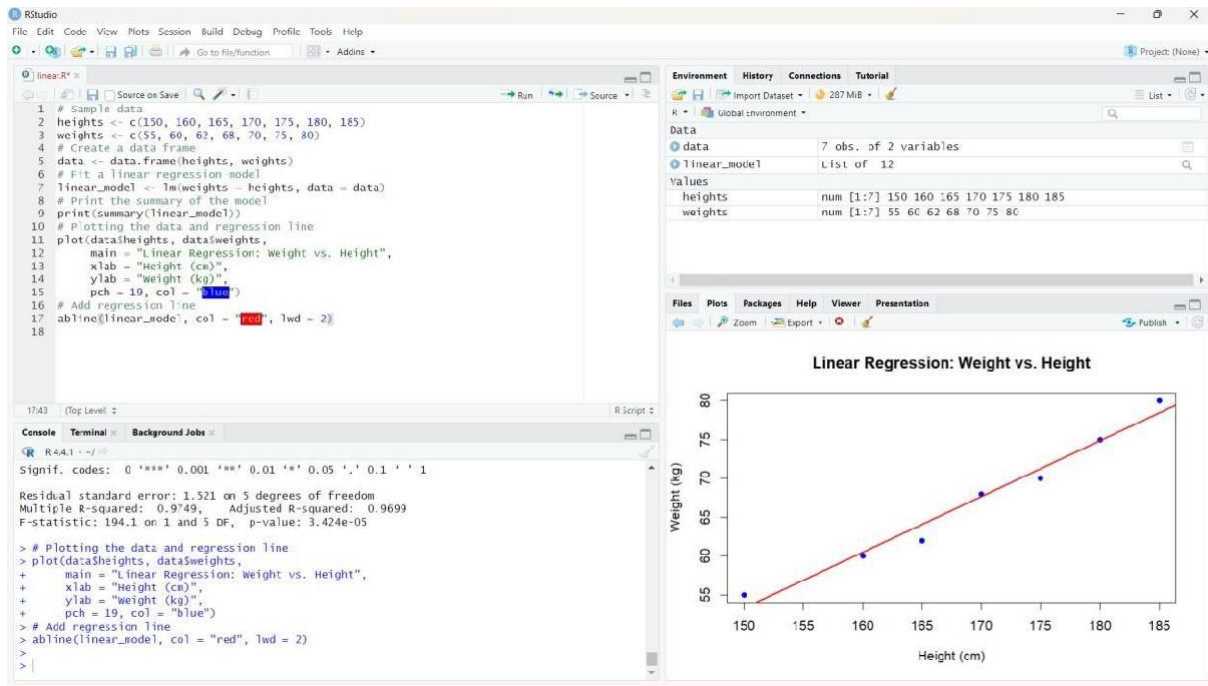
b) __Logistic regression__

Logistic Regression data(mtcars) mtcars$am <- factor(mtcars$am, levels = c(0, 1),

labels = c("Automatic", "Manual")) logistic_model <- glm(am ~ mpg, data = mtcars,

family = binomial) print(summary(logistic_model))

Plotting Logistic Regression predicted_probs

<predict(logistic_model, type = "response") print(predicted_probs)

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 =

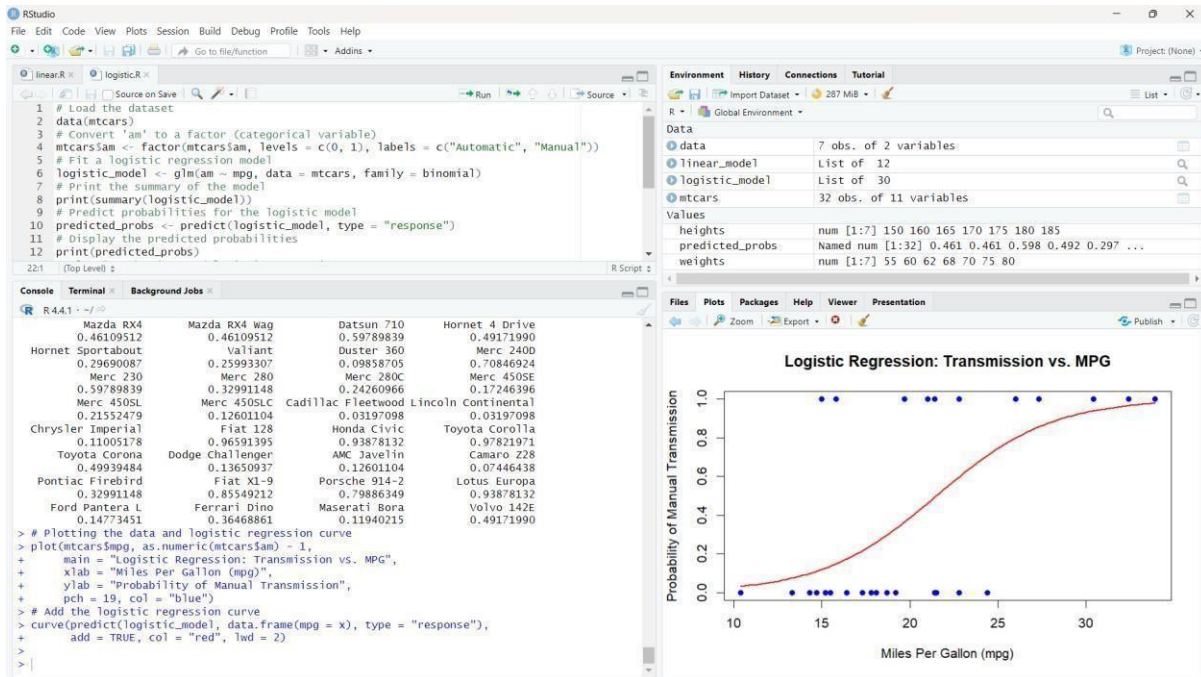"blue") curve(predict(logistic_model, data.frame(mpg = x), type =

"response"),

add = TRUE, col = "red", lwd = 2)

OUTPUT:

RESULT:

Thus the R program to implement Linear and Logistic Regression has been executed and verified successfully.