

Implement Linear and Logistic Regression**Aim:**

To implement Linear and Logistic Regression using R and RStudio.

Procedure:**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)
```

```
E:\Admin\Documents\Linear_regression_1.R - R Editor
# Sample data
Exercise_time <- c(50, 30, 45, 17, 25, 34, 22)
Calories_burnt <- c(500, 310, 450, 175, 260, 340, 210)

# Create a data frame
data <- data.frame(Exercise_time, Calories_burnt)

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

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

# Plotting the data and regression line
plot(data$Exercise_time, data$Calories_burnt,
main = "Linear Regression: Calories_burntvs. Exercise_time",
xlab = "Exercise_time(minutes)",
ylab = "Calories_burnt(kcal)",
pch = 19, col = "blue")

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

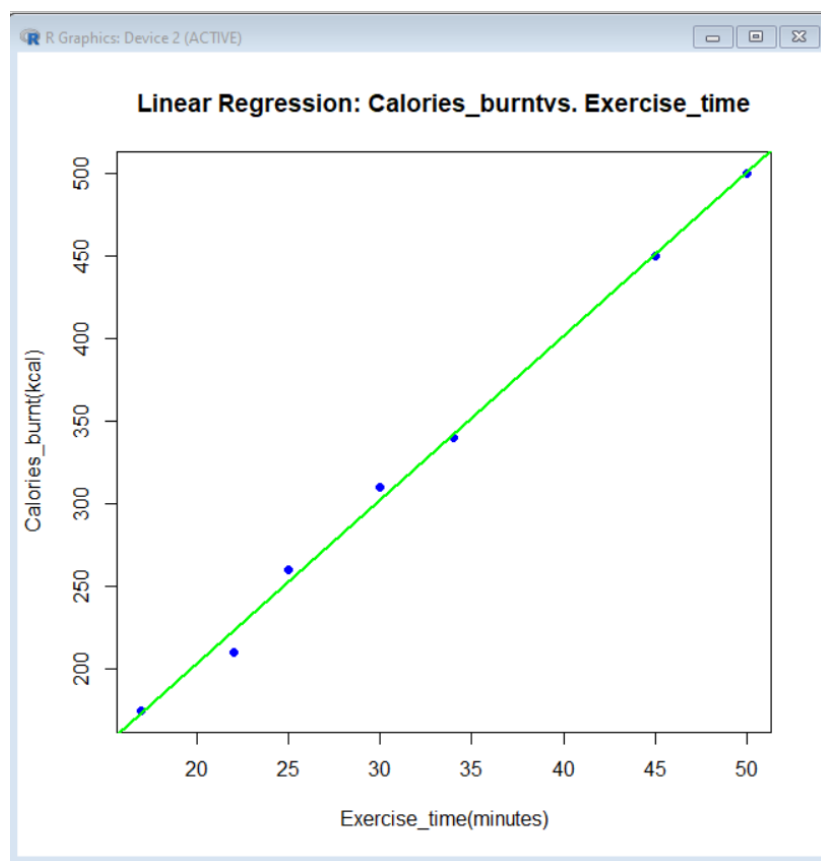
Output:

```
Call:
lm(formula = Calories_burnt ~ Exercise_time, data = data)

Residuals:
    1      2      3      4      5      6      7 
-0.8393  7.7237 -1.1986  1.7897  7.3645 -1.9889 -12.8511

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.4317     8.6681   0.511   0.631
Exercise_time   9.9282     0.2567  38.670 2.18e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.594 on 5 degrees of freedom
Multiple R-squared:  0.9967,    Adjusted R-squared:  0.996
F-statistic: 1495 on 1 and 5 DF, p-value: 2.179e-07
```



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)
```

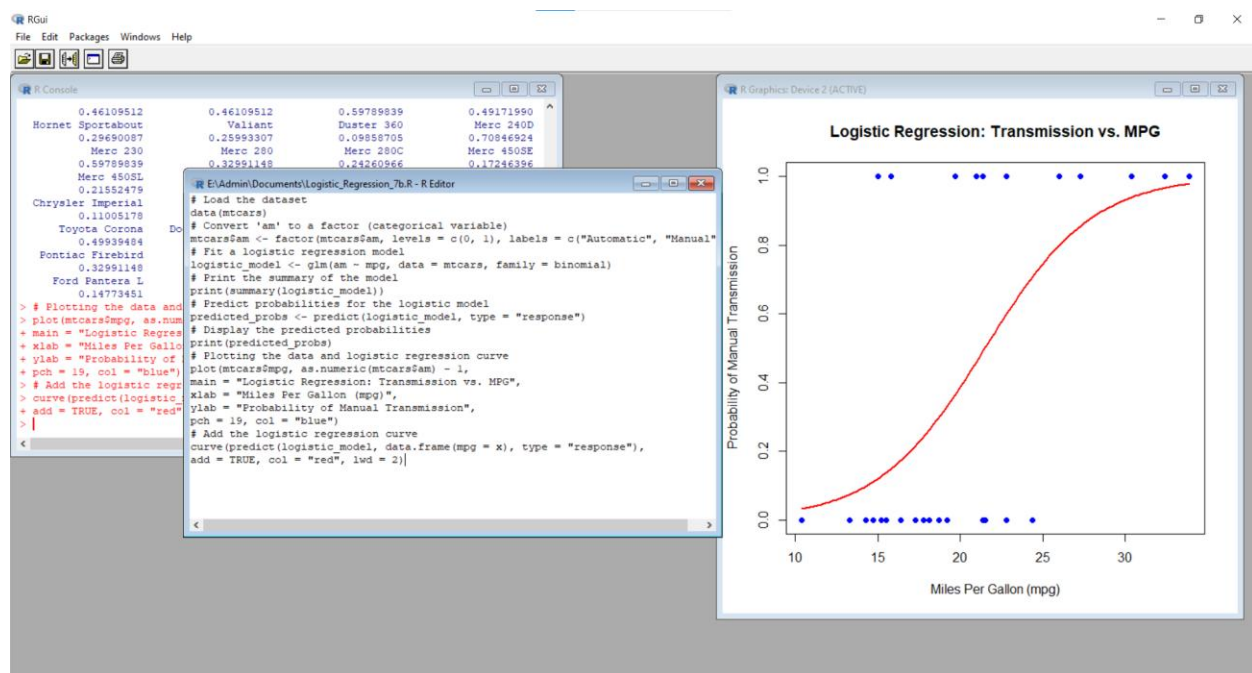
```
# Plotting the data and logistic regression curve

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:



Result:

Thus, Linear and Logistic Regression using R and RStudio was completed successfully.