Implement Linear and Logistic Regression in R
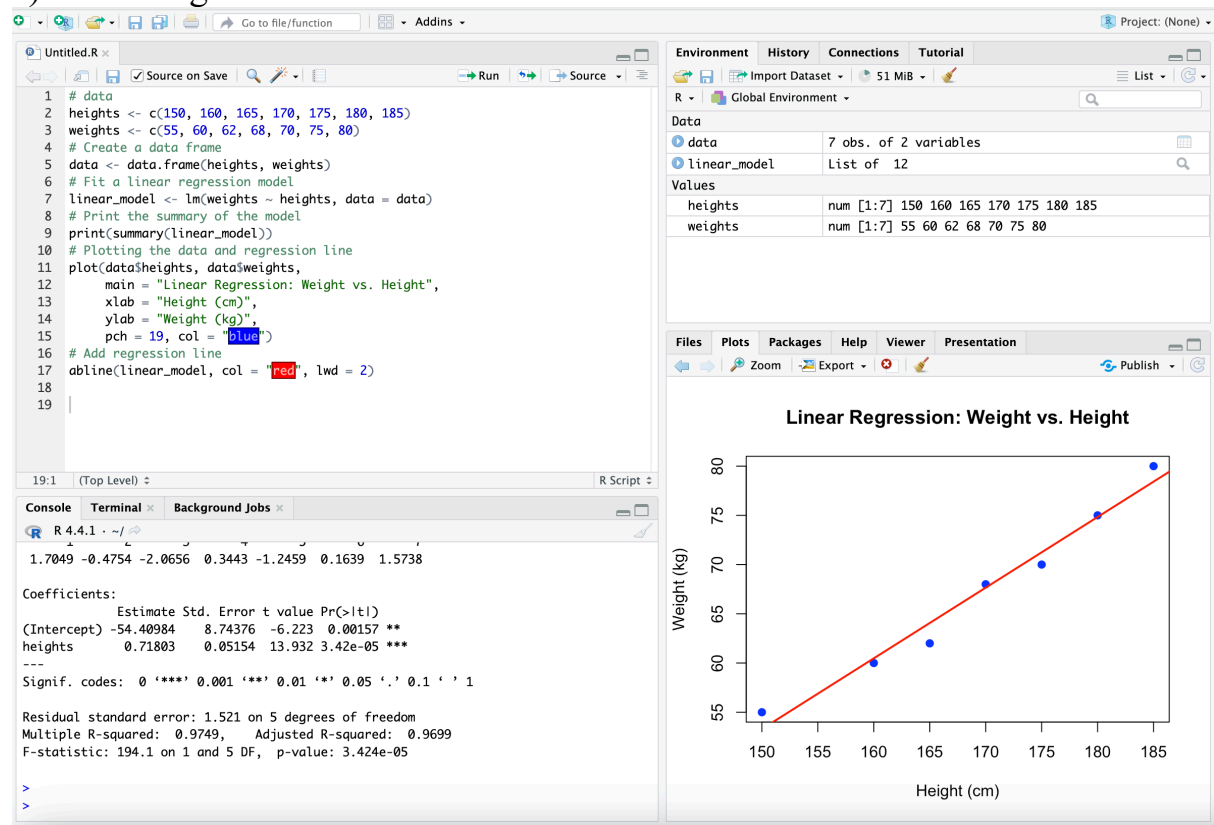
AIM:

    To Implement Linear and Logistic Regression using R
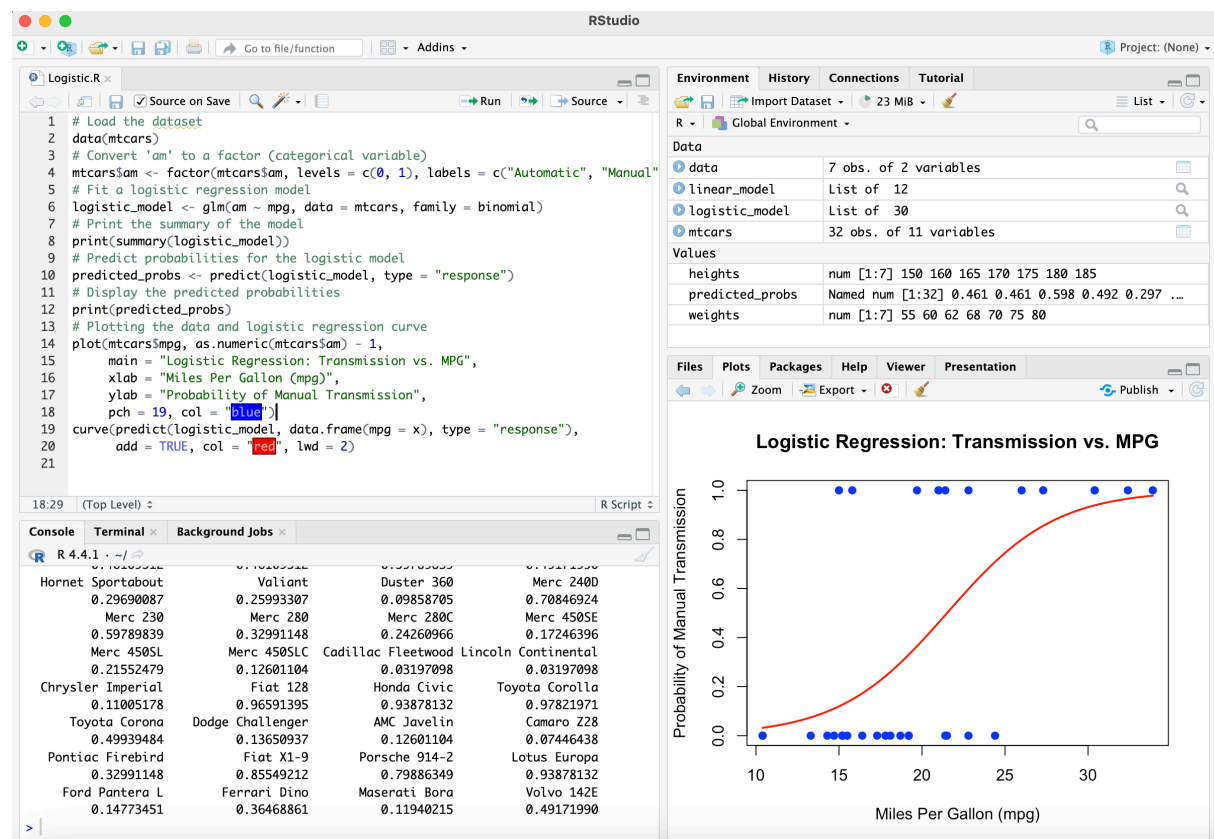
PROCEDURE:

1. Collect and load the dataset from sources like CSV files or databases.
2. Clean and preprocess the data, including handling missing values and encoding categorical variables.
3. Split the dataset into training and testing sets to evaluate model performance.
4. Normalize or standardize the features to ensure consistent scaling.
5. Choose the appropriate model: Linear Regression for continuous outcomes.
6. Train the model on the training data using the `fit` method.
7. Make predictions on the testing data using the `predict` method.
8. Evaluate the model using metrics like Mean Squared Error (MSE) for Linear Regression or accuracy and confusion matrix for Logistic Regression.
9. Visualize the results with plots, such as scatter plots for Linear Regression or decision boundaries for Logistic Regression.
10. Fine-tune the model by adjusting hyperparameters or applying regularization Techniques.

OUTPUT:

a) Linear Regression:


```r
1   # 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
```

Console output:

```
 1.7049 -0.4754 -2.0656  0.3443 -1.2459  0.1639  1.5738

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -54.40984    8.74376  -6.223  0.00157 **
heights       0.71803    0.05154  13.932 3.42e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.521 on 5 degrees of freedom
Multiple R-squared:  0.9749,    Adjusted R-squared:  0.9699
F-statistic: 194.1 on 1 and 5 DF,  p-value: 3.424e-05
```

b) Logistic Regression:


```r
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  curve(predict(logistic_model, data.frame(mpg = x), type = "response"),
20        add = TRUE, col = "red", lwd = 2)
21
```

Console output:

```
   Hornet Sportabout             Valiant          Duster 360           Merc 240D
          0.29690087          0.25993307          0.09858705          0.70846924
             Merc 230            Merc 280            Merc 280C           Merc 450SE
          0.59789839          0.32991148          0.24260966          0.17246396
            Merc 450SL        Merc 450SLC Cadillac Fleetwood Lincoln Continental
          0.21552479          0.12601104          0.03197098          0.03197098
      Chrysler Imperial           Fiat 128          Honda Civic       Toyota Corolla
          0.11005178          0.96591395          0.93878132          0.97821971
          Toyota Corona    Dodge Challenger          AMC Javelin           Camaro Z28
          0.49939842          0.13650937          0.12601104          0.07446438
       Pontiac Firebird           Fiat X1-9        Porsche 914-2         Lotus Europa
          0.32991148          0.85549212          0.79886349          0.93878132
         Ford Pantera L        Ferrari Dino        Maserati Bora           Volvo 142E
          0.14773451          0.36468861          0.11940215          0.49171990
```

RESULT:

      Thus to Implement Linear and Logistic Regression using R has been successfully executed.