

Ex No 7**Implement Linear and Logistic Regression in R****AIM:**

To Implement Linear and Logistic Regression using R

PROCEDURE:

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

CODE:**LinearRegression.R:**

```
# 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)
```

LogisticRegression.R:

```
# 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")
```

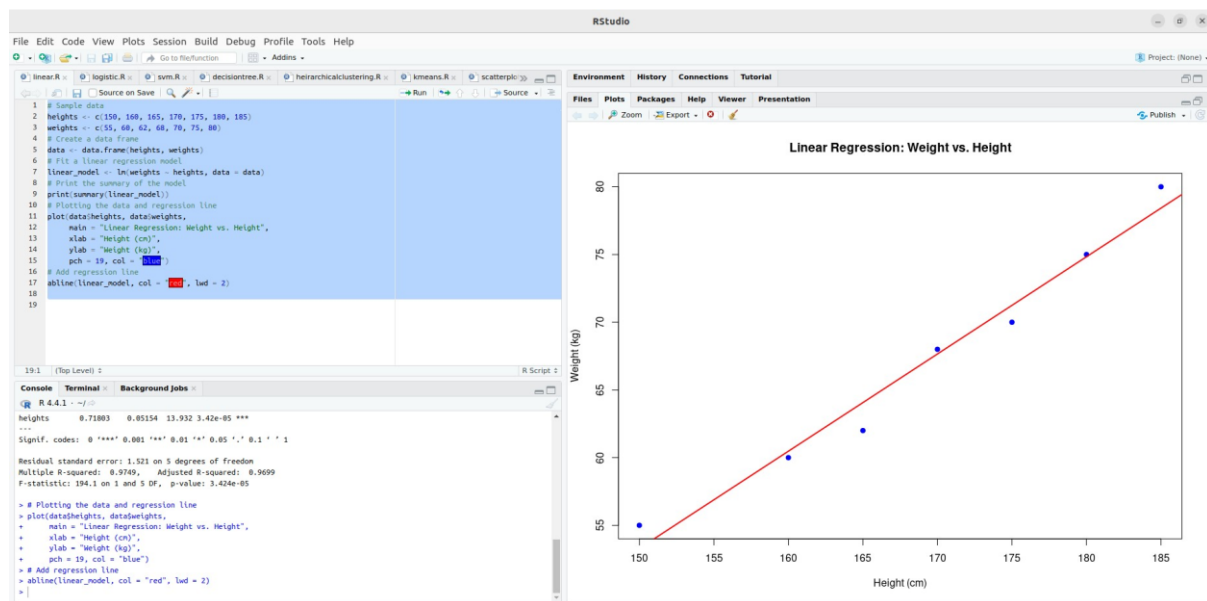
```
# Add the logistic regression curve
```

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

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

OUTPUT:

Linear Regression:



Logistic Regression:

