Exp. No : 7

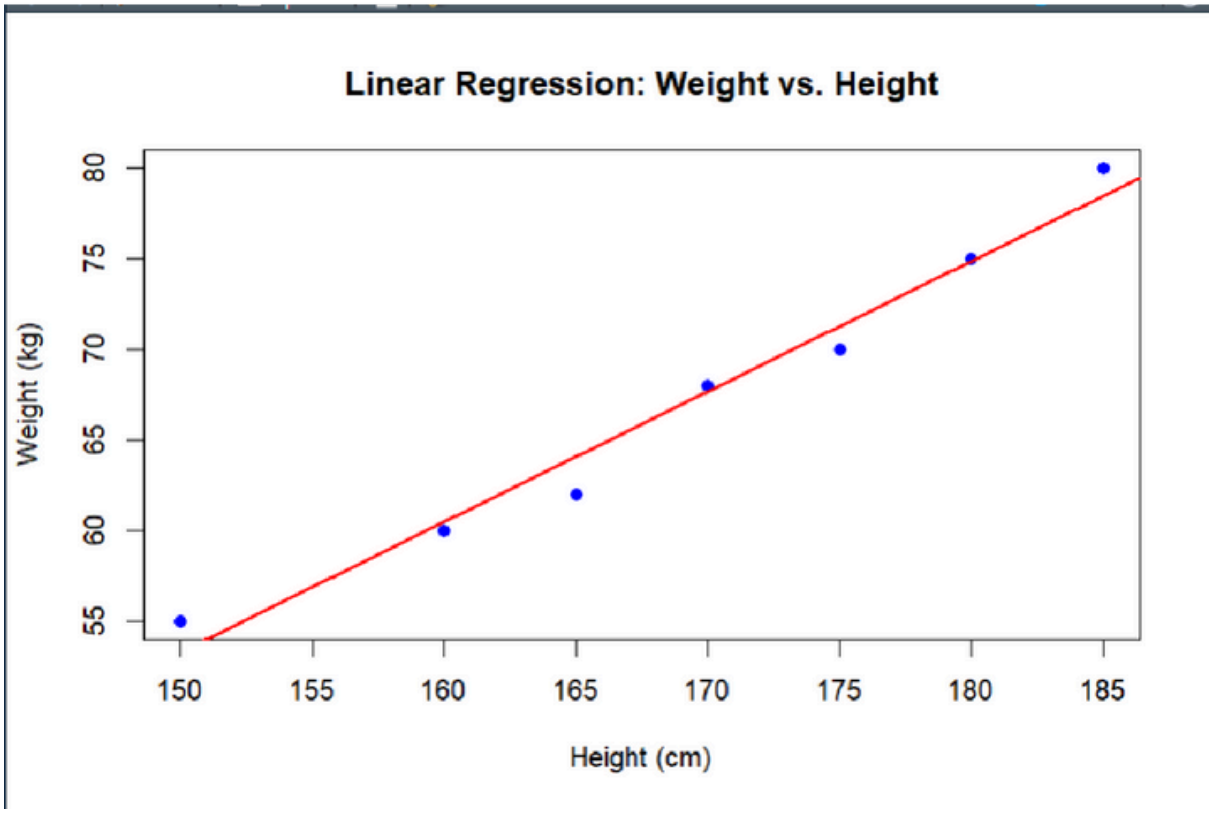
Implementing Linear and Logistic Regression

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:

```
Console   Terminal ×   Background Jobs ×
R 4.4.1 · ~/

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

> # 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)
>
```



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:

```
Residual deviance: 29.675  on 30  degrees of freedom
AIC: 33.675

Number of Fisher Scoring iterations: 5

          Mazda RX4       Mazda RX4 Wag          Datsun 710
         0.46109512          0.46109512          0.59789839
     Hornet 4 Drive   Hornet Sportabout             Valiant
         0.49171990          0.29690087          0.25993307
         Duster 360           Merc 240D            Merc 230
         0.09858705          0.70846924          0.59789839
           Merc 280           Merc 280C           Merc 450SE
         0.32991148          0.24260966          0.17246396
          Merc 450SL         Merc 450SLC  Cadillac Fleetwood
         0.21552479          0.12601104          0.03197098
Lincoln Continental   Chrysler Imperial            Fiat 128
         0.03197098          0.11005178          0.96591395
        Honda Civic      Toyota Corolla       Toyota Corona
         0.93878132          0.97821971          0.49939484
   Dodge Challenger         AMC Javelin          Camaro Z28
         0.13650937          0.12601104          0.07446438
   Pontiac Firebird           Fiat X1-9       Porsche 914-2
         0.32991148          0.85549212          0.79886349
      Lotus Europa     Ford Pantera L        Ferrari Dino
         0.93878132          0.14773451          0.36468861
     Maserati Bora          Volvo 142E
         0.11940215          0.49171990
```