Ex no: 7

Implement Linear and Logistic Regression

Aim:

To implement Linear and Logistic regression in R programming.

Procedure:

a) Linear Regression:

1. Define two numeric vectors for heights and weights.
2. Create a data frame using the heights and weights vectors.
3. Fit a linear regression model using the `lm()` function with weights as the dependent variable and heights as the independent variable.
4. Print the summary of the linear model to display coefficients, R-squared value, and p-values.
5. Plot a scatter plot of the heights and weights to visualize the data.
6. Add axis labels and customize the appearance of the scatter plot.
7. Use the `abline()` function to add the linear regression line to the plot.
8. Analyze the coefficients to understand the relationship between height and weight.
9. Review the R-squared value to assess how well the model fits the data.
10. Optionally use the model for predictions with new height values using the `predict()` function.

b) Logistic Regression:

1. Load the `mtcars` dataset into the environment.
2. Convert the `am` variable from numeric to a categorical factor with levels "Automatic" and "Manual."
3. Fit a logistic regression model using `glm()` with `am` as the dependent variable and `mpg` (miles per gallon) as the independent variable, specifying the binomial family.
4. Print the summary of the logistic model to display coefficients, p-values, and model fit information.
5. Use the `predict()` function to calculate the predicted probabilities of manual transmission based on `mpg` values.
6. Print the predicted probabilities for each observation in the dataset.

7. Plot the data, using `mpg` on the x-axis and the actual transmission type (converted to 0 or 1) on the y-axis.
8. Label the x-axis as "Miles Per Gallon (mpg)" and the y-axis as "Probability of Manual Transmission."
9. Use the `curve()` function to add the fitted logistic regression curve to the plot, representing the predicted probabilities over different `mpg` values.
10. Customize the plot appearance by adding color, point markers, and line thickness for clarity.

Program:

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)
```

b) Logistic Regression:

```
# Load the dataset
data(mtcars)
```

```r
# 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:

a) **Linear Regression:**

Console Terminal Background Jobs

R 4.4.1 · ~/

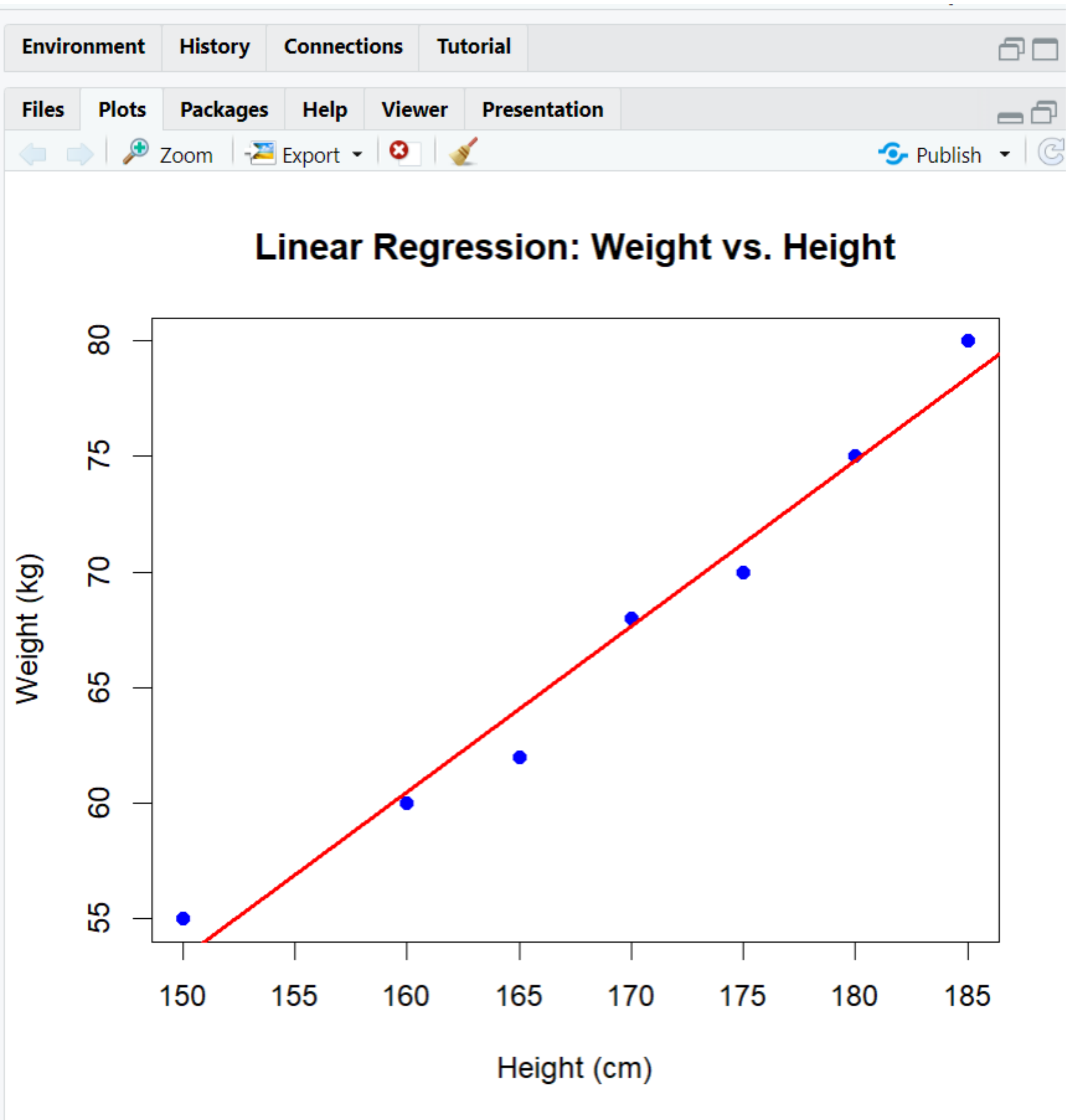
```
> # Sample data
> heights <- c(150, 160, 165, 170, 175, 180, 185)

> 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))

Call:
lm(formula = weights ~ heights, data = data)

Residuals:
      1       2       3       4       5       6       7 
 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


> # Plotting the data and regression line
> plot(data$heights, data$weights,
+      main = "Linear Regression: Weight vs. Height",
+      xlab = "Heig ..." ... [TRUNCATED]

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

Environment History Connections Tutorial

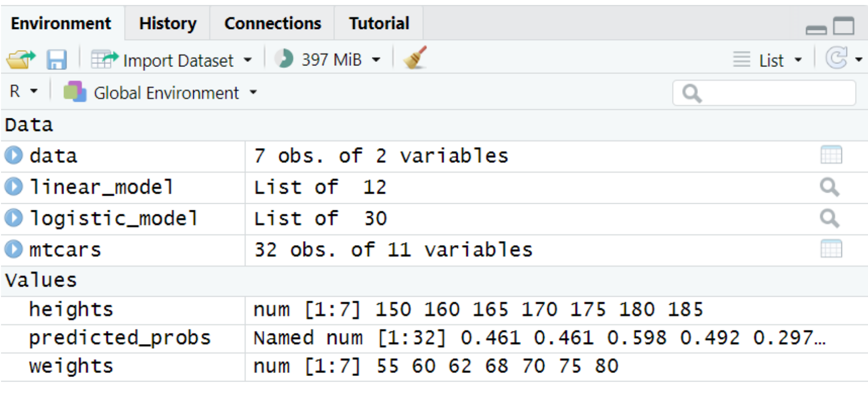
Import Dataset 397 MiB List

R Global Environment

Data	
data	7 obs. of 2 variables
linear_model	List of 12
Values	
heights	num [1:7] 150 160 165 170 175 180 185
weights	num [1:7] 55 60 62 68 70 75 80

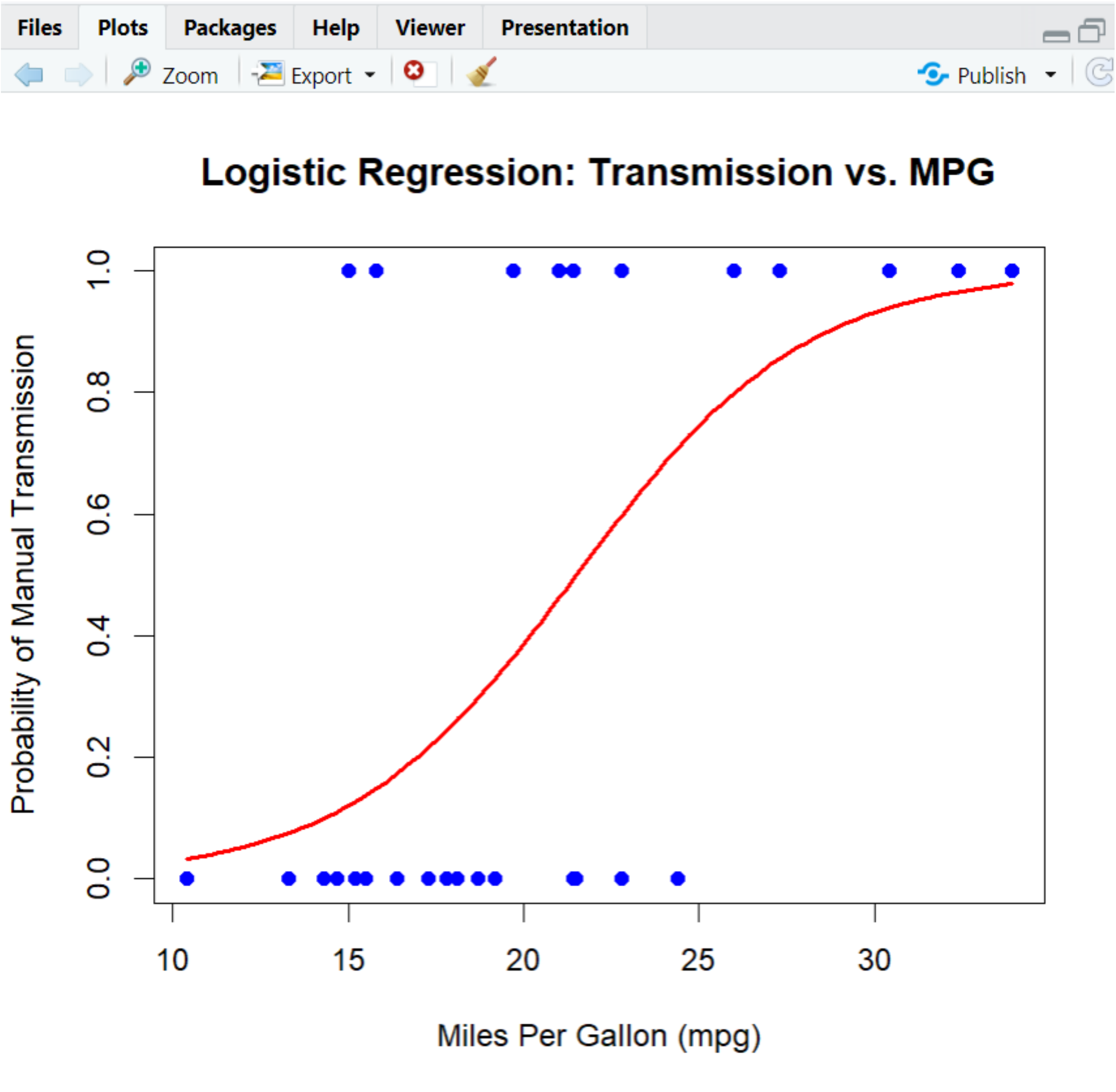


b) Logistic Regression:



```
> # Load the dataset
> data(mtcars)

> # Convert 'am' to a factor (categorical variable)
> mtcars$am <- factor(mtcars$am, levels = c(0, 1), labels = c("Automatic", "Manua
l"))

> # Fit a logistic regression model
> logistic_model <- glm(am ~ mpg, data = mtcars, family = binomial)

> # Print the summary of the model
> print(summary(logistic_model))

Call:
glm(formula = am ~ mpg, family = binomial, data = mtcars)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -6.6035     2.3514  -2.808  0.00498 **
mpg           0.3070     0.1148   2.673  0.00751 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.230  on 31  degrees of freedom
Residual deviance: 29.675  on 30  degrees of freedom
AIC: 33.675

Number of Fisher Scoring iterations: 5

> # Predict probabilities for the logistic model
> predicted_probs <- predict(logistic_model, type = "response")

> # Display the predicted probabilities
> print(predicted_probs)
          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
```

```
> # Plotting the data and logistic regression curve
> plot(mtcars$mpg, as.numeric(mtcars$am) - 1,
+      main = "Logistic Regression: Transmission vs. ..." ... [TRUNCATED]

> # Add the logistic regression curve
> curve(predict(logistic_model, data.frame(mpg = x), type = "response"),
+       add = TRUE, col = "red", lwd = .... [TRUNCATED]
> 
```



Result:

Thus the implementation of Linear and Logistic regression has been executed successfully.