Sheth L.U.J. & Sir M.V. College

Manual Method_Linear Regression

Aim :- To study the relationship between the **number of playlists** and **song streams** and to predict streams using **Simple Linear Regression** in R.

Dataset

- **File name:** Song.csv
- **Source:** Local CSV file
- **Variables used:**
  - **Independent Variable (X):** Number of Playlists
  - **Dependent Variable (Y):** Streams (converted into millions)

Theory

Linear Regression is a statistical technique used to model the relationship between a dependent variable and an independent variable by fitting a straight line.

Regression Equation:

$$Y = \beta_0 + \beta_1 X$$

Where:

- Y = Streams (in millions)
- X = Number of Playlists
- β_0 = Intercept
- β_1 = Slope

Name :- Priya Gupta
Roll No :- S081

Sheth L.U.J. & Sir M.V. College

Procedure

1. Loaded the CSV dataset into R using read.csv().
2. Converted required columns (playlist and streams) into numeric format.
3. Removed missing values using na.omit().
4. Converted streams into millions for better interpretation.
5. Constructed the design matrix manually.
6. Calculated regression coefficients using matrix method:

$$\beta = (X^T X)^{-1} X^T Y$$

7. Calculated predicted values and residuals.
8. Computed **R-squared** to measure goodness of fit.
9. Predicted streams for a new playlist value.
10. Visualized data using scatter plot and regression line.

Results

Regression Coefficients

- Intercept (β_0) = 167.638
- Slope (β_1) = 0.0815

Regression Equation:

$$\text{Streams (in millions)} = 167.638 + 0.0815 \times \text{(Number of Playlists)}$$

R-squared Value

$$R^2 = 0.7809$$

Interpretation:
Approximately **78.09%** of the variation in song streams is explained by the number of playlists, indicating a **strong positive relationship**.

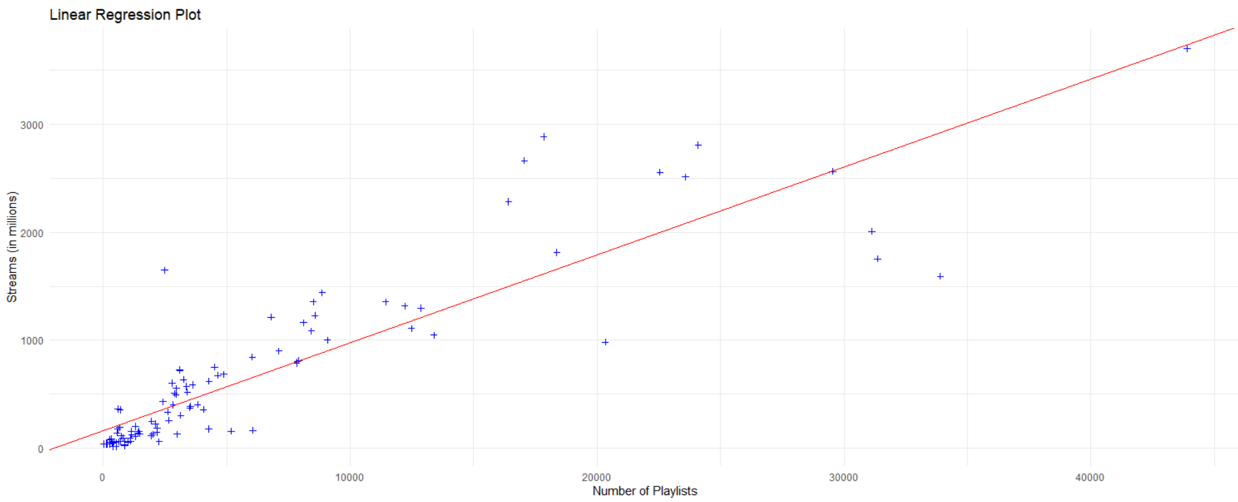
Prediction

Name :- Priya Gupta
Roll No :- S081

Sheth L.U.J. & Sir M.V. College

For **2000 playlists**:
Predicted Streams=330.62 million

Graph



Graph Description:

- Blue points represent actual data values.
- Red line represents the regression line.
- The upward slope shows a positive linear relationship between playlists and streams.

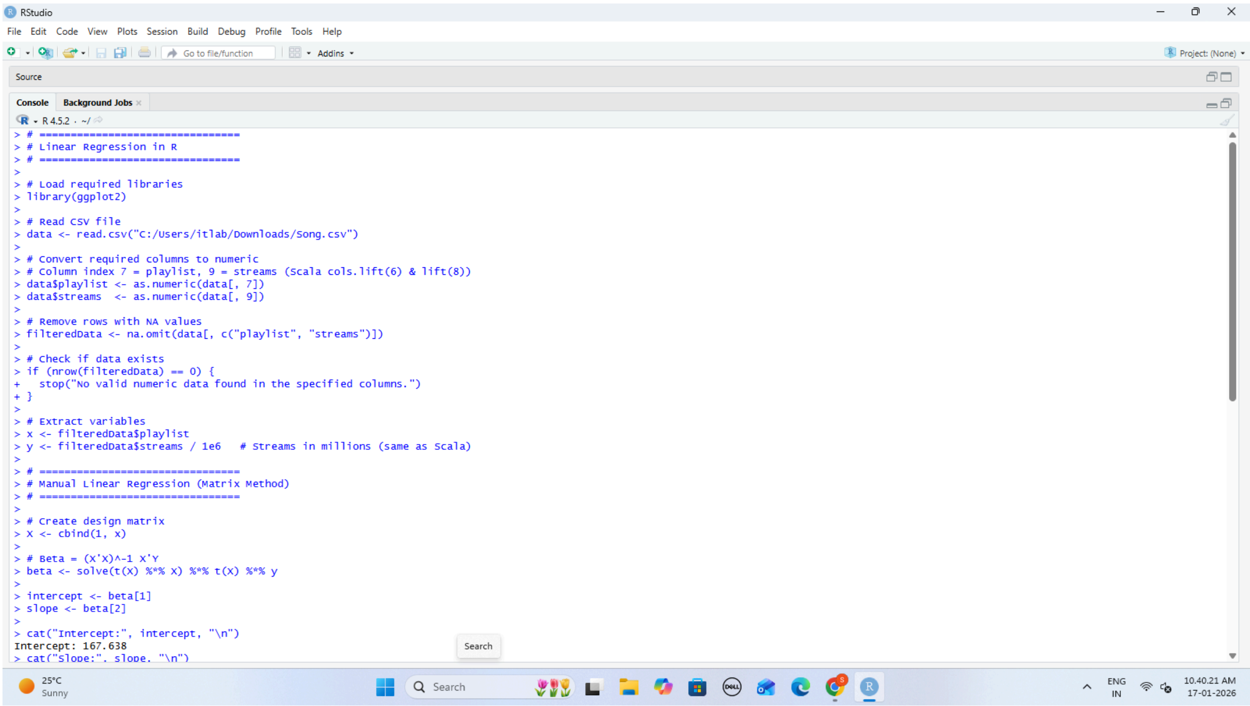
Conclusion

The analysis shows that the number of playlists has a significant positive impact on song streams. As the number of playlists increases, the streams also increase. The high R-squared value confirms that the linear regression model fits the data well and can be used for prediction.

Name :- Priya Gupta
Roll No :- S081

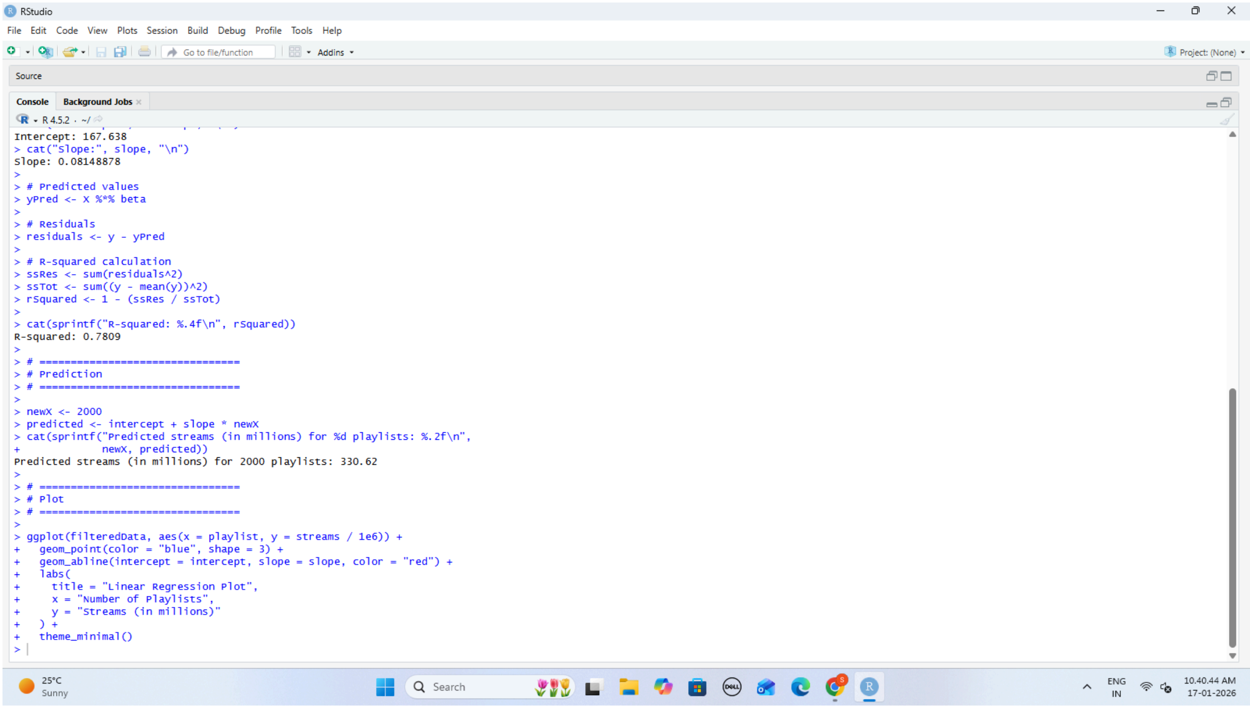
Sheth L.U.J. & Sir M.V. College

<u>Screenshots</u>



Name :- Priya Gupta
Roll No :- S081

Sheth L.U.J. & Sir M.V. College



```
Intercept: 167.638
> cat("Slope:", slope, "\n")
Slope: 0.08148878
>
> # Predicted values
> yPred <- X %*% beta
>
> # Residuals
> residuals <- y - yPred
>
> # R-squared calculation
> ssRes <- sum(residuals^2)
> ssTot <- sum((y - mean(y))^2)
> rSquared <- 1 - (ssRes / ssTot)
>
> cat(sprintf("R-squared: %.4f\n", rSquared))
R-squared: 0.7809
>
> # ================================
> # Prediction
> # ================================
>
> newX <- 2000
> predicted <- intercept + slope * newX
> cat(sprintf("Predicted streams (in millions) for %d playlists: %.2f\n",
+             newX, predicted))
Predicted streams (in millions) for 2000 playlists: 330.62
>
> # ================================
> # Plot
> # ================================
>
> ggplot(filteredData, aes(x = playlist, y = streams / 1e6)) +
+   geom_point(color = "blue", shape = 3) +
+   geom_abline(intercept = intercept, slope = slope, color = "red") +
+   labs(
+     title = "Linear Regression Plot",
+     x = "Number of Playlists",
+     y = "Streams (in millions)"
+   ) +
+   theme_minimal()
>
```

Name :- Priya Gupta
Roll No :- S081

Sheth L.U.J. & Sir M.V. College

Manual Method_Logistic Regression

Aim :- To implement **Logistic Regression** using **Gradient Descent** in R and predict whether a student will **Pass or Fail** based on the number of **Hours Studied**.

Dataset Description

- **Source:** CSV file (hours_scores.csv)
- **Total Observations:** 25
- **Columns Used:**
  - Hours – Number of hours studied (Independent Variable)
  - Scores – Marks obtained
- A new binary column **Pass** was created:
  - Pass = 1, if Scores ≥ 50
  - Fail = 0, if Scores < 50

Objective of Logistic Regression

Since the output variable (Pass/Fail) is **binary**, Logistic Regression is used instead of Linear Regression. It predicts the **probability** of passing based on study hours.

Methodology / Steps Performed

Step 1: Data Loading

The dataset was loaded using read.csv() and column names were verified to ensure correctness.

Step 2: Creation of Binary Output Variable

Name :- Priya Gupta
Roll No :- S081

Sheth L.U.J. & Sir M.V. College

A new variable Pass was created using the condition:

- Pass = 1 (Score ≥ 50)
- Fail = 0 (Score < 50)

Step 3: Feature Normalization

To improve convergence of Gradient Descent, the Hours variable was normalized using:

Formula:

$$X_{norm} = \frac{X - \mu}{\sigma}$$

Where:

- μ = Mean of Hours
- σ = Standard Deviation of Hours

Step 4: Logistic Regression Model

Logistic Regression uses the **Sigmoid Function**:

Sigmoid Formula:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Where:

$$z = \theta_0 + \theta_1 X$$

Name :- Priya Gupta
Roll No :- S081

Step 5: Cost Function

The **Log Loss (Cross Entropy Loss)** function was used:

Formula:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right]$$

Step 6: Gradient Descent Optimization

Model parameters were updated iteratively using:

Gradient Descent Formula:

Gradient Descent Formula:

$$\theta := \theta - \alpha\frac{1}{m}X^T(h_\theta(X) - y)$$

- Learning Rate (α) = 0.1
- Iterations = 5000

Model Training Output

Cost Reduction (Convergence)

The cost function continuously decreased, showing successful learning:

```
+ }
Iteration 500: Cost = 0.1376
Iteration 1000: Cost = 0.1183
Iteration 1500: Cost = 0.1107
Iteration 2000: Cost = 0.1065
Iteration 2500: Cost = 0.1039
Iteration 3000: Cost = 0.1021
Iteration 3500: Cost = 0.1008
Iteration 4000: Cost = 0.0998
Iteration 4500: Cost = 0.0990
Iteration 5000: Cost = 0.0984
>
```

This confirms that the model converged properly.

Final Model Parameters

$$\theta_0 = 0.1640$$

$$\theta_1 = 7.3527$$

A high value of θ_1 indicates that **study hours strongly influence passing probability.**

Graphical Representation

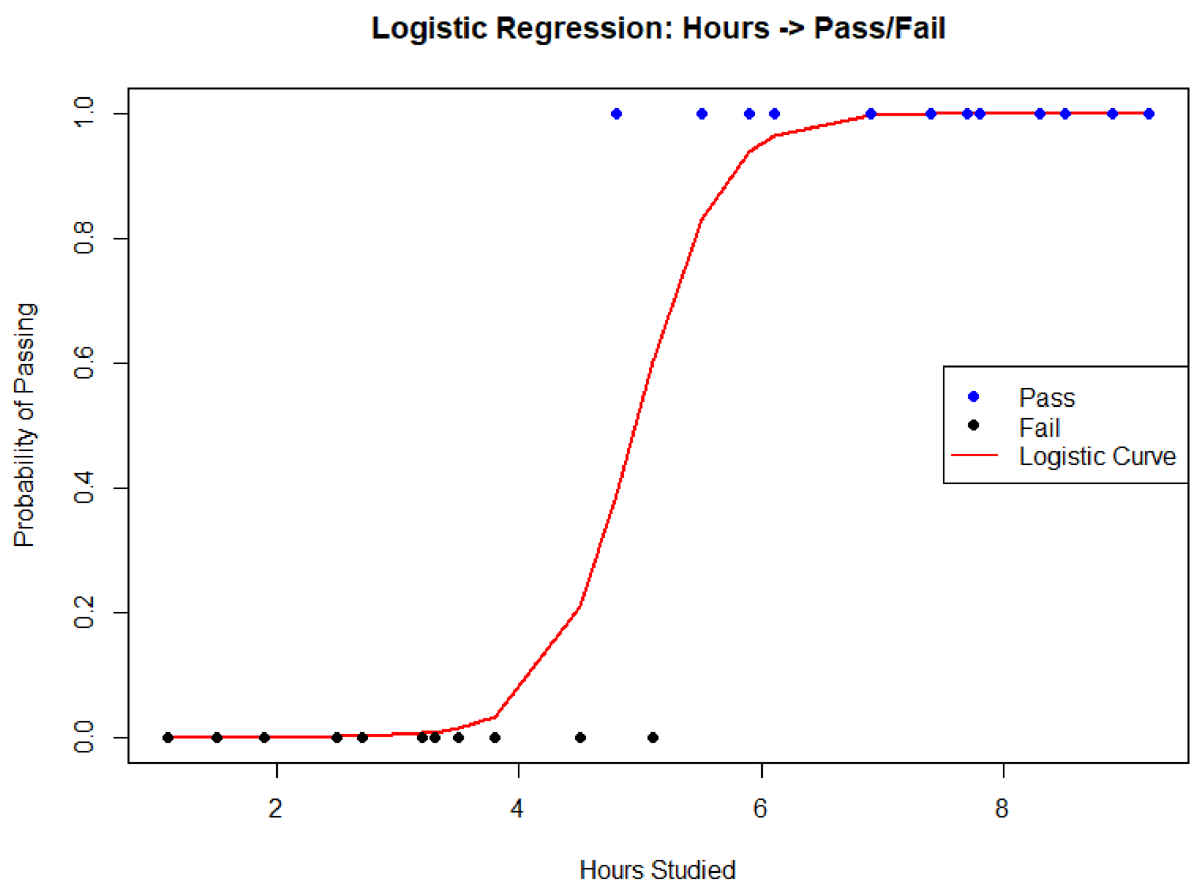
Description of the Graph

- **X-axis:** Hours Studied
- **Y-axis:** Probability of Passing
- **Red Curve:** Logistic Regression Curve
- **Blue Points:** Passed students
- **Black Points:** Failed students

Observation from Graph:

- Below ~4 hours → Probability of passing is very low
- Around 5 hours → Sharp increase in passing probability

- Above 6 hours → Probability approaches 1 (almost certain pass)



Result

The Logistic Regression model successfully predicts **Pass/Fail outcomes** based on study hours. The probability of passing increases significantly as the number of study hours increases.

Conclusion

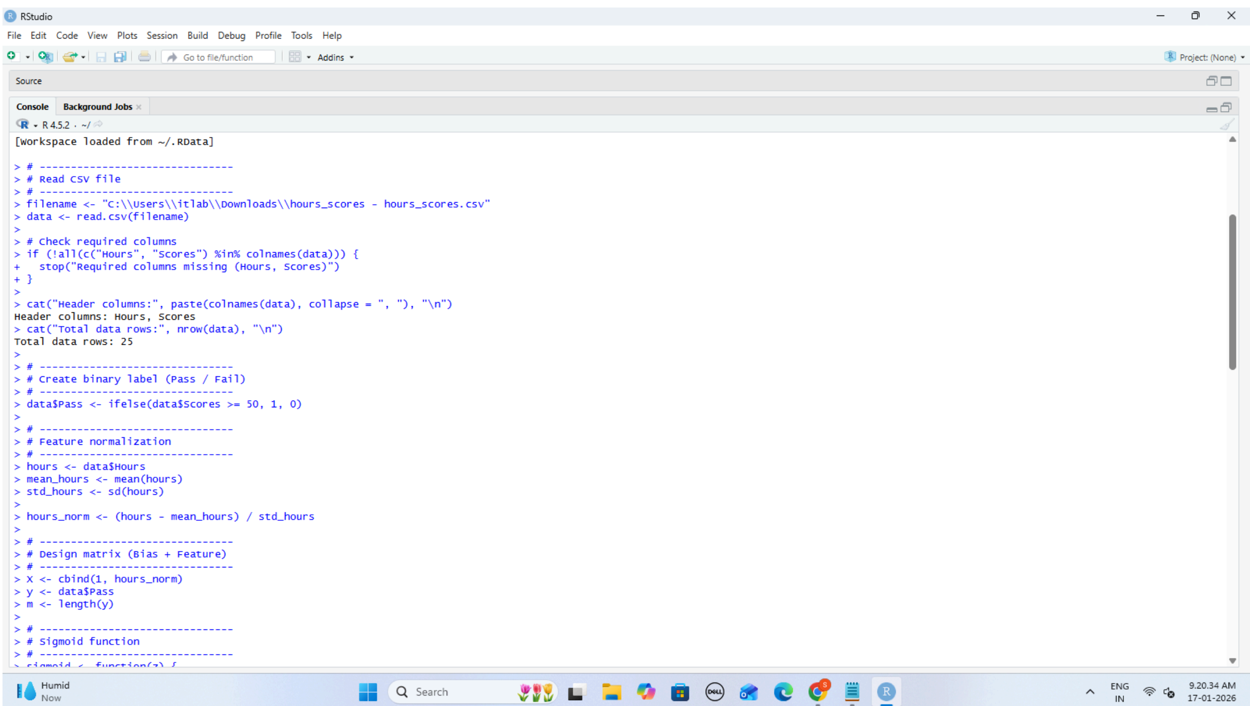
- Logistic Regression is suitable for **binary classification problems**
- Feature normalization improves learning efficiency

Name :- Priya Gupta
Roll No :- S081

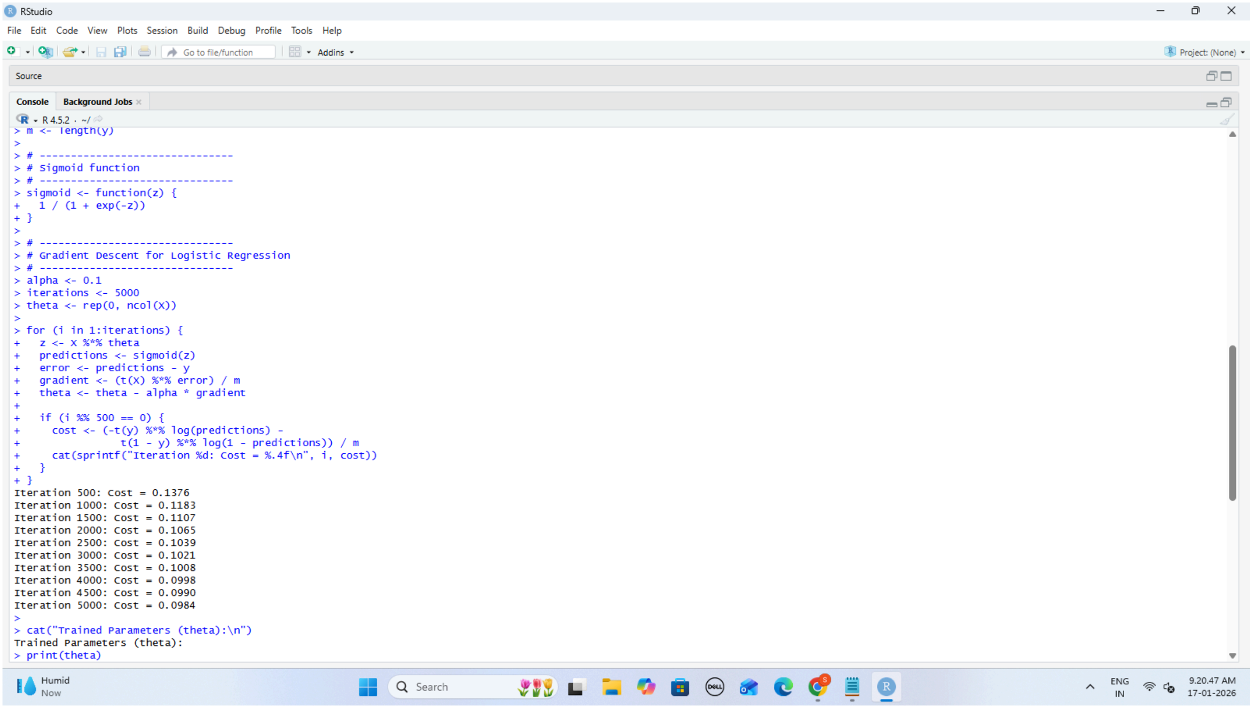
- Gradient Descent effectively minimized the cost function
- Study hours have a **strong positive impact** on exam success

Thus, the objective of predicting Pass/Fail status using Logistic Regression was **successfully achieved**.

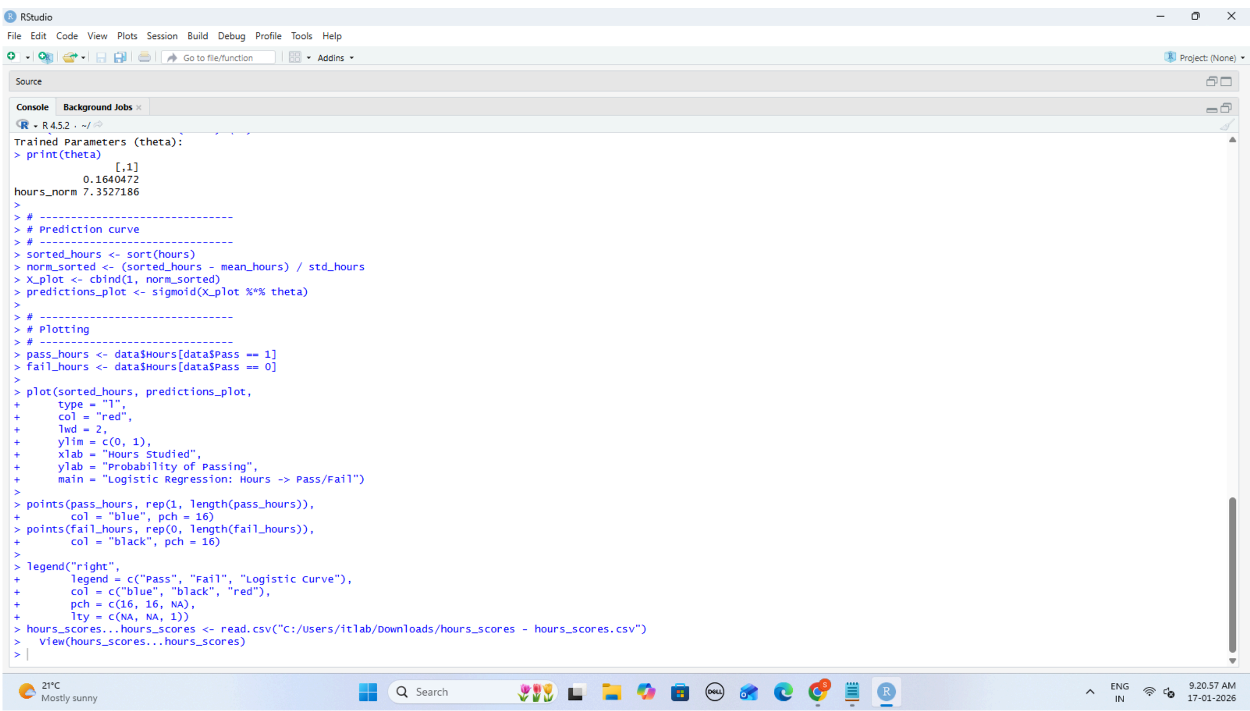
Screenshots



Sheth L.U.J. & Sir M.V. College



```
> m <- length(y)
>
> # ------------------------------
> # Sigmoid function
> # ------------------------------
> sigmoid <- function(z) {
+   1 / (1 + exp(-z))
+ }
>
> # ------------------------------
> # Gradient Descent for Logistic Regression
> # ------------------------------
> alpha <- 0.1
> iterations <- 5000
> theta <- rep(0, ncol(X))
>
> for (i in 1:iterations) {
+   z <- X %*% theta
+   predictions <- sigmoid(z)
+   error <- predictions - y
+   gradient <- (t(X) %*% error) / m
+   theta <- theta - alpha * gradient
+
+   if (i %% 500 == 0) {
+     cost <- (-t(y) %*% log(predictions) -
+               t(1 - y) %*% log(1 - predictions)) / m
+     cat(sprintf("Iteration %d: Cost = %.4f\n", i, cost))
+   }
+ }
Iteration 500: Cost = 0.1376
Iteration 1000: Cost = 0.1183
Iteration 1500: Cost = 0.1107
Iteration 2000: Cost = 0.1065
Iteration 2500: Cost = 0.1039
Iteration 3000: Cost = 0.1021
Iteration 3500: Cost = 0.1008
Iteration 4000: Cost = 0.0998
Iteration 4500: Cost = 0.0990
Iteration 5000: Cost = 0.0984
>
> cat("Trained Parameters (theta):\n")
Trained Parameters (theta):
> print(theta)
```



```
Trained Parameters (theta):
> print(theta)
                 [,1]
            0.1640472
hours_norm 7.3527186
>
> # ------------------------------
> # Prediction curve
> # ------------------------------
> sorted_hours <- sort(hours)
> norm_sorted <- (sorted_hours - mean_hours) / std_hours
> X_plot <- cbind(1, norm_sorted)
> predictions_plot <- sigmoid(X_plot %*% theta)
>
> # ------------------------------
> # Plotting
> # ------------------------------
> pass_hours <- data$Hours[data$Pass == 1]
> fail_hours <- data$Hours[data$Pass == 0]
>
> plot(sorted_hours, predictions_plot,
+      type = "l",
+      col = "red",
+      lwd = 2,
+      ylim = c(0, 1),
+      xlab = "Hours Studied",
+      ylab = "Probability of Passing",
+      main = "Logistic Regression: Hours -> Pass/Fail")
>
> points(pass_hours, rep(1, length(pass_hours)),
+        col = "blue", pch = 16)
> points(fail_hours, rep(0, length(fail_hours)),
+        col = "black", pch = 16)
>
> legend("right",
+        legend = c("Pass", "Fail", "Logistic Curve"),
+        col = c("blue", "black", "red"),
+        pch = c(16, 16, NA),
+        lty = c(NA, NA, 1))
> hours_scores...hours_scores <- read.csv("C:/Users/itlab/Downloads/hours_scores - hours_scores.csv")
>   View(hours_scores...hours_scores)
> |
```

Name :- Priya Gupta
Roll No :- S081

Sheth L.U.J. & Sir M.V. College

Name :- Priya Gupta
Roll No :- S081