

# **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
  - $\beta_0$  = Intercept
  - $\beta_1$  = 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 ( $\beta_0$ ) = 167.638
- Slope ( $\beta_1$ ) = 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**

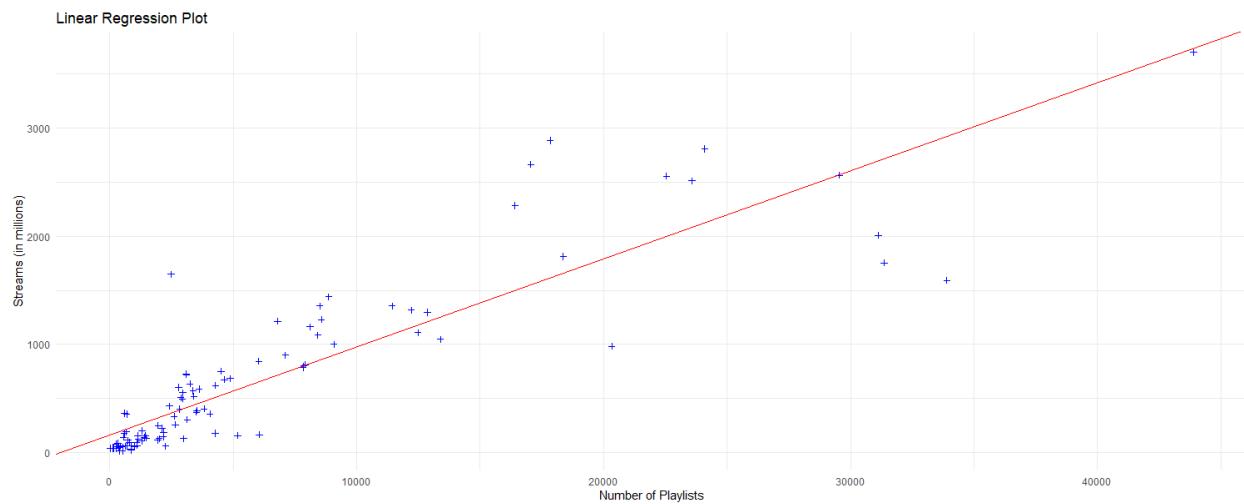
# Sheth L.U.J. & Sir M.V. College

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

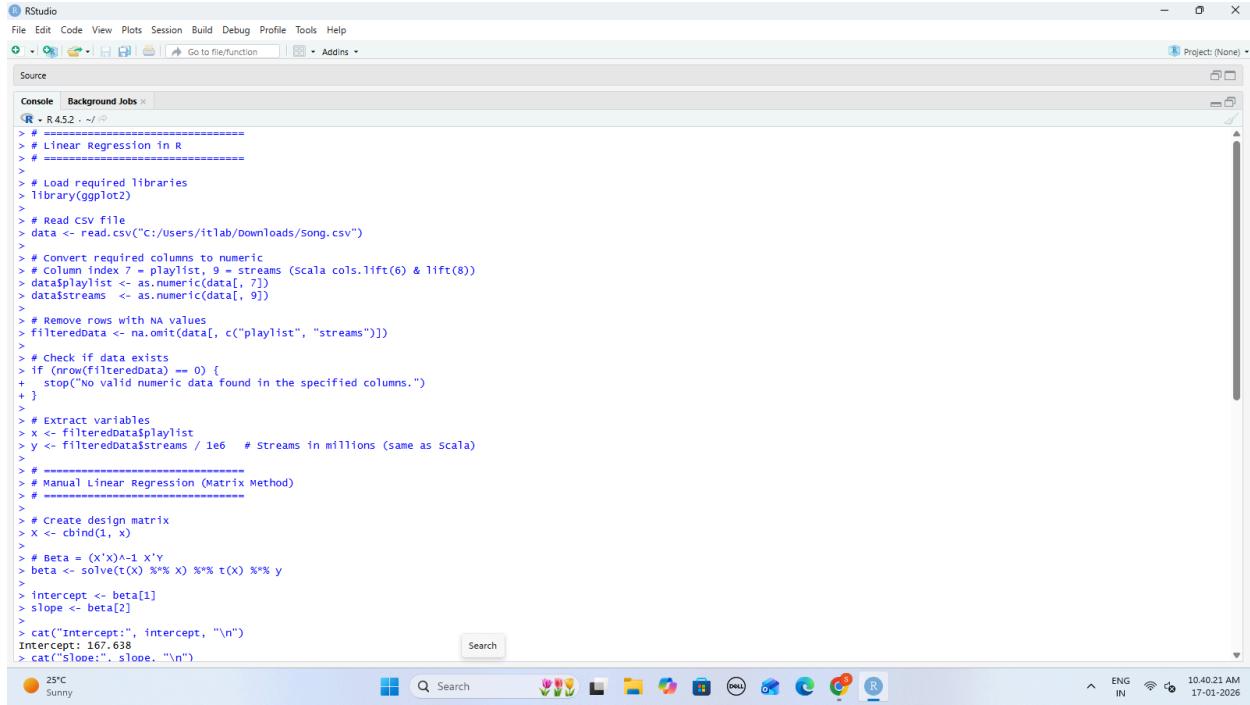
The analysis shows that the number of playlists has a significant positive impact on song streams. As the number of playlists increases, the streams also increase. The high R-squared value confirms that the linear regression model fits the data well and can be used for prediction.

---

**Name :- Priya Gupta**  
**Roll No :- S081**

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

## Screenshots



The screenshot shows the RStudio interface with the following R code in the console:

```
R # =====
> # Linear Regression in R
> # =====
>
> # Load required libraries
> library(ggplot2)
>
> # Read CSV file
> data <- read.csv("C:/users/itlab/Downloads/Song.csv")
>
> # Convert required columns to numeric
> # column index 7 = playlist, 9 = streams (scala cols.lift(6) & lift(8))
> data$playlist <- as.numeric(data[, 7])
> datastreams <- as.numeric(data[, 9])
>
> # Remove rows with NA values
> filteredData <- na.omit(data[, c("playlist", "streams")])
>
> # Check if data exists
> if (nrow(filteredData) == 0) {
+   stop("No valid numeric data found in the specified columns.")
+ }
>
> # Extract variables
> x <- filteredData$playlist
> y <- filteredData$streams / 1e6  # Streams in millions (same as scala)
>
> # =====
> # Manual Linear Regression (Matrix Method)
> # =====
>
> # Create design matrix
> X <- cbind(1, x)
>
> # Beta = (X'X)^-1 X'y
> beta <- solve(t(X) %*% X) %*% t(X) %*% y
>
> intercept <- beta[1]
> slope <- beta[2]
>
> cat("Intercept:", intercept, "\n")
Intercept: 167.638
> cat("Slope:", slope, "\n")
```

The RStudio interface includes a menu bar (File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help), a toolbar with various icons, and a status bar at the bottom showing weather (25°C, Sunny), network (ENG IN), battery (10.40.21 AM), and date (17-01-2026).

**Name :- Priya Gupta**  
**Roll No :- S081**

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

The screenshot shows the RStudio interface with the following details:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Project:** Project: (None)
- Console Tab:** Shows R code for linear regression analysis. The code includes calculating the intercept, slope, residuals, R-squared value, and a prediction for 2000 playlists. It then uses ggplot2 to create a scatter plot with a linear regression line.
- Background Jobs Tab:** Shows a single job named "R 4.5.2 - ~/" with the status "Running".
- Plots Tab:** Shows a scatter plot titled "Linear Regression Plot" with "Number of Playlists" on the x-axis and "streams (in millions)" on the y-axis. A blue scatter of points is shown with a red linear regression line.
- Taskbar:** Includes icons for Search, Start button, File Explorer, File Manager, Task View, Dell logo, Taskbar, and R icon.
- System Tray:** Shows weather (25°C, Sunny), network (ENG IN), battery (10.40 AM), and date (17-01-2026).

**Name :- Priya Gupta**  
**Roll No :- S081**

# **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  $\geq$  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  $\geq 50$ )
  - 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:

- $\mu$  = Mean of Hours
  - $\sigma$  = 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 ( $\alpha$ ) = 0.1
  - Iterations = 5000
- 

**Model Training Output**

**Cost Reduction (Convergence)**

**Name :- Priya Gupta**  
**Roll No :- S081**

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

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  $\theta_1$  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

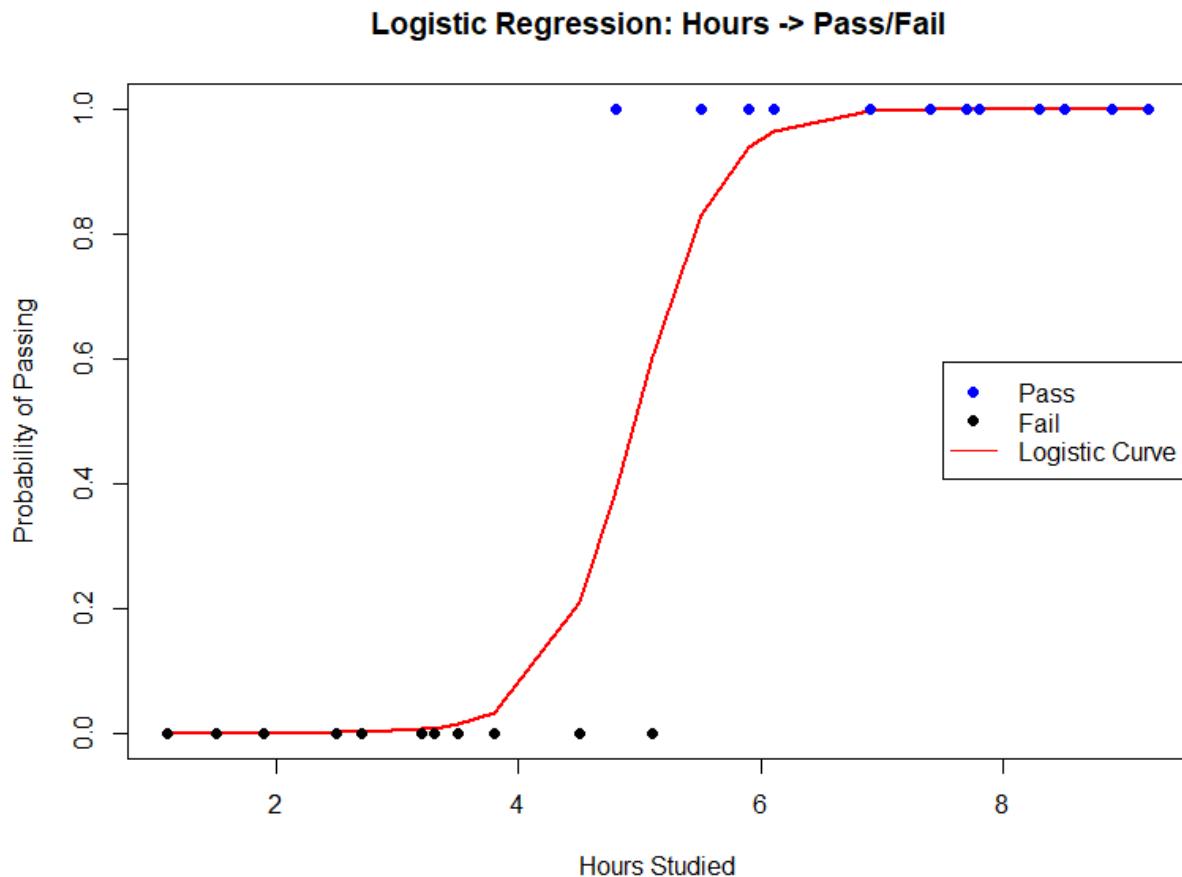
### Observation from Graph:

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

**Name :- Priya Gupta**  
**Roll No :- S081**

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

- 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**

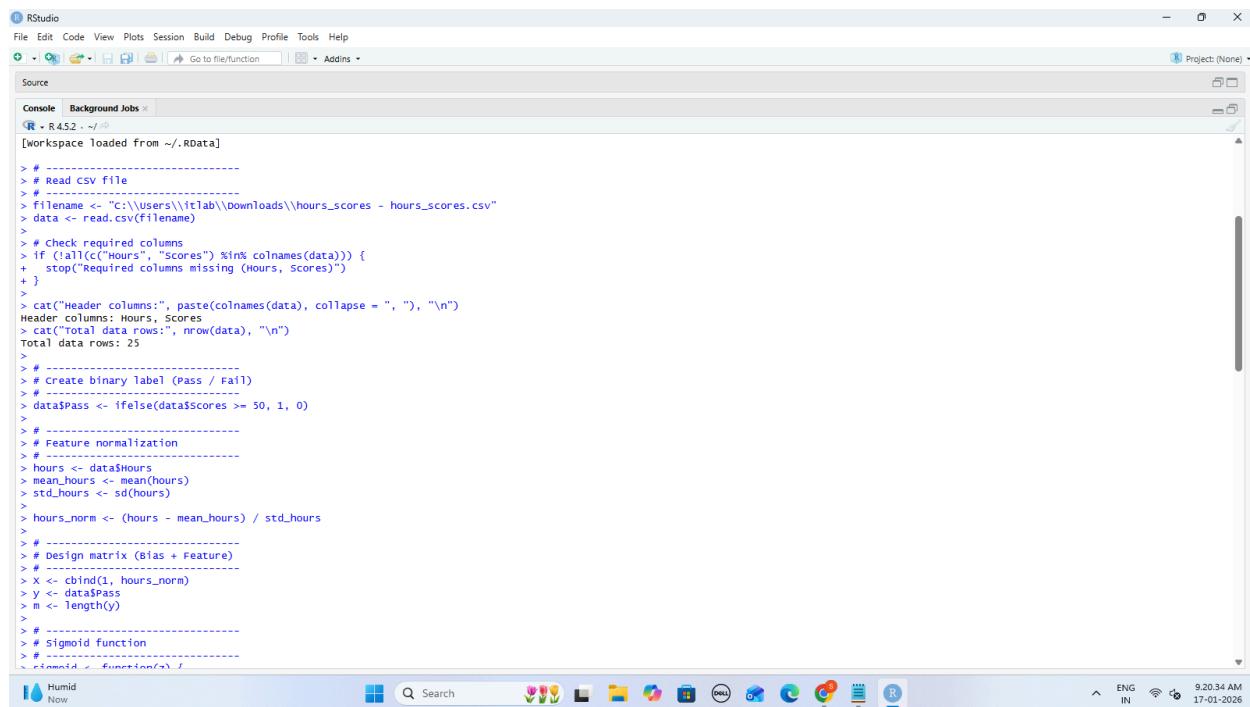
# Sheth L.U.J. & Sir M.V. College

- 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



The screenshot shows the RStudio interface with the following details:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Includes icons for file operations like Open, Save, Print, and a search bar labeled "Go to file/function".
- Console Tab:** Shows the R session output. The code is for reading a CSV file named "hours\_scores.csv" and performing various data preprocessing steps like feature normalization and creating a binary target variable "Pass". It also includes a design matrix and a sigmoid function.
- Background Jobs:** A tab showing "R 4.5.2 - ~" with the message "[workspace loaded from ~/RData]."
- Project Tab:** Shows "(None)".
- Bottom Status Bar:** Displays system information including "ENG IN", "9.20.34 AM", and the date "17-01-2026".

**Name :- Priya Gupta**  
**Roll No :- So81**

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

RStudio  
File Edit Code View Plots Session Build Debug Profile Tools Help  
Project: (None)  
Source  
Console | Background Jobs x  
R v4.5.2 - ~/  
> 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)

RStudio  
File Edit Code View Plots Session Build Debug Profile Tools Help  
Project: (None)  
Source  
Console | Background Jobs x  
R v4.5.2 - ~/  
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 <- dataHours[dataPass == 1]  
> fail\_hours <- dataHours[dataPass == 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 <- read.csv("C:/Users/itlab/downloads/hours\_scores - hours\_scores.csv")  
> view(hours\_scores...hours\_scores)

Name :- Priya Gupta  
Roll No :- So81

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

**Name :- Priya Gupta  
Roll No :- S081**