# How to Easily Perform Least Squares Regression in R

Authored by

**stats writer**

November 22, 2025

The Method of Least Squares (LMS) implemented within the statistical environment of R is an exceptionally powerful technique utilized globally for fitting statistical models, particularly those involving linear and certain types of nonlinear relationships. This methodology provides the foundational mathematical basis for standard regression analysis, allowing analysts and researchers to quantify the association between one or more independent variables and a dependent variable. The process involves minimizing the sum of the squared differences, known as residuals, between the observed data points and the values predicted by the model, thereby identifying the line or plane that best represents the overall trend in the data.

To effectively leverage the Method of Least Squares in R, a clear systematic approach is necessary. Initially, the structure of the model must be defined by carefully specifying the response (dependent) and explanatory (independent) variables. Subsequently, the core fitting process is executed using R's built-in lm() function, which stands for linear model. Once the model is fitted, the summary() function is crucial for obtaining a comprehensive statistical evaluation of the model parameters, including coefficient estimates, standard errors, and significance tests. Finally, visual inspection using R's powerful plotting capabilities, typically involving the plot() function, allows for confirmation of the model fit and helps identify potential violations of underlying assumptions. This integrated approach ensures robust data analysis and provides profound insights into the underlying relationships between variables.

## Understanding the Core Principle of Least Squares

The fundamental concept behind the method of least squares is to determine the unique regression line that minimizes the collective error between the observed data points and the fitted line. This technique calculates the distances (vertical residuals) from each data point to the proposed line, squares those distances, and sums them up. By selecting the line that yields the smallest possible sum of squared residuals, we achieve the statistically "best fit" for the given dataset. This principle is mathematically elegant and serves as the foundation for linear modeling across countless scientific and business applications where understanding correlation and causality is essential.

To better grasp the underlying mathematical derivation and geometric intuition behind minimizing the squared errors, studying the theoretical background is highly recommended. The following section provides a practical approach to implementing this theory using the capabilities inherent in the R statistical environment.

## Implementing Linear Models using the R `lm()` Function

To apply the robust machinery of the method of least squares for fitting a regression line in R, we rely almost exclusively on the built-in **lm()** function. This function is the primary tool for estimating

parameters of linear models and is designed to handle both simple linear regression (one predictor) and multiple linear regression (multiple predictors). Its efficiency and ease of use make it the standard approach for analysts working with numerical data in R.

The structure of the **lm()** function requires the user to specify the relationship between the variables using a standard R formula interface. This formula defines which variable is being predicted (the response) and which variable(s) are doing the predicting (the predictors). The resulting output object contains all necessary information about the model, including the calculated coefficients and statistical metrics needed for evaluation.

This powerful function uses the following fundamental syntax, which defines the model relationship and specifies the data frame where the variables reside:

**model <- lm(response ~ predictor, data=df)**

In this typical usage, **model** is the name assigned to the fitted linear model object. The formula specifies that the 'response' variable is modeled as a function of the 'predictor' variable. The **data=df** argument ensures that R looks for these variables within the specified data frame, denoted here as 'df'. The following comprehensive example illustrates exactly how to apply this function successfully within the R environment, moving from raw data creation to final visualization.

## Example: Applying Least Squares Regression in R

Let us consider a common scenario in educational statistics: analyzing the correlation between the number of hours a student dedicates to studying and their resulting examination score. Suppose we have compiled the performance data for 15 distinct students from a particular course. We aim to determine the best-fit regression line using the **method of least squares**, which will allow us to predict scores based on study time.

We begin by creating the necessary data frame in R. This structure organizes our observations, pairing the independent variable, **hours studied**, with the dependent variable, **exam score**. This organization is critical for the subsequent statistical modeling process.

**#create data frame**
**df <- data.frame(hours=c(1, 2, 4, 5, 5, 6, 6, 7, 8, 10, 11, 11, 12, 12, 14),**
**score=c(64, 66, 76, 73, 74, 81, 83, 82, 80, 88, 84, 82, 91, 93, 89))**

#view first six rows of data frame for verification
head(df)

hours score

1 1 64
2 2 66
3 4 76
4 5 73
5 5 74
6 6 81

The resulting data frame, named **df**, is now structured appropriately for applying linear modeling techniques. We can confirm that **hours** is the predictor variable and **score** is the response variable, setting the stage for the calculation of the least squares estimates.

## Fitting the Regression Model and Analyzing the Summary

With the data prepared, we proceed to use the **lm()** function to fit the regression line. We define the model formula as **score ~ hours**, indicating that we are modeling the exam score based on the hours studied. The **lm() function** automatically employs the iterative minimization process inherent to the method of least squares to find the optimal intercept and slope values that minimize the error.

Once the model object is created, we use the **summary()** function to extract the crucial statistical output. This summary provides far more than just the coefficient estimates; it includes diagnostic information about the model fit, the standard errors associated with the estimates, t-values, p-values, and overall metrics like R-squared.

**#use method of least squares to fit regression line**
**model <- lm(score ~ hours, data=df)**

#view regression model summary
summary(model)

Call:
lm(formula = score ~ hours, data = df)

Residuals:
Min 1Q Median 3Q Max
-5.140 -3.219 -1.193 2.816 5.772

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 65.334 2.106 31.023 1.41e-13 ***
hours 1.982 0.248 7.995 2.25e-06 ***

---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.641 on 13 degrees of freedom
Multiple R-squared: 0.831, Adjusted R-squared: 0.818
F-statistic: 63.91 on 1 and 13 DF, p-value: 2.253e-06

The output confirms that the model fitting was successful. Key aspects to scrutinize include the **Residuals** section, which provides insight into the distribution of the errors, and the **Multiple R-squared** value (0.831), which suggests that approximately 83.1% of the variation in exam scores can be explained by the variation in hours studied, demonstrating a strong predictive relationship. The small p-values associated with both the Intercept and the Hours coefficient confirm that these parameters are statistically significant contributors to the model.

## Interpreting the Estimated Coefficients

The most important results for interpretation are found within the **Coefficients** table under the **Estimate** column. These values represent the parameters of our newly fitted linear equation. By extracting these estimates, we can formally construct the equation of the fitted regression line:

Exam Score = 65.334 + 1.982 × (Hours Studied)

A clear interpretation of each coefficient is vital for drawing meaningful conclusions from the statistical analysis. These coefficients quantify the exact nature of the relationship between the predictor and the response variable:

**Intercept (65.334)**: This is the estimated expected value of the exam score when the predictor variable, hours studied, is equal to zero. In this context, it suggests that a student who studies **hours** is expected to receive an exam score of **65.334**. This serves as the baseline prediction of the model.
**Hours Coefficient (1.982)**: This value represents the slope of the regression line. It indicates the change in the predicted exam score for every one-unit increase in the hours studied. Specifically, for each additional hour dedicated to studying, the expected exam score increases by **1.982** points. This positive coefficient confirms a beneficial linear relationship between study time and performance.

Using this derived equation, we gain a predictive tool. For instance, we can estimate the expected exam score for a hypothetical student who studies for five hours. We substitute the value into the equation:

Exam Score = 65.334 + 1.982 × (5) = 75.244

Thus, a student studying for 5 hours is predicted to achieve a score of approximately 75.244.

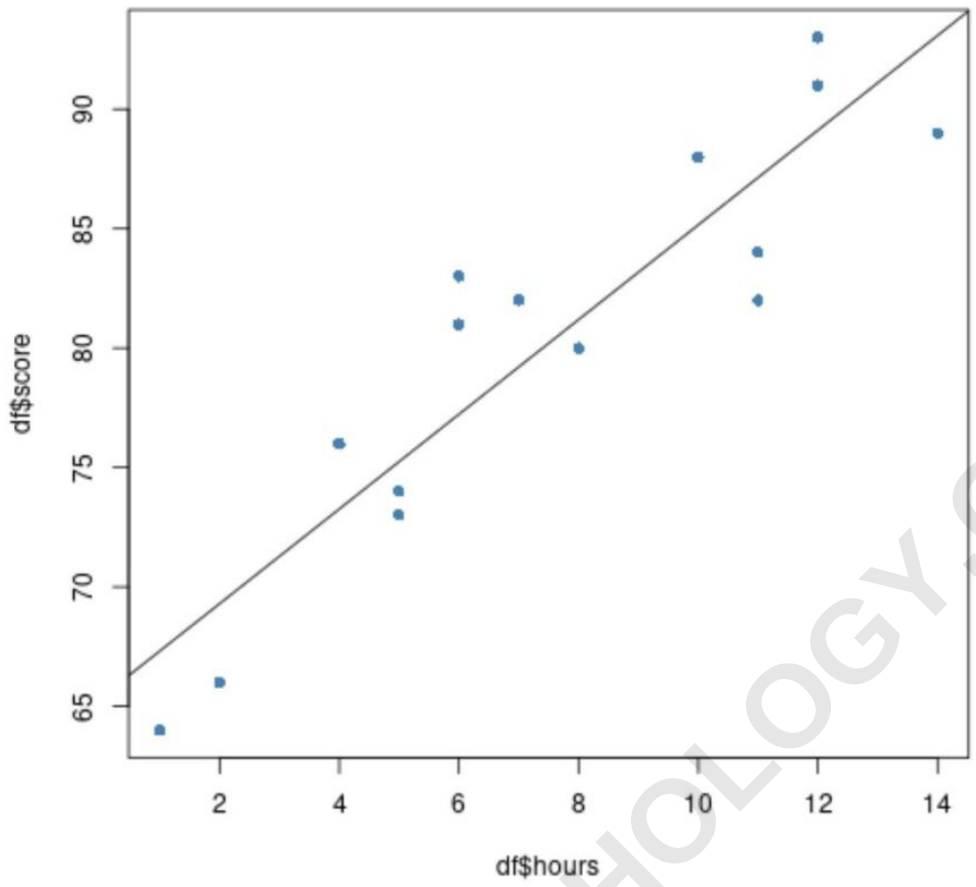
## Visualizing the Fitted Regression Line

The final and often most informative step in regression analysis is the visualization of the results. Creating a scatter plot allows us to view the original data points alongside the least squares regression line, providing an immediate graphical assessment of how well the model aligns with the observed data. This step helps confirm the assumption of linearity and illustrates the predictive power visually.

In R, we use the **plot()** function to generate the initial scatter plot of the data points and then utilize the **abline()** function, feeding it the fitted **model** object, to automatically draw the calculated regression line onto the plot.

```
#create scatter plot of data
plot(df$hours, df$score, pch=16, col='steelblue')

#add fitted regression line to scatter plot
abline(model)
```






In this resulting visualization, the steel blue circles represent the observed data points for the 15 students, while the solid black line represents the fitted regression line calculated using the lm() function. The visual proximity of the data points to the line confirms the strong fit indicated by the high R-squared value, demonstrating the efficacy of the Least Squares method in summarizing this relationship.

This systematic process--from data definition and model fitting using **lm()**, to coefficient interpretation using **summary()**, and finally visual validation--epitomizes the standard workflow for robust linear modeling in R.