How to Easily Perform OLS Regression in R with a Simple Example

Authored by

stats writer

November 27, 2025

Ordinary least squares (OLS) regression is a foundational statistical technique employed across various fields to model the relationship between a dependent variable and one or more independent variables. Specifically, OLS works by minimizing the sum of the squared differences between the observed and predicted responses, thereby determining the optimal parameter values that allow a linear equation to best fit the observed data points. This method is indispensable for making predictions and understanding correlational strength.

When working in the statistical programming environment, R, the key function for performing OLS regression is `lm()` (for linear model). This versatile function takes a formula and a dataset, calculates the necessary statistics, and outputs a model object containing critical details such as estimated coefficients, standard errors, t-values, and p-values.

For instance, to model the relationship between miles per gallon (mpg) and the weight (wt) and number of cylinders (cyl) using R's built-in `mtcars` dataset, one would execute the command: `lm(mpg ~ wt + cyl, data=mtcars)`. Understanding the output of this function is essential for interpreting the predictive power and statistical significance of the variables in the model.

Understanding the OLS Model Equation

At its core, OLS regression aims to find the line of best fit. For simple linear regression--involving only one predictor variable--this line is mathematically represented by a specific linear equation:

? = b0 + b1x

Understanding the components of this formula is critical for model interpretation:

?: This represents the estimated response value (the predicted value of the dependent variable).
b0: Known as the intercept, this is the expected value of the response variable when the predictor variable (x) is zero.
b1: This is the slope coefficient, quantifying the change in the response variable for every one-unit increase in the predictor variable.

By establishing this equation, we gain powerful insights into the association between the predictor and response variables. Not only does this equation describe the observed relationship, but it also provides a framework for predicting the outcome (?) based on new, unobserved values of the predictor (x). The subsequent steps provide a detailed walkthrough of applying this methodology in R.

Step 1: Preparing and Creating the Dataset in R

To demonstrate the application of OLS regression, we will begin by constructing a synthetic dataset within the R environment. This dataset will simulate the performance of 15 fictional

students, tracking two key variables critical for our analysis: the total hours studied and the resulting exam score.

In this simple linear model, the variable we wish to predict--the **Exam score**--will serve as our response (dependent) variable. Conversely, the **Total hours studied** will be designated as the predictor (independent) variable. Our goal is to determine if and how strongly study hours influence test performance and to quantify that relationship.

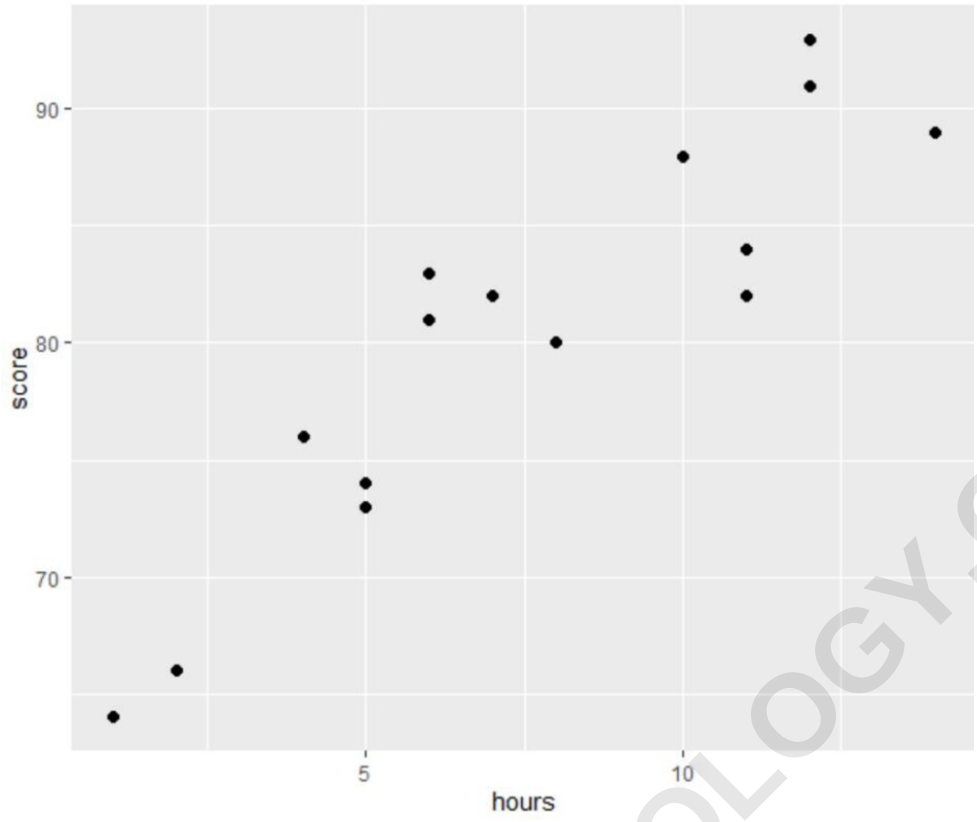
The following R code chunk illustrates how to use the `data.frame()` function to generate and structure this sample data, followed by a quick check using `head()` to ensure the data frame, named `df`, was created correctly:

```
#create dataset
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 dataset
head(df)

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

Step 2: Visual Inspection and Outlier Detection

Before fitting any regression model, it is a crucial best practice to visualize the data. This preliminary step helps confirm the assumption of linearity--that is, whether the relationship between the predictor (hours) and the response (score) appears to follow a straight line. We utilize the powerful `ggplot2` package in R to generate a scatter plot:

```
library(ggplot2)

#create scatter plot
ggplot(df, aes(x=hours, y=score)) +
geom_point(size=2)
```



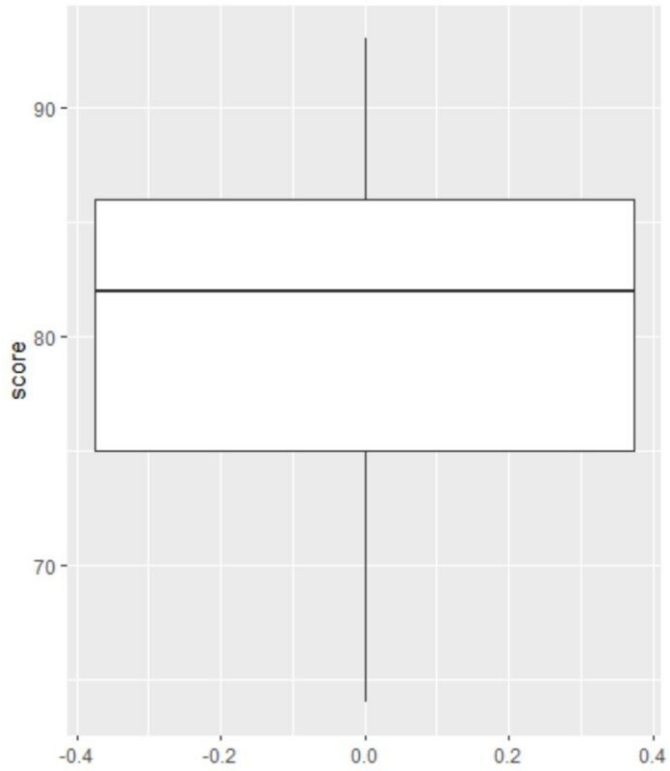
The scatter plot clearly illustrates a positive, linear trend: as the number of hours studied increases, the corresponding exam score tends to increase as well. Since the data points cluster around what would be a straight line, the linearity assumption required for OLS regression appears to be satisfied.

Another essential check involves examining the distribution of the response variable (score) for potential extreme values, or **outliers**. Outliers can significantly skew regression coefficients and inflate error estimates. We generate a boxplot using `ggplot2` on the score variable to visually identify any such points.

Note: In statistical visualization, R often identifies an observation as an outlier if it falls outside the range of 1.5 times the Interquartile Range (IQR) above the third quartile (Q3) or below the first quartile (Q1). If an observation meets this criteria, it is typically marked by a small, isolated circle on the boxplot.

library(ggplot2)

#create scatter plot
ggplot(df, aes(y=score)) +
geom_boxplot()



Since the boxplot above does not display any small circles beyond the whiskers, we can confidently conclude that our dataset contains no statistical outliers, allowing us to proceed to model fitting without needing initial data transformations or removal of influential points.

Step 3: Executing the OLS Regression Model in R

With the data prepared and inspected, the next logical step is to fit the simple linear equation using the `lm()` function in R. We define the model, specifying `score` as the response variable and `hours` as the predictor variable. The output generated by the `summary()` function provides a comprehensive overview of the model fit and the statistical significance of the estimated coefficients.

#fit simple linear regression model
model <- lm(score~hours, data=df)

#view model summary
summary(model)

Call:
lm(formula = score ~ hours)

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
```

Based on the coefficient estimates from the summary output, we can construct the precise fitted regression equation that describes the relationship for our sample data: **Score = 65.334 + 1.982 * (Hours)**. This equation provides immediate and quantifiable insight. The intercept of **65.334** suggests that a student who studies for zero hours is expected to achieve an average exam score of 65.334 points. More importantly, the slope coefficient of **1.982** indicates that for every additional hour a student dedicates to studying, the expected exam score increases by an average of 1.982 points.

This regression equation is not just descriptive; it is also predictive. We can use it to estimate outcomes for new data points. For example, if a student studies for 10 hours, we can calculate the expected score: Score = 65.334 + 1.982 * (10) = **85.154**. This ability to forecast outcomes based on known predictor values is the primary utility of linear regression modeling.

Interpreting the Model Summary Statistics

The lower section of the R summary output contains critical statistical metrics necessary for assessing the model's validity and performance. A thorough interpretation involves understanding what each component represents:

Pr(>|t|) (P-value): This value assesses the statistical significance of each predictor variable. Since the p-value associated with the *hours* variable (2.25e-06) is far below the conventional significance level of 0.05, we reject the null hypothesis. This confirms a highly statistically significant association between the number of hours studied and the exam score.

Multiple R-squared: This coefficient of determination (0.831) indicates that **83.1%** of the total variation observed in the exam scores can be successfully explained by the number of hours studied. Generally, a higher R-squared value suggests that the predictor variable provides a better

fit for explaining the response variable variation.

Residual Standard Error (RSE): The RSE (3.641) quantifies the typical distance that the actual observed exam scores fall from the regression line. It is measured in the units of the response variable. A lower RSE implies a tighter fit, meaning the model's predictions are, on average, within 3.641 points of the true score.

F-statistic & Overall P-value: The F-statistic (63.91) and its corresponding p-value (2.253e-06) test the overall significance of the entire regression model. Since this p-value is extremely small (less than 0.05), we conclude that the model as a whole is statistically significant, confirming that *hours* is a useful predictor for explaining the variation in *score*.

Step 4: Validating Model Assumptions with Residual Analysis

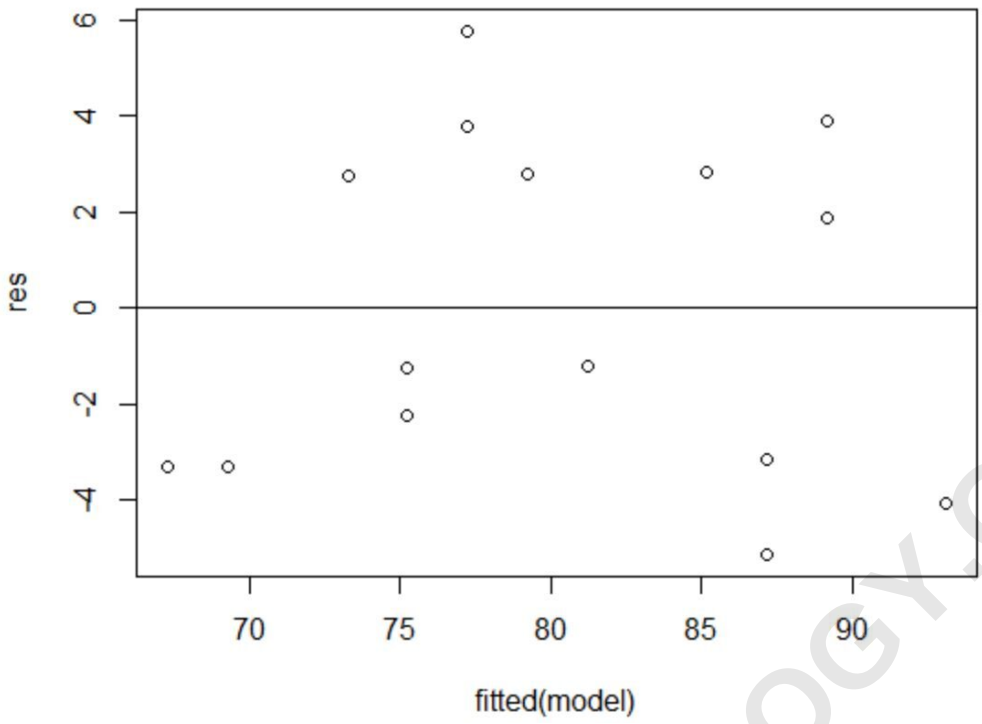
A key requirement for relying on the output of an OLS regression model is verifying that the underlying statistical assumptions regarding the model residuals have been met. The two most important assumptions to check are **homoscedasticity** (constant variance) and **normality** (normal distribution of errors). Failure to meet these assumptions can lead to unreliable standard errors and invalid statistical inferences.

Checking for Homoscedasticity (Constant Variance)

The assumption of Homoscedasticity requires that the variance of the residuals--the differences between the observed and predicted values--remains approximately equal across all levels of the predictor variables. We assess this assumption visually using a **Residuals vs. Fitted Plot**.

In this plot, the x-axis represents the fitted (predicted) values, and the y-axis displays the calculated residuals. If the assumption is met, the residuals should appear randomly scattered, forming a horizontal band centered around the zero line, showing no clear fan shape or funnel pattern.

```
#define residuals
res <- resid(model)

#produce residual vs. fitted plot
plot(fitted(model), res)

#add a horizontal line at 0
abline(0,0)
```



As observed in the plot, the data points are scattered randomly and evenly around the zero line, indicating no systematic pattern in the error variance. We can confidently conclude that the assumption of homoscedasticity is satisfied for this model.

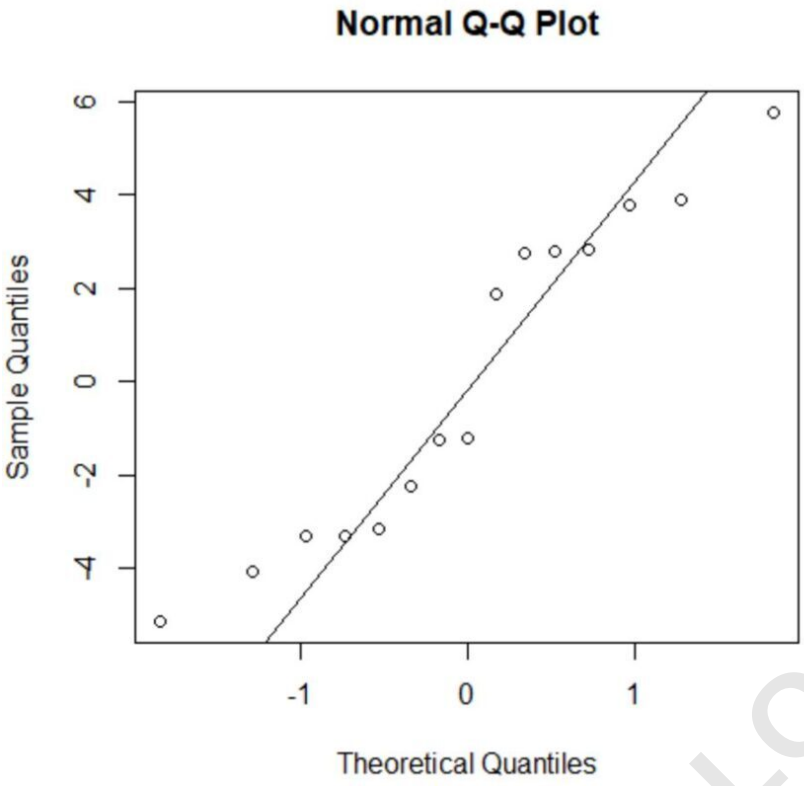
Checking for Normality of Residuals

The second major assumption is Normality, which mandates that the residuals of the regression model must be approximately normally distributed. We verify this using a **Quantile-Quantile (Q-Q) Plot**.

A Q-Q plot compares the quantiles of the residuals to the quantiles of a theoretical normal distribution. If the residuals are normally distributed, the points should closely follow the straight 45-degree diagonal line. Deviations, especially at the extremes (tails), suggest a violation of normality.

#create Q-Q plot for residuals
qqnorm(res)

#add a straight diagonal line to the plot
qqline(res)



While the points exhibit minor deviations at the very edges, they generally align well with the reference line. These small deviations are typical in real-world data and are not severe enough to invalidate the model. We can reasonably assume that the normality assumption is met.

Conclusion on Model Validity

Since both the normality and homoscedasticity assumptions for the regression linear equation have been verified through residual analysis, we confirm that our OLS model is statistically robust and the statistical inferences drawn from the model summary (Step 3) are reliable for this dataset.

Note: Should residual analysis reveal serious violations of these core assumptions, methods such as data transformation (e.g., log transformation) or employing alternative regression techniques (e.g., generalized linear models) would be necessary to achieve a reliable result.