Learning Linear Regression with the lm() Function in R

Authored by

Mohammed looti

November 3, 2025

The **lm() function** in R is the foundational tool used by analysts and statisticians to fit **linear regression models**. Understanding how to utilize this function effectively is crucial for modeling relationships between variables, predicting outcomes, and interpreting statistical significance across diverse fields, including finance, biology, and social sciences. This guide provides a comprehensive, step-by-step walkthrough of fitting, summarizing, plotting, and utilizing linear models in R.

The **lm()** function follows a standard structure that defines the relationship being tested and the source of the data. Mastering this syntax is the first step toward conducting robust statistical analysis.

The function uses the following basic syntax:

lm(formula, data, ...)

Here is a detailed breakdown of the required arguments:

formula: This argument specifies the relationship between the dependent (response) variable and the independent (predictor) variables. It is written using R's standard formula notation, such as `y ~ x1 + x2`, where `y` is the response and `x1` and `x2` are the predictors.
data: This specifies the data frame object that contains the variables defined in the formula. Providing the data argument ensures that R correctly locates all variables used in the model specification.

The following detailed example demonstrates the entire workflow of working with linear models in R, from initial model fitting to making practical predictions. We will cover five essential steps:

Fitting the initial regression model using **lm()**.
Viewing the detailed statistical summary of the model fit using **summary()**.
Analyzing the assumptions of the model through **diagnostic plots**.
Visualizing the fitted regression line against the raw data.
Generating predictions for new observations.

Fit the Linear Regression Model

To begin, we must define a dataset that contains both the predictor and response variables. We will create a simple data frame called `df` to illustrate the relationship between a single predictor variable, `x`, and a response variable, `y`. This example serves as the foundation for our simple linear regression analysis.

The subsequent code shows how to use the **lm()** function to fit a simple linear regression model in R. We specify the formula `y ~ x`, which posits that `y` is linearly dependent on `x`, and instruct the

function to draw the data from the `df` object.

 # Define the example data frame
df = data.frame(x=c(1, 3, 3, 4, 5, 5, 6, 8, 9, 12),
y=c(12, 14, 14, 13, 17, 19, 22, 26, 24, 22))

# Fit the simple linear regression model using 'x' as predictor and 'y' as response variable
model <- lm(y ~ x, data=df)

Upon execution, the variable `model` now stores all the necessary information about the fitted regression, including the calculated coefficients, residuals, and model statistics. This object is the key resource for all subsequent analyses.

View the Summary of the Regression Model

After fitting the model, the next crucial step is examining the statistical output to assess the model's performance and the significance of the individual predictors. We achieve this using the built-in **summary()** function, which processes the `model` object and returns a comprehensive report detailing key statistical metrics.

Interpreting the output of the **summary()** function allows us to rigorously evaluate the statistical validity and practical utility of our fitted linear model. Key sections to focus on include the Residuals, Coefficients, and Model Fit Statistics, as shown below:

 # View detailed summary of the regression model
summary(model)

Call:
lm(formula = y ~ x, data = df)

Residuals:
Min 1Q Median 3Q Max
-4.4793 -0.9772 -0.4772 1.4388 4.6328

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.1432 1.9104 5.833 0.00039 ***
x 1.2780 0.2984 4.284 0.00267 **

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

Residual standard error: 2.929 on 8 degrees of freedom

Multiple R-squared: 0.6964, Adjusted R-squared: 0.6584
F-statistic: 18.35 on 1 and 8 DF, p-value: 0.002675

The statistical output provides several critical pieces of information necessary for drawing conclusions about the relationship between `x` and `y`:

F-statistic and P-value: The overall model fit is tested using the **F-statistic** (18.35), which has a corresponding **p-value** of .002675. Since this p-value is significantly less than the conventional significance level of .05, we conclude that the model as a whole is **statistically significant**, meaning that `x` explains a non-random portion of the variance in `y`.
Multiple R-squared: The value of **Multiple R-squared** (.6964) indicates the proportion of the total variance in the response variable (y) that is explained by the predictor variable (x). In this case, 69.64% of the variation in `y` is accounted for by the linear relationship with `x`.
Coefficient Estimate of x: The coefficient estimate for `x` is 1.2780. This is the slope of the regression line. It tells us that, on average, a one-unit increase in the predictor variable `x` is associated with an increase of 1.2780 units in the response variable `y`, assuming all other factors remain constant. The intercept (11.1432) represents the predicted value of `y` when `x` is zero.

Using the coefficient estimates for the intercept and the predictor, we can formally write the estimated regression equation, which mathematically describes the best-fit line through our data:

y = 11.1432 + 1.2780 * (x)

View Diagnostic Plots of the Model

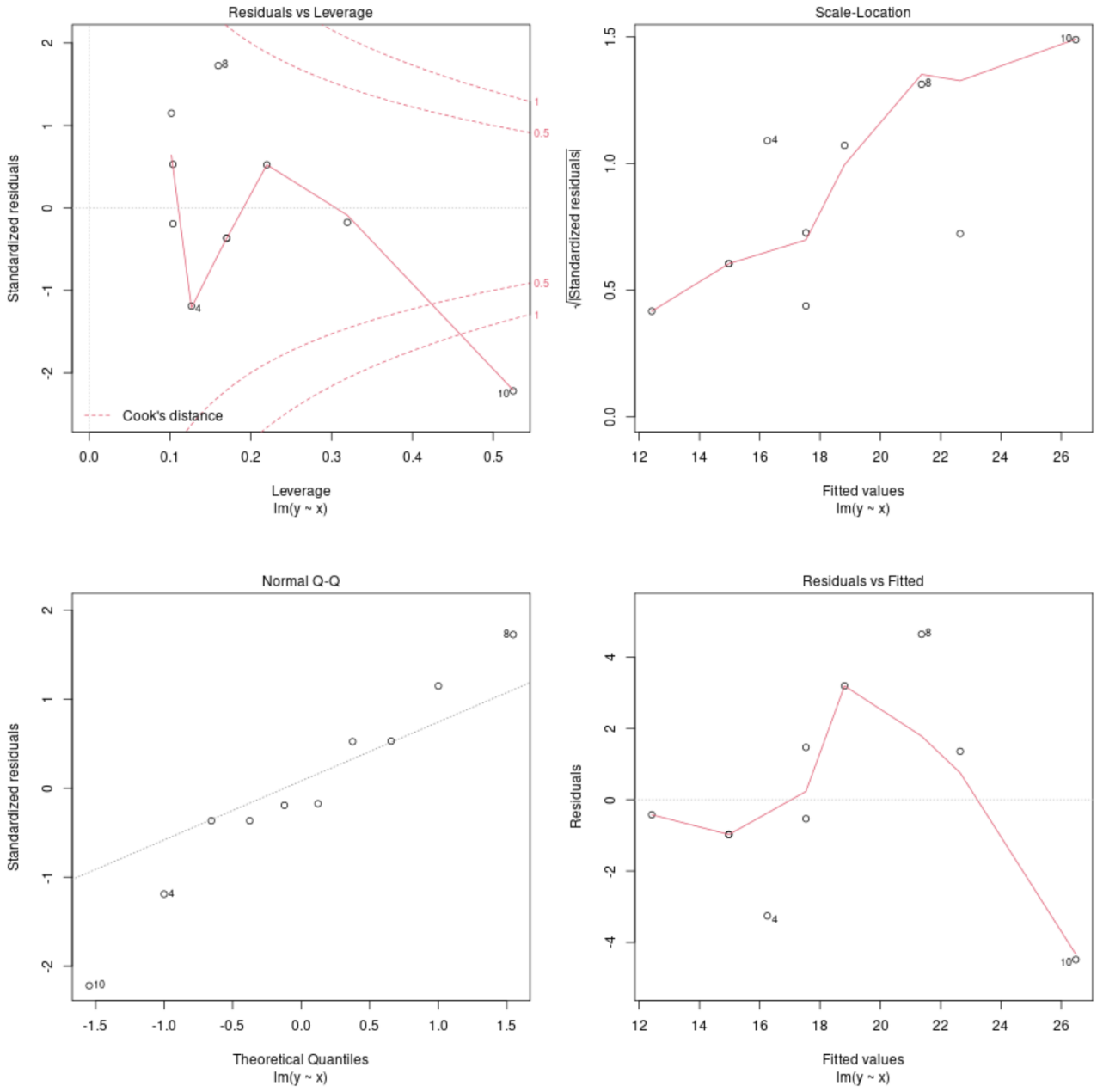
While the summary statistics confirm the significance and explanatory power of the model, they do not tell us whether the fundamental assumptions of linear regression have been met. Violating these assumptions--such as linearity, independence of errors, normality of residuals, and homoscedasticity (constant variance)--can invalidate the statistical inferences drawn from the model. Therefore, generating and examining **diagnostic plots** is a critical step.

The **plot()** function, when applied directly to the `model` object, automatically generates four standard diagnostic plots. These visualizations provide essential graphical checks on the model's fit and the behavior of the residuals.

```
# Create standard diagnostic plots
plot(model)
```

These plots allow us to analyze the model's structural integrity to determine if the linear model is appropriate for the data and if the assumptions are reasonably satisfied. For example, the

Residuals vs. Fitted plot helps check for linearity and homoscedasticity, while the Normal Q-Q plot assesses the normality of the residuals.



Thorough interpretation of these graphical diagnostics is essential before finalizing any analysis. If the plots reveal severe violations, the analyst may need to consider data transformations or alternative modeling techniques.

Plot the Fitted Regression Model

After confirming the statistical significance and checking the model assumptions, visualizing the

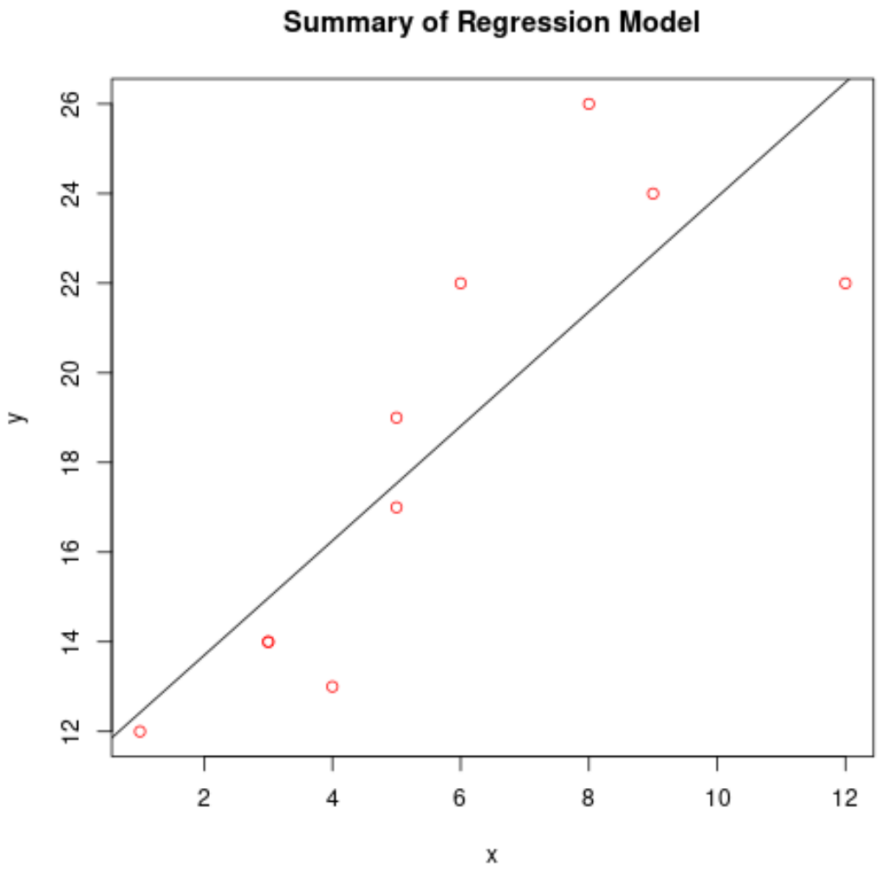
regression line provides an intuitive understanding of the relationship captured by the model. This step involves creating a scatterplot of the raw data points and then overlaying the calculated regression line defined by the **lm()** output.

We first use the standard **plot()** function to create a scatterplot of our raw data (`df$x` vs. `df$y`). Then, the **abline()** function is used to add the fitted line. When passed the `model` object, **abline()** automatically extracts the intercept and slope coefficients to draw the mathematically derived best-fit line directly onto the existing plot.

```
# Create scatterplot of raw data
plot(df$x, df$y, col='red', main='Summary of Regression Model', xlab='x', ylab='y')

# Add the fitted regression line derived from the lm() model
abline(model)
```

This visualization confirms that the regression line minimizes the distance between the line and the observed data points, offering a clear graphical representation of the linear relationship described by the equation y = 11.1432 + 1.2780*(x).



Use the Regression Model to Make Predictions

The ultimate goal of many modeling efforts is prediction. Once a linear model is fitted and validated, it can be used to forecast the response variable (y) for new, unseen values of the predictor variable (x). This process is executed efficiently using the **predict()** function in R.

To use **predict()**, we must first define the new data point(s) we wish to forecast. Crucially, this new data must be supplied to the function in the form of a data frame, ensuring that the variable names match those used in the original model (in this case, x).

 # Define a new observation where x = 5
new <- data.frame(x=c(5))

# Use the fitted model to predict the value for the new observation
predict(model, newdata = new)

1
17.5332

The result shows that based on our fitted **linear regression model**, an observation with a predictor value of `x = 5` is predicted to have a response value of **17.5332**. This demonstrates the practical application of the **lm()** function in generating reliable forecasts based on established statistical relationships.

Additional Resources

To further deepen your understanding of linear modeling and R programming, consider exploring the official R documentation for the **lm()** function, as well as comprehensive tutorials on statistical inference and model diagnostics.