Learning Guide: Interpreting Regression Coefficients from R’s lm() Function

Authored by

Mohammed looti

October 30, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Learning Guide: Interpreting Regression Coefficients from R’s lm() Function*. PSYCHOLOGICAL STATISTICS. Retrieved from https://statistics.arabpsychology.com/?p=6116

Understanding Regression Coefficients in R

When performing linear regression in R, the primary tool is often the **`lm()` function**. This powerful function allows you to fit linear models to your data. A crucial part of interpreting any linear model involves understanding its regression coefficients. These coefficients represent the estimated change in the dependent variable for a one-unit change in the corresponding independent variable, holding all other predictors constant.

Effectively extracting these coefficients is fundamental for model interpretation, reporting, and building predictive equations. This guide will demonstrate two distinct methods to retrieve regression coefficients from an `lm()` object in R, catering to different levels of detail you might require for your analysis.

Method 1: Isolating Regression Coefficients

The simplest way to obtain just the numerical values of the regression coefficients is by directly accessing the `coefficients` component of your fitted model object. This method is ideal when you need to quickly retrieve the estimated values for the intercept and each predictor variable without additional statistical details.

To implement this, you can use the dollar sign (`$`) operator, which is a common way to extract specific components from R objects. Assuming your fitted linear model is stored in a variable named `model`, the syntax is straightforward:

model$coefficients

This command will return a named numeric vector containing the estimated coefficient for the intercept and each independent variable included in your linear regression model.

Method 2: Accessing Detailed Coefficient Statistics

While the first method provides the coefficient estimates, often you need more comprehensive information for statistical inference and model evaluation. This includes the standard error, t-statistic, and p-value associated with each coefficient. These statistics are vital for assessing the precision of the estimates and the statistical significance of each predictor.

To retrieve this detailed output, you should first apply the **`summary()` function** to your linear model object. The `summary()` function provides a comprehensive overview of the model fit. From this summary object, you can then extract the coefficients table, which contains all the aforementioned statistics. The syntax for this method is as follows:

summary(model)$coefficients

This command will yield a matrix where each row corresponds to a coefficient (including the intercept), and columns provide the estimate, its standard error, the t-value, and the associated p-value.

Practical Example: A Multiple Linear Regression Model

Let's illustrate these methods with a practical example. We will create a sample dataset and fit a multiple linear regression model in R. Our goal is to predict an individual's "rating" based on their "points", "assists", and "rebounds" in a hypothetical scenario. This example will clearly demonstrate how to apply the commands discussed above.

First, we define a data frame named `df` containing our variables. Then, we use the **`lm()` function** to fit a model where `rating` is the dependent variable, and `points`, `assists`, and `rebounds` are the independent predictor variables. The syntax for this process is shown below:

#create data frame
df <- data.frame(rating=c(67, 75, 79, 85, 90, 96, 97),
points=c(8, 12, 16, 15, 22, 28, 24),
assists=c(4, 6, 6, 5, 3, 8, 7),
rebounds=c(1, 4, 3, 3, 2, 6, 7))

#fit multiple linear regression model
model <- lm(rating ~ points + assists + rebounds, data=df)

This code successfully creates our sample dataset and fits the desired multiple linear regression model, storing the results in the `model` object.

Interpreting the Full Model Summary

Before diving into specific coefficient extraction, it's beneficial to view the entire summary of our fitted model using the **`summary()` function**. This provides a holistic view of the model's performance and the statistical significance of its components. Understanding the full summary context helps in interpreting the individual coefficients more accurately.

Executing `summary(model)` will output a detailed report, including residuals, the coefficients table, R-squared values, and the F-statistic. This comprehensive output is crucial for a complete understanding of your linear regression analysis.

#view model summary

summary(model)

Call:
lm(formula = rating ~ points + assists + rebounds, data = df)

Residuals:
1 2 3 4 5 6 7
-1.5902 -1.7181 0.2413 4.8597 -1.0201 -0.6082 -0.1644

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 66.4355 6.6932 9.926 0.00218 **
points 1.2152 0.2788 4.359 0.02232 *
assists -2.5968 1.6263 -1.597 0.20860
rebounds 2.8202 1.6118 1.750 0.17847

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

Residual standard error: 3.193 on 3 degrees of freedom
Multiple R-squared: 0.9589, Adjusted R-squared: 0.9179
F-statistic: 23.35 on 3 and 3 DF, p-value: 0.01396

From this summary, we can observe the estimates for each coefficient, alongside their statistical significance indicators. This is the foundation upon which we can perform more targeted extractions.

Extracting and Applying Coefficients

As demonstrated in Method 1, retrieving only the regression coefficients is straightforward using `model$coefficients`. This is particularly useful when you intend to construct the fitted regression equation or use the coefficients for further calculations programmatically.

Applying this to our example model, we can see the intercept and the coefficients for `points`, `assists`, and `rebounds`:

#view only regression coefficients of model
model$coefficients

(Intercept) points assists rebounds
66.435519 1.215203 -2.596789 2.820224

Using these extracted values, we can formally write out the fitted regression equation for predicting `rating`:

Rating = 66.43551 + 1.21520(points) - 2.59678(assists) + 2.82022(rebounds)

This equation indicates, for example, that for every one-unit increase in `points`, the predicted `rating` increases by approximately 1.215 units, assuming `assists` and `rebounds` remain constant. Conversely, an increase in `assists` is associated with a decrease in rating.

Advanced Extraction: Specific Statistics

Beyond simply listing the coefficients, you will often need to access the associated statistical metrics like standard errors, t-values, and especially p-values, to evaluate the significance of each predictor. As discussed in Method 2, `summary(model)$coefficients` provides this detailed matrix.

#view regression coefficients with standard errors, t-statistics, and p-values
summary(model)$coefficients

Estimate Std. Error t value Pr(>|t|)
(Intercept) 66.435519 6.6931808 9.925852 0.002175313
points 1.215203 0.2787838 4.358942 0.022315418
assists -2.596789 1.6262899 -1.596757 0.208600183
rebounds 2.820224 1.6117911 1.749745 0.178471275

This matrix is highly flexible, allowing you to extract specific values programmatically. For instance, to isolate the p-value for a particular variable, such as `points`, you can use matrix indexing:

#view p-value for points variable
summary(model)$coefficients

0.02231542

Alternatively, if you need to access all the p-values for every regression coefficient in the model, you can specify the column name without a row name. This will return a vector of all p-values, which is useful for quick checks on significance across all predictors:

#view p-value for all variables
summary(model)$coefficients

(Intercept) points assists rebounds

0.002175313 0.022315418 0.208600183 0.178471275

These p-values are crucial for determining which predictors have a statistically significant relationship with the response variable, given the other predictors in the model. You can use similar indexing syntax to access any other value within the coefficients matrix, such as the estimates or standard errors for specific variables.

Conclusion and Further Learning

Mastering the extraction of regression coefficients from an `lm()` object in R is an essential skill for anyone working with statistical models. Whether you need just the coefficient estimates for building a regression equation or a detailed table including standard errors, t-statistics, and p-values for rigorous statistical inference, R provides straightforward methods to obtain this information.

By understanding and utilizing `model$coefficients` and `summary(model)$coefficients`, you can efficiently analyze and interpret your linear models, making informed decisions based on your data. Remember that the ability to precisely extract these values enhances your capability to perform advanced data analysis and reporting.

Additional Resources

To deepen your understanding of linear regression and advanced data manipulation in R, consider exploring the following tutorials and documentation: