

Learning R: How to Calculate and Interpret R-Squared in Linear Regression Models

Authored by
Mohammed Iooti

October 27, 2025

RECOMMENDED CITATION

Mohammed Iooti (2025). *Learning R: How to Calculate and Interpret R-Squared in Linear Regression Models*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=4289>

The Importance of R-squared and Adjusted R-squared in Statistical Modeling

When conducting [linear regression](#) analysis in [R](#), two indispensable metrics for assessing model quality are the **R-squared** and **Adjusted R-squared** values. These statistics serve as crucial indicators of how effectively a statistical model captures and explains the variability inherent in the observed data. The **R-squared**, frequently referred to as the [coefficient of determination](#), provides a quantifiable measure of the proportion of the variance in the dependent variable that can be systematically predicted from the independent variables included in the model. In essence, a higher **R-squared** value suggests that a larger percentage of the outcome's variation is explained by the chosen predictors.

Despite its fundamental utility, the raw **R-squared** metric presents a known analytical challenge: it possesses an inherent tendency to increase monotonically as additional predictors are incorporated into the model. This increase occurs regardless of whether those new predictors genuinely enhance the model's explanatory power or are statistically significant. This characteristic can potentially lead to an inflated or overly optimistic evaluation of model fit, particularly when comparing models that differ in their complexity or the sheer number of independent variables utilized. This limitation necessitates a more refined measurement.

To address this inherent bias, the [Adjusted R-squared](#) statistic was developed. This sophisticated metric modifies the traditional **R-squared** by incorporating a penalty based on the number of predictors used and the size of the dataset (degrees of freedom). By penalizing the inclusion of extraneous or poorly performing variables, the **Adjusted R-squared** provides a more conservative and reliable measure for comparing the performance of competing models with varying sets of independent variables. For robust statistical reporting and analysis, precise extraction of these metrics is often mandatory.

The [lm\(\) function](#) in R is the foundational tool for fitting linear models, and its statistical output, which is fully accessible via the [summary\(\) function](#), is the source for these essential R-squared values. This guide focuses on the technical process required to programmatically isolate and extract both the **R-squared** and **Adjusted R-squared** values from a fitted `lm()` model object within an R environment.

Accessing Model Fit Statistics Programmatically in R

To efficiently retrieve the **R-squared** and **Adjusted R-squared** metrics from a linear model fitted using [lm\(\) model](#) in R, we must interact directly with the resulting summary object. Once a linear model has been successfully fitted and stored in an object--conventionally named `model`--the immediate subsequent step is to execute the [summary\(\) function](#) on this object. This operation does not merely print output to the console; rather, it generates a comprehensive list object that

encapsulates all the statistical findings, including the coefficients, residuals, and, crucially, the R-squared metrics.

The structure of this summary object is standardized, allowing direct access to specific components. Specifically, the raw **R-squared** value is stored under the list component named `r.squared`, while the **Adjusted R-squared** value resides within the component labeled `adj.r.squared`. The ability to access these elements individually and precisely is enabled by R's powerful list subsetting mechanism: the `$` operator. This operator allows data scientists and analysts to extract named elements from list structures, bypassing the need to parse the full text output.

This programmatic approach is essential for automating reporting workflows, integrating results into larger data pipelines, or applying conditional logic based on model performance thresholds. Below is the exact, concise syntax required to extract these two critical model fit statistics directly from the summary object:

```
# Extract R-squared value  
summary(model)$r.squared
```

```
# Extract Adjusted R-squared value  
summary(model)$adj.r.squared
```

The forthcoming practical demonstration will integrate this syntax into a complete R workflow, starting from the preparation of the input data and culminating in the highly targeted, programmatic extraction of these specific metrics. This ensures a clear understanding of the necessary steps and contextual application within a typical statistical analysis setting.

Demonstration: Setting Up and Fitting a Multiple Linear Regression

To practically illustrate the extraction technique for R-squared values, we will construct and fit a [multiple linear regression](#) model. For this scenario, we hypothesize that a quantitative 'rating' score is influenced by a combination of three independent variables: 'points', 'assists', and 'rebounds'. This framework is commonly used when analyzing phenomena where a single continuous outcome variable is simultaneously determined by the interplay of two or more predictor variables, whether they are continuous or categorical in nature.

The initial requirement is to define a sample [data frame](#) in R, which will function as our experimental dataset. This structure simulates real-world observational data, providing the necessary variables for our regression analysis. For pedagogical clarity and ease of verification, we utilize a small, well-defined collection of simulated records. The creation of this structured input is a prerequisite for any modeling exercise in R, ensuring that the `lm()` function can correctly

identify and process the predictor and response variables.

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)
```

Following the successful creation of our [data frame](#), we proceed to fit the [multiple linear regression](#) model using R's powerful [lm\(\) function](#). The formula syntax, `rating ~ points + assists + rebounds`, explicitly defines 'rating' as the outcome variable that we aim to predict, based on the linear combination of 'points', 'assists', and 'rebounds' as predictors. The inclusion of the `data=df` argument instructs the function to search for all specified variables exclusively within our defined data frame, `df`. The resulting object, assigned to the variable `model`, now holds all the fitted parameters, statistical residuals, and underlying infrastructure necessary for detailed model evaluation.

Detailed Examination of the Regression Summary Output

To gain a comprehensive diagnostic perspective on our fitted [regression model](#), standard statistical methodology dictates the use of the [summary\(\) function](#). This function provides an elaborate statistical report that encompasses several critical components. This includes the estimated parameters (coefficients) for each predictor, measures of uncertainty (standard errors), associated t-values, and the critical [p-values](#) used to determine individual predictor significance. Most relevant to this discussion, the summary prominently displays the overall model fit statistics, namely the [R-squared](#) and [Adjusted R-squared](#) values.

Executing the command `summary(model)` generates the full statistical output for our regression model, allowing us to inspect the performance metrics visually. This detailed output not only verifies the coefficient estimates but also provides the necessary context for judging the model's overall efficacy, stability, and predictive strength relative to the sample data.

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

Within this comprehensive summary, analysts should focus on the metrics located toward the bottom. These figures quantify the model's global explanatory power. Of particular note are the [F-statistic](#) and its corresponding [p-value](#), which together test the null hypothesis that all regression coefficients are zero--effectively assessing the overall statistical significance of the entire model. Furthermore, the [Residual standard error](#), reported alongside its [degrees of freedom](#), offers a measure of the typical size of the error, or the average difference between the observed and predicted values.

Interpreting the Model Fit Statistics

The primary focus for evaluating model fit lies in the R-squared values presented in the summary output. These statistics provide a clear, standardized measure of how well the combination of predictors ('points', 'assists', 'rebounds') accounts for the variation observed in the dependent variable ('rating'). The values obtained from our specific example model are highlighted as follows:

R-squared: 0.9589

Adjusted R-squared: 0.9179

An observed **R-squared** value of 0.9589 is interpreted to mean that approximately 95.89% of the total variability in the 'rating' score is successfully explained by the combined effect of the three independent variables included in our model. This is generally considered an exceptionally strong fit. However, the [Adjusted R-squared](#) provides a slightly lower, more cautious estimate of 0.9179. This difference is attributable to the adjustment mechanism, which factors in the small sample size

and the inclusion of multiple predictors, thereby offering a more realistic assessment of predictive utility when the model is generalized to new data.

The presence of a high **Adjusted R-squared** value is particularly reassuring, as it suggests that the strong fit is not merely a consequence of having many variables, but rather a reflection of the genuine explanatory power contributed by the chosen predictors. When comparing this model to a potential alternative model with a different set of predictors, the **Adjusted R-squared** would be the preferred metric for making an accurate, objective comparison. Understanding the nuances between these two metrics is crucial for sound statistical inference and reliable model selection.

Programmatic Extraction for Automated Workflows

While visual inspection of the full model summary is vital for initial diagnostics and verification, many data analysis tasks require the precise numerical value of the **R-squared** or **Adjusted R-squared** for automated processing. Such requirements arise in scenarios like implementing model selection criteria within custom R scripts, performing cross-validation analyses, or generating highly customized reports where only the final statistic needs to be displayed. R facilitates this process through direct access to the underlying list components of the summary object.

To extract solely the numerical value of the **R-squared** statistic, thereby isolating it from the extensive textual output, the analyst can directly target the `r.squared` component of the `summary(model)` object. This is achieved using the standard R subsetting operator, the dollar sign (`$`), which is perfectly suited for accessing named elements within the list structure returned by the `summary` function.

```
# Extract R-squared value of regression model  
summary(model)$r.squared
```

```
0.9589274
```

Conversely, should the objective be to obtain the slightly more rigorous **Adjusted R-squared** value--which accounts for model complexity--the extraction mechanism remains identical, but targets the distinct `adj.r.squared` component of the summary object. This consistent and straightforward approach ensures that precise model performance metrics can be retrieved efficiently for downstream computational use.

```
# Extract adjusted R-squared value of regression model  
summary(model)$adj.r.squared
```

```
0.9178548
```

Crucially, these programmatically extracted numerical values for **R-squared** and **Adjusted R-squared** match the figures reported in the comprehensive regression output summary examined previously. This robust consistency underscores the utility of direct extraction, enabling more streamlined, reproducible, and automated data analysis workflows, which are cornerstones of modern statistical practice.

Conclusion and Further Resources

Mastering the efficient extraction of key model statistics, such as **R-squared** and **Adjusted R-squared**, from [lm\(\) models](#) is an essential skill set for anyone performing [linear regression](#) analysis in R. These metrics are fundamental to accurately quantifying model performance, assessing the true explanatory power of chosen predictors, and making empirically supported decisions regarding the refinement or selection of statistical models. By employing the succinct syntax, `summary(model)$r.squared` and `summary(model)$adj.r.squared`, analysts can programmatically isolate these vital values with speed and accuracy, thereby dramatically enhancing the efficiency of their analysis and reporting cycles.

This capability of direct extraction facilitates the seamless incorporation of model fit statistics into automated data processing scripts, custom analytical functions, or dynamic reporting platforms. Achieving this level of flexibility not only ensures the reproducibility of statistical analyses but also provides sophisticated control over how complex regression results are presented and interpreted across diverse applications.

For those seeking to deepen their understanding of R for statistical modeling and advanced data manipulation, the following resources provide additional insights into common tasks and more sophisticated techniques: