

Learn How to Calculate Adjusted R-Squared in R for Regression Analysis

Authored by
Mohammed looti

November 6, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Learn How to Calculate Adjusted R-Squared in R for Regression Analysis*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=11783>

The Core Concepts: R-Squared Versus Adjusted R-Squared

In the realm of statistical modeling, particularly when dealing with [linear regression](#), model evaluation is paramount. The primary metric for quantifying model fit is the [R-squared](#) (R^2), officially known as the coefficient of determination. This metric provides a crucial measure of the proportion of the variance in the [response variable](#) that is predictably explained by the ensemble of [predictor variables](#) included within the model structure. It essentially tells us how well the regression line approximates the actual data points.

The R-squared value exists within a strictly defined range, spanning from 0 to 1. A value approaching the lower limit of 0 suggests that the chosen predictor variables offer negligible, if any, statistical power in explaining the variability observed in the response variable. Conversely, an R-squared value nearing 1 indicates an extremely strong fit, implying that the model captures nearly all the inherent variance. While R-squared is foundational, it suffers from a significant statistical bias, making it an insufficient metric for comparing models with different numbers of predictors.

The fundamental flaw of standard **R-squared** is its inherent tendency to increase monotonically as additional predictor variables are introduced into the model. This increase occurs regardless of whether the new variable contributes meaningfully or statistically significantly to the model's explanatory power. This vulnerability often encourages a practice known as "overfitting," where complex models appear to perform excellently on the training data but fail drastically when applied to new, unseen data. To counteract this critical weakness and provide a more robust assessment of model quality, the metric known as [adjusted R-squared](#) was conceptualized and developed.

The Necessity of Adjustment: Accounting for Model Complexity

The **adjusted R-squared** modifies the standard R-squared calculation by systematically penalizing the inclusion of extraneous or unnecessary predictor variables. This modification ensures that adding a new predictor only results in an increase in the adjusted R-squared if the variable's contribution to the model's explanatory power is substantial enough to offset the penalty imposed for increasing the model's complexity. Consequently, the adjusted R-squared is considered a far more reliable indicator when comparing competing models that utilize different sets of input variables.

The adjustment mechanism leverages the concept of statistical [degrees of freedom](#), providing a calculation that is explicitly sensitive to both the sample size and the number of predictors. Unlike standard R-squared, the adjusted R-squared can potentially decrease if a newly introduced variable does not enhance the model sufficiently. This feature strongly promotes the principle of parsimony--favoring the simplest model that adequately explains the data--making it an indispensable tool in rigorous statistical analysis.

Mathematically, the **adjusted R-squared** is derived directly from the standard R-squared using the following well-established formula. This calculation vividly demonstrates how the sample size and the count of predictor variables are factored into the final result, providing a transparent measure of fit quality that accounts for the model's structural complexity:

Adjusted R² = 1 -

Understanding the specific components of this formula is critical for appreciating how the penalty is applied:

R²: Represents the standard coefficient of determination, or the Multiple R-squared, calculated for the model.

n: Denotes the total number of observations, or data points, available in the sample used for model fitting.

k: Represents the number of independent (or [predictor variables](#)) utilized in the current regression model.

Because **adjusted R-squared** penalizes unnecessary complexity by incorporating the number of predictors (k), it serves as a robust and highly valuable metric for comparing non-nested models and for performing automated feature selection processes, guiding researchers toward the most efficient and generalized model structure.

Preparing the Data and Model in R

This guide focuses on the practical execution of calculating and extracting the adjusted R² within the statistical programming environment [R](#). To illustrate these techniques, we will construct a multiple [linear regression model](#) using the readily available, built-in **mtcars** dataset, which is a common reference for teaching statistical methods in R. This dataset contains information on 32 automobiles.

Our objective is to model the car's horsepower (hp) based on four distinct characteristics, which will serve as our [predictor variables](#): miles per gallon (mpg), vehicle weight (wt), rear axle ratio (drat), and quarter-mile time (qsec). We use the native `lm()` function in **R** to define and fit this model.

```
model <- lm(hp ~ mpg + wt + drat + qsec, data=mtcars)
```

Upon successful execution of the command above, the regression results are encapsulated within the object named `model`. This object holds all the necessary information, including the coefficients, residuals, and the raw statistics required to derive the adjusted R-squared value. The following sections detail three distinct and reliable methods for accessing this critical fit metric within the **R** environment.

Method 1: Comprehensive Output Using summary()

The most straightforward and universally employed method for obtaining the adjusted R-squared in **R** involves invoking the `summary()` function on the fitted model object. This function is designed to return a highly detailed, human-readable statistical report. The output encompasses a wealth of diagnostic information, including coefficient estimates, associated standard errors, t-statistics, p-values, and, crucially, the overall goodness-of-fit metrics.

When the `summary()` function is applied to our `model1` object, the resulting output provides an immediate and transparent view of all key statistical findings. The advantage of this method is that it contextualizes the adjusted R-squared alongside the standard R-squared and other important diagnostic statistics like the F-statistic and the residual standard error, enabling a holistic assessment of the model's performance and significance.

To view the complete model summary, execute the following command:

summary(model)

Call:

```
lm(formula = hp ~ mpg + wt + drat + qsec, data = mtcars)
```

Residuals:

Min 1Q Median 3Q Max

-48.801 -16.007 -5.482 11.614 97.338

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 473.779 105.213 4.503 0.000116 ***

mpg -2.877 2.381 -1.209 0.237319

wt 26.037 13.514 1.927 0.064600 .

drat 4.819 15.952 0.302 0.764910

qsec -20.751 3.993 -5.197 1.79e-05 ***

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

Residual standard error: 32.25 on 27 degrees of freedom

Multiple R-squared: 0.8073, Adjusted R-squared: 0.7787

F-statistic: 28.27 on 4 and 27 DF, p-value: 2.647e-09

By reviewing the final lines of the statistical summary, we can directly compare the standard R-squared value, which is **0.8073**, with the [adjusted R-squared](#) value of **0.7787**. The slight reduction

in the adjusted value confirms the penalty applied for utilizing four predictors, indicating a more realistic estimate of the model's predictive capability on a generalized population.

Method 2: Programmatic Extraction for Automation

While Method 1 is excellent for manual inspection, advanced statistical workflows, automated reporting, and scripting often necessitate obtaining only the raw numerical value of the adjusted R-squared, without the extensive surrounding statistical tables. This programmatic approach is highly efficient and minimizes the processing overhead required for data extraction.

The `summary()` function in **R** actually returns a structured list object, which can be manipulated just like any other list in the programming language. Within this list, the adjusted R-squared value is stored under the specific element name `adj.r.squared`. By applying the summary function and immediately accessing this element using the dollar-sign notation (`$`), we can isolate and retrieve the precise numerical result.

This streamlined command is the preferred method for analysts integrating model performance metrics into larger automated data pipelines or custom functions that require clean, numeric input:

```
summary(model)$adj.r.squared
```

```
0.7787005
```

The output confirms the exact value (0.7787005) directly. This ability to precisely target and extract specific model statistics is a core feature of effective data manipulation within **R** programming.

Method 3: Defining a Custom Calculation Function

For users who prioritize complete transparency over built-in functionality, or for those who need to integrate specific custom adjustments, writing a function that explicitly implements the mathematical formula for adjusted R-squared is an excellent alternative. This method ensures that the calculation logic is entirely visible and customizable, offering maximum control over the statistical process.

The custom function defined below replicates the exact mathematical structure: $1 - \frac{1 - R^2}{n - k}$. To achieve this, we utilize two key **R** functions: `nobs()`, which accurately determines the total number of observations (n) from the model object, and `length(x$coefficients)`, which determines the total number of coefficients (including the intercept), allowing us to accurately derive the count of [predictor variables](#) (k). Note that since the formula requires 'k' (number of predictors), and `length(x$coefficients)` returns $k+1$ (predictors plus intercept), we must subtract 1 from the coefficient length to isolate 'k'.

```
# Define function to calculate adjusted R-squared based on the explicit formula
adj_r2 <- function(x) {
  return (1 - ((1-summary(x)$r.squared)*(nobs(x)-1)/(nobs(x)-length(x$coefficients)-1)))
}

# Use the custom function to calculate adjusted R-squared of the model
adj_r2(model)

0.7787005
numeric(0)
```

Executing this custom function on the `model` object yields the identical result, 0.7787005, verifying the internal consistency of **R**'s calculations. This demonstrates that whether relying on the built-in summary statistics or implementing the fundamental mathematical formula, the adjusted R-squared metric remains a reliable and consistent measure of model fit adjusted for complexity.

Furthering Your Expertise in R Regression Analysis

Mastering the calculation and interpretation of [adjusted R-squared](#) is a significant step toward becoming proficient in regression analysis. However, effective statistical modeling requires a broader understanding of various regression techniques and diagnostic tools. Analysts should strive to fully understand concepts such as multicollinearity, heteroscedasticity, and the practical implications of transforming variables to improve model assumptions.

To deepen your analytical skills and explore more advanced statistical modeling concepts using the power of **R**, consider reviewing the following specialized tutorials which cover the foundational implementations of linear and polynomial modeling:

[How to Perform Simple Linear Regression in R](#)

[How to Perform Multiple Linear Regression in R](#)

[How to Perform Polynomial Regression in R](#)