How to perform a Lack of Fit Test in R (Step-by-Step)

Authored by

stats writer

December 9, 2025

Performing a Lack of Fit Test in R is a crucial step in validating the suitability of a regression model. The process requires careful definition of the models, fitting them to the observed data, and then using the specialized `anova()` function to conduct a formal comparison between a complex model (the full model) and a simpler, restricted version (the reduced model).

The primary goal is to assess whether the increased complexity introduced by the full model--typically including higher-order or interaction terms--provides a statistically significant improvement in explaining the data's variation compared to the reduced model. We calculate the Lack of Fit Test statistic, which follows an F-distribution, and compare it to the F critical value or use the resulting P-value to determine if the additional complexity is justified by the data.

A **lack of fit test** is fundamentally utilized in statistical analysis to formally evaluate whether a comprehensive, or "full," model provides a statistically superior fit to a dataset when pitted against a more parsimonious, or "reduced," version of that model. This methodology is particularly relevant when attempting to identify the true underlying structure of the relationship between variables.

This test is essential because simply maximizing the coefficient of determination (R-squared) by adding more terms can lead to overfitting. The lack of fit test provides an objective, hypothesis-driven framework to justify the inclusion of those extra terms. It effectively partitions the total residual variation into two components: pure error and lack of fit error.

Introduction to the Lack of Fit Test Methodology

In practical statistical modeling, we often encounter situations where a simple linear relationship may not adequately describe the observed data. For instance, consider a scenario where we are analyzing the relationship between the **number of hours studied** and the resulting **exam score** achieved by students at a university. While a linear model might provide a baseline prediction, the relationship might actually follow a curvilinear path--scores might increase rapidly after a certain threshold of study time, suggesting a non-linear component.

To formally test if this curvature is significant, we define two competing regression models. The structure of these models is critical for the test: the reduced model must be nested within the full model, meaning the full model contains all the terms of the reduced model plus the additional term(s) whose impact we wish to evaluate.

In our example, we hypothesize that a quadratic relationship might be more appropriate than a simple linear one. This leads us to define the following nested pair of models:

Full Model (Quadratic Regression): Score = β0 + B1(hours) + B2(hours)2

Reduced Model (Simple Linear Regression): Score = β0 + B1(hours)

The formal hypothesis test is structured as follows: the Null Hypothesis (H0) posits that the reduced model is adequate (i.e., the coefficient B2 is zero, and the quadratic term adds no significant value), while the Alternative Hypothesis (HA) states that the full model provides a significantly better fit.

Step 1: Creating and Visualizing the Sample Dataset in R

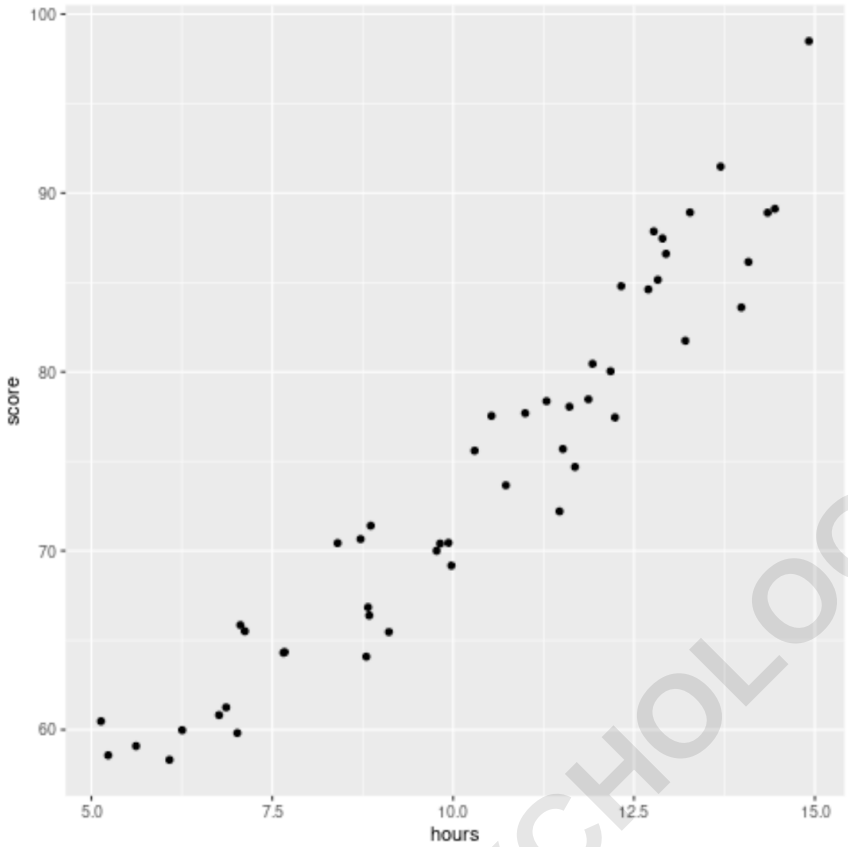
Before we can compare models, we must generate a dataset that exhibits the non-linear properties we intend to detect. We will use R's data manipulation capabilities to simulate data for 50 students, recording both their study hours and corresponding exam scores. Setting a seed ensures the reproducibility of this example.

```
#make this example reproducible
set.seed(1)

#create dataset
df <- data.frame(hours = runif(50, 5, 15), score=50)
df$score = df$score + df$hours^3/150 + df$hours*runif(50, 1, 2)

#view first six rows of data
head(df)

  hours    score
1 7.655087 64.30191
2 8.721239 70.65430
3 10.728534 73.66114
4 14.082078 86.14630
5 7.016819 59.81595
6 13.983897 83.60510
```

The synthetic generation process incorporates a cubic term (`df$hours^3/150`), ensuring that the underlying structure is indeed non-linear, which should lead to a significant result in the lack of fit test. After generating the data, the next critical step is visualization. A scatterplot allows us to visually inspect the relationship and confirm whether a straight line (linear model) would miss crucial trends in the data.

```
#load ggplot2 visualization package
library(ggplot2)

#create scatterplot
ggplot(df, aes(x=hours, y=score)) +
```

geom_point()



Upon visual inspection of the scatterplot, the points appear to follow a noticeable curve, particularly at higher values of hours studied, reinforcing the initial suspicion that a linear model alone would likely result in systematic error, thus confirming the need for a formal lack of fit assessment.

Step 2: Fitting the Competing Regression Models

The next procedural step involves using the standard R function for fitting linear models, `lm()`, to specify and fit both the full (quadratic) and the reduced (linear) models to our dataset, `df`. It is vital to use the `poly()` function when defining the full model in this context, as it ensures that the orthogonal polynomials are used, which can simplify the interpretation of the coefficients and comparisons between models.

#fit full model
full <- lm(score ~ poly(hours,2), data=df)

#fit reduced model
reduced <- lm(score ~ hours, data=df)

The `full` model incorporates the independent variable `hours` up to the second degree (`poly(hours, 2)`), allowing for quadratic curvature. Conversely, the `reduced` model restricts the relationship to be strictly linear, using only the first degree of the `hours` variable. These two fitted objects now contain all the necessary statistical information--such as the Residual Sum of Squares (RSS) and Degrees of Freedom (Df)--required to perform the lack of fit comparison.

Step 3: Executing the Lack of Fit Test using anova()

The formal lack of fit assessment is performed in R by supplying the fitted model objects to the `anova()` command. When `anova()` is provided with two nested model objects, it automatically executes a sequential Analysis of Variance (ANOVA), testing whether the inclusion of the additional terms in the more complex model (the full model) results in a statistically significant reduction in the residual error.

#lack of fit test
anova(full, reduced)

Analysis of Variance Table

Model 1: score ~ poly(hours, 2)
Model 2: score ~ hours
Res.Df RSS Df Sum of Sq F Pr(>F)
1 47 368.48
2 48 451.22 -1 -82.744 10.554 0.002144 **

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

Interpreting the Analysis of Variance (ANOVA) Results

The output from the `anova()` function provides a comprehensive summary table necessary for drawing a statistical conclusion. We are primarily focused on the comparison between the two models presented in the second row of the table, where the difference between Model 1 (Full) and Model 2 (Reduced) is quantified.

The crucial columns for interpretation are:

Res.Df: The Residual Degrees of Freedom. We observe that Model 1 (Full) has 47 degrees of freedom, while Model 2 (Reduced) has 48. The difference (Df = -1) reflects the single additional parameter (the B2 coefficient for the quadratic term) included in the full model.
RSS: The Residual Sum of Squares. This metric measures the unexplained variance. The full model has a lower RSS (368.48) than the reduced model (451.22), indicating that the quadratic

term successfully reduced the overall error.

Sum of Sq: The amount of variance explained by the additional term. The value of -82.744 represents the difference in RSS (451.22 - 368.48), quantifying the specific reduction in residual error achieved by adding the quadratic term.

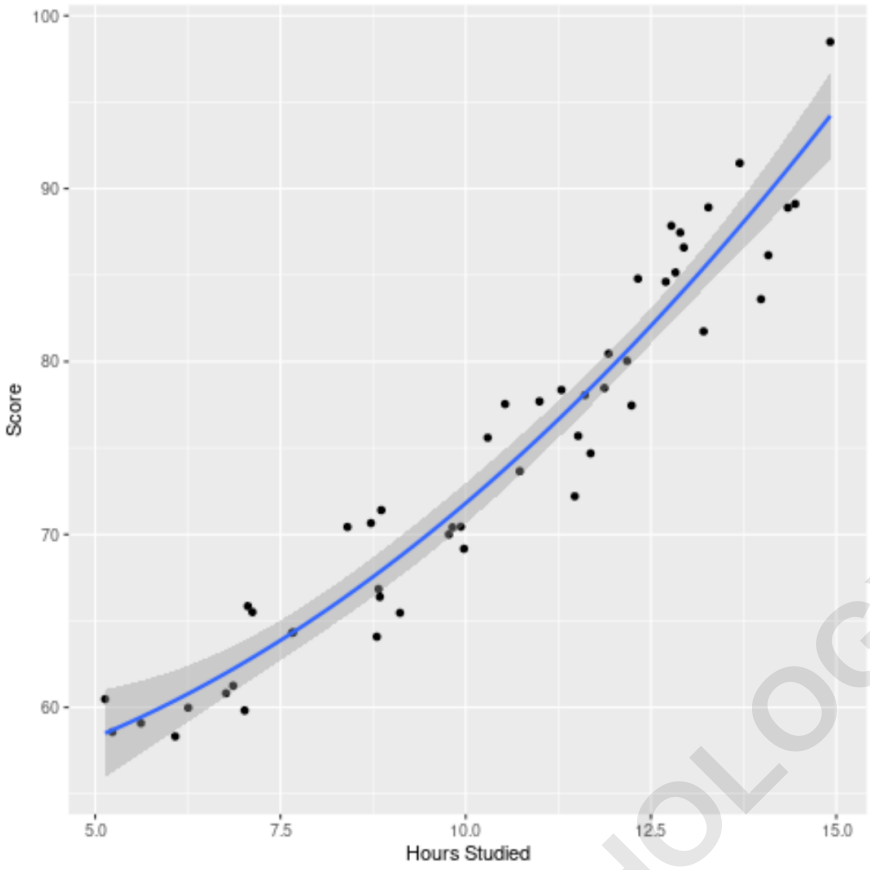
F and Pr(>F): The calculated F-statistic (10.554) and the corresponding P-value (0.002144). The F-statistic tests the ratio of the variance explained by the additional term (lack of fit) to the remaining unexplained variance (pure error).

Since the P-value (0.002144) is substantially smaller than the conventional significance level of $\alpha = 0.05$, we reject the Null Hypothesis. This rejection leads to the formal conclusion that the full, quadratic model offers a significantly better fit to the data than the simple linear model. In essence, the lack of fit in the reduced model is statistically significant, justifying the increased complexity provided by the quadratic term.

Step 4: Visualizing the Final, Superior Model

Having established statistically that the full (quadratic) model is superior, the final step involves visualizing this chosen model to confirm that its curve accurately captures the trend in the original data points. We use `ggplot2` again, but this time we integrate the `stat_smooth()` function to overlay the fitted polynomial line.

```
ggplot(df, aes(x=hours, y=score)) +
geom_point() +
stat_smooth(method='lm', formula = y ~ poly(x,2), size = 1) +
xlab('Hours Studied') +
ylab('Score')
```



The visualization clearly demonstrates how the quadratic curve smoothly traces the central tendency of the data, conforming to the non-linear pattern identified in Step 1. This visual evidence supports the statistical finding from the lack of fit test, confirming that the quadratic model is appropriate for predicting exam scores based on hours studied in this dataset.

Context and Application of Lack of Fit Testing

While the Lack of Fit Test is a powerful tool for comparing nested regression models, its formal application often relies on having replicated data points (multiple observations at the same predictor values). In cases where true replicates are not available, statisticians often use techniques like the pure error sum of squares approximation, or rely on generalized methods like comparing model residuals.

Furthermore, the Lack of Fit Test should be used in conjunction with other model diagnostic tools. Although this test confirmed the quadratic term was significant, other selection criteria, such as the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC), could also be employed to balance model fit against complexity, ensuring the final chosen model is both statistically sound and parsimonious.

How to Perform Polynomial Regression in R