

Learning Linear Hypothesis Testing with the `linearHypothesis()` Function in R

Authored by
Mohammed looti

May 5, 2026

RECOMMENDED CITATION

Mohammed looti (2026). *Learning Linear Hypothesis Testing with the `linearHypothesis()` Function in R*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=3552>

The Importance of Joint Hypothesis Testing in Regression

In advanced [regression analysis](#), researchers frequently encounter situations where they need to assess the collective impact of multiple predictors rather than just their individual effects. While standard statistical summaries provide individual t-tests for each predictor's [regression coefficient](#), these tests cannot adequately address complex restrictions or combined significance. This is precisely where the concept of [linear hypotheses](#) becomes fundamental, allowing analysts to evaluate whether a combination of coefficients simultaneously meets certain conditions, such as being jointly equal to zero or equal to each other.

To effectively handle these intricate statistical comparisons within the [R](#) environment, the highly respected [car package](#) offers the robust [linearHypothesis\(\)](#) function. This versatile function is essential for conducting rigorous [statistical inference](#) across a wide range of model types, including standard linear models (like [ordinary least squares \(OLS\)](#)) and more complex frameworks such as [generalized linear models \(GLMs\)](#). Utilizing this function moves the analysis beyond simple marginal significance checks to provide a more comprehensive view of model structure and predictor relationships.

Mastering the application of **`linearHypothesis()`** is crucial for analysts aiming to perform comprehensive and defensible statistical research in R. This function provides a powerful, unified framework for hypothesis testing that is far more flexible than relying solely on the output of the model summary. In the following sections, we will walk through a practical example, demonstrating how to specify and test a joint hypothesis to assess the combined significance of multiple predictors in a multiple regression model.

Defining the Syntax and Structure of `linearHypothesis()`

The core power of the **`linearHypothesis()`** function lies in its intuitive and flexible syntax, which facilitates the testing of nearly any linear combination of model parameters. At its most basic level, the function requires two fundamental inputs: the fitted model object and a character vector defining the restrictions or constraints to be tested against the model coefficients.

The general structure for defining and executing a joint hypothesis test is presented below:

```
linearHypothesis(fit, c("var1=0", "var2=0"))
```

Here, the argument `fit` must be a model object that has already been estimated, typically generated by R functions such as **`lm()`** for linear models or **`glm()`** for generalized linear models. The subsequent argument, the character vector `c("var1=0", "var2=0")`, explicitly specifies the null hypothesis. Each string within this vector represents an independent constraint placed upon

the model's [regression coefficients](#). The constraints are tested **jointly**, meaning the test evaluates whether **all** specified restrictions hold true simultaneously.

For instance, the command `c("var1=0", "var2=0")` is designed to test whether the coefficients corresponding to `var1` and `var2` are statistically indistinguishable from zero when considered together. Rejecting this joint null hypothesis implies that at least one of these two variables significantly contributes to the model. This capability is essential for model reduction and determining the overall significance of a group of related predictors, providing a crucial layer of statistical rigor often overlooked when relying solely on individual parameter tests.

Setting Up the Sample Data for Regression Modeling

To provide a clear, reproducible demonstration of `linearHypothesis()`, we must first construct a representative sample [data frame](#) in R. This dataset simulates a common scenario in educational research, where we are examining the factors that influence academic outcomes. Our simulated data tracks 10 hypothetical students and contains three critical variables: `score` (the final exam score, our dependent variable), `hours` (the total number of hours dedicated to studying), and `prac_exams` (the count of practice examinations completed). This controlled setup ensures that the subsequent regression analysis and hypothesis testing are easily verifiable.

The process of creating a simple, structured dataset is a foundational step in any statistical workflow, guaranteeing transparency and allowing others to replicate the analysis. Our goal is to assess how study time and practice habits contribute to overall academic performance, setting the stage for fitting a multiple regression model that incorporates both predictors. The R code below executes the creation of the data frame and displays its contents, confirming the structure of our variables.

#create data frame

```
df <- data.frame(score=c(77, 79, 84, 85, 88, 99, 95, 90, 92, 94),
hours=c(1, 1, 2, 3, 2, 4, 4, 2, 3, 3),
prac_exams=c(2, 4, 4, 2, 4, 5, 4, 3, 2, 1))
```

#view data frame

```
df
```

```
score hours prac_exams
1 77 1 2
2 79 1 4
3 84 2 4
4 85 3 2
5 88 2 4
```

```
6 99 4 5
7 95 4 4
8 90 2 3
9 92 3 2
10 94 3 1
```

As confirmed by the output, we have successfully created a data structure with 10 observations and the three specified variables, ready for use. The data frame `df` now serves as the input for fitting our explanatory model, moving us closer to testing our hypotheses about the influence of study habits on exam scores.

Fitting and Evaluating the Multiple Linear Regression Model

Following the data preparation stage, the next logical step involves fitting a [multiple linear regression model](#). This model aims to quantify the relationship between the dependent variable (`score`) and our two independent variables (`hours` and `prac_exams`). This statistical technique allows us to estimate the unique contribution of each predictor while holding the other constant, providing a controlled assessment of their influence.

The mathematical formulation underlying our model is expressed as: **Exam score = $\beta_0 + \beta_1(\text{hours}) + \beta_2(\text{practice exams}) + \epsilon$** . In this equation, β_0 represents the estimated intercept, while β_1 and β_2 are the estimated [regression coefficients](#) (or [\$\beta\$ \(beta\)](#) values) that quantify the expected change in the exam score for a one-unit change in the respective predictor. We use R's fundamental linear model function, `lm()`, to calculate these estimates, storing the results in the object `fit`.

After fitting the model, we inspect the `summary(fit)` output to obtain preliminary insights into the model's performance and the individual significance of the predictors:

```
#fit multiple linear regression model
fit <- lm(score ~ hours + prac_exams, data=df)

#view summary of model
summary(fit)

Call:
lm(formula = score ~ hours + prac_exams, data = df)

Residuals:
Min 1Q Median 3Q Max
-5.8366 -2.0875 0.1381 2.0652 4.6381
```

Coefficients:

```
Estimate Std. Error t value Pr(>|t|)
(Intercept) 72.7393 3.9455 18.436 3.42e-07 ***
hours 5.8093 1.1161 5.205 0.00125 **
prac_exams 0.3346 0.9369 0.357 0.73150
```

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

Residual standard error: 3.59 on 7 degrees of freedom

Multiple R-squared: 0.8004, Adjusted R-squared: 0.7434

F-statistic: 14.03 on 2 and 7 DF, p-value: 0.003553

The summary reveals that the `hours` variable exhibits strong statistical significance ([p-value](#) < 0.01), suggesting a reliable positive relationship with exam scores. Conversely, the `prac_exams` variable appears non-significant individually ([p-value](#) > 0.05). The overall model, however, is statistically significant (global [F-statistic](#) p-value of 0.003553), and the R-squared value indicates that approximately 80% of the variance in scores is accounted for by these two predictors. This discrepancy--where one variable is individually significant but the overall model is strong--motivates the necessity of a joint hypothesis test to precisely determine the collective contribution of both predictors.

Executing the Joint Hypothesis Test using `linearHypothesis()`

The preliminary individual tests indicate that study hours are important, but we are still left with a critical question: Do `hours` and `prac_exams` contribute significantly to the model *together*? We want to test the joint null hypothesis that the [regression coefficients](#) for both variables are simultaneously zero. This test compares the fit of our full model against a more restricted model where these two predictors are forcibly removed.

To perform this rigorous comparison, we must utilize the [linearHypothesis\(\)](#) function, which requires the [car package](#) to be loaded. We pass our fitted model object (`fit`) and the constraints--that the coefficients for `hours` and `prac_exams` are zero--to the function as a character vector. The output provides the results of an [F-test](#) comparing the unrestricted model (Model 2) to the restricted model (Model 1).

`library(car)`

```
#perform hypothesis test for hours=0 and prac_exams=0
linearHypothesis(fit, c("hours=0", "prac_exams=0"))
```

Linear hypothesis test

Hypothesis:

hours = 0

prac_exams = 0

Model 1: restricted model

Model 2: score ~ hours + prac_exams

```
Res.Df RSS Df Sum of Sq F Pr(>F)
1 9 452.10
2 7 90.24 2 361.86 14.035 0.003553 **
---
```

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

The resulting table presents the detailed comparison. **Model 1**, the restricted version, has a residual sum of squares (RSS) of 452.10, representing the variance remaining when both predictors are excluded. **Model 2**, the full model, has a significantly smaller RSS of 90.24. The difference in RSS (361.86) is the variation explained by the joint inclusion of `hours` and `prac_exams`. This reduction in unexplained variance is summarized by the calculated [F test statistic](#) and its corresponding [p-value](#), which form the basis of our conclusion.

Interpreting the Joint Hypothesis Test Results

The interpretation of the `linearHypothesis()` output is vital for drawing statistically sound conclusions about the model's structure. The key results provided are the F test statistic (14.035) and the associated p-value (0.003553). These values directly test our formal hypotheses:

H0 (Null Hypothesis): The [regression coefficients](#) for `hours` and `prac_exams` are simultaneously equal to zero (i.e., they have no joint linear effect).

HA (Alternative Hypothesis): At least one of the [regression coefficients](#) is not equal to zero (i.e., they collectively contribute to explaining the score variance).

To determine whether we should reject the [null hypothesis](#), we compare the calculated p-value to our chosen [significance level \(alpha\)](#), which is conventionally set at 0.05. Since our p-value (0.003553) is substantially smaller than 0.05, we have strong statistical evidence to **reject the null hypothesis**.

The rejection of H0 leads to the conclusion that the variables `hours` and `prac_exams`, when evaluated together, significantly contribute to explaining the variation in students' final exam scores. This result is particularly important because, while `prac_exams` appeared insignificant on its own in the model summary, the joint test confirms that the combined effect of the study factors is highly relevant. This underscores the power of joint hypothesis testing: it can confirm the

collective relevance of a group of predictors, even if one or more members of that group do not meet the threshold for individual statistical significance.

Additional Resources for Regression Analysis

For those interested in deepening their expertise in linear modeling, multivariate testing, and the R programming language, the following official and authoritative resources are highly recommended. These links provide comprehensive documentation and tutorials that expand upon the concepts of statistical hypothesis testing and model fitting demonstrated in this guide.

[Quick-R: Linear Regression](#)

[R Documentation: lm\(\) function](#)

[R Documentation: linearHypothesis\(\) function](#)