

How to Perform One-Way ANOVA in R Using `aov()` and `anova()`

Authored by
stats writer

November 28, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Perform One-Way ANOVA in R Using `aov()` and `anova()`*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=100724>

The statistical environment R provides two distinct functions, `aov()` and `anova()`, which are frequently confused due to their similar naming convention. While both relate to the Analysis of Variance (ANOVA) framework, their primary applications, output structures, and underlying computational methods are fundamentally different. Understanding these distinctions is critical for accurately modeling statistical relationships in R.

The core difference lies in their purpose: `aov()` is designed specifically for fitting traditional ANOVA models, particularly those based on the general linear model, and returning an object suitable for summarizing variance decomposition. Conversely, `anova()` is a generic function used for comparing two or more fitted statistical models--often regression models--to assess whether a more complex model provides a statistically significant improvement in fit over a simpler, nested alternative. This distinction moves beyond merely Type I vs. Type II sums of squares, addressing the very nature of the statistical question being asked.

Traditionally, `aov()` in R calculates sequential (Type I) sums of squares, meaning the order of terms in the model formula matters significantly. This is typically appropriate for balanced designs, such as standard One-Way ANOVA where all groups have equal sample sizes. When comparing nested models, `anova()` determines the incremental improvement of adding predictors, a classic application often referred to as a lack-of-fit test. The following detailed discussion and practical examples clarify these distinct uses.

The Primary Role of `aov()`: Fitting ANOVA Models

The function `aov()`, short for "Analysis of Variance," is the standard tool in R for fitting models that assess differences in group means, primarily within the context of experimental designs. When using `aov()`, the user is typically interested in decomposing the total variability observed in a dependent variable into components attributable to specific factors or independent variables. The result is an object of class `aov` (which also inherits from `lm`), containing the structure necessary for subsequent analysis of variance tables.

The core utility of `aov()` lies in its ability to handle complex factorial designs, including models with interaction effects, blocking variables, and repeated measures. Although it relies on the same underlying linear model mathematics as `lm()` (Linear Model), `aov()` specifically formats the output and internal structure to make the standard ANOVA summary table easily accessible via the `summary()` function. When you execute `summary(aov_fit)`, R presents the traditional breakdown: degrees of freedom (Df), sum of squares (Sum Sq), mean square (Mean Sq), F-statistic (F value), and the associated p-value ($\text{Pr}(>F)$).

It is crucial to note that `aov()`, by default, calculates sequential sums of squares, also known as Type I sums of squares. This means that the variation explained by the first term in the model formula is calculated first, then the variation explained by the second term given the first, and so

on. If the design is unbalanced (unequal sample sizes or non-orthogonal factors), changing the order of terms in the formula can alter the resulting F-statistics and p-values, which is a key characteristic of Type I analysis. For balanced designs, where the predictors are uncorrelated, the order does not matter.

Theoretical Context of `aov()`: Type I Sums of Squares

The concept of Type I sums of squares, inherently linked to the default operation of `aov()`, defines how variance is partitioned among predictors. In a Type I analysis, each factor is assessed based on the variance it explains after accounting for all factors that precede it in the model specification. This sequential nature provides a clear, hierarchical view of variance decomposition, which is highly interpretable in designed experiments where researchers deliberately introduce factors in a specific, theoretically meaningful order.

However, the reliance on sequential testing means that researchers must be meticulous when constructing their model formula, especially when dealing with observational data or experimental designs that have become unbalanced due to attrition or practical constraints. For instance, if an interaction term is placed before its main effects, the interaction term would absorb variance that might otherwise be attributed to the main effects, potentially leading to misleading conclusions about the significance of the main factors.

For situations involving complex or highly unbalanced designs, researchers often prefer Type II or Type III sums of squares, which test main effects adjusted for other main effects (Type II) or adjusted for all other terms including interactions (Type III). While R's base `aov()` function does not directly calculate Type II or Type III sums of squares, these can be obtained using specialized packages like `car` (specifically, the `Anova()` function within that package, note the capital 'A'), which offer more flexibility for different statistical testing contexts.

Detailed Example 1: Implementing a One-Way ANOVA using `aov()`

To illustrate the practical application of `aov()`, consider a scenario where we investigate the impact of different exercise programs on weight loss. This is a classic One-Way ANOVA design, suitable for analysis using `aov()`.

Suppose an experiment recruits 90 individuals, randomly assigning 30 people to each of three distinct programs (A, B, or C) for one month. The goal is to determine if the mean weight loss significantly differs across the three programs. Since this is a balanced design (30 participants per group), the Type I sums of squares approach provided by `aov()` is appropriate and straightforward.

The following code block demonstrates how to generate simulated data reflective of this

experiment and subsequently fit the ANOVA model using the `aov()` function, followed by generating the standard summary table:

#make this example reproducible

```
set.seed(0)
```

```
#create data frame
```

```
df <- data.frame(program = rep(c("A", "B", "C"), each=30),
```

```
weight_loss = c(runif(30, 0, 3),
```

```
runif(30, 0, 5),
```

```
runif(30, 1, 7)))
```

```
#fit one-way anova using aov()
```

```
fit <- aov(weight_loss ~ program, data=df)
```

```
#view results
```

```
summary(fit)
```

```
Df Sum Sq Mean Sq F value Pr(>F)
```

```
program 2 98.93 49.46 30.83 7.55e-11 ***
```

```
Residuals 87 139.57 1.60
```

```
---
```

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

Interpreting the Output of `aov()`

The output generated by `summary(fit)` provides a comprehensive breakdown of the variance. The row labeled "program" represents the variation explained by the differences between the three exercise programs, while the "Residuals" row represents the unexplained, within-group variation (error).

We observe a Degrees of Freedom (Df) of 2 for the program factor ($k-1$, where $k=3$ programs) and 87 for the residuals ($N-k$, where $N=90$ total observations). The Sum of Squares (Sum Sq) for the program is 98.93, indicating the total variation in weight loss attributable to the different programs. Dividing this by the Df yields the Mean Square (Mean Sq), which is 49.46.

The critical element is the F value, calculated as the ratio of the Mean Square for the factor (49.46) to the Mean Square for the residuals (1.60). An F value of 30.83 is derived. This value is used to calculate the p-value, denoted as $\text{Pr}(>F)$. In this case, the p-value is extremely small (7.55e-11, or 0.0000000000755).

Since this p-value is far less than the conventional significance level of 0.05, we reject the null hypothesis of the One-Way ANOVA. We conclude that there is a statistically significant difference in the mean weight loss between the three exercise programs. This analysis confirms that `aov()` is the appropriate function for fitting and summarizing standard ANOVA designs.

The Primary Role of `anova()`: Comparing Nested Models

Unlike `aov()`, the function `anova()` in R is not primarily used to fit a model, but rather to evaluate the statistical comparison between two or more existing, nested statistical models. This function is an S3 generic method, meaning its behavior changes depending on the class of the arguments provided. When used with multiple models--typically those fitted using `lm()` (linear models) or `glm()` (generalized linear models)--it performs a likelihood ratio test or a sequence of F-tests to assess whether the more complex model offers a statistically significant improvement in fit over the simpler model(s).

A crucial prerequisite for using `anova(model1, model2)` is that the models must be "nested." This means that the simpler model (the reduced model) must be a specific case of the more complex model (the full model), where the reduced model can be obtained by setting one or more coefficients in the full model to zero. For instance, if the full model includes a quadratic term for a predictor, the reduced model might only include the linear term for that predictor. The `anova()` test then assesses whether the additional predictor(s) in the full model significantly account for more variance than the reduced model.

This application is highly valuable in regression models when researchers are engaged in model selection or assessing whether non-linear components, interaction terms, or additional predictors contribute meaningfully to the explanatory power of the model. The output focuses not on the overall variance decomposition of a single model, but on the difference in residual sums of squares (RSS) between the models being compared.

Understanding Nested Regression Models

The concept of nested models is central to the proper utilization of the `anova()` function. When comparing Model A (the reduced model) and Model B (the full model), Model A must be nested within Model B. If the models are not nested, the comparison made by `anova()` is invalid and the resulting statistical interpretation is meaningless. For linear models, the comparison relies on calculating the difference in the residual sums of squares (RSS) between the two models.

The null hypothesis (H_0) for the `anova()` comparison is that the additional parameters in the full model (Model B) are zero--meaning the reduced model (Model A) is sufficient to explain the data. The alternative hypothesis (H_A) is that at least one of the added parameters is non-zero, indicating the full model provides a significantly better fit.

The comparison results in an F-statistic, which compares the reduction in error variance achieved by the full model relative to the added degrees of freedom. This method is often used for specific testing procedures, such as assessing the significance of a single block of variables or performing a lack-of-fit test, which determines if a simple linear relationship adequately captures the trend compared to a more complex polynomial or non-linear relationship.

Detailed Example 2: Utilizing `anova()` for Model Comparison (Lack of Fit Test)

Consider a scenario in predictive modeling where we attempt to predict a student's exam score based on the number of hours studied. We want to determine if incorporating a quadratic term for study hours significantly improves the prediction accuracy over a simple linear model. This requires comparing a full model and a reduced model, which are nested.

We define our models as follows:

Full Model: $\text{Score} = \beta_0 + \beta_1(\text{hours}) + \beta_2(\text{hours})^2$

Reduced Model: $\text{Score} = \beta_0 + \beta_1(\text{hours})$

The reduced model is nested within the full model because it is equivalent to setting the parameter β_2 to zero in the full model. The `anova()` function will test the null hypothesis $H_0: \beta_2 = 0$.

The following R code first fits both the full (quadratic) and reduced (linear) regression models using `lm()`, and then uses `anova()` to perform the comparison:

#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 head 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

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

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

#perform lack of fit test using anova()
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 `anova()` Output for Model Significance

The output table from `anova(full, reduced)` clearly contrasts the two models. Model 1 is the full model, and Model 2 is the reduced model. The table shows the Residual Degrees of Freedom (Res.Df) and the Residual Sum of Squares (RSS) for both models.

Model 1 (Full): **Res.Df** = 47, **RSS** = 368.48. This represents the unexplained variance when using the quadratic model.

Model 2 (Reduced): **Res.Df** = 48, **RSS** = 451.22. This represents the unexplained variance when using only the linear model.

The critical row in the output corresponds to the comparison (Model 2, or the test row). The difference in Df is 1 (48 - 47), representing the one additional parameter (β_2) included in the full model. The difference in Sum of Squares (Sum of Sq) is 82.744 (451.22 - 368.48), representing the amount of variation explained uniquely by the addition of the quadratic term.

The F-statistic (10.554) is derived from comparing this reduction in RSS to the estimated error variance of the full model. The associated p-value (Pr(>F)) is 0.002144.

Since the calculated p-value (0.002144) is substantially less than the standard threshold of 0.05, we reject the null hypothesis. We conclude that the full model, which includes the quadratic term, offers a statistically significantly better fit to the data than the simpler linear model. Therefore, the

non-linear relationship suggested by the quadratic term is essential for accurately modeling the effect of study hours on exam score.

Summary of Key Differences

While both `aov()` and `anova()` are foundational functions for variance analysis in R, their intended uses are distinct and generally non-interchangeable. Choosing the correct function depends entirely on the statistical objective:

`aov()` Purpose: Primary function is to **fit** an ANOVA model. It takes predictors and a response variable and returns a model object designed for variance decomposition via the `summary()` method. It defaults to calculating Type I (sequential) sums of squares. This is ideal for analyzing results from designed experiments, such as One-Way ANOVA or factorial designs.

`anova()` Purpose: Primary function is to **compare** two or more fitted statistical models (e.g., regression models). It calculates the F-statistic or likelihood ratio test comparing the residual variances of nested models to determine if the added complexity is statistically justified. This is essential for model selection, lack-of-fit testing, and sequential testing of specific parameter blocks.

In essence, `aov()` creates the structure for the analysis of variance, whereas `anova()` provides the framework for comparing different model structures, irrespective of whether those structures were created via `lm()` or `aov()`. Understanding this division of labor ensures robust and statistically sound analysis when working with linear models in R.