How to Perform a Two-Way ANOVA in Python?

Authored by

stats writer

December 25, 2025

The Two-Way ANOVA (Analysis of Variance) is a powerful statistical tool used to analyze the relationship between two categorical independent variables (factors) and a continuous dependent variable (response). In Python, this analysis is efficiently implemented using the `statsmodels.formula.api` package.

This package provides access to the `ols` class, which stands for Ordinary Least Squares. The `ols` class is fundamental because it allows us to fit a Linear Model to our experimental data using algebraic formulas, mirroring the structure often used in R. Once the model is fitted, the two-way ANOVA is performed using the `anova_lm()` method. This function takes the fitted linear model object and generates a comprehensive ANOVA summary table, providing the critical statistics needed to analyze the results, including F-statistics and p-values.

The core goal of utilizing the Two-Way ANOVA is twofold: first, to ascertain whether each independent factor individually influences the response variable (main effects); and second, to determine if the effect of one factor changes depending on the level of the other factor (the interaction effect). This tutorial provides a detailed, step-by-step guide on how to structure the data, execute the statistical analysis, and interpret the results of a Two-Way ANOVA using Python.

Understanding the Two-Way ANOVA Model

The Two-Way ANOVA is an extension of the basic one-way ANOVA, allowing researchers to study the effects of two factors simultaneously. It is typically employed when the experimental design involves crossing two factors, meaning every level of the first factor is combined with every level of the second factor. This design allows for a deeper understanding of complex relationships in the data set.

Statistically, the Two-Way ANOVA decomposes the total variance observed in the response variable into three distinct components: the variance attributable to the first factor (Factor A), the variance attributable to the second factor (Factor B), and the variance attributable to the unique combination of the two factors (the interaction term, A × B). The ability to isolate the interaction term is a significant advantage over simply running two separate one-way ANOVAs, as interacting factors might show no individual effect but a strong combined effect, or vice-versa.

When conducting a Two-Way ANOVA, we test three primary sets of null hypotheses (H?):

Main Effect of Factor A (Watering Frequency): H?: The population means across the levels of Factor A are equal.
Main Effect of Factor B (Sunlight Exposure): H?: The population means across the levels of Factor B are equal.
Interaction Effect (A × B): H?: There is no statistically significant interaction between Factor A and Factor B affecting the response variable.

Prerequisites and Python Libraries

To perform statistical analysis in Python, a foundational set of libraries is required. For data manipulation and structuring, we rely heavily on Pandas and NumPy. Pandas is crucial for creating and managing the experimental data structure (the DataFrame), while NumPy provides the necessary tools for numerical computations, particularly when generating synthetic data sets or handling array operations.

The statistical heavy lifting is handled by statsmodels. Specifically, we utilize the `statsmodels.formula.api` submodule, which offers a syntax highly familiar to users of R. By using the `ols` class (for Ordinary Least Squares), we can define our complex statistical model simply by writing a formula string, such as `response ~ factor1 + factor2 + factor1:factor2`, which dramatically simplifies the modeling process compared to matrix-based approaches.

It is important to understand that ANOVA, despite being defined by comparing means, is fundamentally a specific application of the General Linear Model. The `ols` function fits this linear model by minimizing the sum of the squared differences between the observed data and the values predicted by the model. The `anova_lm()` method then takes the variance partitioning derived from this fitted linear model to calculate the necessary F-statistics and p-values required for the ANOVA summary table.

Setting Up the Experimental Data: The Botanist Study

We will use a practical example to illustrate the process. A botanist seeks to understand how two factors--watering frequency and sunlight exposure--influence plant growth. The experiment involves 30 seeds, which are grown for two months under controlled conditions. The response variable, the height of the plant (in inches), is collected at the end of the experimental period. This setup is a classic example where a Two-Way ANOVA is the appropriate analysis technique.

To prepare the data for analysis in Python, we must organize it into a Pandas DataFrame. This structure ensures that each observation is linked correctly to its corresponding experimental conditions. Our DataFrame will include three variables representing the experiment's structure:

water: A categorical factor indicating the watering schedule (daily or weekly).
sun: A categorical factor indicating the level of sunlight exposure (low, medium, or high).
height: The continuous response variable (plant height in inches).

The code below sets up the experimental data, ensuring a balanced design where observations are distributed across all factor combinations:

Step 1: Enter the data.

import numpy as np
import pandas as pd

#create data
df = pd.DataFrame({'water': np.repeat(, 15),
'sun': np.tile(np.repeat(, 5), 2),
'height': })

#view first ten rows of data
df

water sun height
0 daily low 6
1 daily low 6
2 daily low 6
3 daily low 5
4 daily low 6
5 daily med 5
6 daily med 5
7 daily med 6
8 daily med 4
9 daily med 5

Formulating the Statistical Model

The next critical step is defining the statistical model that represents our experimental hypothesis. In the `statsmodels` formula language, this is done using a simple string notation. For a Two-Way ANOVA, the formula must specify the response variable, the two factors, and crucially, the interaction term between the factors.

The formula structure used for the plant growth study is `'height ~ C(water) + C(sun) + C(water):C(sun)'`. The left side of the tilde (~) denotes the dependent variable (`height`). The right side defines the independent terms: the main effect of `water`, the main effect of `sun`, and the interaction effect `water:sun`. The colon operator (`:`) specifically requests the multiplicative interaction between the two specified factors.

The function `C()` wraps the factor names (`water` and `sun`) and explicitly tells `ols` that these variables should be treated as **categorical factors**, regardless of how they are stored in the Pandas DataFrame. This ensures that the model correctly generates dummy variables for the

levels of each factor, which is essential for performing ANOVA correctly within the Linear Model framework.

When executing the ANOVA, we must also consider the Type of Sum of Squares (SS). Statistical software often defaults to Type I, II, or III SS. Type I is sequential, Type II is suitable when interaction effects are non-significant, and Type III is typically used when interaction effects are significant or in unbalanced designs. For this analysis, we will specify `typ=2` (Type II SS) in the `anova_lm` function. Type II SS tests the main effects after accounting for the other main effect, but not the interaction term, making it a robust choice when interaction is suspected but not confirmed as significant.

Executing the Two-Way ANOVA in Python

With the data structured and the model defined, we proceed to fit the model using the `ols` function from the `statsmodels.formula.api` module. This step generates a fitted Linear Model object that contains all the coefficients and residuals needed for variance analysis.

The `.fit()` method executes the Ordinary Least Squares estimation process, calculating the parameters that best fit the data according to the specified formula. The resulting model object, stored here as `model`, is the input for the final ANOVA calculation.

The function `sm.stats.anova_lm(model, typ=2)` takes the fitted model and applies the ANOVA decomposition, using Type II Sum of Squares as specified. This output is a table containing the degrees of freedom, sums of squares, F-statistics, and the corresponding p-values (PR(>F)) for each effect included in the model.

Step 2: Perform the two-way ANOVA.

import statsmodels.api as sm
from statsmodels.formula.api import ols

#perform two-way ANOVA
model = ols('height ~ C(water) + C(sun) + C(water):C(sun)', data=df).fit()
sm.stats.anova_lm(model, typ=2)

sum_sq df F PR(>F)
C(water) 8.533333 1.0 16.0000 0.000527
C(sun) 24.866667 2.0 23.3125 0.000002
C(water):C(sun) 2.466667 2.0 2.3125 0.120667
Residual 12.800000 24.0 NaN NaN

Detailed Interpretation of ANOVA Output

The generated ANOVA table provides all the critical metrics required to test the null hypotheses established earlier. To interpret these results, we focus on the F-statistic and the corresponding p-value, labeled `PR(>F)`. The F-statistic is the test statistic in ANOVA, representing the ratio of the variance explained by the factor to the unexplained variance (the Residual Mean Square).

The `sum_sq` column shows the Sum of Squares for each factor, indicating how much variation in plant height is accounted for by that specific factor (or the interaction). The `df` column shows the Degrees of Freedom associated with each effect. Finally, the critical column is `PR(>F)`, which is the p-value. This value represents the probability of observing the data (or more extreme results) if the null hypothesis were true.

Our standard threshold for statistical significance is typically an alpha level (α) of 0.05. If the p-value is less than 0.05, we reject the null hypothesis, concluding that the factor has a statistically significant effect on the dependent variable.

Analyzing Main Effects and Interaction Effects

By examining the `PR(>F)` column, we can draw conclusions about the effects of watering frequency, sunlight exposure, and their combination:

Watering Frequency (C(water)): The p-value is 0.000527. Since 0.000527 is much smaller than our 0.05 threshold, we reject the null hypothesis. This indicates that **watering frequency has a statistically significant effect on plant height.**
Sunlight Exposure (C(sun)): The p-value is 0.000002. This extremely small value leads us to strongly reject the null hypothesis. We conclude that **sunlight exposure has a statistically significant effect on plant height.**
Interaction Effect (C(water):C(sun)): The p-value is 0.120667. Since 0.120667 is greater than 0.05, we fail to reject the null hypothesis. This means that **there is no statistically significant interaction effect** between watering frequency and sunlight exposure on plant height.

The interpretation suggests that both main factors independently impact plant growth. However, the lack of a significant interaction effect is crucial: it means that the effect of switching from "daily" to "weekly" watering is roughly the same regardless of whether the plant received low, medium, or high sunlight exposure. Conversely, the effect of increasing sunlight level is consistent whether the plant is watered daily or weekly.

In summary, the experiment confirms that both watering schedule and light level are critical determinants of final plant height. The botanical implication is that researchers can study the effects of these factors independently, as they do not appear to modify each other's influence on

growth.

Next Steps: Understanding Differences through Post-Hoc Analysis

Although the ANOVA results confirm that statistically significant differences exist among the levels of watering and sunlight, the test itself does not specify *where* those differences lie. For instance, the significant result for `C(sun)` tells us that 'low', 'medium', and 'high' sunlight levels do not all produce the same average height, but it does not tell us if 'low' is significantly different from 'medium', or if 'medium' is different from 'high'.

To determine these specific group differences, we must conduct Post-Hoc Tests (or multiple comparison tests). These tests are essential when a factor has more than two levels, as is the case for the `sun` factor (low, med, high).

The most common and robust post-hoc procedure is the Tukey's Honestly Significant Difference (HSD) test. This test compares all possible pairs of means while maintaining control over the family-wise error rate, ensuring that the overall probability of making a Type I error across all comparisons remains at the desired alpha level (e.g., 0.05). In Python, post-hoc analysis can be executed using the `pairwise_tukeyhsd` function, also available within the statsmodels library, to provide precise confidence intervals and p-values for every pair comparison.

Note: Although the ANOVA results tell us that watering frequency and sunlight exposure have a statistically significant effect on plant height, we would need to perform Post-Hoc Tests to determine exactly how different levels of water and sunlight affect plant height. This provides granular detail beyond the general finding of significance provided by the F-test.

The following tutorials explain how to perform other common tasks in Python: