

How to Perform a Three-Way ANOVA in Python: A Step-by-Step Guide

Authored by
stats writer

November 28, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Perform a Three-Way ANOVA in Python: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=101000>

Introduction to the Three-Way ANOVA

The **three-way ANOVA** (Analysis of Variance) is a sophisticated statistical method used by analysts and researchers to simultaneously evaluate the impact of three independent categorical variables (factors) on a single continuous dependent variable. The primary purpose of this technique is to determine whether there is a **statistically significant difference** among the mean scores of the dependent variable when the population is segmented based on the levels of these three factors. Unlike simpler designs, the three-way ANOVA allows for the isolation and testing of complex interaction effects, providing a much deeper understanding of the underlying data structure.

This form of multivariate analysis is critical when conducting experiments where outcomes are believed to be influenced not just by individual factors, but by the specific combinations of these factors. Successfully implementing this analysis in a production environment, such as data science pipelines, often relies on powerful, open-source programming tools. Python, through its robust statistical libraries, particularly **statsmodels**, provides an accessible and flexible framework for executing and interpreting these complex models with high precision.

The following example provides a detailed walkthrough, demonstrating how to properly structure the data, specify the full factorial model, execute the three-way ANOVA using Python, and rigorously interpret the final statistical output, including the crucial F-statistics and p-values associated with both main and interaction effects.

Case Study: Three-Way ANOVA for Athletic Performance

Consider a scenario in sports science where a researcher aims to investigate the factors driving improvement in vertical jump height among college basketball players. The primary intervention is the comparison of two distinct **training programs** (Program A vs. Program B). This represents the first independent factor.

However, the researcher hypothesizes that the effectiveness of the training program may not be universal. They suspect that the player's **gender** (Male or Female) and their competitive **division** (NCAA Division I or Division II) may also influence the degree of improvement, either independently or in combination with the training program. These three variables--program, gender, and division--constitute the three factors defining the experimental design.

The ultimate objective is to perform a three-way ANOVA to determine, firstly, the main effects of training program, gender, and division on jumping height improvement, and secondly, whether any significant two-way or three-way interaction effects exist, which would indicate that the impact of one factor depends conditionally on the level of the others. The subsequent steps detail the data creation and modeling process necessary to achieve this goal within the Python environment.

Step 1: Data Structuring using Pandas and NumPy

The foundation of any sound statistical analysis in Python is the accurate creation and structuring of the dataset. We rely on the **NumPy** library for efficient array operations and the **Pandas** library for generating the organizational structure, specifically the **Pandas DataFrame**, which is essential for statistical modeling in Python. Since this is a balanced factorial design (equal observations in each cell), we must ensure the factors are distributed correctly across the 40 total observations.

We define four columns: the three independent factors (`program`, `gender`, `division`) and the continuous dependent variable (`height`, representing the measured improvement in jumping height). The use of NumPy functions like `repeat()` and `tile()` ensures a systematic and balanced allocation of factor levels, which is highly beneficial for robust ANOVA calculations. The categorical factors are encoded numerically (1 and 2 for program and division) or alphabetically (M and F for gender), while the dependent variable holds the raw measurement scores.

The code below executes the data creation and provides a preview of the initial rows, confirming that the data is correctly structured and ready for statistical processing.

```
import numpy as np
import pandas as pd

#create DataFrame
df = pd.DataFrame({'program': np.repeat(, 20),
'gender': np.tile(np.repeat(, 10), 2),
'division': np.tile(np.repeat(, 5), 4),
'height': })

#view first ten rows of DataFrame
df

program gender division height
0 1 M 1 7
1 1 M 1 7
2 1 M 1 8
3 1 M 1 8
4 1 M 1 7
5 1 M 2 6
6 1 M 2 6
7 1 M 2 5
8 1 M 2 6
9 1 M 2 5
```

Step 2: Model Specification and Execution in statsmodels

Once the data is prepared, the next step involves defining and fitting the statistical model. The **statsmodels** library is utilized, specifically the `ols()` (Ordinary Least Squares) function from the formula API, as ANOVA is mathematically equivalent to a regression model using dummy variables for categorical predictors. It is crucial to use the `C()` notation around the factor names (``program``, ``gender``, ``division``). This explicitly tells `statsmodels` to treat these variables as categorical factors, ensuring correct calculation of the Sum of Squares.

The model formula must be exhaustive for a three-way ANOVA, including all main effects and all possible interaction terms. The formula includes three main effects, three two-way interaction terms, and one single three-way interaction term. The full model specification is designed to partition the total variance in ``height`` across all possible experimental sources of variation.

After fitting the OLS model to the data, we use the `anova_lm()` function to generate the ANOVA table. By specifying `typ=2`, we request Type II Sum of Squares. Type II tests the effect of a factor after accounting for all other main effects and lower-order interaction effects that do not contain the factor. This method is highly recommended for factorial designs, particularly when interactions are expected to be non-significant, as it ensures robust testing of the main effects.

```
import statsmodels.api as sm
```

```
from statsmodels.formula.api import ols
```

```
#perform three-way ANOVA
```

```
model = ols("""height ~ C(program) + C(gender) + C(division) +
C(program):C(gender) + C(program):C(division) + C(gender):C(division) +
C(program):C(gender):C(division)""", data=df).fit()
```

```
sm.stats.anova_lm(model, typ=2)
```

```
sum_sq df F PR(>F)
```

```
C(program) 3.610000e+01 1.0 6.563636e+01 2.983934e-09
```

```
C(gender) 6.760000e+01 1.0 1.229091e+02 1.714432e-12
```

```
C(division) 1.960000e+01 1.0 3.563636e-01 1.185218e-06
```

```
C(program):C(gender) 2.621672e-30 1.0 4.766677e-30 1.000000e+00
```

```
C(program):C(division) 4.000000e-01 1.0 7.272727e-01 4.001069e-01
```

```
C(gender):C(division) 1.000000e-01 1.0 1.818182e-01 6.726702e-01
```

```
C(program):C(gender):C(division) 1.000000e-01 1.0 1.818182e-01 6.726702e-01
```

```
Residual 1.760000e+01 32.0 NaN NaN
```

Step 3: Interpreting Interaction and Main Effects

The resulting ANOVA table provides several key metrics for each source of variation (rows in the table). The critical columns for hypothesis testing are the F column, which provides the calculated **F-statistic**, and the $PR(>F)$ column, which gives the corresponding **p-value**. Interpretation should always begin with the highest-order interaction term.

The three-way interaction, $C(\text{program}):C(\text{gender}):C(\text{division})$, tests whether the interaction between the program and gender changes depending on the division. Its p-value is approximately 0.6727. Since this value is significantly larger than the conventional alpha level of 0.05, we fail to reject the null hypothesis. We conclude that the three-way interaction is not statistically significant. Similarly, all three two-way interaction terms (``program:gender``, ``program:division``, and ``gender:division``) also show non-significant p-values (ranging from 0.4001 to 1.0000). The lack of significant interaction effects greatly simplifies the interpretation, as we can analyze the main effects independently.

We now focus on the main effects. We assess the statistical significance of each individual factor based on their respective p-values in the $PR(>F)$ column:

P-value of **program**: 2.9839×10^{-9}

P-value of **gender**: 1.7144×10^{-12}

P-value of **division**: 1.1852×10^{-6}

Analyzing the Significance of Main Factors

Since all three calculated p-values for the main effects are extremely small (approaching zero) and are far below the standard significance level ($\alpha = 0.05$), we must reject the null hypothesis for each factor. This leads to the robust conclusion that the training program, the player's gender, and their competitive division all exert independent, **statistically significant** influences on the observed increase in jumping height.

A significant main effect for the **program** indicates that the mean improvement in height differs between Program 1 and Program 2, irrespective of the player's gender or division. Similarly, the highly significant main effect for **gender** means that male players and female players achieve different average increases in jumping height, regardless of the program they followed or their division. Finally, the significant main effect for **division** demonstrates that Division I players, on average, experienced a different level of height improvement compared to Division II players, holding the other factors constant.

The primary takeaway from this analysis is twofold: all chosen factors are important predictors of the outcome, and since there are no significant interactions, the effect of one factor (e.g., gender)

does not depend on the level of another (e.g., program). This allows the researcher to address each variable's influence separately when formulating recommendations for training optimization.

Summary and Next Steps

In conclusion, the three-way ANOVA successfully identified that training program, gender, and division are all significant factors influencing jumping height improvement. The lack of significant interaction effects suggests that the benefit derived from a particular training program is consistent across genders and divisions. The researcher can thus confidently state that all three variables are important determinants of performance increase.

While the ANOVA confirms that differences exist, it does not specify which groups are statistically different from which others (e.g., whether Program 1 is definitively better than Program 2). To determine specific pairwise mean differences, the researcher would need to proceed with post-hoc tests, such as Tukey's Honestly Significant Difference (HSD) test, or by performing specific planned contrasts. These follow-up analyses would provide the necessary detail to move from a general conclusion of significance to actionable insights regarding the best training practices.

Further Reading on ANOVA Models in Python

For analysts looking to explore other variations of the Analysis of Variance framework, **statsmodels** supports a wide range of designs, including one-way, two-way, and Repeated Measures ANOVA. The following tutorials explain how to fit other ANOVA models in Python: