

# Learning Three-Way ANOVA with Python: A Step-by-Step Guide

Authored by  
**Mohammed loot**

October 28, 2025

## RECOMMENDED CITATION

Mohammed loot (2025). *Learning Three-Way ANOVA with Python: A Step-by-Step Guide*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=5071>

In the complex landscape of [statistical analysis](#), researchers often face the challenge of evaluating how multiple independent variables simultaneously influence a single outcome. When dealing with three categorical predictor variables, the appropriate and highly powerful technique is the [three-way ANOVA](#) (Analysis of Variance). This sophisticated method is designed to determine if there are [statistically significant](#) differences between the [means](#) of a continuous outcome variable, while simultaneously accounting for the unique contributions and combined effects of three independent [factors](#).

The three-way ANOVA represents a crucial expansion of simpler ANOVA models (one-way and two-way), providing a profound ability to investigate intricate relationships. It not only quantifies the individual impact of each factor but, more importantly, uncovers crucial [interaction effects](#). An interaction occurs when the effect of one factor on the response variable is dependent upon the level of another factor. This comprehensive modeling capability is indispensable for analysts seeking nuanced insights and a complete understanding of the underlying data mechanisms.

This authoritative guide is structured to lead you through the practical application of conducting a three-way ANOVA within the [Python](#) environment. We will utilize Python's specialized, robust statistical libraries to streamline the entire process, covering essential steps from meticulous data preparation and model execution to the rigorous interpretation of the resulting statistical output, ensuring you can confidently draw meaningful and evidence-based conclusions from your research.

## Deconstructing the Three-Way ANOVA Model

At its core, ANOVA serves as a fundamental statistical test aimed at comparing the means across three or more distinct groups. Where a one-way ANOVA limits the scope to a single factor and a two-way ANOVA considers two factors, the three-way ANOVA is specifically engineered for scenarios involving three [categorical variables](#) (the independent factors) and one continuous dependent variable. The overarching goal remains the assessment of mean differences in the dependent variable across all combinations of levels defined by the three independent factors, evaluating both their isolated impact and their joint influence.

When executing a three-way ANOVA, the analysis is partitioned into two primary categories of effects: the [main effects](#) and the interaction effects. A **main effect** isolates the influence of one independent factor on the dependent variable, averaging this effect across all levels of the other two factors. For instance, if 'dosage' is one factor, its main effect quantifies the overall difference in response attributed solely to variations in dosage, irrespective of 'age group' or 'gender'. Establishing significant main effects confirms that the levels of that factor inherently cause differences in the outcome.

Conversely, **interaction effects** are pivotal for understanding complexity, as they indicate that the

relationship between one factor and the dependent variable shifts based on the level of another factor. In a three-way design, we scrutinize three distinct types of interactions: three two-way interactions (e.g., Factor A x Factor B, Factor A x Factor C, Factor B x Factor C) and one critical three-way interaction (Factor A x Factor B x Factor C). A statistically significant three-way interaction suggests that the two-way relationship between two factors itself changes depending on the specific level of the third factor. Identifying and interpreting these interactions is crucial for attaining a truly complete and nuanced understanding of the data dynamics.

## Establishing the Python Environment for Statistical Modeling

To effectively conduct advanced statistical procedures, such as ANOVA, within Python, specialized, high-performance libraries are required. Our workflow relies primarily on two essential tools: [pandas](#), which excels at data manipulation and structuring, and [statsmodels](#), the foundational library providing the robust statistical modeling framework necessary for ANOVA calculations. Pandas facilitates the organization of raw data into the familiar, powerful DataFrame structure, which is optimal for tabular data processing. Statsmodels, in turn, offers the specific functions needed to define, fit, and evaluate complex statistical models.

Before initiating the analysis, it is imperative that these core libraries are installed within your Python environment. If they are not yet present, installation is straightforward using the pip package manager. Furthermore, the [NumPy](#) library is a prerequisite for many statistical operations, often bundled with pandas but essential for array handling and numerical efficiency:

```
pip install pandas
pip install statsmodels
pip install numpy
```

Once these dependencies are confirmed, you can proceed with importing the necessary modules and functions at the start of your script, ensuring a seamless and efficient transition into the data loading and analytical stages of the three-way ANOVA.

## Practical Case Study: Analyzing Athletic Performance Improvement

To provide a clear, relatable illustration of the three-way ANOVA, we will apply the technique to a realistic research scenario. Consider a sports science researcher investigating methods to maximize the vertical jumping height of university basketball players. The researcher is testing two distinct training protocols (Program 1 vs. Program 2) but recognizes that the effectiveness of these programs might not be uniform across all athletes; performance improvements are likely mediated by inherent demographic and classification variables.

The core hypothesis centers on two mediating variables: the athlete's **gender** (Male or Female)

and their athletic **division** classification (Division I or Division II). The researcher meticulously collects data on the observed improvement in jumping height (the continuous dependent variable) for athletes assigned to specific combinations of training program, gender, and division. The explicit objective of this study is to perform a three-way ANOVA to rigorously assess how the training program, gender, and division independently (main effects) and jointly (interaction effects) contribute to the observed variability in jumping height improvement.

This case study establishes a perfectly balanced experimental design ideal for this analysis: the dependent variable is continuous (height improvement), and we have three independent, categorical factors (Program, Gender, Division). By integrating these three factors into a single model, we move beyond simple comparisons, gaining the ability to discern whether, for instance, Program 1 is only superior for Division I males, or if its efficacy holds consistently across all athlete groups.

## Step 1: Structuring and Preparing Data in Python

The foundational step for any rigorous statistical analysis is the organization of data into a format suitable for modeling. In Python, the [pandas DataFrame](#) is the industry standard structure, facilitating easy management and manipulation of variables arranged in labeled columns.

For our case study, we construct a DataFrame containing four columns: `program` (coded as 1 or 2), `gender` (coded as 'M' or 'F'), `division` (coded as 1 or 2), and `height` (the measured jumping height improvement). We leverage the efficient array manipulation capabilities of the [NumPy](#) library to generate the factor levels. Specifically, we use `numpy.repeat` and `numpy.tile` to ensure the categorical factors are structured correctly for a balanced experimental design, guaranteeing an equal number of observations are assigned to each possible combination of conditions (cells).

The following Python code demonstrates the construction of this dataset. After the DataFrame is initialized, we display the initial rows to visually confirm that the structure and content accurately represent the experimental setup, preparing the data for the modeling phase:

```
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: Executing the Three-Way ANOVA in Python

With the data correctly structured in a pandas DataFrame, we proceed to execute the three-way ANOVA using the powerful tools provided by the `statsmodels` library. This process involves two key steps: first, specifying the [linear model](#) using the `ols()` (Ordinary Least Squares) function, and second, generating the ANOVA table using the `anova_lm()` function.

The model formula is the heart of this step, explicitly defining how the dependent variable (`height`) is predicted by the three independent factors (`program`, `gender`, `division`). We must include all possible terms: the three main effects, the three two-way interactions, and the single three-way interaction. Crucially, the `C()` wrapper around the factor names signals to `statsmodels` that these variables are nominal or categorical, forcing the library to treat them appropriately within the ANOVA framework rather than as continuous numerical variables.

After fitting the model using `ols().fit()`, the resulting object is passed to `sm.stats.anova_lm()`. We specify the argument `typ=2`, which calculates Type II sums of squares. Type II is the generally preferred method, particularly for designs involving interactions, because it tests the main effects after accounting for all other main effects and two-way interactions, providing a more robust measure of the unique contribution of each factor while respecting the hierarchy of the model:

```
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: Rigorous Interpretation of ANOVA Results

The output generated by the `anova_lm()` function provides the essential ANOVA summary table, which serves as the foundation for our statistical conclusions. Each row reports the results for a specific source of variation--a main effect or an interaction term--and includes key statistical metrics: `sum_sq` ([sum of squares](#)), `df` ([degrees of freedom](#)), `F` (the calculated [F-statistic](#)), and the highly critical `Pr(>F)`, which is the [p-value](#) associated with the test.

The p-value is the primary tool for hypothesis testing in ANOVA; it quantifies the probability of observing the data, or more extreme data, if the null hypothesis (that the means are equal, or the effect is zero) were true. We use a predetermined significance level, typically  $\alpha = 0.05$ . If a p-value falls below 0.05, we reject the null hypothesis, concluding that the factor or interaction term has a statistically significant effect on the dependent variable. Conversely, a p-value greater than 0.05 suggests insufficient evidence to reject the null hypothesis.

We begin the systematic interpretation by examining the interaction terms, as significant interactions often qualify the interpretation of main effects. Reviewing the output, we observe the following p-values for all interaction terms:

```

Three-way interaction (C(program):C(gender):C(division)): p-value = 6.726702e-01 (0.673)
Two-way interaction (C(program):C(gender)): p-value = 1.000000e+00 (1.000)
Two-way interaction (C(program):C(division)): p-value = 4.001069e-01 (0.400)
Two-way interaction (C(gender):C(division)): p-value = 6.726702e-01 (0.673)

```

Since all interaction p-values are considerably above the 0.05 threshold, we conclude that none of the two-way or the three-way interactions are statistically significant. This finding simplifies our

interpretation; it implies that the efficacy of the training program is consistent regardless of the athlete's gender or division, and similarly, the effects of gender and division are independent of each other.

We can now confidently interpret the main effects, knowing they are not overridden by complex interactions. Examining the main effects results:

P-value for **program**: 2.983934e-09 (a value extremely close to zero)

P-value for **gender**: 1.714432e-12 (an extremely small value)

P-value for **division**: 1.185218e-06 (a very small value)

As every one of these p-values is substantially smaller than 0.05, we decisively reject the null hypothesis for all three main effects. Our conclusion is that the training program used, the gender of the athlete, and their athletic division each exert a statistically significant, independent influence on the jumping height improvement. In summary, the three-way ANOVA reveals that differences in training protocols, inherent gender characteristics, and athletic classification divisions are all vital, non-interacting predictors of performance improvement in this study.

## Concluding Thoughts and Further Analytical Steps

Executing a three-way ANOVA in [Python](#) is a highly structured and efficient process, provided one masters the data setup and the proper application of the `statsmodels` library. This guide has furnished a comprehensive roadmap, enabling you to transition smoothly from initial data preparation to advanced statistical modeling and meaningful result interpretation, equipping you to apply this robust technique to multifactorial research questions across various domains.

While we found no significant interactions in this specific case, in many real-world datasets, interactions are present and require follow-up post-hoc analysis (e.g., simple main effects tests) to pinpoint where the significant differences lie. For researchers interested in broadening their statistical toolkit, the `statsmodels` library offers extensive capabilities beyond this specific model, including repeated measures ANOVA and mixed-effects models, which can handle even more complex experimental designs.

For continuing education in statistical modeling, exploring alternative ANOVA structures is highly recommended:

The following tutorials explain how to fit other ANOVA models in Python: