How to perform an ANCOVA in Python?

Authored by

stats writer

December 25, 2025

The Analysis of Covariance (ANCOVA) is a powerful statistical tool utilized in data science and research to combine features of both ANOVA (Analysis of Variance) and linear regression. In the Python ecosystem, this analysis is frequently executed using specialized statistical libraries like statsmodels or pingouin, which simplify the process of fitting complex linear models. These libraries allow researchers to model a continuous response variable while controlling for the influence of one or more continuous independent variables, known as covariates, alongside categorical factors. The output, typically accessed via a summary method, provides critical insights, including statistical metrics like **F-statistics**, model coefficients, and **p-values**, enabling a comprehensive evaluation of the relationships between variables.

Understanding the Analysis of Covariance (ANCOVA)

ANCOVA, or Analysis of Covariance, serves a vital purpose in experimental and quasi-experimental design: determining whether mean differences exist across three or more independent groups, but only after statistically accounting for the impact of extraneous continuous variables. This technique allows for a more precise assessment of the categorical factor's effect by removing variance that is attributable to the covariate. Essentially, it enhances the statistical power of the analysis compared to a standard ANOVA, particularly when the covariate significantly correlates with the dependent variable, allowing for a cleaner interpretation of group differences.

The fundamental mechanism of ANCOVA involves creating a linear model where the dependent variable is predicted by the categorical grouping variable (the factor) and the continuous covariate. By partitioning the variance in this manner, ANCOVA effectively 'adjusts' the group means. This adjustment simulates a scenario where all subjects possess the same average covariate score, thereby neutralizing the covariate's confounding influence. This rigorous approach ensures that any statistically significant findings are genuinely related to the factor being tested and not merely due to pre-existing differences captured by the covariate.

Choosing to use ANCOVA over a standard ANOVA is justified when researchers suspect that a continuous variable, which cannot be manipulated or controlled experimentally, might significantly influence the outcome measure. For example, if studying the effectiveness of different teaching methods (categorical factor) on test scores (dependent variable), the student's pre-test knowledge or baseline grade (covariate) would introduce unwanted noise. ANCOVA efficiently controls for this pre-existing knowledge, isolating the true effect of the teaching method itself.

The Critical Role of Covariates

In statistical modeling, a covariate is a continuous variable that is measured but not typically manipulated by the researcher, acting as a potential source of variation that needs to be controlled. The inclusion of a covariate in the ANCOVA model serves two primary, interrelated goals. First, it

significantly reduces the **error variance** (the unexplained variation) within the model, leading to tighter confidence intervals and more reliable estimates of the factor effects. Second, by accounting for the variance explained by the covariate, ANCOVA increases the statistical power to detect true differences between the experimental groups, assuming such differences genuinely exist.

For a covariate to be effective in ANCOVA, it must meet certain criteria. Crucially, the covariate should be measured reliably and should correlate significantly with the dependent variable. However, it should ideally be independent of the categorical factor variable (the groups). If the groups significantly differ on the covariate, this might suggest that the covariate is acting as a confounder, complicating the interpretation and potentially violating key ANCOVA assumptions, particularly the assumption of homogeneity of regression slopes, which must be carefully tested.

In practical Python implementations, identifying and correctly specifying the covariate is key to setting up the model. Libraries like Pingouin require the user to explicitly define which variable serves as the covariate (`covar`), which is the dependent variable (`dv`), and which variable defines the groups (`between`). Mislabeling these variables will lead to an incorrect model specification and ultimately, faulty statistical conclusions regarding the adjusted group means and the significance of the factor effects.

Prerequisites and Python Environment Setup

Performing ANCOVA in Python requires a foundational understanding of data handling using the **pandas** library and access to specialized statistical packages. While **statsmodels** offers extensive capabilities for generalized linear models, the **pingouin** library provides a highly user-friendly and streamlined function specifically designed for ANCOVA, which is often preferred for its simplicity in standard applications. Before proceeding with the analysis, ensure that the necessary libraries are installed within your Python environment.

The primary libraries needed include **numpy** for numerical operations, **pandas** for data structures and manipulation, and **pingouin** for the statistical test itself. If these packages are not already available, they must be installed using the Python package installer, `pip`. Although the installation command is typically run outside of a standard Python script, it is critical for setting up the environment:

pip install pandas
pip install numpy
pip install pingouin

Once the environment is properly configured, the subsequent steps focus on loading or generating

the data structure into a **pandas DataFrame**, which is the standard format for handling tabular data in Python analyses. This structure is essential because both **numpy** and **pingouin** functions are optimized to interact seamlessly with DataFrame objects, ensuring efficient data processing and model execution. The clarity and organization of the DataFrame directly impact the ability to correctly define the dependent variable, factor, and covariate in the ANCOVA function call.

Case Study: Investigating Study Techniques and Grades

To illustrate the application of ANCOVA, consider a pedagogical research scenario. A teacher is evaluating the efficacy of three distinct studying techniques (Technique A, B, and C) on final exam scores. However, the teacher recognizes that the students' existing knowledge level, reflected by their current grade in the class prior to the intervention, significantly influences the exam score. Therefore, to accurately assess the unique impact of the studying techniques, the teacher must control for this pre-existing academic status.

The goal of the ANCOVA in this context is to determine if there are statistically significant differences in the mean exam scores across the three study technique groups **after** statistically adjusting for the students' initial current grades. If a significant effect is found for the factor (studying technique), it implies that one or more techniques are truly superior, irrespective of the student's starting point.

The variables in this specific study are precisely defined according to the ANCOVA requirements:

Factor variable (Categorical Independent Variable): The qualitative grouping variable, which is the **studying technique** (A, B, or C). This is the primary variable of interest whose effect we seek to measure.
Covariate (Continuous Variable): The variable used for statistical control, which is the student's **current grade** in the class. This variable accounts for extraneous variance.
Response variable (Dependent Variable): The outcome measure being analyzed, which is the student's **exam score**. This must be a continuous variable.

By using this structure, the analysis seeks to isolate the effect of the instructional intervention (the technique) from the inherent variability introduced by the students' baseline performance (the current grade).

Step 1: Data Preparation using Pandas

The initial and most fundamental step in any Python statistical analysis is the preparation and structuring of the data. For ANCOVA, the data must be organized into a format suitable for the chosen statistical library, which typically means using a **pandas DataFrame**. This DataFrame must contain three essential columns corresponding to the factor variable, the covariate, and the

response variable.

We begin by importing the necessary libraries, **numpy** for efficient array generation and **pandas** for DataFrame creation. The sample data is constructed by repeating the categorical levels ('A', 'B', 'C') for the `technique` variable and assigning corresponding values for `current_grade` and `exam_score`. Ensuring that the data is correctly paired--each exam score and current grade corresponds to a specific study technique--is crucial for the integrity of the subsequent statistical model.

import numpy as np
import pandas as pd

#create data
df = pd.DataFrame({'technique': np.repeat(, 5),
'current_grade': ,
'exam_score': })
#view data
df

technique current_grade exam_score
0 A 67 77
1 A 88 89
2 A 75 72
3 A 77 74
4 A 85 69
5 B 92 78
6 B 69 88
7 B 77 93
8 B 74 94
9 B 88 90
10 C 96 85
11 C 91 81
12 C 88 83
13 C 82 88
14 C 80 79

Examining the output of the DataFrame confirms that we have 15 observations distributed evenly across the three techniques (five per group). This structure is now ready for input into the **pingouin** library's ANCOVA function. It is important to remember that data accuracy and proper labeling in this stage directly translate to the validity of the statistical inferences drawn later.

Step 2: Executing the ANCOVA Model with Pingouin

With the data correctly structured in the pandas DataFrame, the next step involves utilizing the specialized `ancova()` function provided by the **pingouin** library. This library is designed to simplify common statistical tests in Python, abstracting much of the complexity inherent in lower-level libraries. If **pingouin** was not previously installed, the command to install it must be run first, as shown in the prerequisites section.

The `ancova()` function requires four primary arguments for proper execution: the DataFrame itself (`data`), the name of the dependent variable (`dv`), the name of the covariate (`covar`), and the name of the categorical factor variable (`between`). Explicitly defining these roles ensures that the model correctly partitions the variance of the dependent variable (`exam_score`) into components explained by the factor (`technique`), the covariate (`current_grade`), and the unexplained residual error.

from pingouin import ancova

#perform ANCOVA
ancova(data=df, dv='exam_score', covar='current_grade', between='technique')

Source SS DF F p-unc np2
0 technique 390.575130 2 4.80997 0.03155 0.46653
1 current_grade 4.193886 1 0.10329 0.75393 0.00930
2 Residual 446.606114 11 NaN NaN NaN

The output is a clear Analysis of Variance table, which summarizes the results for each specified source of variation (`technique` and `current_grade`) as well as the `Residual` error. This table provides all the critical statistics required for hypothesis testing and interpretation, including the Sum of Squares (SS), Degrees of Freedom (DF), the F-statistic, the uncorrected p-value (p-unc), and the partial eta-squared (np2).

Step 3: Interpreting the ANCOVA Output

The final step involves rigorously interpreting the statistical results provided by the ANCOVA output table. The primary focus of interpretation is typically the row corresponding to the categorical factor variable--in this case, `technique`--and the associated **p-unc** value, which indicates the probability of observing the data (or more extreme data) if the null hypothesis were true.

The null hypothesis in this ANCOVA states that the mean exam scores across the three studying techniques are equal, even after adjusting for the students' current grades. To evaluate this, we examine the output for the `technique` factor: the F-statistic is 4.80997, associated with 2 degrees

of freedom (DF) in the numerator and 11 degrees of freedom for the residual error. The crucial metric is the uncorrected **p-value** (p-unc), which is reported as **0.03155**.

By convention, researchers often use an alpha level (significance threshold) of 0.05. Since the calculated p-value (0.03155) is less than 0.05, we possess sufficient evidence to reject the null hypothesis. This rejection signifies that there is a statistically significant difference in the mean exam scores between the studying techniques, even after the variance attributed to the students' pre-existing current grades has been statistically controlled. This conclusion strongly supports the teacher's hypothesis that the study technique itself has a discernible impact on performance.

It is also insightful to examine the results for the covariate, `current_grade`. In this example, the covariate has an F-statistic of 0.10329 and a p-value of 0.75393. Since this p-value is significantly greater than 0.05, the relationship between the current grade and the exam score, once the technique differences are accounted for, is not statistically significant. This outcome is somewhat unusual in applied ANCOVA, as covariates are typically selected precisely because they are expected to significantly correlate with the dependent variable. If the covariate's effect is non-significant, the analysis still remains valid, but the benefit gained from controlling for that specific covariate is minimal.

Understanding Key ANCOVA Assumptions

While the execution of ANCOVA in Python is straightforward, the validity of the results hinges on satisfying several underlying statistical assumptions. Failing to meet these assumptions can lead to biased estimates and inaccurate hypothesis testing. Therefore, an expert analysis must always include diagnostic checks for these core requirements:

Normality of Residuals: The residuals (the differences between the observed and predicted values) for the dependent variable must be approximately normally distributed within each group. This can be checked using Q-Q plots or formal tests like the Shapiro-Wilk test on the residual values extracted from the model.
Homogeneity of Variances (Homoscedasticity): The variance of the residuals should be equal across all factor groups. This is typically assessed using **Levene's test** or **Bartlett's test**. Violation of this assumption is particularly problematic when group sizes are unequal.
Linearity of the Relationship: ANCOVA assumes that the relationship between the dependent variable (exam score) and the covariate (current grade) is linear within each of the factor groups. This assumption can be visually confirmed by plotting the dependent variable against the covariate separately for each group.
Homogeneity of Regression Slopes: This is arguably the most critical and unique assumption of ANCOVA. It mandates that the relationship (the slope of the regression line) between the covariate and the dependent variable must be the same across all levels of the factor variable. If this

assumption is violated (indicating an interaction effect between the factor and the covariate), ANCOVA is inappropriate, and a more complex **two-way ANOVA with interaction** or moderation analysis should be employed instead.

In a real-world scenario, the implementation of ANCOVA in Python would involve preliminary steps to test these assumptions using functions available in **pingouin** or **statsmodels**. For instance, testing for the homogeneity of regression slopes often involves adding an interaction term (Factor x Covariate) to the linear model and checking its significance. Only if all these assumptions are reasonably met can the interpretation of the F-statistic and p-value be considered robust and reliable.