

# Learn How to Perform a Two-Sample T-Test in Python

Authored by  
**Mohammed loot**

November 8, 2025

## RECOMMENDED CITATION

Mohammed loot (2025). *Learn How to Perform a Two-Sample T-Test in Python*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=12696>

The [two-sample t-test](#) stands as a cornerstone of [statistical hypothesis testing](#), providing researchers with a rigorous method to assess whether the difference observed between two sample averages is statistically reliable or simply the result of random variation. This essential inferential procedure is specifically designed to determine if a significant difference exists between the [means of two independent populations](#). Its application spans diverse scientific disciplines, including biology, finance, and social sciences, whenever a comparison of two distinct groups based on a continuous metric is required.

Before one can confidently proceed with the analysis, it is imperative to confirm that the underlying assumptions of the test are reasonably met. These critical assumptions typically include the independence of data points, random sampling from the respective populations, and an approximate normal distribution of the data. Furthermore, a crucial preliminary decision involves assessing the population variances: whether they are assumed equal (leading to the standard independent two-sample t-test) or unequal (necessitating the use of [Welch's t-test](#)). This comprehensive guide walks you through the entire process, demonstrating how to execute a two-sample t-test efficiently and accurately within the highly capable statistical environment offered by the Python ecosystem.

Mastering this technique in Python allows data professionals and researchers to draw robust, evidence-based conclusions from comparative experimental data. We will utilize key libraries like [NumPy](#) for data handling and [SciPy](#) for the statistical heavy lifting, ensuring a clear, reproducible methodology from data setup to final interpretation.

## Case Study: Comparing Plant Heights (Species A vs. Species B)

To illustrate the practical application of the two-sample t-test, let us examine a common scenario in agricultural research. Imagine a team of scientists investigating whether two genetically distinct species of plants, designated Species A and Species B, exhibit the same average maximum height. The core research question hinges on determining if any observed height differences are statistically significant enough to warrant different cultivation techniques or further genetic exploration.

To gather reliable evidence, the researchers implemented a meticulously structured sampling plan, collecting a [simple random sample](#) of 20 plants from the population of Species A and 20 plants from the population of Species B. These resulting measurements constitute the raw data required for the statistical comparison. The goal is to move beyond the sample observations and make an inference about the true population parameters ( $\mu_A$  and  $\mu_B$ ).

The subsequent steps will detail the procedure for applying the principles of [hypothesis testing](#). We aim to conclusively determine if the disparities found in the sample means are substantial enough to reject the idea that the two species are, on average, the same height. This robust procedure

ensures the validity and reliability of our conclusions concerning the population means.

## Step 1: Setting Up the Python Environment and Data Structures

The first essential phase of any rigorous statistical analysis involves preparing the necessary computational tools and structuring the raw data appropriately. Python's scientific stack provides unparalleled resources for this task. We rely on the [NumPy](#) library, which is the foundational standard for numerical computation and array manipulation, to efficiently store and manage our plant height measurements.

For the two-sample comparison, the data must be clearly segmented into two distinct [NumPy](#) arrays, corresponding precisely to the observations for Species A (Group 1) and Species B (Group 2). This clear structure is vital because the core statistical function, `ttest_ind()`, requires these two separate inputs to accurately calculate the test statistics based on the observed differences between the group means and their respective variances. The measurements below represent the heights (in centimeters) recorded for the 20 sampled plants of each species.

### **import numpy as np**

```
group1 = np.array()  
group2 = np.array()
```

With the height data successfully loaded into these structured arrays, the initial setup is complete. However, before proceeding directly to the calculation of the t-statistic, we must perform a crucial pre-test analysis to evaluate the variability within each group, as this determination directly influences the statistical formula we must employ.

## Pre-Analysis: Evaluating Variance Homogeneity

A key methodological step when preparing for a two-sample t-test is assessing whether the population variances of the two groups can be reasonably assumed to be equal--a property known as [homoscedasticity](#). If we assume equal variances, we pool the variance estimates, leading to the standard t-test. Conversely, if the variances are unequal, the standard test becomes less reliable, requiring us to opt for [Welch's t-test](#), which adjusts for this disparity and is generally more robust.

While formal statistical tests, such as Levene's test or Bartlett's test, offer precise methods for variance comparison, a common and highly practical rule of thumb is to calculate the ratio of the larger sample variance to the smaller sample variance. If this ratio remains below a threshold, typically 4:1, researchers often proceed with the assumption of equal population variances. If the ratio substantially exceeds 4, it strongly suggests that the variances are different, making [Welch's t-test](#) the appropriate choice.

We begin by calculating the variance for each sample using the robust variance function provided by [NumPy](#). This calculation generates the necessary metrics for applying the 4:1 rule and making our methodological decision:

```
#find variance for each group  
print(np.var(group1), np.var(group2))
```

```
7.73 12.26
```

With the calculated sample variances (Group 1 variance = 7.73; Group 2 variance = 12.26), we can determine the ratio:  $12.26 \div 7.73 = 1.586$ . Since this ratio (**1.586**) is significantly less than the established threshold of 4, we can confidently assume that the population variances are equal. This crucial finding dictates that we will use the standard two-sample independent t-test, ensuring the `equal_var` parameter is set to `True` during the execution phase.

## Step 2: Executing the Two-Sample T-Test using SciPy

The core computational step involves calculating the [t-statistic](#) and the corresponding [p-value](#). For this, we leverage the `ttest_ind()` function, which is a key component of the `scipy.stats` module within the [SciPy](#) library. [SciPy](#) is the bedrock library for scientific and technical computing in Python, offering specialized functionality for statistics, optimization, and integration.

The function's syntax is highly intuitive, requiring only the two data arrays and a specification regarding our variance assumption, which we determined in the previous step:

```
ttest_ind(a, b, equal_var=True)
```

The key parameters govern the function's behavior:

**a:** The [NumPy](#) array containing the sample observations for the first group (Species A).

**b:** The [NumPy](#) array containing the sample observations for the second group (Species B).

**equal\_var:** A Boolean flag. When set to **True** (as determined by our variance homogeneity test), the function executes the standard two-sample independent t-test. Setting it to **False** would execute [Welch's t-test](#), which is necessary when unequal population variances are suspected.

Following our pre-test analysis, we perform the standard t-test by setting `equal_var=True`:

```
import scipy.stats as stats
```

```
#perform two sample t-test with equal variances
```

```
stats.ttest_ind(a=group1, b=group2, equal_var=True)
```

```
(statistic=-0.6337, pvalue=0.53005)
```

The output delivers the two essential metrics needed for inference: the calculated [t-statistic](#) (**-0.6337**) and the corresponding two-sided [p-value](#) (**0.53005**). The [t-statistic](#) measures the difference between the sample means relative to the variation within the samples, while the [p-value](#) quantifies the probability of observing such a difference (or a more extreme one) if the [null hypothesis](#) were true.

### Step 3: Interpreting the Statistical Outcomes and Drawing Conclusions

The final and most crucial stage involves interpreting the numerical output within the context of the original research question. Since we performed a two-sided test--meaning we are testing for a difference in either direction--the formal statistical hypotheses are structured as follows:

**H<sub>0</sub> (The Null Hypothesis):**  $\mu_1 = \mu_2$ . This hypothesis states there is no true difference; the mean heights of the two plant populations are equal.

**H<sub>A</sub> (The Alternative Hypothesis):**  $\mu_1 \neq \mu_2$ . This hypothesis states that a statistically significant difference exists between the mean heights of the two plant populations.

To determine whether we reject or fail to reject the [null hypothesis](#), we compare the calculated [p-value](#) against a predetermined significance level, denoted as  $\alpha$  (alpha). For nearly all scientific endeavors,  $\alpha$  is conventionally set at 0.05. The decision rule is straightforward: if the [p-value](#) is less than  $\alpha$  (0.05), we reject the [null hypothesis](#). If the [p-value](#) is greater than  $\alpha$ , we fail to reject the [null hypothesis](#).

In this specific analysis, the calculated p-value is **0.53005**. Given that 0.53005 is substantially greater than our chosen significance level of  $\alpha = 0.05$ , the conclusion is to **fail to reject the null hypothesis**. Statistically speaking, the observed variance in the sample means between Species A and Species B is likely attributable to random sampling variability rather than a genuine, underlying difference in the population means. We conclude that there is insufficient statistical evidence, at the 5% significance level, to support the claim that the mean heights of the two plant populations are different.

### Expanding Your Statistical Toolkit in Python

The two-sample t-test is but one crucial technique in the wider field of inferential statistics. For those seeking to broaden their understanding of hypothesis testing or to address different experimental designs using the power of Python's scientific libraries, exploring related t-tests is

highly recommended. These resources offer valuable guidance on adjusting your methodology for different data structures and research questions:

[How to Conduct a One Sample T-Test in Python](#)

[How to Conduct a Paired Samples T-Test in Python](#)