How to Easily Calculate Sample and Population Variance in Python

Authored by

stats writer

December 3, 2025

In the realm of data analysis and statistics, understanding the spread or dispersion of data points is fundamental. One of the most critical measures used to quantify this dispersion is the **variance**. Calculating both sample and population variance accurately is a necessary task for data scientists and analysts, and the **Python** programming language offers streamlined tools to perform these calculations efficiently.

Specifically, the built-in **statistics module** provides specialized functions--`variance()` for sample variance and `pvariance()` for population variance. These functions accept a dataset, typically supplied as a list or iterable, and return the corresponding measure of spread. While one theoretically could calculate variance for a single data point (resulting in zero variance), these functions are designed to calculate the standard measure of dispersion across an entire dataset, providing a comprehensive view of variability.

Understanding Variance in Data Science

The **variance** is defined as the average of the squared differences from the Mean. It serves as a crucial metric for evaluating how far a set of numbers is spread out from their average value. A high variance indicates that the data points are very spread out from the mean, while a low variance suggests that the data points tend to be clustered closely around the mean. This metric is used extensively in fields like finance, quality control, and hypothesis testing.

Before diving into the Python implementation, it is essential to distinguish between the two primary types of variance calculation: population variance and sample variance. The distinction lies entirely in whether the dataset represents every single observation of interest (the population) or only a subset of those observations (the sample).

The Mathematical Foundation: Defining Variance

Although modern programming languages like **Python** simplify the computation, understanding the underlying mathematical formulas provides critical insight into how these measures differ and why the results can vary slightly between a population calculation and a sample calculation.

The mathematical approach involves summing the squared deviations of each data point from the mean and then dividing that sum by either the population size (N) or the sample size minus one (n-1).

Calculating Population Variance: The Formula and Interpretation

When calculating the **population variance**, we assume that the dataset contains every member of the group being studied. Population variance is symbolized by σ^2 (sigma squared).

$\sigma^2 = \Sigma (xi - \mu)2 / N$

In this formula, the variables represent the following statistical components:

Σ: The summation symbol, indicating the total sum of the values.
μ: The **Population mean**, representing the average value of the entire dataset.
xi: The ith element (or individual observation) from the total population.
N: The total **Population size** (the count of observations).

This formula provides an exact measure of the dispersion, as all relevant data points are included in the calculation.

Calculating Sample Variance: Addressing Degrees of Freedom

The **sample variance** is used when we only have a subset of data taken from a larger population. Since a sample is inherently an estimate of the true population, the calculation requires a slight adjustment to provide an unbiased estimate of the true population variance. Sample variance is symbolized by **s^2**.

s2 = Σ (xi - x)2 / (n-1)

The components of the sample variance formula are defined as follows:

x: The **Sample mean**, the average calculated from the sample dataset.
xi: The ith element (observation) taken from the sample.
n: The **Sample size** (the count of observations in the sample).

The critical difference is the denominator, `(n-1)`, which represents the degrees of freedom. Dividing by `(n-1)` rather than `n` corrects for the fact that the sample mean (x) is used as an estimate, ensuring the sample variance is an unbiased estimator of the population variance.

Python's `statistics` Module: Tools for Variance Calculation

The **statistics module** is a standard library in **Python** that makes calculating these measures straightforward. It abstracts the complex formulas, allowing developers to quickly derive variance measures for any iterable dataset. We utilize the `variance` function for sample variance and the `pvariance` function for population variance. These functions are highly efficient and reliable for standard statistical calculations.

Below is a quick demonstration of importing these functions and how they are typically called:

from statistics import variance, pvariance

```
# Assume 'x' is a predefined list of numerical data

# calculate sample variance
variance(x)

# calculate population variance
pvariance(x)
```

The following detailed examples illustrate the practical application of these two functions using a consistent dataset.

Practical Implementation: Calculating Sample Variance in Python

To calculate the sample variance, we must import the `variance` function from the **statistics module**. This function automatically applies the `(n-1)` degrees of freedom correction, providing the unbiased estimate required for sample data. We will define a list of data points and then pass this list directly to the function.

This approach is suitable whenever your data is a small subset intended to represent a much larger, inaccessible group. For instance, if these numbers represent the scores of 15 students selected randomly from an entire university, the sample variance calculation would be appropriate.

```
from statistics import variance

# define data
data =

# calculate sample variance
variance(data)
```

22.067

Upon execution, the sample variance for this specific dataset is found to be **22.067**. This high value indicates a moderate spread of scores around the sample mean.

Practical Implementation: Calculating Population Variance in Python

Conversely, if we assume the exact same list of 15 scores constitutes the entire universe of data we care about--the definitive population--we must use the `pvariance` function. This function calculates the **population variance** by dividing the sum of squared differences by the total count of observations (N), without applying the degrees of freedom adjustment.

For example, if the data represents the entire class enrollment of a small seminar (and we are only concerned with that seminar), then this calculation is mathematically accurate for describing the dispersion within that specific group.

from statistics import pvariance

define data
data =

# calculate population variance
pvariance(data)

20.596

By treating the data as a population, the calculated variance value is **20.596**. As expected, this result is slightly lower than the sample variance calculated previously (22.067).

Sample vs. Population: Choosing the Right Calculation

The decision between `variance()` and `pvariance()` is perhaps the most crucial step in accurately analyzing statistical dispersion. Selecting the wrong function can lead to biased or misinformed conclusions. Keep these fundamental guidelines in mind when approaching any statistical calculation involving variance:

You should calculate the **population variance** when the dataset you're working with represents an entire population, meaning every value that you are interested in is included in your data structure.
You should calculate the **sample variance** when the dataset represents a subset (a sample) taken from a larger population of interest, and your goal is to infer characteristics about that larger population.
It is a fundamental statistical principle that the sample variance for a given array of data will always be larger than the population variance for the same array of data. This is due to the inherent uncertainty when estimating population parameters from a sample, which is corrected by dividing by `(n-1)`, resulting in a larger estimated **variance**.

Conclusion: Further Measures of Spread in Python

Calculating **variance** in **Python** using the **statistics module** is highly efficient and necessary for characterizing data dispersion. However, variance is only one of many measures of spread. Analysts often rely on standard deviation (the square root of variance), range, and interquartile range to gain a complete picture of data distribution.

For those looking to deepen their understanding of data variability, exploring tutorials on standard deviation (calculated using `stdev()` and `pstdev()` in the same library) is the next logical step, as standard deviation is generally preferred for interpretation due to being in the original units of the data.

The following tutorials explain how to calculate other measures of spread in Python: