How can the Coefficient of Variation be calculated using Python?

Authored by

stats writer

April 24, 2024

The Coefficient of Variation is a statistical measure that is used to evaluate the variability of a dataset relative to its mean. It is typically calculated as the standard deviation divided by the mean, and is expressed as a percentage. In Python, the Coefficient of Variation can be calculated by first importing the necessary libraries such as numpy and statistics. The dataset can then be inputted and the mean and standard deviation can be calculated using built-in functions. Finally, the Coefficient of Variation can be computed by dividing the standard deviation by the mean and multiplying by 100. This process can be easily implemented in Python, making it a convenient tool for calculating the Coefficient of Variation for various datasets.

Calculate the Coefficient of Variation in Python

A coefficient of variation, often abbreviated as *CV*, is a way to measure how spread out values are in a dataset relative to the mean. It is calculated as:

CV = σ / μ

where:

σ: The standard deviation of datasetμ: The mean of dataset

In plain English, the coefficient of variation is simply the ratio between the standard deviation and the mean.

When to Use the Coefficient of Variation

The coefficient of variation is often used to compare the variation between two different datasets.

In the real world, it's often used in finance to compare the mean expected return of an investment relative to the expected standard deviation of the investment. This allows investors to compare the risk-return trade-off between investments.

For example, suppose an investor is considering investing in the following two mutual funds:

Mutual Fund A: mean = 9%, standard deviation = 12.4%

Mutual Fund B: mean = 5%, standard deviation = 8.2%

Upon calculating the coefficient of variation for each fund, the investor finds:

CV for Mutual Fund A = 12.4% /9% = 1.38

CV for Mutual Fund B = 8.2% / 5% = 1.64

Since Mutual Fund A has a lower coefficient of variation, it offers a better mean return relative to the standard deviation.

How to Calculate the Coefficient of Variation in Python

To calculate the coefficient of variation for a dataset in

Python, you can use the following syntax:

```
import numpy as np

cv = lambda x: np.std(x, ddof=1) / np.mean(x) * 100
```

The following examples show how to use this syntax in practice.

Example 1: Coefficient of Variation for a Single Array

The following code shows how to calculate CV for a single array:

```
#create vector of data
data =

#define function to calculate cv
cv = lambda x: np.std(x, ddof=1) / np.mean(x) * 100
#calculate CV
cv(data)

9.234518
```

The coefficient of variation turns out to be 9.23.

Example 2: Coefficient of Variation for Several Vectors

The following code shows how to calculate the CV for several columns in a pandas DataFrame:

```
import numpy as npimport pandas as pd

#define function to calculate cv
cv = lambda x: np.std(x, ddof=1) / np.mean(x) * 100

#create pandas DataFrame
df = pd.DataFrame({'a': ,
'b': ,
'c': })

#calculate CV for each column in data frame
df.apply(cv)
```

```
a 11.012892
b 8.330843
c 7.154009
dtype: float64
```

Note that missing values will simply be ignored when calculating the coefficient of variation:

```
import numpy as npimport pandas as pd

#define function to calculate cv
cv = lambda x: np.std(x, ddof=1) / np.mean(x) * 100

#create pandas DataFrame
df = pd.DataFrame({'a': ,
'b': ,
'c': })

#calculate CV for each column in data frame
df.apply(cv)

a 11.012892
b 8.497612
c 5.860924
dtype: float64
```