

How to Calculate Correlation in Python Using Pandas and NumPy

Authored by
stats writer

March 15, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Calculate Correlation in Python Using Pandas and NumPy*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=136066>

Calculating correlation in Python involves using the statistical method of measuring the relationship between two variables. This can be done by using the "corr" function from the "pandas" library or the "corrcoef" function from the "numpy" library. These functions will return a correlation coefficient value that indicates the strength and direction of the relationship between the two variables. Additionally, visual representations such as scatter plots or heatmaps can also be used to better understand the correlation between the variables. By using these methods, one can easily determine the level of correlation between two variables in a given dataset.

Calculate Correlation in Python

One way to quantify the relationship between two variables is to use the , which is a measure of the linear association between two variables. It always takes on a value between -1 and 1 where:

**-1 indicates a perfectly negative linear correlation between two variables
0 indicates no linear correlation between two variables
1 indicates a perfectly positive linear correlation between two variables**

The further away the correlation coefficient is from zero, the stronger the relationship between the two variables.

This tutorial explains how to calculate the correlation between variables in Python.

How to Calculate Correlation in Python

To calculate the correlation between two variables in

Python, we can use the Numpy corrcoef() function.

```
import numpy as np
```

```
np.random.seed(100)
```

```
#create array of 50 random integers between 0 and 10
```

```
var1 = np.random.randint(0, 10, 50)
```

```
#create a positively correlated array with some random noise
```

```
var2 = var1 + np.random.normal(0, 10, 50)
```

```
#calculate the correlation between the two arrays
```

```
np.corrcoef(var1, var2)
```

```
]
```

We can see that the correlation coefficient between these two variables is 0.335, which is a positive correlation.

By default, this function produces a matrix of correlation coefficients. If we only wanted to return the correlation coefficient between the two variables, we could use the following syntax:

```
np.corrcoef(var1, var2)
```

```
0.335
```

To test if this correlation is statistically significant, we can calculate the p-value associated with the Pearson correlation coefficient by using the Scipy `pearsonr()` function, which returns the Pearson correlation coefficient along with the two-tailed p-value.

```
from scipy.stats.stats import pearsonr
```

```
pearsonr(var1, var2)
```

```
(0.335, 0.017398)
```

The correlation coefficient is 0.335 and the two-tailed p-value is .017. Since this p-value is less than .05, we would conclude that there is a statistically significant correlation between the two variables.

If you're interested in calculating the correlation between several variables in a Pandas DataFrame, you can simply use the `.corr()` function.

```
import pandas as pd
```

```
data = pd.DataFrame(np.random.randint(0, 10, size=(5, 3)), columns=)
```

```
data
```

```
A B C
```

```
0 8 0 9
```

```
1 4 0 7
```

```
2 9 6 8
```

```
3 1 8 1
```

```
4 8 0 8
```

```
#calculate correlation coefficients for all pairwise combinations
```

```
data.corr()
```

```
A B C
```

```
A 1.000000 -0.775567 -0.493769
```

```
B -0.775567 1.000000 0.000000
```

```
C -0.493769 0.000000 1.000000
```

And if you're only interested in calculating the correlation between two specific variables in the DataFrame, you can specify the variables:

```
data.corr(data)
```

```
-0.775567
```

The following tutorials explain how to perform other common tasks in Python:

ARABPSYCHOLOGY.COM