How to Calculate Cross Correlation in Python

BY ZACH BOBBITT MARCH 26, 2021

Cross correlation is a way to measure the degree of similarity between a time series and a lagged version of another time series.

This type of correlation is useful to calculate because it can tell us if the values of one time series are predictive of the future values of another time series. In other words, it can tell us if one time series is a leading indicator for another time series.

This type of correlation is used in many different fields, including:

Business: Marketing spend is often considered to be a leading indicator for future revenue of businesses. For example, if a business spends an abnormally high amount of money on marketing during one quarter, then total revenue is expected to be high *x* quarters later.

Economics: The consumer confidence index (CCI) is considered to be a leading indicator for the gross domestic product (GDP) of a country. For example, if CCI is high during a given month, the GDP is likely to be higher *x* months later.

The following example shows how to calculate the cross correlation between two time series in Python.

Example: How to Calculate Cross Correlation in Python

Suppose we have the following time series in Python that show the total marketing spend (in thousands) for a certain company along with the the total revenue (in thousands) during 12 consecutive months:

```
import numpy as np

#define data
marketing = np.array([3, 4, 5, 5, 7, 9, 13, 15, 12,
revenue = np.array([21, 19, 22, 24, 25, 29, 30, 34,
```

We can calculate the cross correlation for every lag between the two time series by using the **ccf()** function from the statsmodels package as follows:

```
import statsmodels.api as sm

#calculate cross correlation
sm.tsa.stattools.ccf(marketing, revenue, adjusted=Fa

array([ 0.77109358,  0.46238654,  0.19352232, -0.060
       -0.44531104, -0.49159463, -0.35783655, -0.150
        0.01587722,  0.0070399 ])
```

Here's how to interpret this output:

- The cross correlation at lag 0 is **0.771**.
- The cross correlation at lag 1 is **0.462**.

- The cross correlation at lag 2 is **0.194**.
- The cross correlation at lag 3 is **-0.061**.

And so on.

Notice that the correlation between the two time series becomes less and less positive as the number of lags increases. This tells us that marketing spend during a given month is quite predictive of revenue one or two months later, but not predictive of revenue beyond more than two months.

This intuitively makes sense – we would expect that high marketing spend during a given month is predictive of increased revenue during the next two months, but not necessarily predictive of revenue several months into the future.

Additional Resources

How to Calculate Autocorrelation in Python
How to Calculate Partial Correlation in Python
How to Calculate Point-Biserial Correlation in Python

POSTED IN PROGRAMMING •

Zach Bobbitt

Hey there. My name is Zach Bobbitt. I have a Masters of Science degree in Applied Statistics and I've worked on machine learning algorithms for professional businesses in both healthcare and retail. I'm passionate about statistics, machine learning, and