

# How can I calculate rolling correlation in Pandas? Can you provide some examples?

Authored by  
**stats writer**

April 21, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I calculate rolling correlation in Pandas? Can you provide some examples?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=137819>

To calculate rolling correlation in Pandas, the user can utilize the "rolling" function along with the "corr" function to compute the correlation between two time series data. This method allows for the calculation of correlations over a specified window or period, providing a more dynamic and accurate representation of the relationship between variables. Some examples of using this method would be to analyze the correlation between stock prices and market trends over a certain time frame, or to examine the relationship between temperature and sales data over a specific period. By utilizing Pandas' rolling correlation function, users can gain valuable insights into the changing dynamics between variables over time.

## Calculate Rolling Correlation in Pandas (With Examples)

**Rolling correlations are correlations between two time series on a rolling window. One benefit of this type of correlation is that you can visualize the correlation between two time series over time.**

**This tutorial explains how to calculate and visualize rolling correlations for a pandas DataFrame in Python.**

**How to Calculate Rolling Correlations in Pandas**

**Suppose we have the following data frame that display the total number of products sold for two different products (x and y) during a 15-month period:**

```
import pandas as pd  
import numpy as np
```

```
#create DataFrame
```

```
df = pd.DataFrame({'month': np.arange(1, 16),  
'x': ,  
'y': })
```

```
#view first six rows
```

```
df.head()
```

```
month x y
```

```
1 1 13 22
```

```
2 2 15 24
```

```
3 3 16 23
```

```
4 4 15 27
```

```
5 5 17 26
```

```
6 6 20 26
```

To calculate a rolling correlation in pandas, we can use the rolling.corr() function.

This function uses the following syntax:

```
df.rolling(width).corr(df)
```

where:

**df:** Name of the data frame  
**width:** Integer specifying the window width for the rolling correlation  
**x, y:** The two

**column names to calculate the rolling correlation between**

**Here's how to use this function to calculate the 3-month rolling correlation in sales between product x and product y:**

**#calculate 3-month rolling correlation between sales for x and y**

```
df.rolling(3).corr(df)
```

**0 NaN**

**1 NaN**

**2 0.654654**

**3 -0.693375**

**4 -0.240192**

**5 -0.802955**

**6 0.802955**

**7 0.960769**

**8 0.981981**

**9 0.654654**

**10 0.882498**

**11 0.817057**

**12 -0.944911**

**13 -0.327327**

**14 -0.188982**

**dtype: float64**

**This function returns the correlation between the two product sales for the previous 3 months. For example:**

**The correlation in sales during months 1 through 3 was 0.654654. The correlation in sales during months 2 through 4 was -0.693375. The correlation in sales during months 3 through 5 was -0.240192.**

**And so on.**

**We can easily adjust this formula to calculate the rolling correlation for a different time period. For example, the following code shows how to calculate the 6-month rolling correlation in sales between the two products:**

```
#calculate 6-month rolling correlation between sales for  
x and y
```

```
df.rolling(6).corr(df)
```

```
0 NaN
```

```
1 NaN
```

```
2 NaN
```

```
3 NaN
```

```
4 NaN
5 0.558742
6 0.485855
7 0.693103
8 0.756476
9 0.895929
10 0.906772
11 0.715542
12 0.717374
13 0.768447
14 0.454148
dtype: float64
```

The correlation in sales during months 1 through 6 was 0.558742. The correlation in sales during months 2 through 7 was 0.485855. The correlation in sales during months 3 through 8 was 0.693103.

And so on.

#### Notes

Here are a few notes for the functions used in these examples:

**The width (i.e. the rolling window) should be 3 or greater in order to calculate correlations. You can find the full documentation for the `rolling.corr()` function [here](#).**

**[How to Calculate Rolling Correlation in R](#)**

**[How to Calculate Rolling Correlation in Excel](#)**

ARABPSYCHOLOGY.COM