How can I stack multiple Pandas DataFrames together?

Authored by

stats writer

April 18, 2024

Stacking multiple Pandas DataFrames together refers to the process of combining or merging multiple DataFrames into one larger DataFrame. This can be achieved by using the built-in Pandas functions such as concat, merge, and join. These functions allow for the combination of DataFrames based on common columns or indexes, resulting in a single comprehensive DataFrame. This approach is useful for organizing and analyzing large datasets, as it allows for easier manipulation and comparison of data. By stacking DataFrames together, users can efficiently handle and analyze complex datasets in a structured and organized manner.

Stack Multiple Pandas DataFrames

Often you may wish to stack two or more pandas DataFrames. Fortunately this is easy to do using the pandas concat() function.

This tutorial shows several examples of how to do so.

Example 1: Stack Two Pandas DataFrames

The following code shows how to "stack" two pandas DataFrames on top of each other and create one DataFrame:

```
import pandas as pd

#create two DataFrames
df1 = pd.DataFrame({'player': ,
'points':})

df2 = pd.DataFrame({'player': ,
```

```
'points':})

#"stack" the two DataFrames together
df3 = pd.concat(, ignore_index=True)

#view resulting DataFrame
df3

player points
0 A 12
1 B 5
2 C 13
3 D 17
4 E 27
5 F 24
6 G 26
7 H 27
8 I 27
9 J 12
```


Example 2: Stack Three Pandas DataFrames

Similar code can be used to stack three pandas DataFrames on top of each other to create one DataFrame:

```
import pandas as pd

#create three DataFrames
df1 = pd.DataFrame({'player': ,
'points':})

df2 = pd.DataFrame({'player': ,
'points':})

df3 = pd.DataFrame({'player': ,
'points':})

#"stack" the two DataFrames together
df4 = pd.concat(, ignore_index=True)

#view resulting DataFrame
df4

player points
0 A 12
1 B 5
2 C 13
3 D 17
4 E 27
5 F 24
6 G 26
```

7 H 27

8 I 27

9 J 12

10 K 9

11 L 5

12 M 5

13 N 13

14 O 17

The Importance of ignore_index

Note that in the previous examples we used ignore_index=True.

This tells pandas to ignore the index numbers in each DataFrame and to create a new index ranging from 0 to n-1 for the new DataFrame.

For example, consider what happens when we don't use ignore_index=True when stacking the following two DataFrames:

```
import pandas as pd

#create two DataFrames with indices
df1 = pd.DataFrame({'player': ,
```

```
'points':},
index=)

df2 = pd.DataFrame({'player': ,
'points':},
index=)

#stack the two DataFrames together
df3 = pd.concat()

#view resulting DataFrame
df3

player points
0 A 12
1 B 5
2 C 13
3 D 17
4 E 27
2 F 24
4 G 26
5 H 27
6 I 27
9 J 12
```

The resulting DataFrame kept its original index values from the two DataFrames.

Thus, you should typically use ignore_index=True when stacking two DataFrames unless you have a specific reason for keeping the original index values.

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

How to Add an Empty Column to a Pandas DataFrame
How to Insert a Column Into a Pandas DataFrame
How to Export a Pandas DataFrame to Excel