

How can I add two Pandas DataFrames together, and can you provide an example?

Authored by
stats writer

June 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I add two Pandas DataFrames together, and can you provide an example?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=152975>

Adding two Pandas DataFrames together is a simple and efficient way to combine data from two separate sources into one cohesive dataset. This can be achieved using the "pd.concat()" function, which merges the rows or columns of the two DataFrames based on specific criteria. An example of this would be merging two DataFrames with identical column names, where the resulting DataFrame will have the same number of rows as the original DataFrames, with the columns from both DataFrames combined. This is a useful technique for data analysis and manipulation, allowing for the easy comparison and combination of data from different sources.

Add Two Pandas DataFrames (With Example)

You can use the following basic syntax to add the values in two pandas DataFrames:

```
df3 = df1.add(df2, fill_value=0)
```

This will produce a new DataFrame that contains the sum of the corresponding elements in each individual DataFrame.

If an element exists in one DataFrame and not the other, the existing element will be used in the resulting DataFrame.

The following example shows how to use this syntax in practice.

Example: How to Add Two Pandas DataFrames

Suppose we have the following two pandas

DataFrames:

```
import pandas as pd
```

```
#create first DataFrame
```

```
df1 = pd.DataFrame({'points': ,  
'assists': })
```

```
#view first DataFrame
```

```
print(df1)
```

```
points assists
```

```
0 18 5
```

```
1 22 11
```

```
2 19 7
```

```
3 14 9
```

```
4 11 12
```

```
#create second DataFrame
```

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

```
#view second DataFrame
```

```
print(df2)
```

```
points assists
```

```
0 10 9
1 5 7
2 4 4
3 3 2
4 9 3
5 14 3
```

We can use the following syntax to create a new DataFrame that takes the sum of corresponding elements in each individual DataFrame:

```
#create new DataFrame by adding two DataFrames
df3 = df1.add(df2, fill_value=0)
```

```
#view new DataFrame
print(df3)
```

```
points assists
0 28.0 14.0
1 27.0 18.0
2 23.0 11.0
3 17.0 11.0
4 20.0 15.0
5 14.0 3.0
```

Notice that the resulting DataFrame contains the sum of corresponding elements in each individual DataFrame.

Note that the row with an index value of 5 only existed in the second DataFrame, so the values in this row are simply the values from the second DataFrame.

Also notice that since we performed addition, each of the values in the new DataFrame are represented as float values with one decimal place.

To convert each of these values back to an integer, we can use the `astype()` function:

```
#convert all columns in new DataFrame to integer
```

```
df3 = df3.astype('int64')
```

```
#view updated DataFrame
```

```
print(df3)
```

```
points assists
```

```
0 28 14
```

```
1 27 18
```

```
2 23 11
```

```
3 17 11
```

```
4 20 15
```

5 14 3

Each of the values in the new DataFrame are now integers.

ARABPSYCHOLOGY.COM