

How do I merge multiple DataFrames in Pandas?

Authored by
stats writer

June 29, 2024

RECOMMENDED CITATION

stats writer (2024). *How do I merge multiple DataFrames in Pandas?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=159328>

Pandas is a popular Python library used for data manipulation and analysis. One common task in data analysis is merging multiple DataFrames, which are tables of data, into a single DataFrame. This can be done in Pandas using the "merge" function, which combines data from two or more DataFrames based on common columns or indices. By specifying the type of join to be performed, users can merge DataFrames horizontally or vertically, creating a new combined DataFrame with all the data from the original tables. This functionality allows for efficient and organized data integration, making it a valuable tool for data analysts and scientists.

Merge Multiple DataFrames in Pandas (With Example)

You can use the following syntax to merge multiple DataFrames at once in pandas:

```
import pandas as pd
from functools import reduce

#define list of DataFrames
dfs =

#merge all DataFrames into one
final_df = reduce(lambda left,right:
pd.merge(left,right,on=,
how='outer'), dfs)
```

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

Example: Merge Multiple DataFrames in Pandas

Suppose we have the following three pandas DataFrames that contain information about basketball players on various teams:

```
import pandas as pd
```

```
#create DataFrames
```

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

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

```
df3 = pd.DataFrame({'team': ,  
'rebounds': })
```

```
#view DataFrames
```

```
print(df1)
```

```
team points
```

```
0 A 18
```

```
1 B 22
```

```
2 C 19
```

```
3 D 14
```

```
print(df2)
```

```
team assists
```

```
0 A 4
```

```
1 B 9
```

```
2 C 14
```

```
print(df3)
```

```
team rebounds
```

```
0 C 10
```

```
1 D 17
```

```
2 E 11
```

```
3 F 10
```

We can use the following syntax to merge all three DataFrames into one:

```
from functools import reduce
```

```
#define list of DataFrames
```

```
dfs =
```

```
#merge all DataFrames into one
```

```
final_df = reduce(lambda left,right:
```

```
pd.merge(left,right,on=,
```

```
how='outer'), dfs)
```

```
#view merged DataFrame
```

```
print(final_df)
```

```
team points assists rebounds
```

```
0 A 18.0 4.0 NaN
```

```
1 B 22.0 9.0 NaN
```

```
2 C 19.0 14.0 10.0
```

```
3 D 14.0 NaN 17.0
```

```
4 E NaN NaN 11.0
```

```
5 F NaN NaN 10.0
```

The final result is one DataFrame that contains information from all three DataFrames.

Notice that NaN values are used to fill in empty cells in the final DataFrame.

To use a value other than NaN to fill in empty cells, we can use the `fillna()` function:

```
from functools import reduce
```

```
#define list of DataFrames
```

dfs =

#merge all DataFrames into one

```
final_df = reduce(lambda left,right:  
pd.merge(left,right,on=  
how='outer'), dfs).fillna('none')
```

#view merged DataFrame

```
print(final_df)
```

team points assists rebounds

0 A 18.0 4.0 none

1 B 22.0 9.0 none

2 C 19.0 14.0 10.0

3 D 14.0 none 17.0

4 E none none 11.0

5 F none none 10.0

Each of the empty cells are now filled with 'none' instead of NaN.

Note: You can find the complete documentation for the merge function in pandas .

Additional Resources

The following tutorials explain how to perform other common operations in pandas:

[How to Merge Two Pandas DataFrames on Index](#)

[How to Stack Multiple Pandas DataFrames](#)

ARABPSYCHOLOGY.COM