# How can I merge columns in Pandas that share the same name?

Authored by

**stats writer**

June 25, 2024

Pandas is a popular Python library used for data manipulation and analysis. One common task in data manipulation is merging columns that share the same name. This can be achieved in Pandas by using the "merge" function, which combines two or more data frames based on a shared column or index. By specifying the "on" parameter as the shared column name, the merge will join the columns from both data frames into a single column. This allows for the consolidation of data and simplifies the analysis process. Additionally, Pandas offers various options for handling duplicate or missing data during the merge, providing flexibility and customization for the user. Overall, the process of merging columns in Pandas is a simple and efficient way to combine related data and improve the overall data organization.

## Pandas: Merge Columns Sharing Same Name

## You can use the following basic syntax to merge together columns in a pandas DataFrame that share the same column name:

## #define function to merge columns with same names together
## defsame_merge(x): return ','.join(x.astype(str))

## #define new DataFrame that merges columns with same names together
## df_new = df.groupby(level=0, axis=1).apply(lambda x: x.apply(same_merge, axis=1))

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

## Example: Merge Together Columns Sharing Same Name in Pandas

## Suppose we have the following pandas DataFrame:

```
import pandas as pd
import numpy as np

#create DataFrame
df = pd.DataFrame({'A': ,
'A1': ,
'B': ,
'B1': })

#rename columns so there are duplicate column names
df.columns =

#view DataFrame
print(df)
```

```
A A B B
0 5.0 NaN 2.0 5.0
1 6.0 12.0 7.0 NaN
2 8.0 NaN NaN 6.0
3 NaN 10.0 NaN 15.0
4 4.0 NaN 2.0 1.0
5 NaN 6.0 4.0 NaN
```

6 NaN 4.0 NaN 4.0

Notice that two columns have a name of 'A' and two columns have a name of 'B.'

We can use the following code to merge the columns that have the same column names and concatenate their values together with a comma:

```
#define function to merge columns with same names together
defsame_merge(x): return ','.join(x.astype(str))

#define new DataFrame that merges columns with same names together
df_new = df.groupby(level=0, axis=1).apply(lambda x: x.apply(same_merge, axis=1))

#view new DataFrame
print(df_new)
```

A B
0 5.0 2.0,5.0
1 6.0,12.0 7.0
2 8.0 6.0

3 10.0 15.0
4 4.0 2.0,1.0
5 6.0 4.0
6 4.0 4.0

The new DataFrame has merged together the columns with the same names and concatenated their values together with a comma.

If you would like to use a different separator, simply change the comma separator to something else in the same_merge() function.

For example, the following code shows how to use a semi-colon separator instead:

```
#define function to merge columns with same names together
defsame_merge(x): return ';'.join(x.astype(str))

#define new DataFrame that merges columns with same names together
df_new = df.groupby(level=0, axis=1).apply(lambda x: x.apply(same_merge, axis=1))
```

**#view new DataFrame**
**print(df_new)**

**A B**
**0 5.0 2.0;5.0**
**1 6.0;12.0 7.0**
**2 8.0 6.0**
**3 10.0 15.0**
**4 4.0 2.0;1.0**
**5 6.0 4.0**
**6 4.0 4.0**

**The new DataFrame has merged together the columns with the same names and concatenated their values together with a semi-colon.**

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