

05-Merging-Joining-and-Concatenating

August 5, 2020

1 Merging, Joining, and Concatenating

There are 3 main ways of combining DataFrames together: Merging, Joining and Concatenating. In this lecture we will discuss these 3 methods with examples.

1.0.1 Example DataFrames

```
[3]: import pandas as pd
```

```
[4]: df1 = pd.DataFrame({'A': ['A0', 'A1', 'A2', 'A3'],  
                        'B': ['B0', 'B1', 'B2', 'B3'],  
                        'C': ['C0', 'C1', 'C2', 'C3'],  
                        'D': ['D0', 'D1', 'D2', 'D3']},  
                        index=[0, 1, 2, 3])
```

```
[5]: df2 = pd.DataFrame({'A': ['A4', 'A5', 'A6', 'A7'],  
                        'B': ['B4', 'B5', 'B6', 'B7'],  
                        'C': ['C4', 'C5', 'C6', 'C7'],  
                        'D': ['D4', 'D5', 'D6', 'D7']},  
                        index=[4, 5, 6, 7])
```

```
[6]: df3 = pd.DataFrame({'A': ['A8', 'A9', 'A10', 'A11'],  
                        'B': ['B8', 'B9', 'B10', 'B11'],  
                        'C': ['C8', 'C9', 'C10', 'C11'],  
                        'D': ['D8', 'D9', 'D10', 'D11']},  
                        index=[8, 9, 10, 11])
```

```
[7]: df1
```

```
[7]:
```

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

```
[8]: df2
```

```
[8]:      A    B    C    D
4  A4  B4  C4  D4
5  A5  B5  C5  D5
6  A6  B6  C6  D6
7  A7  B7  C7  D7
```

```
[12]: df3
```

```
[12]:      A    B    C    D
8   A8  B8  C8  D8
9   A9  B9  C9  D9
10  A10 B10 C10 D10
11  A11 B11 C11 D11
```

1.1 Concatenation

Concatenation basically glues together DataFrames. Keep in mind that dimensions should match along the axis you are concatenating on. You can use **pd.concat** and pass in a list of DataFrames to concatenate together:

```
[10]: pd.concat([df1,df2,df3])
```

```
[10]:      A    B    C    D
0   A0  B0  C0  D0
1   A1  B1  C1  D1
2   A2  B2  C2  D2
3   A3  B3  C3  D3
4   A4  B4  C4  D4
5   A5  B5  C5  D5
6   A6  B6  C6  D6
7   A7  B7  C7  D7
8   A8  B8  C8  D8
9   A9  B9  C9  D9
10  A10 B10 C10 D10
11  A11 B11 C11 D11
```

```
[18]: pd.concat([df1,df2,df3],axis=1)
```

```
[18]:      A    B    C    D    A    B    C    D    A    B    C    D
0   A0  B0  C0  D0  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
1   A1  B1  C1  D1  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
2   A2  B2  C2  D2  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
3   A3  B3  C3  D3  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
4  NaN  NaN  NaN  NaN   A4  B4  C4  D4  NaN  NaN  NaN  NaN
5  NaN  NaN  NaN  NaN   A5  B5  C5  D5  NaN  NaN  NaN  NaN
```

6	NaN	NaN	NaN	NaN	A6	B6	C6	D6	NaN	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN	A7	B7	C7	D7	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A8	B8	C8	D8
9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A9	B9	C9	D9
10	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A10	B10	C10	D10
11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A11	B11	C11	D11

1.2 Example DataFrames

```
[28]: left = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3'],
                           'A': ['A0', 'A1', 'A2', 'A3'],
                           'B': ['B0', 'B1', 'B2', 'B3']})

right = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3'],
                      'C': ['C0', 'C1', 'C2', 'C3'],
                      'D': ['D0', 'D1', 'D2', 'D3']})
```

```
[29]: left
```

```
[29]:   A  B key
0  A0 B0 K0
1  A1 B1 K1
2  A2 B2 K2
3  A3 B3 K3
```

```
[30]: right
```

```
[30]:   C  D key
0  C0 D0 K0
1  C1 D1 K1
2  C2 D2 K2
3  C3 D3 K3
```

1.3 Merging

The **merge** function allows you to merge DataFrames together using a similar logic as merging SQL Tables together. For example:

```
[35]: pd.merge(left, right, how='inner', on='key')
```

```
[35]:   A  B key  C  D
0  A0 B0 K0 C0 D0
1  A1 B1 K1 C1 D1
```

```

2  A2  B2  K2  C2  D2
3  A3  B3  K3  C3  D3

```

Or to show a more complicated example:

```

[37]: left = pd.DataFrame({'key1': ['K0', 'K0', 'K1', 'K2'],
                           'key2': ['K0', 'K1', 'K0', 'K1'],
                           'A': ['A0', 'A1', 'A2', 'A3'],
                           'B': ['B0', 'B1', 'B2', 'B3']})

right = pd.DataFrame({'key1': ['K0', 'K1', 'K1', 'K2'],
                      'key2': ['K0', 'K0', 'K0', 'K0'],
                      'C': ['C0', 'C1', 'C2', 'C3'],
                      'D': ['D0', 'D1', 'D2', 'D3']})

```

```

[39]: pd.merge(left, right, on=['key1', 'key2'])

```

```

[39]:
   A  B key1 key2  C  D
0  A0 B0  K0  K0  C0 D0
1  A2 B2  K1  K0  C1 D1
2  A2 B2  K1  K0  C2 D2

```

```

[40]: pd.merge(left, right, how='outer', on=['key1', 'key2'])

```

```

[40]:
   A  B key1 key2  C  D
0  A0 B0  K0  K0  C0 D0
1  A1 B1  K0  K1  NaN NaN
2  A2 B2  K1  K0  C1 D1
3  A2 B2  K1  K0  C2 D2
4  A3 B3  K2  K1  NaN NaN
5  NaN NaN  K2  K0  C3 D3

```

```

[41]: pd.merge(left, right, how='right', on=['key1', 'key2'])

```

```

[41]:
   A  B key1 key2  C  D
0  A0 B0  K0  K0  C0 D0
1  A2 B2  K1  K0  C1 D1
2  A2 B2  K1  K0  C2 D2
3  NaN NaN  K2  K0  C3 D3

```

```

[42]: pd.merge(left, right, how='left', on=['key1', 'key2'])

```

```

[42]:
   A  B key1 key2  C  D
0  A0 B0  K0  K0  C0 D0
1  A1 B1  K0  K1  NaN NaN
2  A2 B2  K1  K0  C1 D1
3  A2 B2  K1  K0  C2 D2

```

```
4  A3  B3   K2   K1  NaN  NaN
```

1.4 Joining

Joining is a convenient method for combining the columns of two potentially differently-indexed DataFrames into a single result DataFrame.

```
[46]: left = pd.DataFrame({'A': ['A0', 'A1', 'A2'],  
                          'B': ['B0', 'B1', 'B2']},  
                          index=['K0', 'K1', 'K2'])  
  
right = pd.DataFrame({'C': ['C0', 'C2', 'C3'],  
                      'D': ['D0', 'D2', 'D3']},  
                      index=['K0', 'K2', 'K3'])
```

```
[47]: left.join(right)
```

```
[47]:
```

	A	B	C	D
K0	A0	B0	C0	D0
K1	A1	B1	NaN	NaN
K2	A2	B2	C2	D2

```
[48]: left.join(right, how='outer')
```

```
[48]:
```

	A	B	C	D
K0	A0	B0	C0	D0
K1	A1	B1	NaN	NaN
K2	A2	B2	C2	D2
K3	NaN	NaN	C3	D3