

▼ 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.

▼ Example DataFrames

```
import pandas as pd
```

```
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])
```

```
df2 = pd.DataFrame({'A': ['A4', 'A5', 'A6', 'A7'],
                    'B': ['B4', 'B5', 'B6', 'B7'],
                    'C': ['C4', 'C5', 'C6', 'C7'],
                    'D': ['D4', 'D5', 'D6', 'D7']},
                    index=[0, 5, 6, 7])
```

```
df3 = pd.DataFrame({'A': ['A8', 'A9', 'A10', 'A11'],
                    'B': ['B8', 'B9', 'B10', 'B11'],
                    'C': ['C8', 'C9', 'C10', 'C11'],
                    'D': ['D8', 'D9', 'D10', 'D11']},
                    index=[0, 9, 10, 11])
```

df1

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

df2

	A	B	C	D
0	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6

df3

	A	B	C	D
0	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

▼ Concatenation

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

▼ Join the two dataframes along rows

```
pd.concat([df1,df2,df3], axis=0)
```

	A	B	C	D
0	A0	B0	C0	D0

▼ Join the two dataframes along columns

```

3      A3      B3      C3      D3
pd.concat([df1,df2,df3],axis=1)

```

	A	B	C	D	A	B	C	D	A	B	C	D
0	A0	B0	C0	D0	A4	B4	C4	D4	A8	B8	C8	D8
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
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
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

▼ Merging

The **merge** function allows us to merge DataFrames together. For example:

Merge method	- SQL Equivalent	- Description
1. left	- LEFT OUTER JOIN	- Use keys from left object
2. right	- RIGHT OUTER JOIN	- Use keys from right object
3. outer	- FULL OUTER JOIN	- Use union of keys
4. inner	- INNER JOIN	- Use intersection of keys

Merge two dataframes along the key value

Merge with inner join

“Inner join produces only the set of records that match in both dataframes df1 and df2.”

▼ Example DataFrames

```
df1 = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3'],
                    'A': ['A0', 'A1', 'A2', 'A3'],
                    'B': ['B0', 'B1', 'B2', 'B3']})
```

```
df2 = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3'],
                    'C': ['C0', 'C1', 'C2', 'C3'],
                    'D': ['D0', 'D1', 'D2', 'D3']})
```

df1

df2

```
pd.merge(df1, df2, how='inner', on='key')
```

▼ One more example:

```
df1 = pd.DataFrame({'key1': ['K0', 'K0', 'K1', 'K2'],
                    'key2': ['K0', 'K1', 'K0', 'K1'],
                    'A': ['A0', 'A1', 'A2', 'A3'],
                    'B': ['B0', 'B1', 'B2', 'B3']})
```

```
df2 = pd.DataFrame({'key1': ['K0', 'K1', 'K1', 'K2'],
                    'key2': ['K0', 'K0', 'K0', 'K0'],
                    'C': ['C0', 'C1', 'C2', 'C3'],
                    'D': ['D0', 'D1', 'D2', 'D3']})
```

df1

df2

```
pd.merge(df1, df2)
```

▼ Merge with outer join

“Outer join produces the set of all records from df1 and df2, with matching records. If there is no match, the missing side will contain null.”

```
pd.merge(df1, df2, how='outer', on=['key1', 'key2'])
```

▼ Merge with right join

“Right join produces a complete set of records from right object (df2), with the matching records in df1. If there is no match, the corresponding side will contain null.”

```
print(df1)
print(df2)
```

```
pd.merge(df1, df2, how='right', on=['key1', 'key2'])
```

▼ Merge with left join

“Left join produces a complete set of records from left object (df1), with the matching records in df2. If there is no match, the corresponding side will contain null.”

```
pd.merge(df1, df2, how='left', on=['key1', 'key2'])
```

▼ Joining

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

```
df1 = pd.DataFrame({'A': ['A0', 'A1', 'A2'],
                    'B': ['B0', 'B1', 'B2']},
                    index=['K0', 'K1', 'K2'])
```

```
df2 = pd.DataFrame({'C': ['C0', 'C2', 'C3'],
                    'D': ['D0', 'D2', 'D3']},
                    index=['K0', 'K2', 'K3'])
```

```
df1
```

```
df2
```

“Join produces a complete set of records from df1, with the matching records in df2. If there is no match, the right side will contain null.”

similar to "merge with left join" but "merge with left join" performs on a column and join performs on indices

```
df1.join(df2)
```

▼ Join with outer argument

"Join with outer argument produces the set of all records from df1 and df2, with matching records. If there is no match, the missing side will contain null."

```
df1.join(df2, how='outer')
```

