

파이썬 라이브러리를 활용한 데이터 분석

8장 데이터 준비하기:
조인, 병합, 변형

2020.07.02 3h

다중 색인, 조인, 병합, 변형

- 데이터틀 합치고 재배열 필요
 - 원천 데이터는 분석하기 어려운 형태로 기록되어 제공
- 주요 내용
 - 계층 색인(다중 색인)
 - **Multi-index**
 - 데이터 합치기
 - **2개 또는 2개 이상의 시리즈나 데이터프레임의 데이터 합치기**
 - Merge
 - Join
 - Concat
 - Combine_first
 - 재형성과 피벗
 - **하나의 테이블의 행, 열, 인덱스 등 구조를 재형성**
 - Stack
 - Unstack
 - Pivot
 - Melt

참고 사이트

- **국내**

- https://freelife1191.github.io/dev/2018/05/07/dev-data_analysis-22.python_data_analysis/
- <https://rfriend.tistory.com/276>

- **국외**

- https://pandas.pydata.org/pandas-docs/stable/user_guide/advanced.html#advanced-hierarchical
- https://pandas.pydata.org/pandas-docs/stable/user_guide/reshaping.html
- <https://towardsdatascience.com/python-pandas-dataframe-join-merge-and-concatenate-84985c29ef78>
- <http://talimi.se/p/pandas/>

파일 ch08-study.ipynb

8장 데이터 준비하기: 조인, 병합, 변형

join

조인 개요

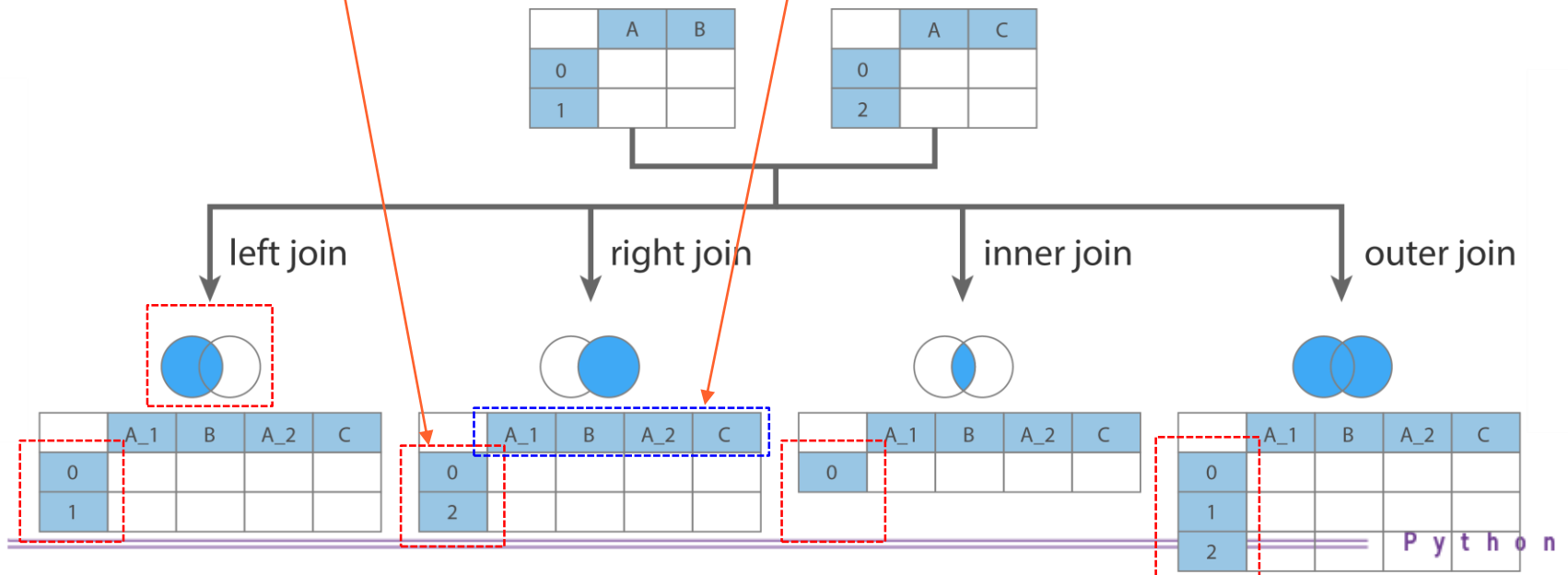
색인으로 병합(merge)

- 컬럼은 겹치지 않게 모든 칼럼은 추가, 빠지는 열이 없음
 - 이름이 겹치면 열 이름 접미어 추가
- 각 행은 행 색인으로 병합, 옵션 how로 4가지 지정

자동으로 지정이 안되
니 직접 지정해야 함

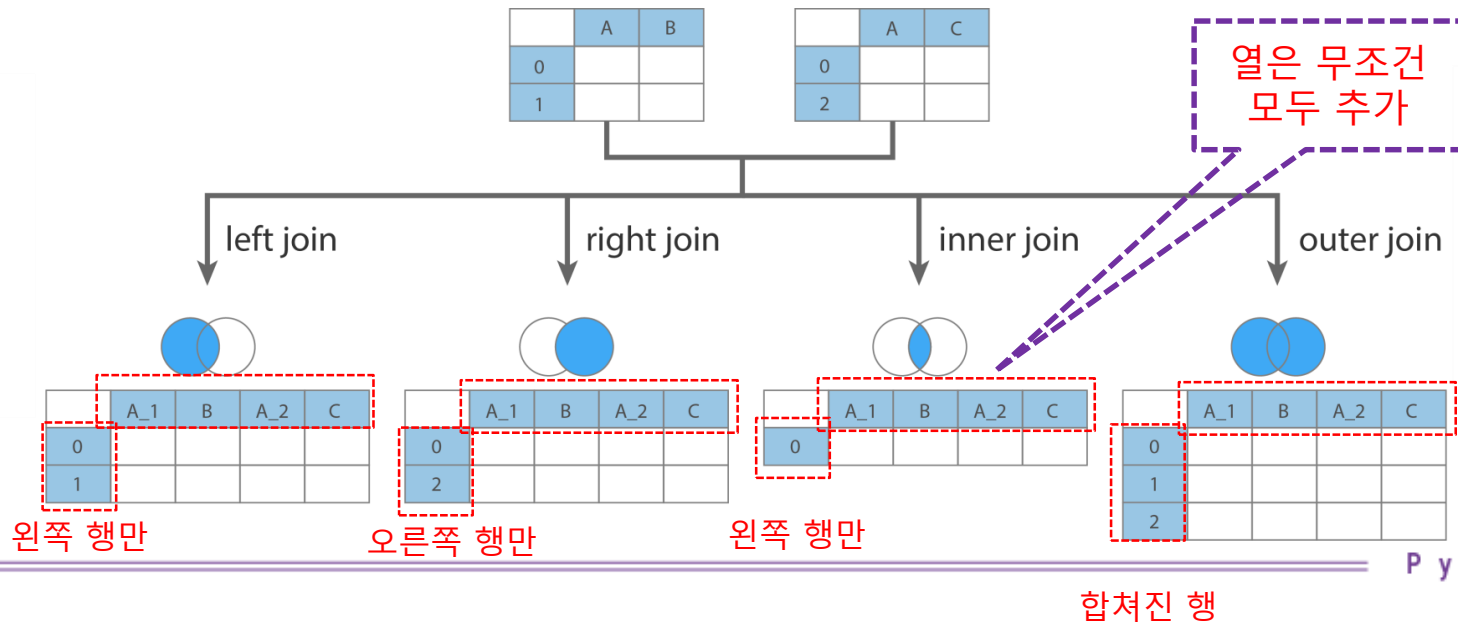
구문

- DataFrame.join(self, other, on=None, how='left', lsuffix="", rsuffix="", sort=False)
 - how: 결과의 색인 선택 기준
 - lsuffix : 중복되는 열의 왼쪽 열이름, 접미어 지정
 - sort: 조인 키에 의한 정렬, 기본은 안됨



조인 방법: how=

- **left(기본)**
 - 왼쪽 데이터프레임의 색인(왼쪽 색인) 모두 사용하여 병합
- **right**
 - 오른쪽 데이터프레임의 색인(오른쪽 색인) 모두 사용하여 병합
- **inner**
 - 두 데이터프레임의 공통된 색인(교집합 색인)만을 사용하여 병합
- **outer**
 - 두 데이터프레임의 색인 모두(합집합 색인) 사용하여 병합

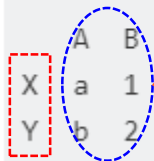


조인 이해

• Join

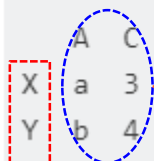
- 각각의 인덱스(행 색인)를 기반으로 데이터 프레임과 결합
 - 겹치는 열이 있으면 join은 왼쪽 데이터 프레임에서 겹치는 열 이름에 접미사를 추가
 - 인자 lsuffix

```
left = pd.DataFrame([[ 'a', 1], [ 'b', 2]], list('XY'), list('AB'))
left
```



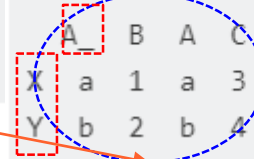
	A	B
X	a	1
Y	b	2

```
right = pd.DataFrame([[ 'a', 3], [ 'b', 4]], list('XY'), list('AC'))
right
```



	A	C
X	a	3
Y	b	4

```
left.join(right, lsuffix='_')
```



	A_	B	A	C
X	a	1	a	3
Y	b	2	b	4

외부 조인

- 색인이 추가되고 열도 모두 추가되어 5 개의 열

In [144]: `right.reset_index() # 색인을 컬럼의 데이터로`

Out[144]:

	index	A	C
0	X	a	3
1	Y	b	4

In [146]: `left`

Out[146]:

	A	B
X	a	1
Y	b	2

In [143]: `left.join(right.reset_index(), lsuffix='_', how='outer')`

Out[143]:

	A_	B	index	A	C
X	a	1.0	NaN	NaN	NaN
Y	b	2.0	NaN	NaN	NaN
0	NaN	NaN	X	a	3.0
1	NaN	NaN	Y	b	4.0

교재: p324

• 외부 조인: 모든 합집합

In [153]: left2

Out[153]:

	Ohio	Nevada
a	1.0	2.0
c	3.0	4.0
e	5.0	6.0

In [134]: left2.join(right2, how='outer')

Out[134]:

	Ohio	Nevada	Missouri	Alabama
a	1.0	2.0	NaN	NaN
b	NaN	NaN	7.0	8.0
c	3.0	4.0	9.0	10.0
d	NaN	NaN	11.0	12.0
e	5.0	6.0	13.0	14.0

In [154]: right2

Out[154]:

	Missouri	Alabama
b	7.0	8.0
c	9.0	10.0
d	11.0	12.0
e	13.0	14.0

왼쪽의 키(인덱스 지정) on=

- 옵션 on=
 - 왼쪽 데이터 프레임의 특정한 열을 조인 키로 사용
 - 여전히 오른쪽은 기본 인덱스를 사용
- 결과의 색인
 - 왼쪽 조인이므로 왼쪽의 인덱스로

```
In [147]: left.reset_index().join(right, on='index', lsuffix='_')
```

```
Out[147]:
```

	index	A_	B	A	C
0	X	a	1	a	3
1	Y	b	2	b	4

열은 모두 참가

```
In [148]: left
```

```
Out[148]:
```

	A	B
X	a	1
Y	b	2

```
In [149]: left.reset_index()
```

```
Out[149]:
```

	index	A	B
0	X	a	1
1	Y	b	2

```
Out[149]: right
```

```
Out[149]:
```

	A	C
X	a	3
Y	b	4

왼쪽의 키(인덱스 지정) on=

• 왼쪽 데이터프레임의 키를 지정하여 왼쪽 조인

- 오른쪽은 기본 색인 사용
- how가 없으므로 왼쪽 조인(색인이 왼쪽만 구성)

```
In [155]: left1
```

```
Out[155]:
```

	key	value
0	a	0
1	b	1
2	a	2
3	a	3
4	b	4
5	c	5

```
In [156]: right1
```

```
Out[156]:
```

	group_val
a	3.5
b	7.0

```
In [157]: left1.join(right1, on='key')
```

```
Out[157]:
```

	key	value	group_val
0	a	0	3.5
1	b	1	7.0
2	a	2	3.5
3	a	3	3.5
4	b	4	7.0
5	c	5	NaN

인자 없는 조인

- 기본은 두 데이터프레임의 색인 대 색인으로 병합
 - How가 없으므로 왼쪽 조인
 - 왼쪽의 색인 만을 사용

In [159]: left2

Out[159]:

	Ohio	Nevada
a	1.0	2.0
c	3.0	4.0
e	5.0	6.0

In [160]: right2

Out[160]:

	Missouri	Alabama
b	7.0	8.0
c	9.0	10.0
d	11.0	12.0
e	13.0	14.0

In [166]: left2.join(right2)

Out[166]:

	Ohio	Nevada	Missouri	Alabama
a	1.0	2.0	NaN	NaN
c	3.0	4.0	9.0	10.0
e	5.0	6.0	13.0	14.0

여러 개를 병합

• 왼쪽 조인

- 오른쪽 인자에 리스트로 활용

```
In [159]: left2
```

```
Out[159]:
```

	Ohio	Nevada
a	1.0	2.0
c	3.0	4.0
e	5.0	6.0

```
In [160]: right2
```

```
Out[160]:
```

	Missouri	Alabama
b	7.0	8.0
c	9.0	10.0
d	11.0	12.0
e	13.0	14.0

```
In [161]: another = pd.DataFrame([[7., 8.], [9., 10.], [11., 12.], [16., 17.]],
                                index=['a', 'c', 'e', 'f'],
                                columns=['New York', 'Oregon'])
another
```

```
Out[161]:
```

	New York	Oregon
a	7.0	8.0
c	9.0	10.0
e	11.0	12.0
f	16.0	17.0

```
In [167]: left2.join([right2, another])
```

```
Out[167]:
```

	Ohio	Nevada	Missouri	Alabama	New York	Oregon
a	1.0	2.0	NaN	NaN	7.0	8.0
c	3.0	4.0	9.0	10.0	9.0	10.0
e	5.0	6.0	13.0	14.0	11.0	12.0

여러 개를 병합

• 외부 조인

- 오른쪽 인자에 리스트로 활용, 모든 행과 열을 합집합

```
In [159]: left2
```

```
Out[159]:
```

	Ohio	Nevada
a	1.0	2.0
c	3.0	4.0
e	5.0	6.0

```
In [160]: right2
```

```
Out[160]:
```

	Missouri	Alabama
b	7.0	8.0
c	9.0	10.0
d	11.0	12.0
e	13.0	14.0

```
In [168]: left2.join([right2, another], how='outer')
```

```
Out[168]:
```

	Ohio	Nevada	Missouri	Alabama	New York	Oregon
a	1.0	2.0	NaN	NaN	7.0	8.0
c	3.0	4.0	9.0	10.0	9.0	10.0
e	5.0	6.0	13.0	14.0	11.0	12.0
b	NaN	NaN	7.0	8.0	NaN	NaN
d	NaN	NaN	11.0	12.0	NaN	NaN
f	NaN	NaN	NaN	NaN	16.0	17.0

```
In [161]: another = pd.DataFrame([[7., 8.], [9., 10.], [11., 12.], [16., 17.]],
                                index=['a', 'c', 'e', 'f'],
                                columns=['New York', 'Oregon'])
another
```

```
Out[161]:
```

	New York	Oregon
a	7.0	8.0
c	9.0	10.0
e	11.0	12.0
f	16.0	17.0

판다스 홈 예제

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.join.html?highlight=join#pandas.DataFrame.join>

```
>>> df = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3', 'K4', 'K5'],
...                     'A': ['A0', 'A1', 'A2', 'A3', 'A4', 'A5']})
```

```
>>> df
  key  A
0  K0  A0
1  K1  A1
2  K2  A2
3  K3  A3
4  K4  A4
5  K5  A5
```

```
>>> other = pd.DataFrame({'key': ['K0', 'K1', 'K2'],
...                       'B': ['B0', 'B1', 'B2']})
```

```
>>> other
  key  B
0  K0  B0
1  K1  B1
2  K2  B2
```

Join DataFrames using their indexes.

```
>>> df.join(other, lsuffix='_caller', rsuffix='_other')
  key_caller  A key_other  B
0         K0  A0         K0  B0
1         K1  A1         K1  B1
2         K2  A2         K2  B2
3         K3  A3         NaN NaN
4         K4  A4         NaN NaN
5         K5  A5         NaN NaN
```



```
>>> df = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3', 'K4', 'K5'],
...                     'A': ['A0', 'A1', 'A2', 'A3', 'A4', 'A5']})
```

```
>>> df
  key  A
0  K0  A0
1  K1  A1
2  K2  A2
3  K3  A3
4  K4  A4
5  K5  A5
```

```
>>> other = pd.DataFrame({'key': ['K0', 'K1', 'K2'],
...                       'B': ['B0', 'B1', 'B2']})
```

```
>>> other
  key  B
0  K0  B0
1  K1  B1
2  K2  B2
```

기본은 왼쪽 조인

```
>>> df.set_index('key').join(other.set_index('key'))
```

```
   A    B
key
K0  A0  B0
K1  A1  B1
K2  A2  B2
K3  A3  NaN
K4  A4  NaN
K5  A5  NaN
```

왼쪽 키를 on으로 지정

- 지정한 열을 색인으로 사용해 join
 - 이 지정 열이 그대로 결과에 열로 사용됨

```
>>> df = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3', 'K4', 'K5'],
...                     'A': ['A0', 'A1', 'A2', 'A3', 'A4', 'A5']})
```

```
>>> df
  key A
0  K0 A0
1  K1 A1
2  K2 A2
3  K3 A3
4  K4 A4
5  K5 A5
```

```
>>> other = pd.DataFrame({'key': ['K0', 'K1', 'K2'],
...                       'B': ['B0', 'B1', 'B2']})
```

```
>>> other
  key B
0  K0 B0
1  K1 B1
2  K2 B2
```

```
>>> df.join(other.set_index('key'), on='key')
  key A   B
0  K0 A0 B0
1  K1 A1 B1
2  K2 A2 B2
3  K3 A3 NaN
4  K4 A4 NaN
5  K5 A5 NaN
```