빅데이터 실습

6주차 3차시

데이터 분석하기 - 그룹집계(Group Aggregate)

데이터 분석하기
그룹집계 (Group Aggregate)

학습개요

1/ 그룹집계의 개념

2/ groupby( ) 함수를 활용한 그룹집계

3/ pivot_table( ) 함수를 활용한 그룹집계

01

그룹집계의 개념

① 그룹집계의 개념

그룹 집계

✓ 전체 판매량을 구하려면?

data

제품명	판매량
A	10
A	15
B	2
B	4
B	6
C	6
C	1

통계함수

data.sum( )

44

① 그룹집계의 개념

그룹 집계

✓ 제품별 판매량을 구하려면?

data

제품명	판매량
A	10
A	15
B	2
B	4
B	6
C	6
C	1

data[data.제품명 == 'A'].sum( ) → 25

data[data.제품명 == 'B'].sum( ) → 12

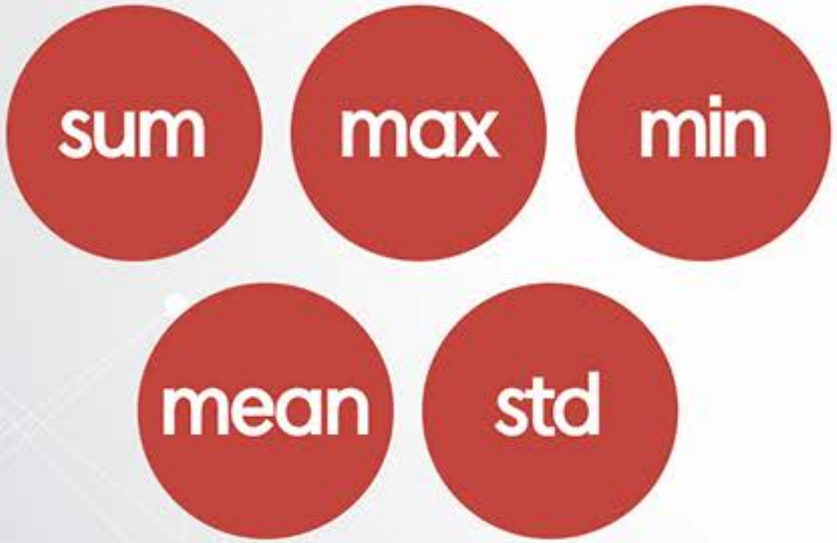
data[data.제품명 == 'C'].sum( ) → 7

특정 컬럼의 값을 기준으로 데이터를 그룹핑하여 그룹별로 집계를 하는 방식은 많이 활용됩니다.

① 그룹집계의 개념

그룹 집계

✓ 데이터를 특정 속성의 값으로 그룹핑하여 집계를 수행함



✓ SQL 구문의 GROUP BY 연산과 유사

① 그룹집계의 개념

그룹 분석을 하는 방법

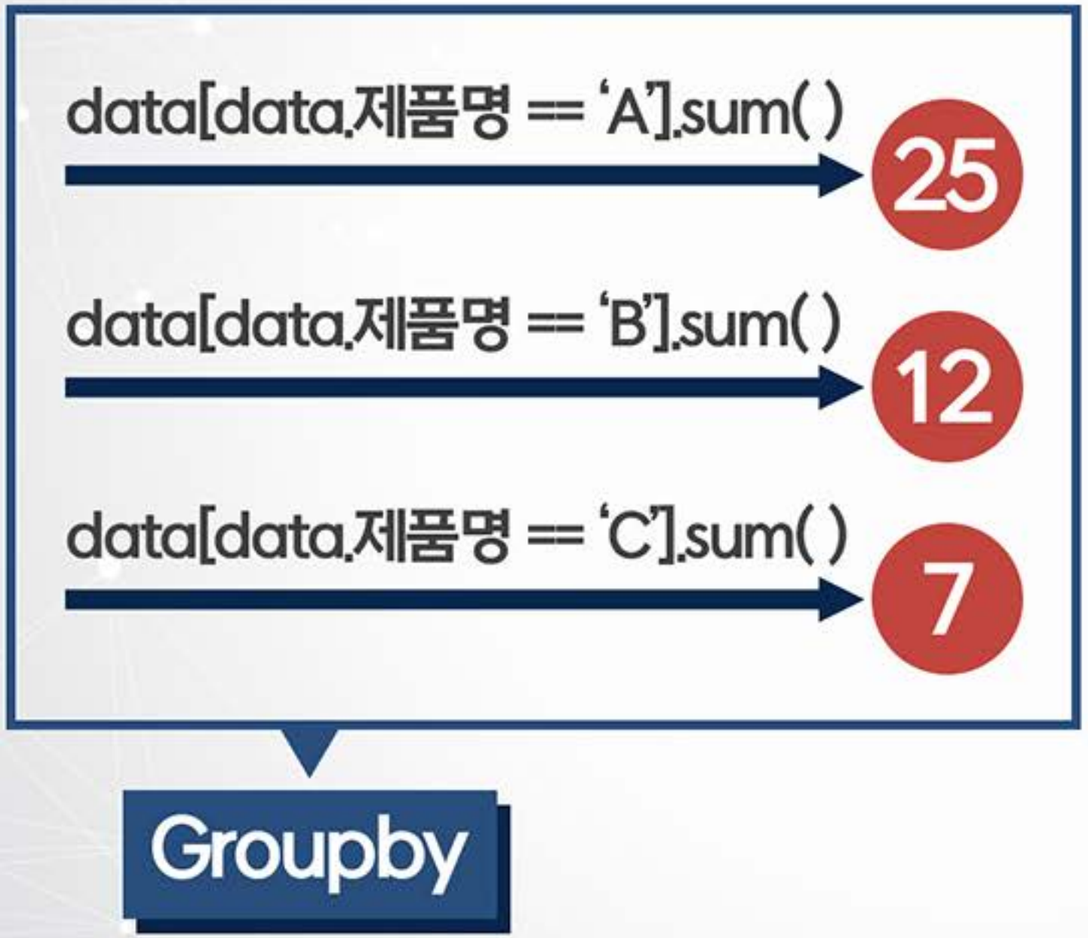
Groupby

Pivot_table

02

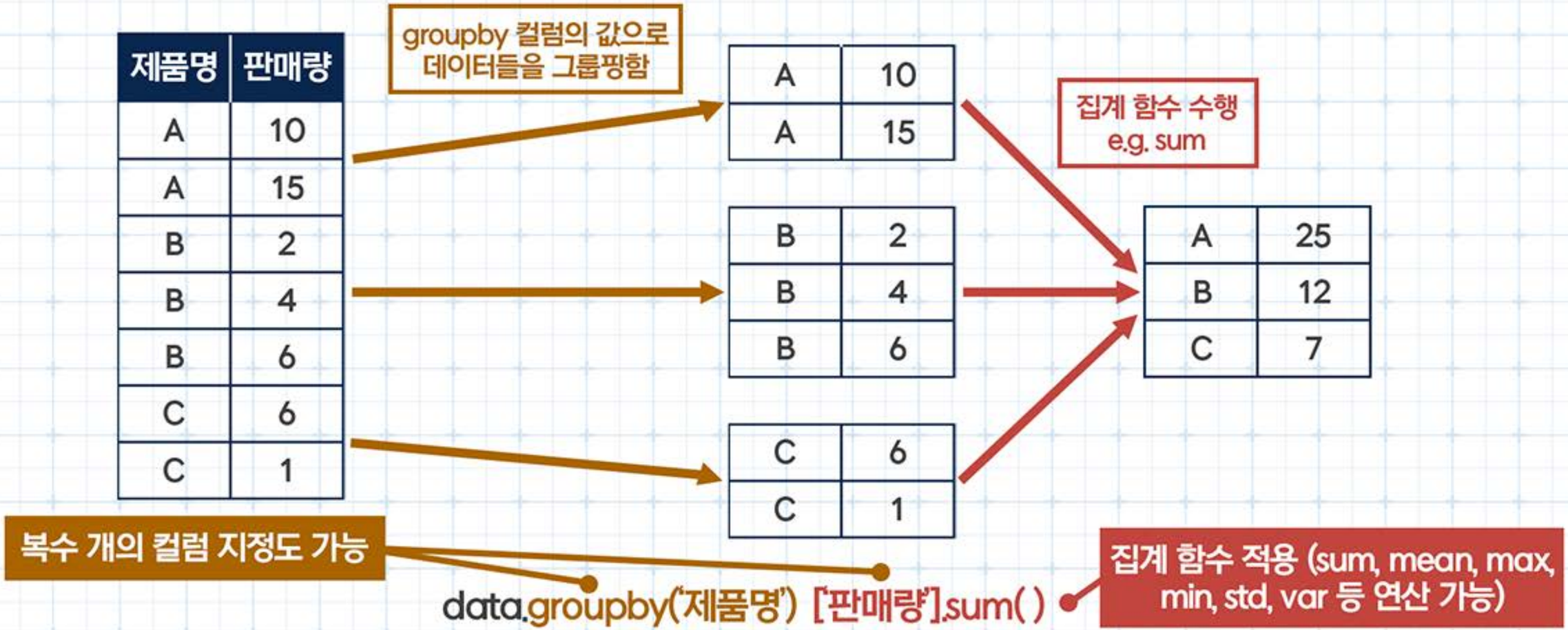
groupby() 함수를 활용한 그룹집계

② groupby( ) 함수를 활용한 그룹집계



② groupby( ) 함수를 활용한 그룹집계

그룹 분석을 하는 방법



groupby() 함수

실습 준비하기

◎ import

Run Code

In [12]:
```
import pandas as pd
from pandas import Series, DataFrame
```

In [29]:
```
# 실습 데이터 적재
data = pd.read_excel('data/인구수예제.xlsx')
data.head()
```

Out[29]:

	도시	자치구	연도	남자인구	여자인구	총인구
0	서울	강남구	2016	61	70	131
1	서울	강남구	2017	119	116	235
2	서울	강남구	2018	134	141	275
3	서울	강남구	2019	119	116	235
4	서울	강남구	2020	115	144	259

In [1]: # 실습 1. 연도별 총인구수 구하기

검색하려면 여기에 입력하십시오.

실습 준비하기

◎ 실습 데이터 적재

✔ 도시의 자치구별, 연도별 남자인구와 여자인구 그리고 총인구

In [1]:
```
import pandas as pd
from pandas import Series, DataFrame
```

In [29]:
```
# 실습 데이터 적재
data = pd.read_excel('data/인구수예제.xlsx')
data.head()
```

Out[29]:

	도시	자치구	연도	남자인구	여자인구	총인구
0	서울	강남구	2016	61	70	131
1	서울	강남구	2017	119	116	235
2	서울	강남구	2018	134	141	275
3	서울	강남구	2019	119	116	235
4	서울	강남구	2020	115	144	259

In [1]:
```
# 실습 1. 연도별 총인구수 구하기
```

groupby() 연도별 총인구수 구하기

◎ 2016, …, 2020년의 연도별 총인구의 sum

✔ 연도별로 그룹핑

Run Code

In [ ]:

In [5]:
```
# 실습 1. 연도별 총인구수 구하기
data.groupby('연도')['총인구'].sum()
```

Out[5]:
```
연도
2016    2427
2017    2308
2018    2382
2019    2193
2020    2332
Name: 총인구, dtype: int64
```

In [2]:
```
# 실습 2. 연도별 전체의 남자인구, 여자인구, 총인구 수 구하기
```

In [3]:
```
# 실습 3. 자치구별로 평균 총인구수 구하기
```

검색하려면 여기에 입력하십시오.

groupby() 연도별 총인구수 구하기

◎ 2016, …, 2020년의 연도별 총인구의 sum

- ✔ groupby 인자에 그룹핑할 기준 컬럼인 '연도' 컬럼을 지정
- ✔ 총인구 컬럼의 합(sum) 수행

```
In [5]: # 실습 1. 연도별 총인구수 구하기
        data.groupby('연도')['총인구'].sum()
```

```
Out[5]: 연도
        2016    2427
        2017    2308
        2018    2382
        2019    2193
        2020    2332
        Name: 총인구, dtype: int64
```

```
In [2]: # 실습 2. 연도별 전체의 남자인구, 여자인구, 총인구 수 구하기
```

```
In [3]: # 실습 3. 자치구별로 평균 총인구수 구하기
```

groupby() 연도별 전체의 남자인구, 여자인구, 총인구수 구하기

◎ 2016, …, 2020년의 연도별 남자인구, 여자인구, 총인구의 sum

✔ 연도별로 그룹핑 후 남자인구, 여자인구, 총인구를 리스트로 전달

```
In [6]: data.groupby('연도')[['남자인구','여자인구','총인구']].sum()
```

Out[6]:

연도	남자인구	여자인구	총인구
2016	1269	1158	2427
2017	1167	1141	2308
2018	1210	1172	2382
2019	1053	1140	2193
2020	1218	1114	2332

```
In [3]: # 실습 3. 자치구별로 평균 총인구수 구하기
```

```
In [4]: # 실습 4.도시/자치구별 평균 총인구수 구하기
```

groupby() 자치구별로 평균 총인구수 구하기

◎ 자치구별로 그룹핑 후, 총인구수의 평균 구하기

Run Code

In [ ]:

In [8]:
```
# 실습 3. 자치구별로 평균 총인구수 구하기
data.groupby('자치구')['총인구'].mean()
```

Out[8]:
```
자치구
강남구      227.0
도봉구      176.2
동래구      214.2
동작구      150.6
서대문구     171.8
송파구
수영구
영등포구
종로구
중구
해운대구
Name: 총인
```

주의하기

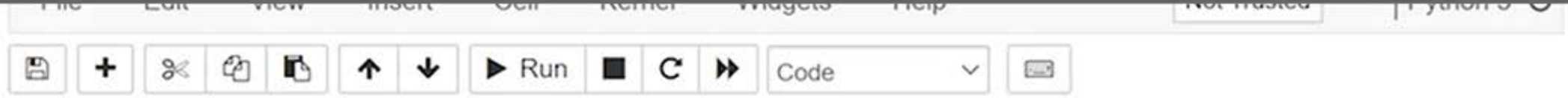
✔ 같은 자치구의 이름이 다른 도시에 있을 수 있기 때문에 도시와 자치구별로 구해야 함

In [4]: # 실습 4 도시/자치구별 평균 총인구수 구하기

검색하려면 여기에 입력하십시오.

groupby() 도시/자치구별로 평균 총인구수 구하기

◎ groupby 인자에 도시, 자치구를 리스트로 담아서 전달




```
해운대구    213.0
Name: 총인구, dtype: float64
```

In [9]:
```
# 실습 4.도시/자치구별 평균 총인구수 구하기
data.groupby(['도시', '자치구'])['총인구'].mean()
```

Out[9]:
```
도시  자치구
부산  동래구      214.2
      수영구      188.6
      중구        193.4
      해운대구    213.0
서울  강남구      227.0
      도봉구      176.2
      동작구      150.6
      서대문구    171.8
      송파구      197.8
      영등포구    228.4
      종로구      189.0
      중구        178.4
Name: 총인구, dtype: float64
```

groupby() 도시/자치구별로 평균 남자인구와 여자인구수 구하기

◎ 총인구수 대신 남자인구, 여자인구를 리스트로 담아 전달

Out[10]:

도시	자치구	남자인구	여자인구
부산	동래구	91.4	122.8
	수영구	90.2	98.4
	중구	97.2	96.2
	해운대구	107.2	105.8
서울	강남구	109	
	도봉구	10	
	동작구	8	
	서대문구	8	
	송파구	11	
	영등포구	112.6	119.6

주의하기

- ✔ grougby의 columns과 집계하고자 하는 columns을 리스트로 담아 여러 개를 전달할 수 있음
- ✔ 집계 함수를 동시에 여러 개 수행할 수는 없음

groupby() 도시별, 연도별로 총인구수 출력

◎ groupby 함수에 도시, 연도를 인자로 담고, 총인구의 합을 구함

Run Code

In [13]:
```
# 실습 6. 도시별, 연도별로 총인구수 출력
data.groupby(['도시','연도'])['총인구'].sum()
```

Out[13]:
```
도시  연도
부산  2016     832
      2017     791
      2018     790
      2019     737
      2020     896
서울  2016    1595
      2017    1517
      2018    1592
      2019    1456
      2020    1436
Name: 총인구, dtype: int64
```

그룹 분석 - pivot_table()

구문
```
data.pivot_table(index = '연도', columns = '도시', aggfunc = 'sum', values = '총인구')
```

groupby() 도시별, 연도별로 총인구수 출력

◎ columns index 계층 색인으로 변경

✔ unstack에 레벨 0 또는 '도시' 명명

Markdown

In [14]:
```
# 실습 6. 도시별, 연도별로 총인구수 출력
data.groupby(['도시','연도'])['총인구'].sum().unstack('도시')
```

Out[14]:

도시	부산	서울
연도		
2016	832	1595
2017	791	1517
2018	790	1592
2019	737	1456
2020	896	1436

그룹 분석 - pivot_table()

구문 `data.pivot_table(index = '연도', columns = '도시', aggfunc = 'sum', values = '총인구')`

② groupby( ) 함수를 활용한 그룹집계

Groupby

집계 결과를
보기 좋게 하기 위해
stack이나 unstack 등의
구문을 추가로 작성해야
되는 경우가 있음

Pivot_table

컬럼 인자들이나
내가 집계하고자 하는
컬럼들은 리스트로 담아서
여러 개를 한 번에
할 수가 있음

03

pivot_table() 함수를 활용한 그룹집계

③ pivot_table() 함수를 활용한 그룹집계

Groupby

집계 결과를
보기 좋게 하기 위해
stack이나 unstack 등의
구문을 추가로 작성해야
되는 경우가 있음

Pivot_table

매우 직관적임

③ pivot_table() 함수를 활용한 그룹집계

구문

```
data.pivot_table(index = '연도', columns = '도시', aggfunc = 'sum', values = '총인구')
```

- `index = '연도'` : 로우 인덱스로 활용할 컬럼
- `columns = '도시'` : 컬럼 인덱스로 활용할 컬럼
- `aggfunc = 'sum'` : 적용할 집계 함수
- `values = '총인구'` : 집계함수를 적용할 컬럼

결과

도시	부산	서울
연도		
2013	699	1306
2014	448	1401
2015	776	1355
2016	567	1141
2017	603	1642

* 모든 인자는 List의 형태로 복수 개 지정 가능

③ pivot_table() 함수를 활용한 그룹집계

연도별, 도시별 총인구의 합계는?

구문

```
data.pivot_table(index = '연도', columns = '도시', aggfunc = 'sum', values = '총인구')
```

결과

도시	부산	서울
연도		
2013	699	1306
2014	448	1401
2015	776	1355
2016	567	1141
2017	603	1642

결과에 보시는 바와 같이 깔끔하게 연도별, 도시별 총인구의 sum을 구할 수가 있습니다.

pivot_table() 도시별, 연도별로 총인구수 출력

Jupyter [W6-3] (실습 전) 그룹집계 (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Run Code

종로구	75.8	113.2
중구	112.8	65.6

In [15]:
```
# 실습 6. 도시별, 연도별로 총인구수 출력
#data.groupby(['도시','연도'])['총인구'].sum().unstack('도시')
data.pivot_table(index = '연도', columns = '도시',
                 aggfunc = 'sum', values = '총인구')
```

Out[15]:

도시	부산	서울
연도		
2016	832	1595
2017	791	1517
2018	790	1592
2019	737	1456
2020	896	1436

검색하려면 여기에 입력하십시오.

pivot_table() 연도별 전체의 남자인구, 여자인구, 총인구수 구하기

◎ pivot_table( ) : groupby 컬럼이 1개(연도)이므로,
index나 columns 인자 중 하나만 사용

2019	1053	1140	2193
2020	1218	1114	2332

In [16]:
```
data.pivot_table(index = '연도', aggfunc = 'sum',
                 values = ['남자인구','여자인구','총인구'])
```

Out[16]:

	남자인구	여자인구	총인구
연도			
2016	1269	1158	2427
2017	1167	1141	2308
2018	1210	1172	2382
2019	1053	1140	2193
2020	1218	1114	2332

In [ ]:

pivot_table() 연도별 전체의 남자인구, 여자인구, 총인구수 구하기

◎ 집계 함수를 2개 이상 사용 가능 (리스트 형태)

2019	1053	1140	2193
2020	1218	1114	2332

In [17]:
```
data.pivot_table(index = '연도', aggfunc = ['sum','mean'] ,
                 values = ['남자인구','여자인구','총인구'])
```

Out[17]:

	sum			mean		
	남자인구	여자인구	총인구	남자인구	여자인구	총인구
연도						
2016	1269	1158	2427	105.750000	96.500000	202.250000
2017	1167	1141	2308	97.250000	95.083333	192.333333
2018	1210	1172	2382	100.833333	97.666667	198.500000
2019	1053	1140	2193	87.750000	95.000000	182.750000
2020	1218	1114	2332	101.500000	92.833333	194.333333

pivot_table()

도시/자치구별로 남자인구의 평균을 구한 후, 남자인구가 가장 많은 도시 및 자치구 Top 5 찾기

◎ 평균 구하기

✔ index : [도시, 자치구], columns : X, 남자인구에 평균(mean) 적용

In [19]:
```
# 실습 7. 도시/자치구 별로 남자인구의 평균을 구한 후,
# 남자 인구가 가장 많은 도시 및 자치구 Top 5 찾기
data.pivot_table(index = ['도시','자치구'],
                 aggfunc = 'mean', values = '남자인구')
```

Out[19]:

		남자인구
도시	자치구	
부산	동래구	91.4
	수영구	90.2
	중구	97.2
	해운대구	107.2
서울	강남구	109.6
	도봉구	101.8
	동작구	87.4

pivot_table()

도시/자치구별로 남자인구의 평균을 구한 후, 남자인구가 가장 많은 도시 및 자치구 Top 5 찾기

◎ Top 5 찾기(정렬)

✔ sort_values : 내림차순으로 남자인구 정렬하고 head(5)

```
In [21]: # 실습 7. 도시/자치구 별로 남자인구의 평균을 구한 후,
         # 남자 인구가 가장 많은 도시 및 자치구 Top 5 찾기
         data.pivot_table(index = ['도시','자치구'],
                          aggfunc = 'mean', values = '남자인구').sort_values(by = '남자인구', asc
```

Out[21]:

		남자인구
도시	자치구	
서울	송파구	115.2
	영등포구	112.8
	중구	112.8
	강남구	109.6
부산	해운대구	107.2

```
In [7]: # 실습 8. 여자가 남자보다 많은 도시/자치구 상위 3개 찾기
```

pivot_table()

도시/자치구별로 남자인구의 평균을 구한 후, 남자인구가 가장 많은 도시 및 자치구 Top 5 찾기

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Run Code

2020 896 1436

In [22]:
```
# 실습 7. 도시/자치구 별로 남자인구의 평균을 구한 후,
# 남자 인구가 가장 많은 도시 및 자치구 Top 5 찾기
data.pivot_table(index = ['도시','자치구'],
                 aggfunc = 'mean', values = '남자인구')₩
            .sort_values(by = '남자인구', ascending = False).head()
```

Out[22]:

		남자인구
도시	자치구	
서울	송파구	115.2
	영등포구	112.8
	중구	112.8
	강남구	109.6
부산	해운대구	107.2

주의하기

✔ 역슬래시(₩)는 하나의 구문을 줄바꿈할 때 사용하며, 구문이 콤마로 구문되는 경우에는 역슬래시(₩) 없이 줄바꿈 가능

검색하려면 여기에 입력하십시오.

오후 7:06 2021-02-09

pivot_table() 여자가 남자보다 많은 도시/자치구 상위 3개 찾기

◎ 여자가 남자보다 많은 도시 찾기(5년 평균)

- ✔ 여자와 남자 인구 차이를 계산하여 새로운 columns(남녀차이) 추가
- ✔ + : 여자 인구가 많은 곳, - : 여자 인구가 적은 곳

```
In [25]: # 문제 8. 여자가 남자보다 많은 도시/자치구 상위 3개 찾기
         data['남녀차이'] = data.여자인구 - data.남자인구
```

```
In [26]: data
```

Out[26]:

	도시	자치구	연도	남자인구	여자인구	총인구	남녀차이
0	서울	강남구	2016	61	70	131	9
1	서울	강남구	2017	119	116	235	-3
2	서울	강남구	2018	134	141	275	7
3	서울	강남구	2019	119	116	235	-3
4	서울	강남구	2020	115	144	259	29
5	서울	서대문구	2016	114	95	209	-19

pivot_table() 여자가 남자보다 많은 도시/자치구 상위 3개 찾기

◎ 여자가 남자보다 많은 도시 찾기(5년 평균)

- ✔ index : 도시, 자치구
- ✔ 남녀차이 columns의 평균 구하기

Code

부산	해운대구	107.2

In [25]:
```
# 실습 8. 여자가 남자보다 많은 도시/자치구 상위 3개 찾기
data['남녀차이'] = data.여자인구 - data.남자인구
```

In [27]:
```
data.pivot_table(index = ['도시','자치구'],
                 aggfunc = 'mean', values = '남녀차이')
```

Out[27]:

		남녀차이
도시	자치구	
부산	동래구	31.4
	수영구	8.2
	중구	-1.0
	해운대구	-1.4

pivot_table() 여자가 남자보다 많은 도시/자치구 상위 3개 찾기

◎ 여자가 남자보다 많은 도시 찾기(5년 평균)

✔ -: 남자 인구가 많은 곳, +: 여자 인구가 많은 곳

In [25]:
```
# 실습 8. 여자가 남자보다 많은 도시/자치구 상위 3개 찾기
data['남녀차이'] = data.여자인구 - data.남자인구
```

In [27]:
```
data.pivot_table(index = ['도시','자치구'],
                 aggfunc = 'mean', values = '남녀차이').W
```

Out[27]:

		남녀차이
도시	자치구	
부산	동래구	31.4
	수영구	8.2
	중구	-1.0
	해운대구	-1.4
서울	강남구	7.8

pivot_table() 여자가 남자보다 많은 도시/자치구 상위 3개 찾기

◎ 상위 3개 찾기(정렬)

✔ sort_values를 이용하여 남녀차이로 내림차순하여 head(3)

In [25]:
```
# 실습 8. 여자가 남자보다 많은 도시/자치구 상위 3개 찾기
data['남녀차이'] = data.여자인구 - data.남자인구
```

In [29]:
```
data.pivot_table(index = ['도시','자치구'],
                 aggfunc = 'mean', values = '남녀차이').W
        sort_values(by = '남녀차이', ascending = False).head(3)
```

Out[29]:

		남녀차이
도시	자치구	
서울	종로구	37.4
부산	동래구	31.4
	수영구	8.2

In [ ]: