빅데이터개론2

파이선 pandas 데이터프레임

2022년 10월 3일

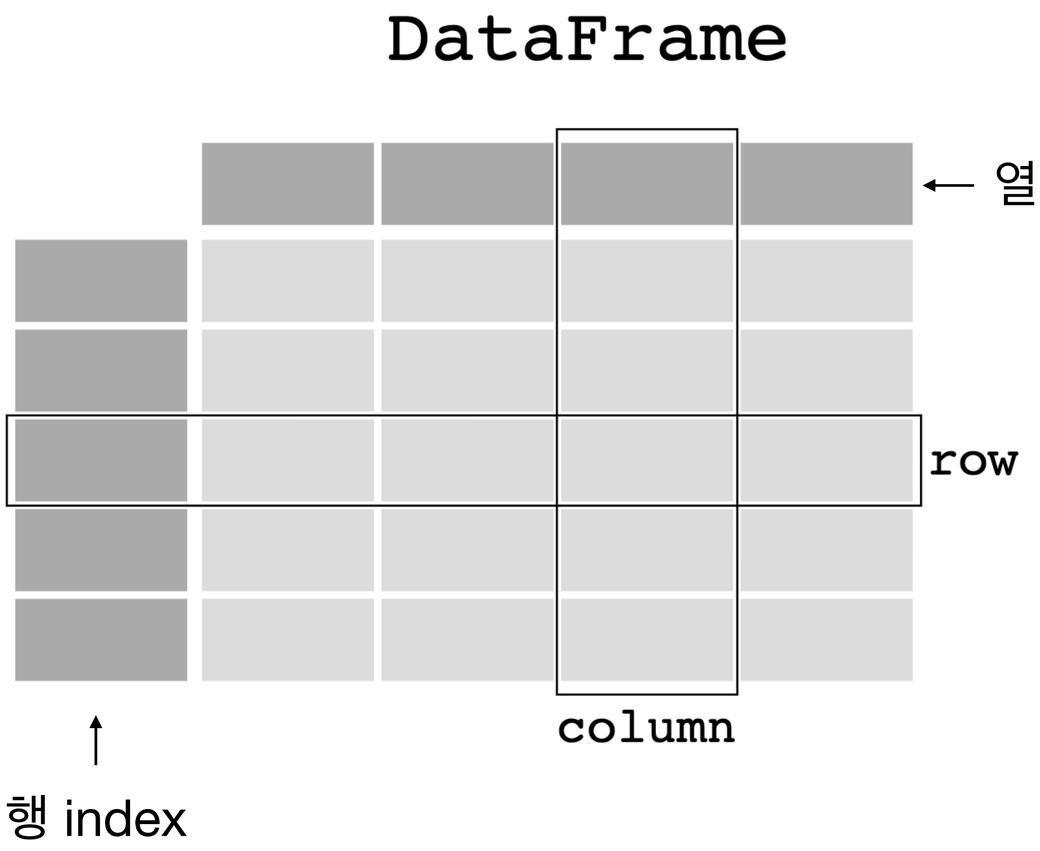
Pandas 데이터프레임

- 구조
- 열과 행의 선택
- 정렬
- 열과 행 지우기
- 열 이름 바꾸기
- 새로운 열 만들기
- 열의 요약
- 그룹의 생성과 요약
- 메소드 - 연쇄적 처리
- 두 데이터프레임의 결합

참조:

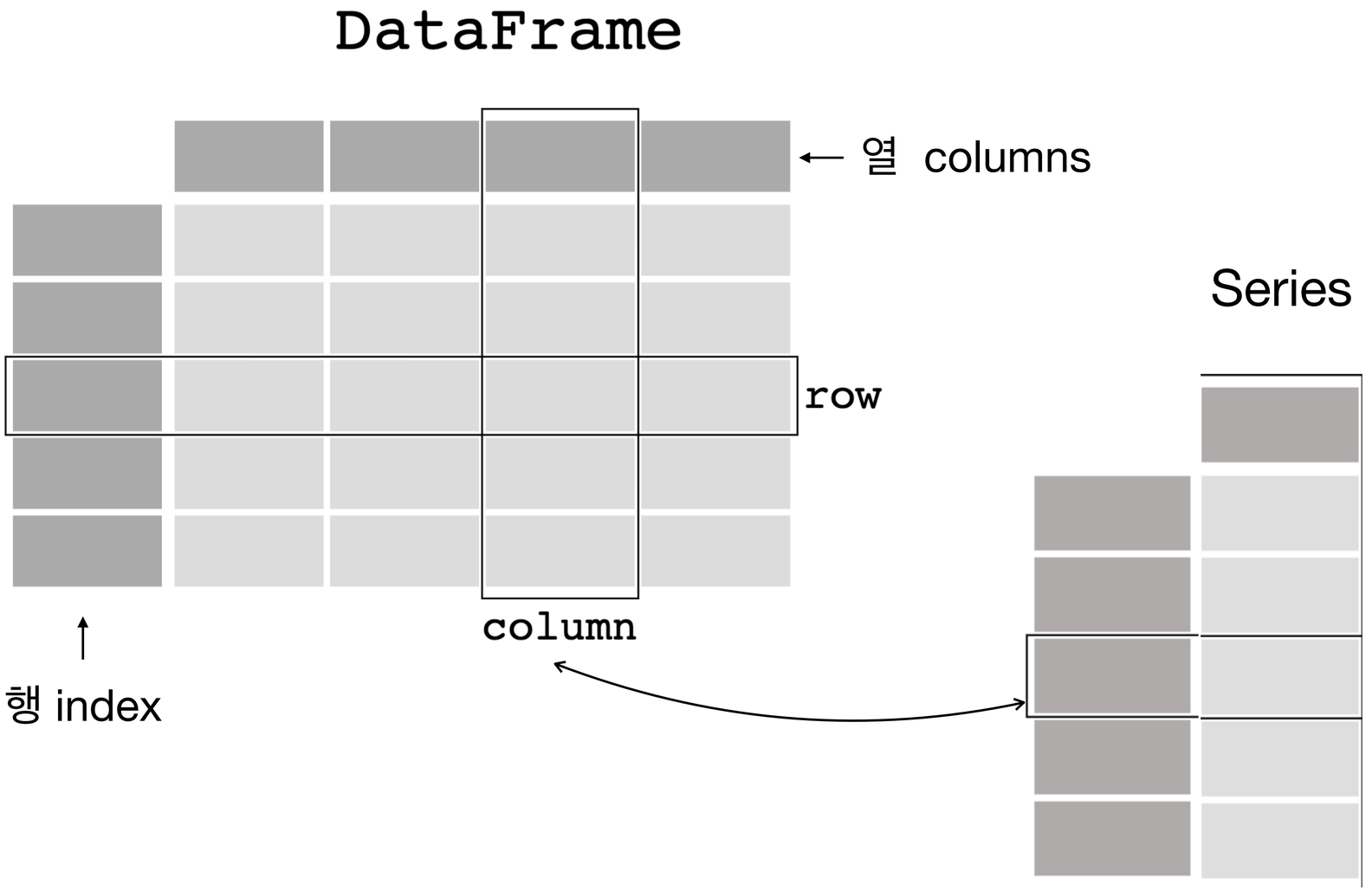
1. pandas Getting started tutorials: https://pandas.pydata.org/docs/getting_started/intro_tutorials/index.html
2. 온라인 교과서: https://uos-bigdata.github.io/bigdatabook/chapters/intro.html

데이터프레임 - 구조



	일시	평균기온(°C)	최고기온 평균(°C)	최저기온 평균(°C)	강수량(mm)
0	2020년	13.2	17.9	9.4	1651.1
1	2019년	13.5	18.5	9.3	891.3
2	2018년	12.9	17.9	8.8	1284.1
3	2017년	13.0	18.1	8.8	1233.2
4	2016년	13.6	18.5	9.4	991.7

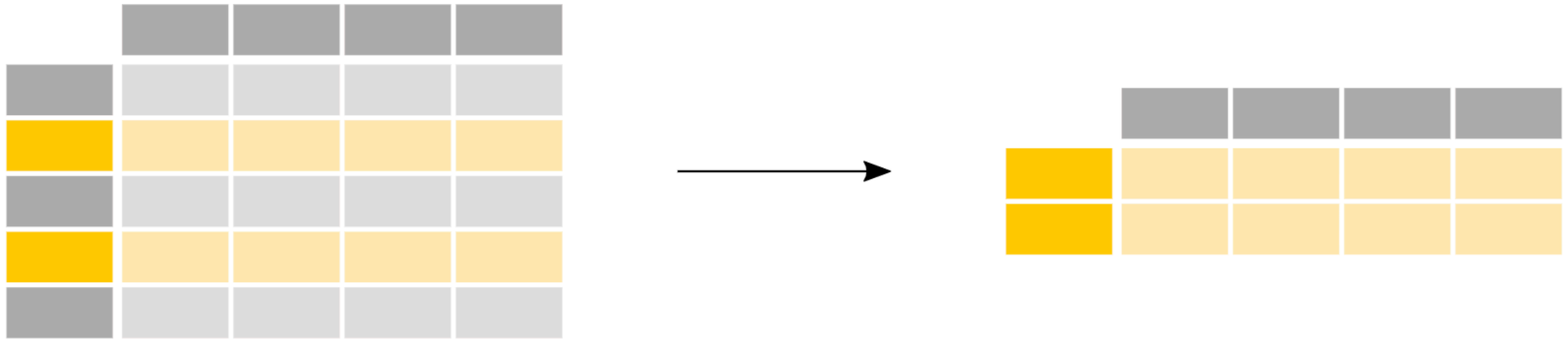
데이터프레임 - 구조



열 선택

```
df[['age', 'height']]
```

행 선택



```
df[ (df['sex'] == 'M') & (df['height'] >= 160.0) ]
```

열과 행 선택 - 조건과 이름

```
df.loc[ (df['height'] >= 170.0) , 'name']
```

열과 행 선택 - 인덱스

```
df.iloc[3,1:3]
```

정렬

```
house.sort_values(by = [ '일반가구_계'], ascending=False)
```

정렬 - inplace

```
house.sort_values(by = [ '일반가구_계'], ascending=False,
                  inplace=True)
```

열 지우기 - inplace

```
house.drop(columns = ["일반가구_계"], inplace= True)
```

행 지우기 - inplace

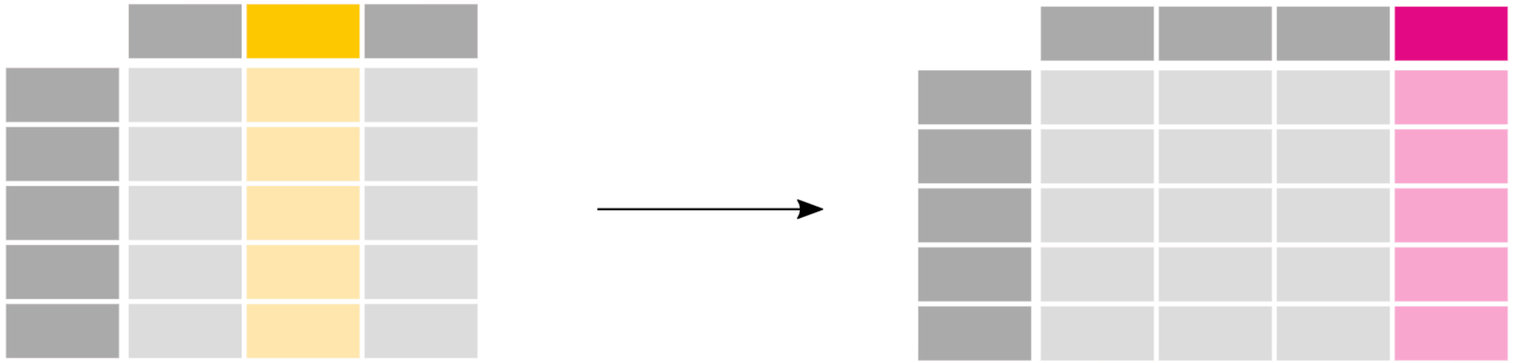
```
index_for_delete = house[house["행정구역별(읍면동)"] == '전국'].index

house.drop( index = index_for_delete, inplace= True )
```

열 이름 바꾸기 - inplace

```
house.rename( columns={
   "시점" : "year", "행정구역별(읍면동)" : "region",
   "1인" : "p1", "2인" : "p2", "3인" : "p2", "4인" : "p4",
   "5인" : "p5", "6인" : "p6", "7인 이상" : "p7plus"},
   inplace=True)
```

```
house.columns = ["year", "region", "p1", "p2", "p2", "p4", "p5", "p6", "p7plus"]
```

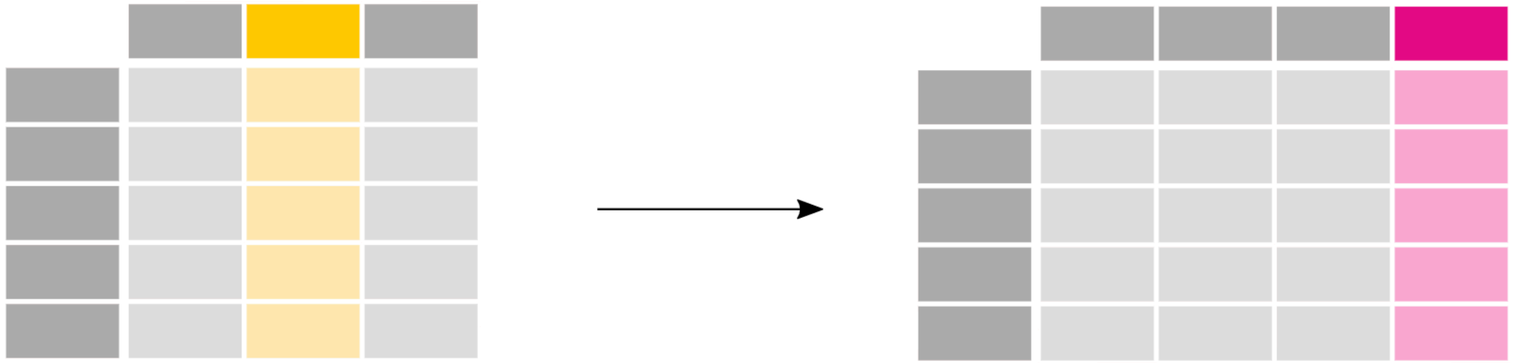
새로운 열 만들기



```
weather["avg_t_F"] = 32.0 + 1.8 * weather['avg_t']
```

```
weather["avg_t_minmax"] = (weather.max_t + weather.min_t) / 2.0
```

새로운 열 만들기 - 메소드

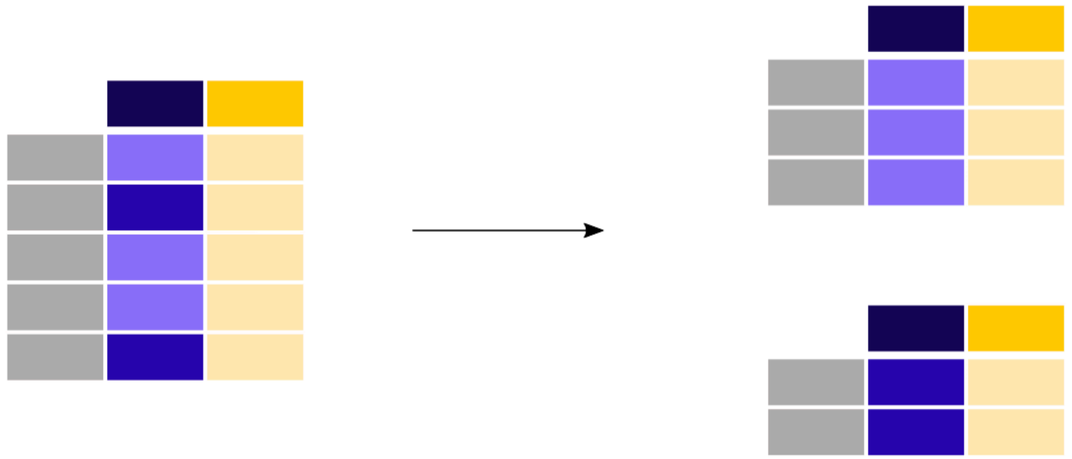


```
weather["avg_t_minmix_2"] = weather[ ["max_t", min_t"] ].sum(axis=1) /2.0
```

열의 요약

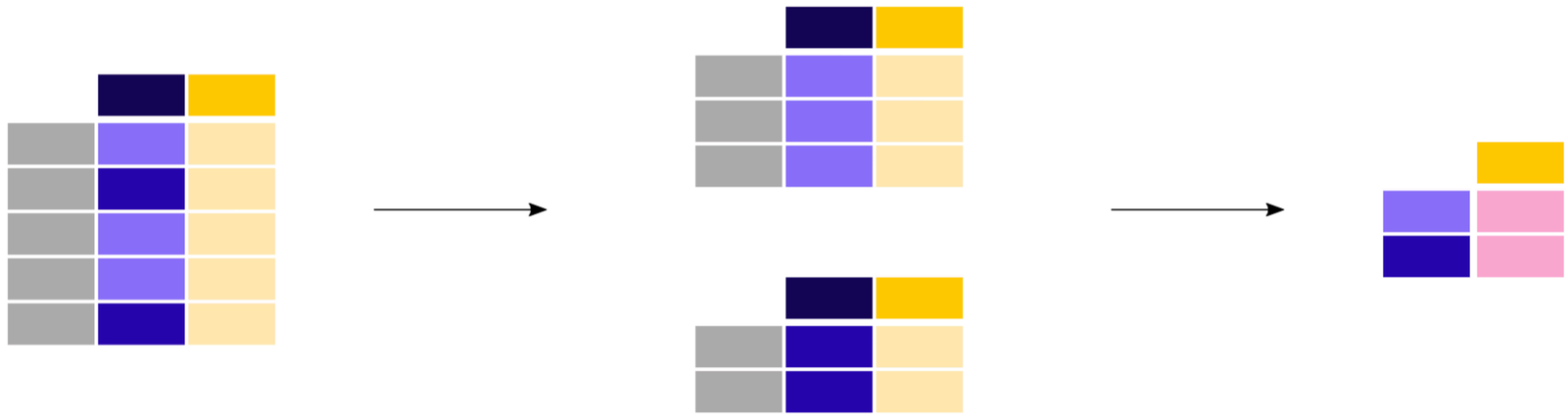
```
weather['avg_t'].mean(axis=0)

weather[['avg_t', 'max_t']].mean(axis=0)

weather[['avg_t', 'max_t']].mean()
```

그룹의 생성



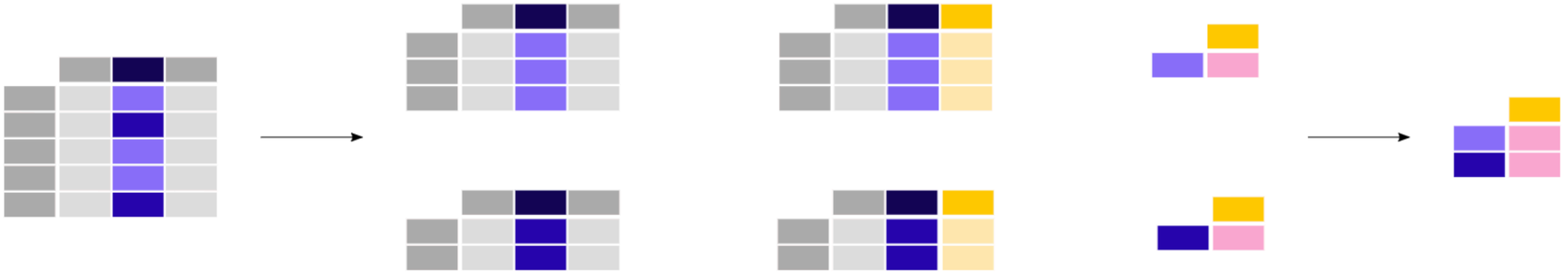
```
house_grp = house.groupby( by=["year"] )
```

그룹별 요약



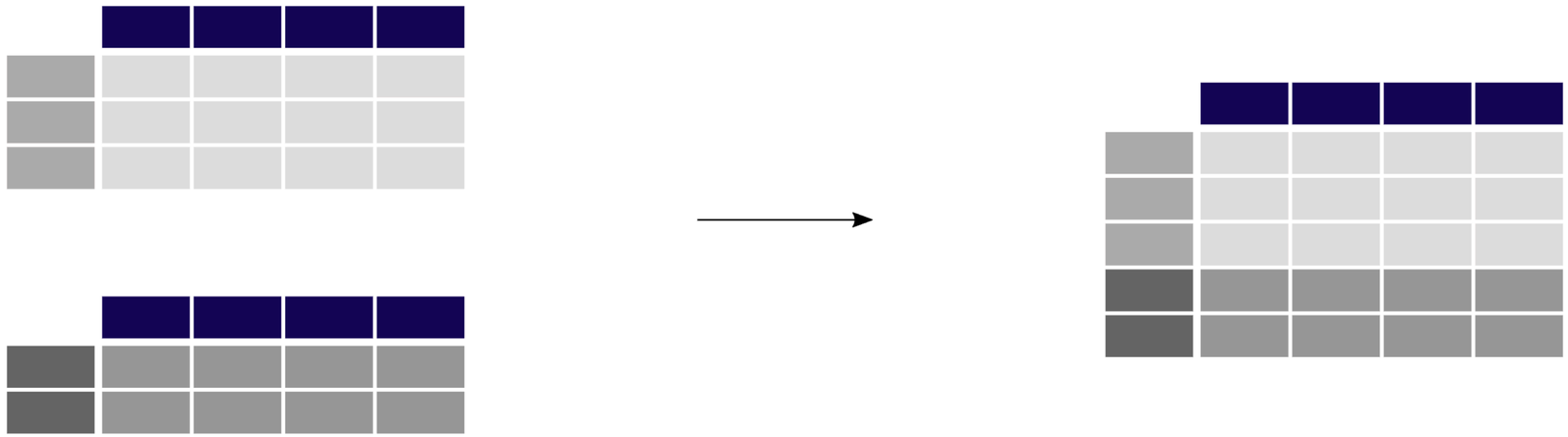
```
house_grp = house.groupby( by=["year"] )

house_grp.sum()
```

메소드 - 연쇄적 처리 (chaining)



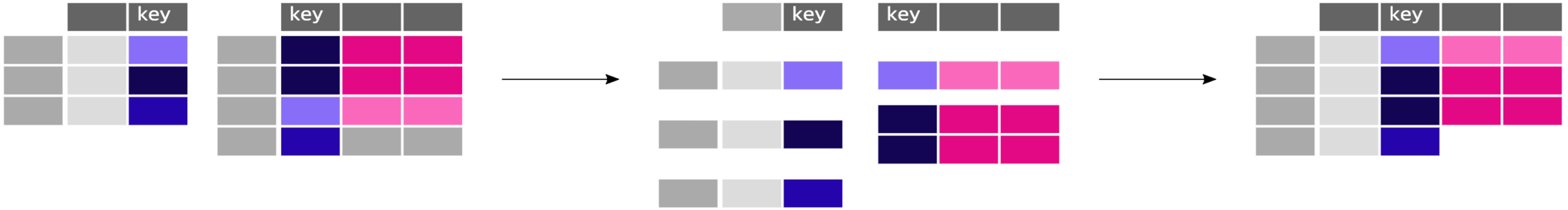
```
house.groupby( by=["year"] ).sum()
```

데이터프레임 - 결함 -행단위



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

데이터프레임 - 결함 -열단위




```
pd.merge(df_left, df_right, on="name", how='left')
pd.merge(df_left, df_right, on="name", how='right')
pd.merge(df_left, df_right, on="name", how='inner')
pd.merge(df_left, df_right, on="name", how='outer')
```