BIG DATA Technology

전국 신규 민간 아파트 분양가격 동향

빅테이터 프로젝트
201844071 김규준

01. 전국 신규 민간 아파트 분양가격 동향

- 공공데이터를 활용해 전혀 다른 두 개의 데이터를 가져와서 전처리 하고 병합
- 수치형 데이터와 범주형 데이터를 시각화
- 판다스를 통해 데이터를 요약하고 분석하기
- 수치형 데이터와 범주형 데이터 다루기
- 막대그래프(bar plot), 선그래프(line plot), 산포도(scatter plot) 등 시각화

02. 데이터분석에 필요한 데이터셋

- http://bit.ly/open-data-set-folder (전국신규민간아파트분양가격동향)

03. 빅데이터 프로젝트 Github주소

- mbc2579 · GitHub

01. 전국 신규 민간 아파트 분양가격 동향

- 데이터 요약하기

```
[11]  df_last.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4335 entries, 0 to 4334
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   지역명         4335 non-null   object
 1   규모구분        4335 non-null   object
 2   연도          4335 non-null   int64 
 3   월           4335 non-null   int64 
 4   분양가격(㎡)  4058 non-null   object
dtypes: int64(2), object(3)
memory usage: 169.5+ KB
```

- 결측치 보기

```
[15]  df_last.isnull()
```

	지역명	규모구분	연도	월	분양가격(㎡)
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
4330	False	False	False	False	False
4331	False	False	False	False	True
4332	False	False	False	False	False
4333	False	False	False	False	True
4334	False	False	False	False	False

4335 rows × 5 columns

01. 전국 신규 민간 아파트 분양가격 동향

- groupby로 데이터 집계하기

[29]
```
# 지역명으로 분양가격의 평균을 구하고 막대그래프(bar)로 시각화
# df.groupby(["인덱스로 사용할 컬럼명"])["계산할 컬럼 값"].연산()
df_last.groupby(["지역명"])["평당분양가격"].mean()
```

```
지역명
강원      7890.750000
경기     13356.895200
경남      9268.778138
경북      8376.536515
광주      9951.535821
대구     11980.895455
대전     10253.333333
부산     12087.121200
서울     23599.976400
세종      9796.516456
울산     10014.902013
인천     11915.320732
전남      7565.316532
전북      7724.235484
제주     11241.276712
충남      8233.651883
충북      7634.655600
Name: 평당분양가격, dtype: float64
```

[30]
```
# 전용면적으로 분양가격의 평균을 구함
df_last.groupby(["전용면적"])["평당분양가격"].mean()
```

```
전용면적
102㎡~         11517.705634
60㎡           10375.137421
60㎡~85㎡       10271.040071
85㎡~102㎡      11097.599573
전체            10276.086207
Name: 평당분양가격, dtype: float64
```

[32]
```
# 연도, 지역명으로 평당분양가격의 평균을 구함

g = df_last.groupby(["연도", "지역명"])["평당분양가격"].mean()
g
# g.unstack().transpose()
```

```
연도    지역명
2015  강원      7188.060000
      경기     11060.940000
      경남      8459.220000
      경북      7464.160000
      광주      7916.700000
                   ...
2019  전남      8219.275862
      전북      8532.260000
      제주     11828.469231
      충남      8748.840000
      충북      7970.875000
Name: 평당분양가격, Length: 85, dtype: float64
```

01. 전국 신규 민간 아파트 분양가격 동향

- pivot table로 데이터 집계하기

```
pd.pivot_table(df_last, index=["지역명"], values=["평당분양가격"], aggfunc="mean")
```

지역명	평당분양가격
강원	7890.750000
경기	13356.895200
경남	9268.778138
경북	8376.536515
광주	9951.535821
대구	11980.895455
대전	10253.333333
부산	12087.121200
서울	23599.976400
세종	9796.516456
울산	10014.902013
인천	11915.320732
전남	7565.316532
전북	7724.235484
제주	11241.276712
충남	8233.651883
충북	7634.655600

```
[38] pd.pivot_table(df_last, index="전용면적", values="평당분양가격")
```

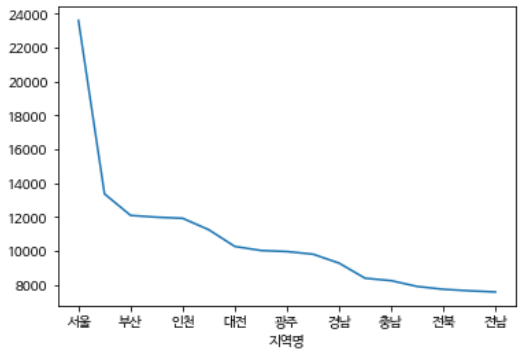
전용면적	평당분양가격
102㎡~	11517.705634
60㎡	10375.137421
60㎡~85㎡	10271.040071
85㎡~102㎡	11097.599573
전체	10276.086207

01. 전국 신규 민간 아파트 분양가격 동향

- Pandas로 시각화 하기 – 선그래프와 막대그래프

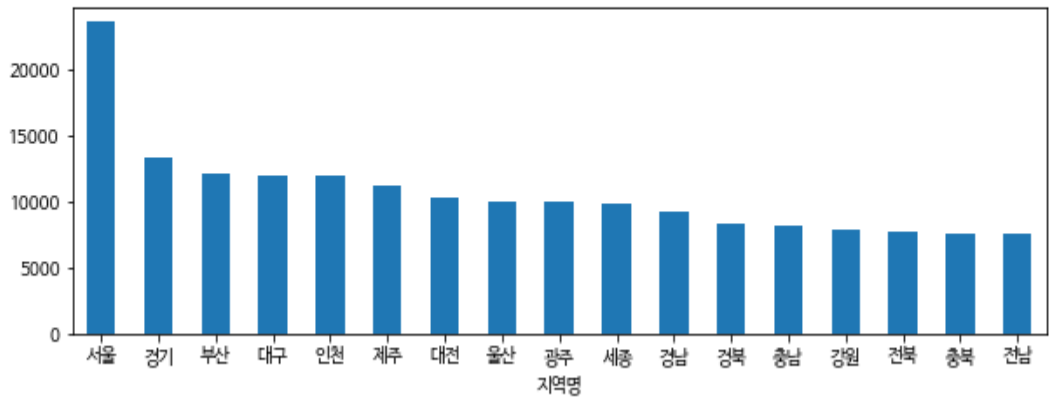
```
[44] # 지역명으로 분양가격의 평균을 구하고 선그래프로 시각화
     g = df_last.groupby(["지역명"])["평당분양가격"].mean().sort_values(ascending=False)
     g.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2f5b7f160>
```




```
[45] # 지역명으로 분양가격의 평균을 구하고 막대그래프(bar)로 시각화
     g.plot.bar(rot=0, figsize=(10, 3))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2f5b17d30>
```



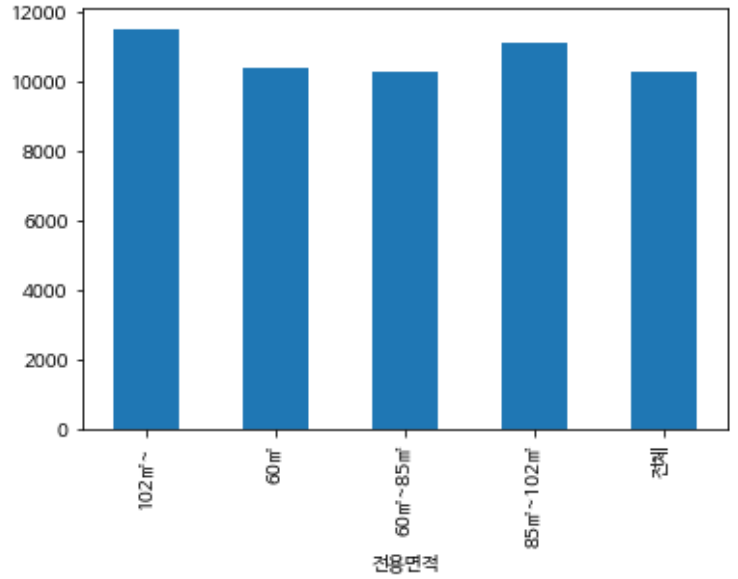
01. 전국 신규 민간 아파트 분양가격 동향

- Pandas로 시각화 하기 – 선그래프와 막대그래프

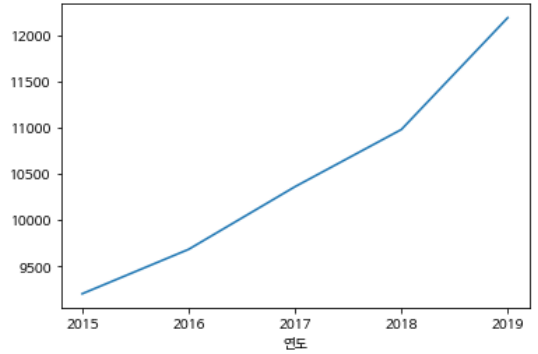
전용면적별 분양가격의 평균값을 구하고 그래프로 출력

```
# 전용면적으로 분양가격의 평균을 구하고 막대그래프(bar)로 시각화
df_last.groupby(["전용면적"])["평당분양가격"].mean().plot.bar()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2f566b700>
```




```
[47] # 연도별 분양가격의 평균을 구하고 막대그래프(bar)로 시각화
     # 연도에 소숫점이 생기지 않게 표시하고자 한다면 ax.xaxis.set_major_locator를 사용해서 integer로 설정
     from matplotlib.ticker import MaxNLocator

     ax = plt.figure().gca()
     df_last.groupby(["연도"])["평당분양가격"].mean().plot()
     ax.xaxis.set_major_locator(MaxNLocator(integer=True))
```

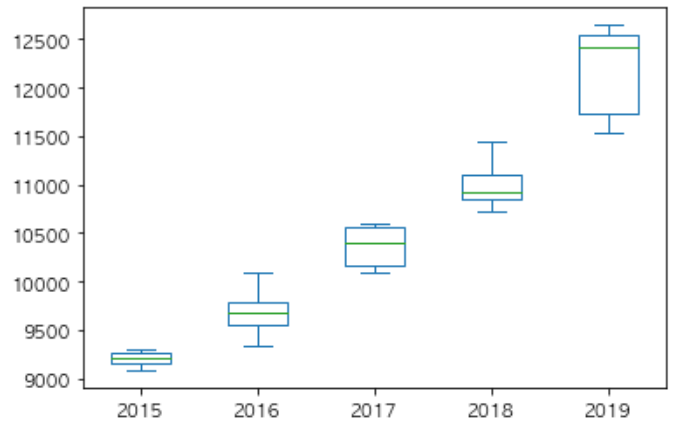


01. 전국 신규 민간 아파트 분양가격 동향

- box-and-whisker plot | diagram

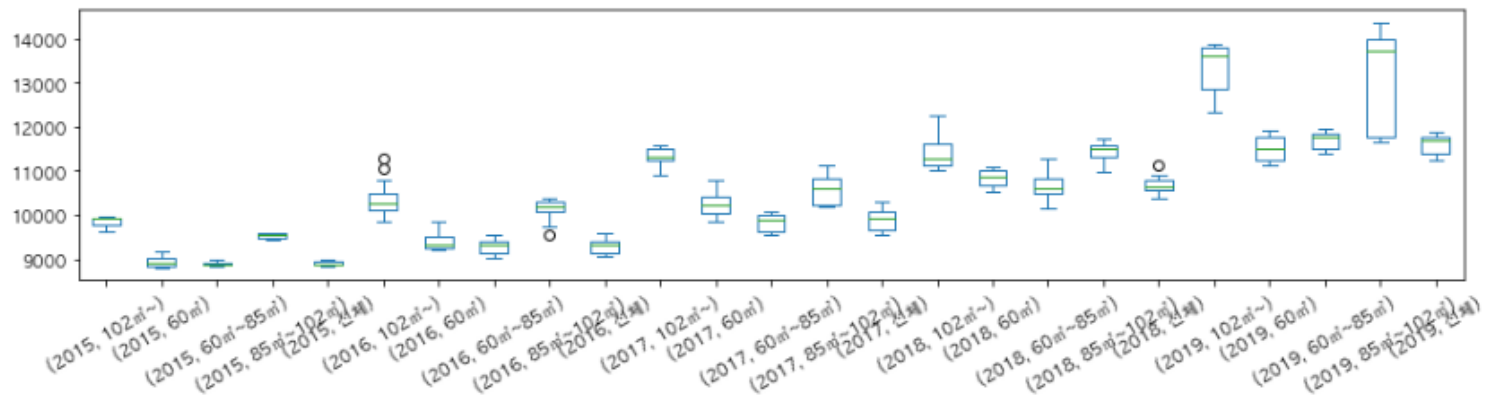
```
[ ]  df_last.pivot_table(index="월", columns="연도", values="평당분양가격").plot.box()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb1a83192d0>
```




```
[ ]  p = df_last.pivot_table(index="월", columns=["연도", "전용면적"], values="평당분양가격")
     p.plot.box(figsize=(15, 3), rot=30)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb1c8e439d0>
```

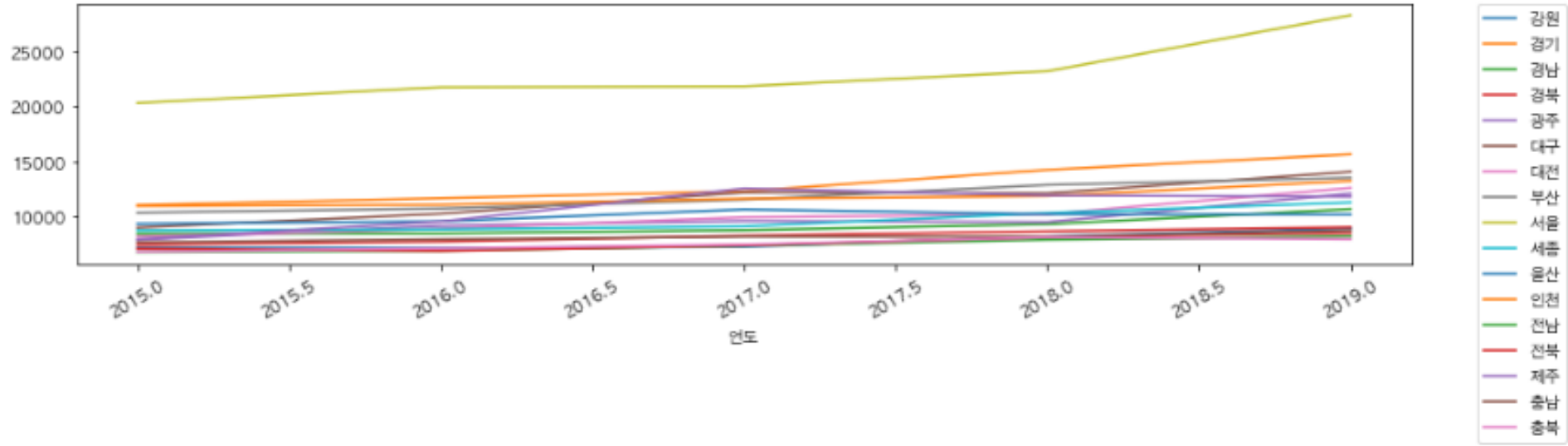


01. 전국 신규 민간 아파트 분양가격 동향

- box-and-whisker plot | diagram

```
[ ]  p = df_last.pivot_table(index="연도", columns="지역명", values="평당분양가격")
     p.plot(figsize=(15, 3), rot=30)
     # 그래프의 밖에 legend 표시하도록 설정
     plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```

```
<matplotlib.legend.Legend at 0x7fb1c8e76510>
```



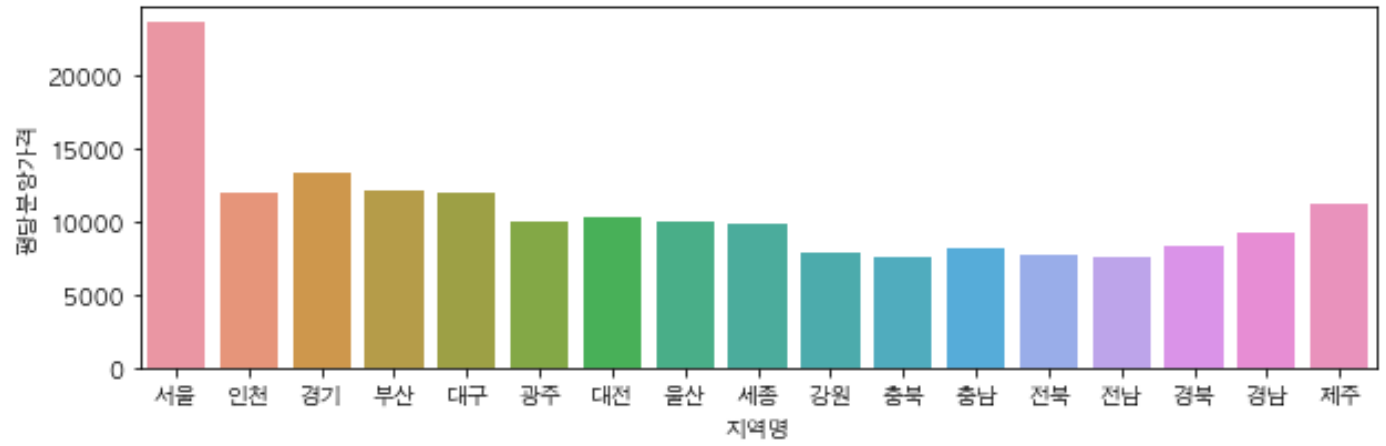
01. 전국 신규 민간 아파트 분양가격 동향

- Seaborn으로 시각화

```
[ ]  import seaborn as sns
     %matplotlib inline
```

```
[ ]  # barplot으로 지역별 평당분양가격을 그려봅니다.
     plt.figure(figsize=(10, 3))
     sns.barplot(data=df_last, x="지역명", y="평당분양가격", ci=None)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb198d8f790>
```

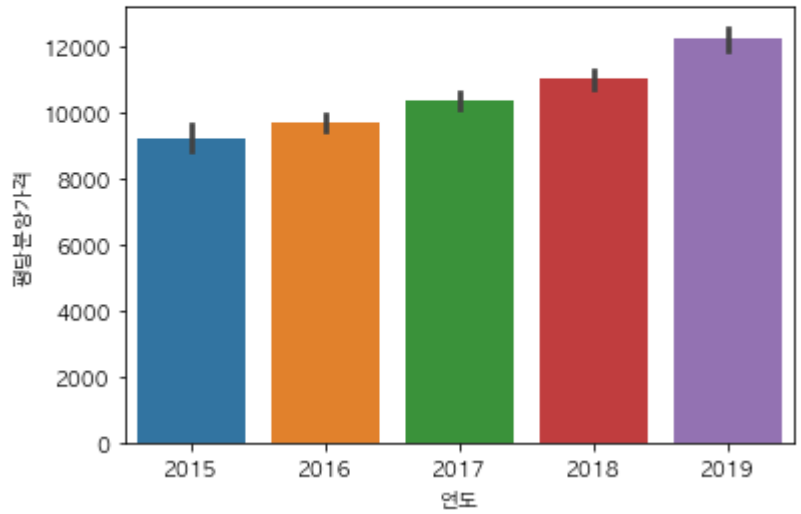


01. 전국 신규 민간 아파트 분양가격 동향

- Seaborn으로 시각화

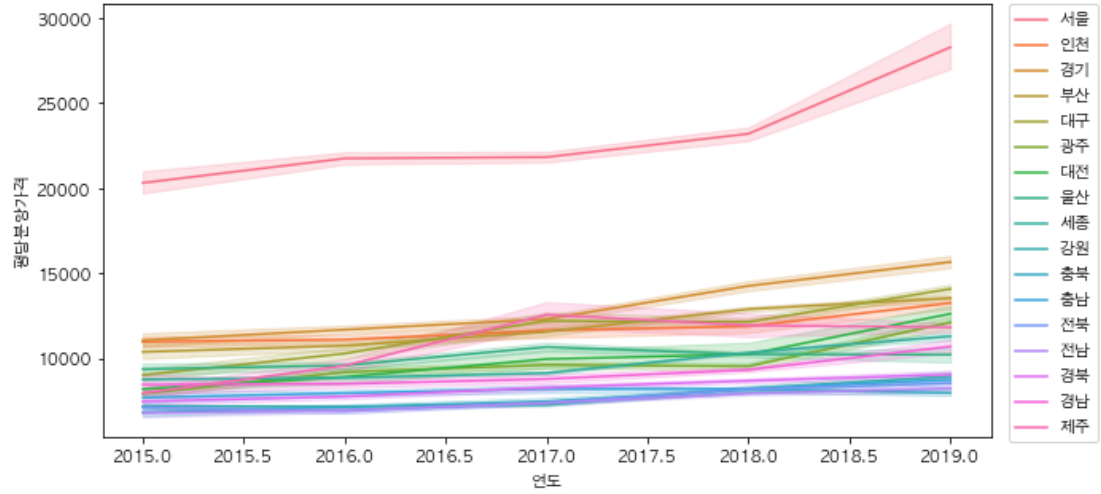
```
[ ]  # barplot으로 연도별 평당분양가격을 그려봅니다.

     sns.barplot(data=df_last, x="연도", y="평당분양가격")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb18965b850>
```




```
[ ]  # lineplot으로 연도별 평당분양가격을 그려봅니다.
     # hue 옵션을 통해 지역별로 다르게 표시해 봅니다.
     plt.figure(figsize=(10, 5))
     sns.lineplot(data=df_last, x="연도", y="평당분양가격", hue="지역명")
     plt.legend(bbox_to_anchor=(1.02, 1), loc=2, borderaxespad=0.)
```

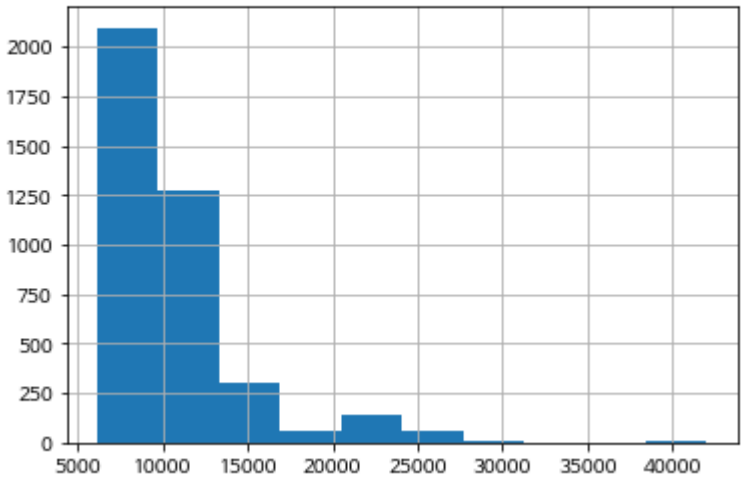
```
<matplotlib.legend.Legend at 0x7fb1b9508c90>
```



01. 전국 신규 민간 아파트 분양가격 동향

- 수치데이터 히스토그램 출력

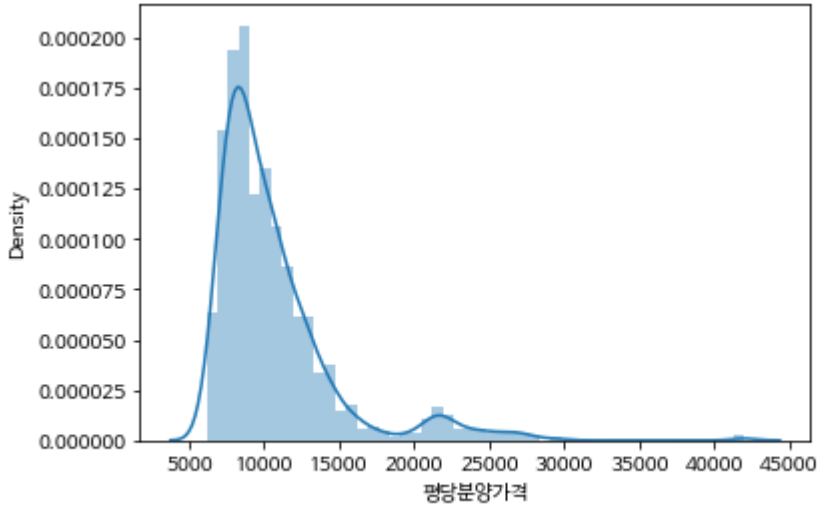
```
[149] h = df_last["평당분양가격"].hist(bins=10)
```




```
[70] # 결측치가 없는 데이터에서 평당분양가격만 가져옴 그리고 price라는 변수에 담음
     # .loc[행]
     # .loc[행, 열]
     price = df_last.loc[df_last["평당분양가격"].notnull(), "평당분양가격"]
```

```
[71] # distplot으로 평당분양가격을 표현

     sns.distplot(price)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2e6c9ee20>
```

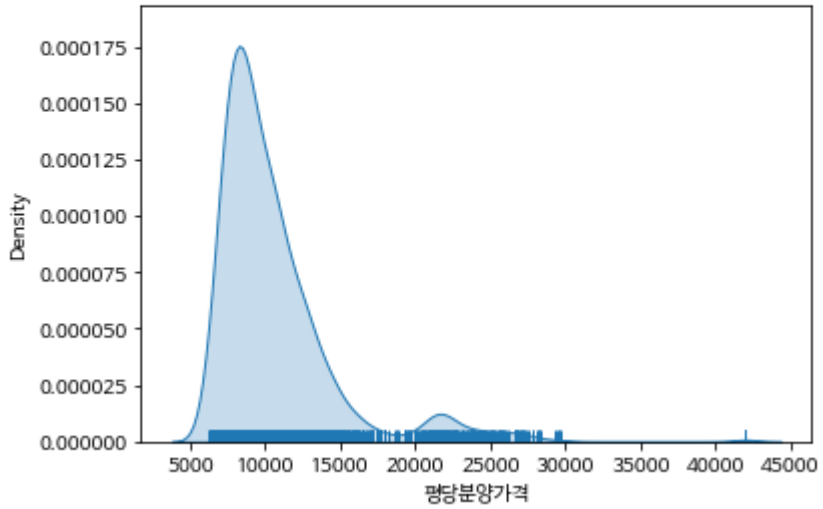


01. 전국 신규 민간 아파트 분양가격 동향

- 수치데이터 히스토그램 출력

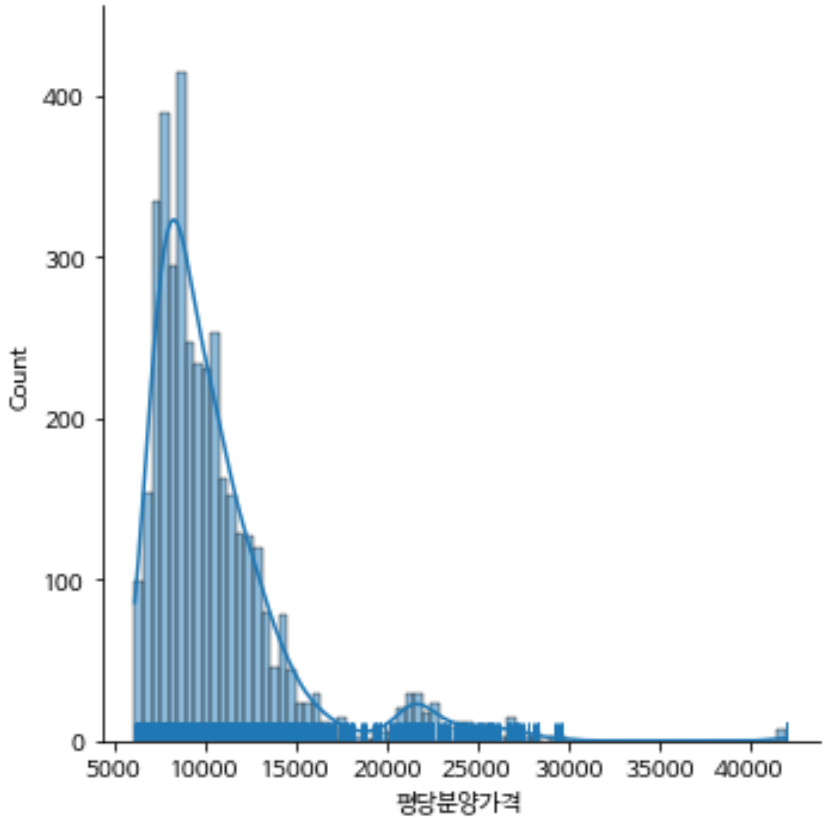
```
[150] sns.kdeplot(price, shade=True)
      sns.rugplot(price)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2e21a7850>
```




```
[73] sns.displot(price, kde=True, rug=True)
```

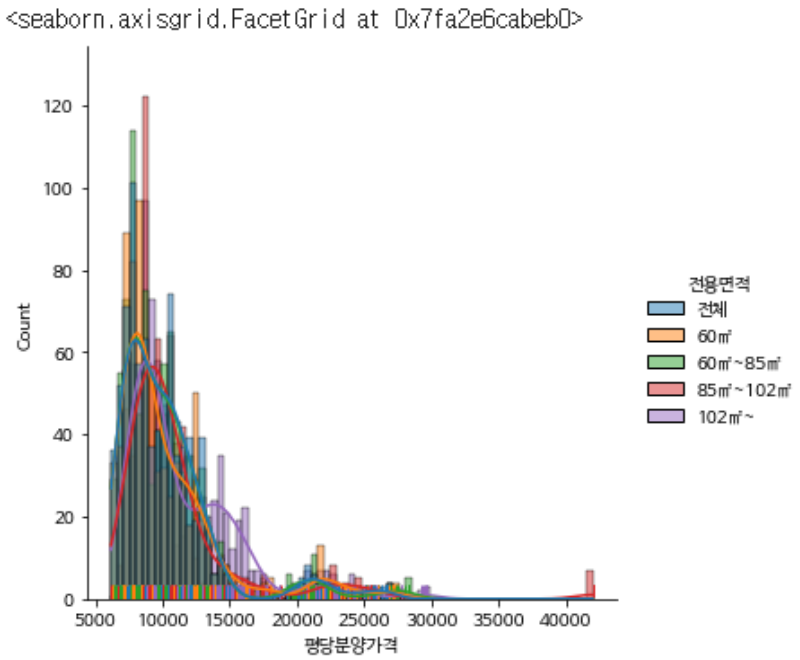
```
<seaborn.axisgrid.FacetGrid at 0x7fa2e69401f0>
```



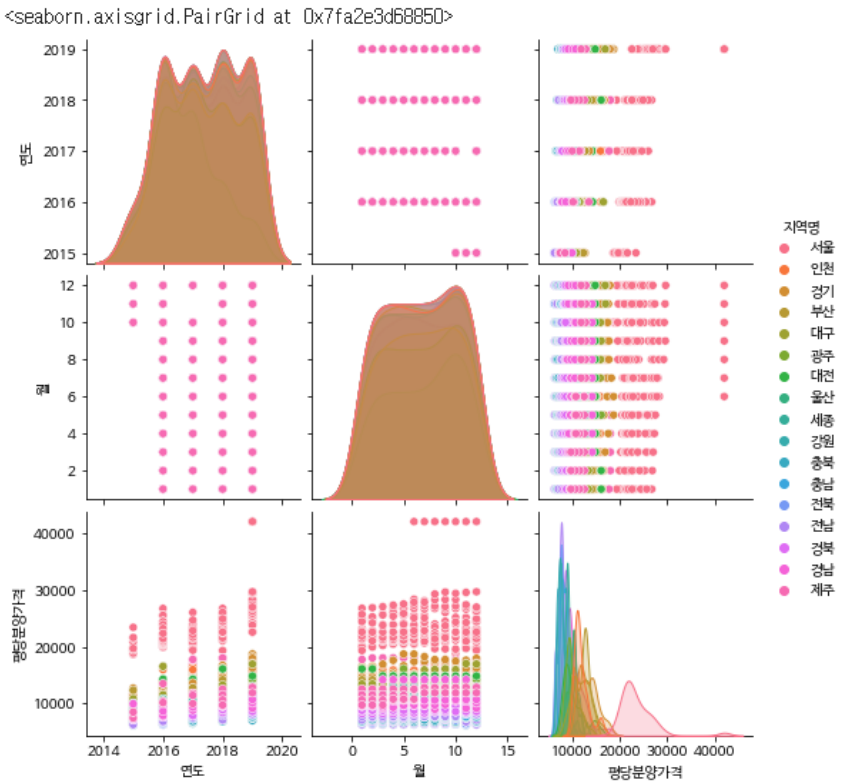
01. 전국 신규 민간 아파트 분양가격 동향

- 수치데이터 히스토그램 출력

```
[74]  sns.displot(data=df_last, x="평당분양가격", kde=True, rug=True, hue="전용면적")
```




```
[78]
df_last_notnull = df_last.loc[df_last["평당분양가격"].notnull(),
                              ["연도", "월", "평당분양가격", "지역명", "전용면적"]]
sns.pairplot(df_last_notnull, hue="지역명")
```



01. 전국 신규 민간 아파트 분양가격 동향

- concat으로 데이터 합치기

[116]
```
# df_first_prepare 와 df_last_prepare 를 합쳐줌

df = pd.concat([df_first_prepare, df_last_prepare])
df.shape
```

```
(1224, 4)
```

[117]
```
# 제대로 합쳐졌는지 미리보기

df
```

	지역명	연도	월	평당분양가격
0	서울	2013	12	18189.0
1	부산	2013	12	8111.0
2	대구	2013	12	8080.0
3	인천	2013	12	10204.0
4	광주	2013	12	6098.0
...	...	...	...	...
4310	전북	2019	12	8144.4
4315	전남	2019	12	8091.6
4320	경북	2019	12	9616.2
4325	경남	2019	12	10107.9
4330	제주	2019	12	12810.6

1224 rows × 4 columns

[118]
```
# 연도별로 데이터가 몇개씩 있는지 value_counts를 통해 셈

df["연도"].value_counts(sort=False)
```

```
2013     17
2014    204
2015    187
2016    204
2017    204
2018    204
2019    204
Name: 연도, dtype: int64
```

01. 전국 신규 민간 아파트 분양가격 동향

- pivot_table 사용

```
[119] # 연도를 인덱스로, 지역명을 컬럼으로 평당분양가격을 피봇테이블로 출력
      t = pd.pivot_table(df, index="연도", columns="지역명",
                         values="평당분양가격").round()
      t
```

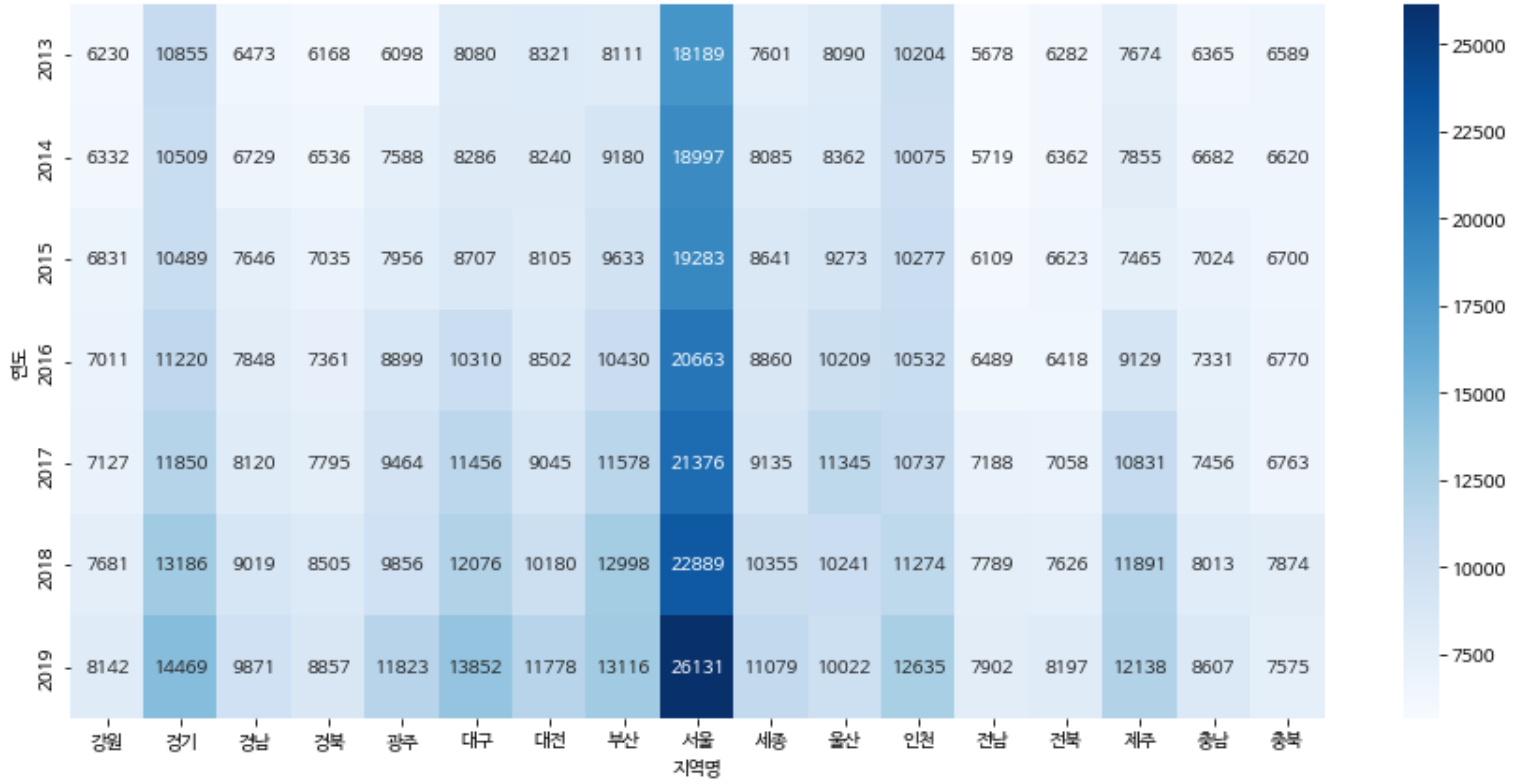
지역명	강원	경기	경남	경북	광주	대구	대전	부산	서울
연도									
2013	6230.0	10855.0	6473.0	6168.0	6098.0	8080.0	8321.0	8111.0	18189.0
2014	6332.0	10509.0	6729.0	6536.0	7588.0	8286.0	8240.0	9180.0	18997.0
2015	6831.0	10489.0	7646.0	7035.0	7956.0	8707.0	8105.0	9633.0	19283.0
2016	7011.0	11220.0	7848.0	7361.0	8899.0	10310.0	8502.0	10430.0	20663.0
2017	7127.0	11850.0	8120.0	7795.0	9464.0	11456.0	9045.0	11578.0	21376.0
2018	7681.0	13186.0	9019.0	8505.0	9856.0	12076.0	10180.0	12998.0	22889.0
2019	8142.0	14469.0	9871.0	8857.0	11823.0	13852.0	11778.0	13116.0	26131.0

01. 전국 신규 민간 아파트 분양가격 동향

- pivot_table 사용

```
[120] # 위에서 그린 피봇테이블을 히트맵으로 표현
      plt.figure(figsize=(15, 7))
      sns.heatmap(t, cmap="Blues", annot=True, fmt=".0f")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2e364a4f0>
```



01. 전국 신규 민간 아파트 분양가격 동향

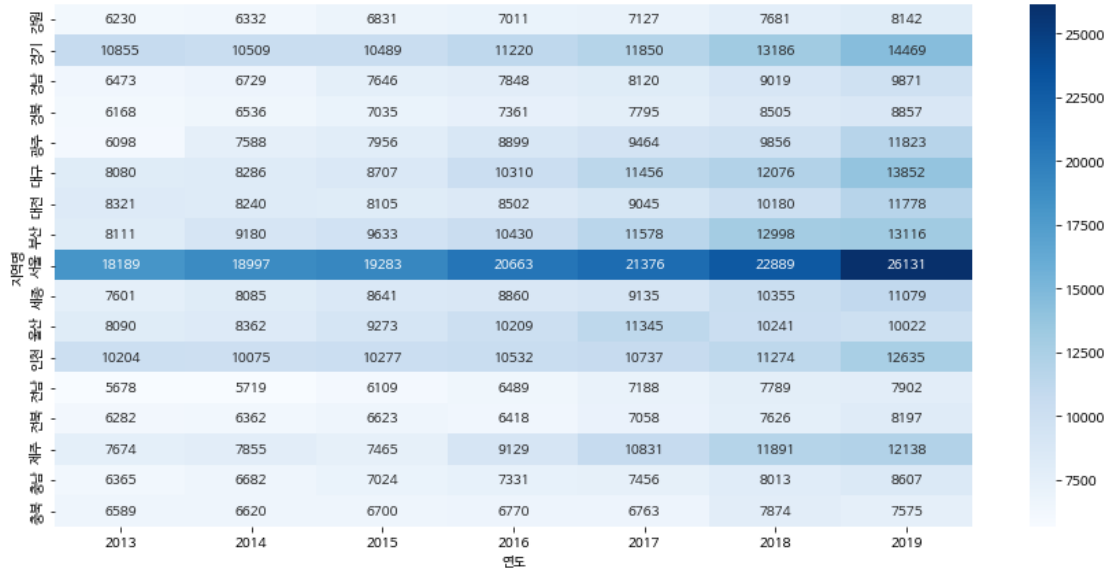
- pivot_table 사용

```
[121] # transpose 를 사용하면 행과 열을 바꿀수있다.

      t.transpose()
```

연도	2013	2014	2015	2016	2017	2018	2019
지역명							
강원	6230.0	6332.0	6831.0	7011.0	7127.0	7681.0	8142.0
경기	10855.0	10509.0	10489.0	11220.0	11850.0	13186.0	14469.0
경남	6473.0	6729.0	7646.0	7848.0	8120.0	9019.0	9871.0
경북	6168.0	6536.0	7035.0	7361.0	7795.0	8505.0	8857.0
광주	6098.0	7588.0	7956.0	8899.0	9464.0	9856.0	11823.0
대구	8080.0	8286.0	8707.0	10310.0	11456.0	12076.0	13852.0
대전	8321.0	8240.0	8105.0	8502.0	9045.0	10180.0	11778.0
부산	8111.0	9180.0	9633.0	10430.0	11578.0	12998.0	13116.0
서울	18189.0	18997.0	19283.0	20663.0	21376.0	22889.0	26131.0
세종	7601.0	8085.0	8641.0	8860.0	9135.0	10355.0	11079.0
울산	8090.0	8362.0	9273.0	10209.0	11345.0	10241.0	10022.0
인천	10204.0	10075.0	10277.0	10532.0	10737.0	11274.0	12635.0
전남	5678.0	5719.0	6109.0	6489.0	7188.0	7789.0	7902.0
전북	6282.0	6362.0	6623.0	6418.0	7058.0	7626.0	8197.0
제주	7674.0	7855.0	7465.0	9129.0	10831.0	11891.0	12138.0
충남	6365.0	6682.0	7024.0	7331.0	7456.0	8013.0	8607.0
충북	6589.0	6620.0	6700.0	6770.0	6763.0	7874.0	7575.0

```
[122] # 바뀐 행과 열을 히트맵으로 표현

      plt.figure(figsize=(15, 7))
      sns.heatmap(t.T, cmap="Blues", annot=True, fmt=".0f")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2e33a3ee0>
```

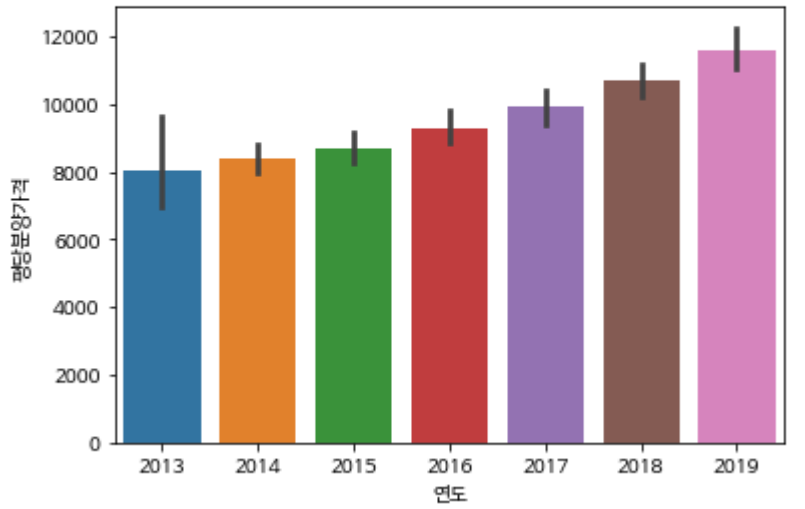


01. 전국 신규 민간 아파트 분양가격 동향

- 연도별 평당분양가격 보기

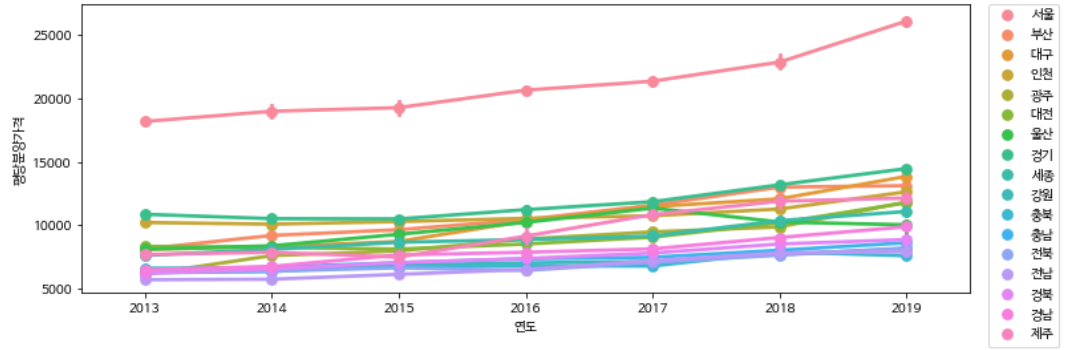
```
[125] # barplot 으로 연도별 평당분양가격 출력
      sns.barplot(data=df, x="연도", y="평당분양가격")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2e2f18970>
```




```
[126] # pointplot 으로 연도별 평당분양가격 출력
      plt.figure(figsize=(12, 4))
      sns.pointplot(data=df, x="연도", y="평당분양가격", hue="지역명")
      plt.legend(bbox_to_anchor=(1.02, 1), loc=2, borderaxespad=0.)
```

```
<matplotlib.legend.Legend at 0x7fa2e3387970>
```

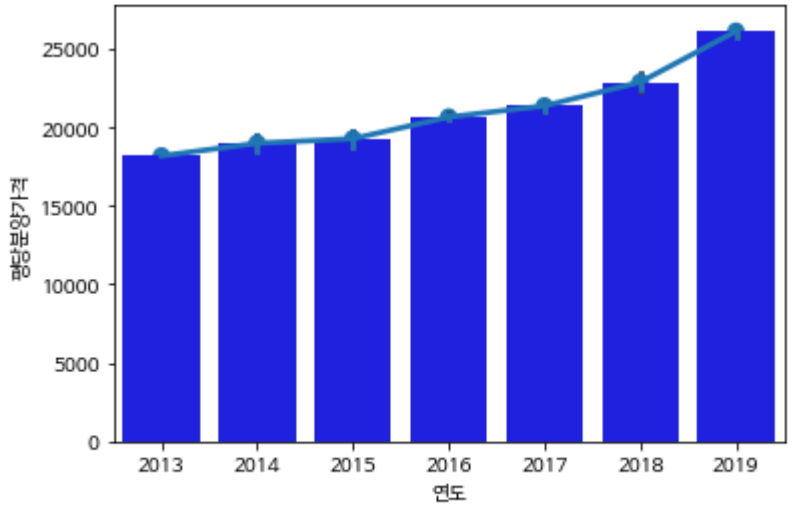


01. 전국 신규 민간 아파트 분양가격 동향

- 연도별, 지역별 평당분양가격 보기

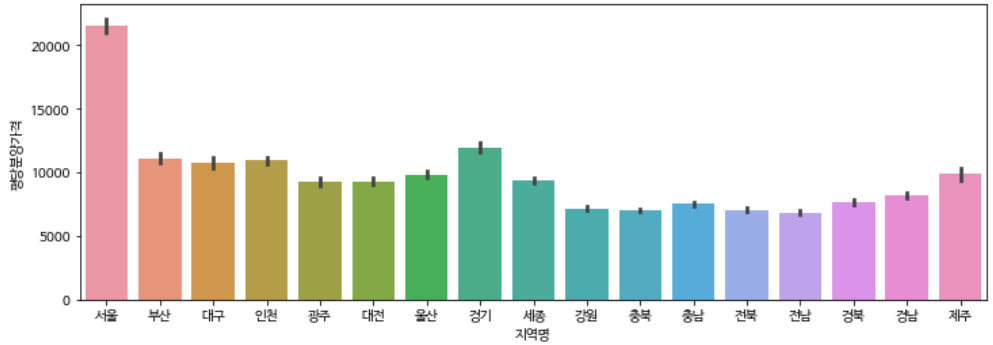
```
[127] # 서울만 barplot 으로 출력
      df_seoul = df[df["지역명"] == "서울"].copy()
      print(df_seoul.shape)

      sns.barplot(data=df_seoul, x="연도", y="평당분양가격", color="b")
      sns.pointplot(data=df_seoul, x="연도", y="평당분양가격")
```

```
(72, 4)
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2e3f463d0>
```




```
[140] # barplot 으로 지역별 평당분양가격을 출력
      plt.figure(figsize=(12, 4))
      sns.barplot(data=df, x="지역명", y="평당분양가격")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2e3fdea60>
```




```
[143] plt.figure(figsize=(12, 4))
      sns.barplot(data=mean_price, x=mean_price.index, y="평당분양가격", palette="Blues_r")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2e2ab3670>
```