Pandas - 데이터 요약, 분석

전국 민간아파트 분양가격 데이터를 기반으로 최근까지 분양가격의 통계등을 분석하고 이를 시각화 하기 위한 기초 데이터 처리를 Pandas 라이브러리를 이용해서 데이터 요약과 분석을 해보겠습니다.

1. 데이터 셋 다운로드

    https://s3-us-west-2.amazonaws.com/secure.notion-static.com/678e2564-318f-49c0-ac77-4fc6b30fb609/__(2013_9_2015_8).csv

    https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c6e6fa64-4514-42d2-9416-c63602f6f1c4/___(2019_12).csv

    ※위 데이터 셋을 Jupyter Notebook App의 작업 디렉터리에 /data 폴더를 생성한 뒤 저장하세요

2. 주피터 노트북 실행하기

- 시작 > Jupyter Notebook (Anaconda3)
- 시작 > Anaconda Prompt (anaconda3)

3. Pandas 라이브러리를 이용한 데이터 요약, 분석하기

3-1. Pandas 불러오기

```
import pandas as pd
```

3-2. 데이터 셋 불러오기(2019년 12월 데이터)

```
# 최근 데이터 셋 - 2019년 기준
df_last = pd.read_csv('data/전국신규민간아파트분양가격동향/주택도시보증공사_전국 평균 분양가격(2019년 12월).csv', encoding='euc-kr')

# 데이터 셋의 (행, 열) 갯수를 확인합니다.
# 방대한 데이터 셋의 경우 PC의 성능에 따라 불러들이는 데 시간이 걸릴수 있으니, 확인하는 용도
df_last.shape

# (4335, 5)
```

3-2. 데이터 - 미리보기(윗쪽부터)

```
#df_last.head? 를 입력후 실행[Shift+Enter]하면, 도움말 호출, 기본 5개
#df_last.head()에서 [Shift+Tab]을 1, 2회 누르면 도움말 호출
df_last.head()
```

	지역명	규모구분	연도	월	분양가격(㎡)
0	서울	전체	2015	10	5841
1	서울	전용면적 60㎡이하	2015	10	5652
2	서울	전용면적 60㎡초과 85㎡이하	2015	10	5882
3	서울	전용면적 85㎡초과 102㎡이하	2015	10	5721
4	서울	전용면적 102㎡초과	2015	10	5879

3-3. 데이터 - 미리보기(아래부터)

```
#df_last.tail? 를 입력후 실행[Shift+Enter]하면, 도움말 호출, 기본 5개
#df_last.tail()에서 [Shift+Tab]을 1, 2회 누르면 도움말 호출
df_last.tail()
```

	지역명	규모구분	연도	월	분양가격(㎡)
4330	제주	전체	2019	12	3882
4331	제주	전용면적 60㎡이하	2019	12	NaN
4332	제주	전용면적 60㎡초과 85㎡이하	2019	12	3898
4333	제주	전용면적 85㎡초과 102㎡이하	2019	12	NaN
4334	제주	전용면적 102㎡초과	2019	12	3601

3-4. 데이터 셋 불러오기(2013년 9월부터 2015년 8월까지 데이터)

```
df_first = pd.read_csv('data/전국신규민간아파트분양가격동향/전국 평균 분양가격(2013년 9월부터 2015년 8월까지).csv', encoding='euc-kr')

# 행, 열의 갯수를 확인
df_first.shape

# (17, 22)
```

3-5. 데이터 - 미리보기(윗쪽부터)

```
df_first.head()
```

	지역	2013년 12월	2014년1월	2014년2월	2014년3월	2014년4월	2014년5월	2014년6월	2014년7월	2014년8월	...	2014년 11월	2014년 12월	2015년1월	2015년2월	2015년3월	2015년4월	2015년5월	2015년6월	2015년7월	2015년8월
0	서울	18189	17925	17925	18016	18098	19446	18867	18742	19274	...	20242	20269	20670	20670	19415	18842	18367	18374	18152	18443
1	부산	8111	8111	9078	8965	9402	9501	9453	9457	9411	...	9208	9208	9204	9235	9279	9327	9345	9515	9559	9581
2	대구	8080	8080	8077	8101	8267	8274	8360	8360	8370	...	8439	8253	8327	8416	8441	8446	8568	8542	8542	8795
3	인천	10204	10204	10408	10408	10000	9844	10058	9974	9973	...	10020	10020	10017	9876	9876	9938	10551	10443	10443	10449
4	광주	6098	7326	7611	7346	7346	7523	7659	7612	7622	...	7752	7748	7752	7756	7861	7914	7877	7881	8089	8231

5 rows × 22 columns

3-6. 데이터 - 미리보기(아래부터)

```
df_first.tail()
```

	지역	2013년 12월	2014 년1월	2014 년2월	2014 년3월	2014 년4월	2014 년5월	2014 년6월	2014 년7월	2014 년8월	...	2014 년11월	2014년 12월	2015 년1월	2015 년2월	2015 년3월	2015 년4월	2015 년5월	2015 년6월	2015 년7월	2015 년8월
12	전북	6282	6281	5946	5966	6277	6306	6351	6319	6436	...	6583	6583	6583	6583	6542	6551	6556	6601	6750	6580
13	전남	5678	5678	5678	5696	5736	5656	5609	5780	5685	...	5768	5784	5784	5833	5825	5940	6050	6243	6286	6289
14	경북	6168	6168	6234	6317	6412	6409	6554	6556	6563	...	6881	6989	6992	6953	6997	7006	6966	6887	7035	7037
15	경남	6473	6485	6502	6610	6599	6610	6615	6613	6606	...	7125	7332	7592	7588	7668	7683	7717	7715	7723	7665
16	제주	7674	7900	7900	7900	7900	7900	7914	7914	7914	...	7724	7739	7739	7739	7826	7285	7285	7343	7343	7343

5 rows × 22 columns

3-7. 데이터 요약하기

```
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
'''

# Dtype - Object (문자 데이터)
# int64 / float64 (숫자 데이터)
# memory usage (메모리 사용량)
```

3-8. 결측치 구하기

```
# .isnull()
# .isna()
# 결측치(NaN)가 있다면 True 반환
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

```
# 결측치 합계 구하기 - .sum()
df_last.isnull().sum()

# 또는 .isna()
# df_last.isna().sum()

'''
지역명           0
규모구분          0
연도            0
월             0
분양가격(㎡)    277
dtype: int64
'''
```

3-9. 데이터 타입 변경하기

```
# df_last.info() 명령을 실행하면, 분양가격(㎡)의 Dtype이 Object(문자)로 되어있고,
# 이 경우, 통계데이터를 구할수 없음 (통계? --> 숫자!)
# 따라서, 데이터 타입을 숫자(int or float)로 변경해야 함

# 분양가격 컬럼을 추가하고,
# pd.to_numeric() 을 이용해 숫자 데이터로 변환
df_last['분양가격'] = pd.to_numeric(df_last['분양가격(㎡)'], errors='coerce')
df_last['분양가격']

'''
0       5841.0
1       5652.0
2       5882.0
3       5721.0
4       5879.0
         ...
4330    3882.0
4331       NaN
4332    3898.0
4333       NaN
4334    3601.0
Name: 분양가격, Length: 4335, dtype: float64
'''
```

3-10. 평당 분양가격 구하기

```
# 평당분양가격 컬럼을 추가하고, 분양가격 * 3.3을 대입
df_last['평당분양가격'] = df_last['분양가격'] * 3.3
df_last
```

※ 분양가격, 평당분양가격 컬럼은 새롭게 추가된 부분입니다.

	지역명	규모구분	연도	월	분양가격(㎡)	분양가격	평당분양가격
0	서울	전체	2015	10	5841	5841.0	19275.3
1	서울	전용면적 60㎡이하	2015	10	5652	5652.0	18651.6
2	서울	전용면적 60㎡초과 85㎡이하	2015	10	5882	5882.0	19410.6
3	서울	전용면적 85㎡초과 102㎡이하	2015	10	5721	5721.0	18879.3
4	서울	전용면적 102㎡초과	2015	10	5879	5879.0	19400.7
...	...	...	...	...	...	...	...
4330	제주	전체	2019	12	3882	3882.0	12810.6
4331	제주	전용면적 60㎡이하	2019	12	NaN	NaN	NaN
4332	제주	전용면적 60㎡초과 85㎡이하	2019	12	3898	3898.0	12863.4
4333	제주	전용면적 85㎡초과 102㎡이하	2019	12	NaN	NaN	NaN
4334	제주	전용면적 102㎡초과	2019	12	3601	3601.0	11883.3

4335 rows × 7 columns

3-11. 분양가격 요약하기

```
df_last.info()
'''
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4335 entries, 0 to 4334
Data columns (total 7 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
 5   분양가격      3957 non-null   float64
 6   평당분양가격    3957 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 237.2+ KB
'''
```

3-12. 변경전 컬럼 분양가격(㎡) -요약하기

```
# Dtype - Object (문자)
df_last['분양가격(㎡)'].describe()

'''
count     4058
unique    1753
top       2221
freq        17
Name: 분양가격(㎡), dtype: object
'''
```

3-13. 변경후 컬럼 분양가격 - 통계 요약하기

```
# 분양가격(㎡) --> 분양가격
# Object      ---> float64
# 숫자 데이터 통계/요약

df_last['분양가격'].describe()
'''
count     3957.000000
mean      3238.128633
std       1264.309933
min       1868.000000
25%       2441.000000
50%       2874.000000
75%       3561.000000
max      12728.000000
Name: 분양가격, dtype: float64
'''
```

3-14. 규모구분 → 전용면적 컬럼으로 변경하기

```
# 규모구분 이라는 컬럼의 중복되지 않은 값 확인하기
df_last['규모구분'].unique()

'''
array(['전체', '전용면적 60㎡이하', '전용면적 60㎡초과 85㎡이하', '전용면적 85㎡초과 102㎡이하',
       '전용면적 102㎡초과'], dtype=object)
'''
```

```
# 규모구분 보다 전용면적 이라고 볼 수 있고, 좀더 직관적으로 느껴지기때문에
# 전용면적이라는 컬럼을 새롭게 추가하고, 기존 데이터의 불필요한 텍스트를 제거합니다.
# .replace vs .str.replace : .replace는 정확하게 일치해야 함.

df_last['전용면적'] = df_last['규모구분'].str.replace('전용면적','')
df_last['전용면적'] = df_last['전용면적'].str.replace('초과', '~')
df_last['전용면적'] = df_last['전용면적'].str.replace('이하', '')
df_last['전용면적'] = df_last['전용면적'].str.replace(' ','').str.strip()
df_last['전용면적']

'''
0             전체
1            60㎡
2        60㎡~85㎡
3       85㎡~102㎡
4          102㎡~
         ...
4330          전체
4331         60㎡
4332     60㎡~85㎡
4333    85㎡~102㎡
4334       102㎡~
Name: 전용면적, Length: 4335, dtype: object
'''
```

3-15. 불필요한 컬럼 - 제거하기

```
# 기존 컬럼을 새로운 컬럼으로 전처리해서 옮겨 주었으므로,
# 규모구분, 분양가격(㎡) 을 제거해주도록 합니다.
# 데이터 셋의 불필요한 컬럼/데이터 삭제시 메모리 사용량이 줄어들고 처리속도가 빨라짐
# df_last.drop()
# axis = 0 (index) / 1 (column)
df_last = df_last.drop(['규모구분', '분양가격(㎡)'], axis=1)
```

```
# 제거가 잘 되었나 확인
df_last.info()

'''
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4335 entries, 0 to 4334
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   지역명     4335 non-null   object
 1   연도      4335 non-null   int64
 2   월       4335 non-null   int64
 3   분양가격    3957 non-null   float64
 4   평당분양가격  3957 non-null   float64
 5   전용면적    4335 non-null   object
dtypes: float64(2), int64(2), object(2)
memory usage: 203.3+ KB
'''
```

3-16. 데이터 집계하기 - groupby

```
# groupby를 이용해 데이터를 집계, 연산해보기
# df_last.groupby(["인덱스로 사용할 컬럼명"])["계산할 컬럼 값"].연산()
# .mean() : 평균값
# .sum() : 합계값
# .max() : 최대값
# .min() : 최소값
# .median() : 중앙값 / 데이터를 일렬로 나열했을때 중간 위치값 (50%)
# .std() : 표준편차

# 숫자데이터 통계 요약하기
df_last.groupby(['지역명','연도'])['평당분양가격'].describe()
```

		count	mean	std	min	25%	50%	75%	max
지역명	연도								
강원	2015	15.0	7188.060000	250.536854	6801.3	7157.700	7164.30	7268.25	7563.6
	2016	52.0	7162.903846	328.461816	6656.1	6925.875	7060.35	7350.75	7926.6
	2017	55.0	7273.560000	391.458701	6639.6	6897.000	7329.30	7428.30	8906.7
	2018	60.0	8219.255000	742.016586	7312.8	7646.925	7877.10	8794.50	9645.9
	2019	60.0	8934.475000	1221.095881	7626.3	8207.925	8347.35	9972.60	11873.4
...	...	...	...	...	...	...	...	...	...
충북	2015	15.0	6828.800000	508.073964	6164.4	6509.250	6656.10	7198.95	7817.7
	2016	60.0	7133.335000	602.198116	6319.5	6676.725	7012.50	7461.30	9421.5
	2017	55.0	7473.120000	816.075386	6520.8	6769.950	7134.60	8250.00	9421.5
	2018	60.0	8149.295000	646.657440	6850.8	7748.400	7880.40	8858.85	9150.9
	2019	60.0	7970.875000	631.082504	7009.2	7555.350	7656.00	8500.80	9203.7

85 rows × 8 columns

```
# 전용면적으로 분양가격의 평균을 구합니다.
# df_last.groupby(["인덱스로 사용할 컬럼명"])["계산할 컬럼 값"].연산()
df_last.groupby(['전용면적'])['분양가격'].mean()
'''
전용면적
102㎡~       3490.213828
```

```
60㎡          3143.981037
60㎡~85㎡      3112.436385
85㎡~102㎡     3362.908962
전체           3113.965517
Name: 분양가격, dtype: float64
'''
```

```
# 지역명, 전용면적으로 평당분양가격의 평균을 구합니다.
df_last.groupby(['지역명','전용면적'])['평당분양가격'].mean()

# 순서를 바꾸려면 (1)
df_last.groupby(['전용면적','지역명'])['평당분양가격'].mean()

# 순서를 바꾸려면 (2)
# df_last.groupby(['전용면적','지역명'])['평당분양가격'].mean().unstack()

# 반올림
# df_last.groupby(['전용면적','지역명'])['평당분양가격'].mean().unstack().round()
```

지역명 전용면적	강원	경기	경남	경북	광주	대구	대전	부산	서울	세종	울산	인천	전남	전북	제주	충남	충북
102㎡~	8311.0	14772.0	10358.0	9157.0	11042.0	13087.0	14877.0	13208.0	23446.0	10107.0	9974.0	14362.0	8168.0	8194.0	10523.0	8689.0	8195.0
60㎡	7567.0	13252.0	8689.0	7883.0	9431.0	11992.0	9176.0	11354.0	23213.0	9324.0	9202.0	11241.0	7210.0	7610.0	14022.0	7911.0	7103.0
60㎡~85㎡	7486.0	12524.0	8619.0	8061.0	9911.0	11779.0	9711.0	11865.0	22787.0	9775.0	10503.0	11384.0	7269.0	7271.0	10621.0	7819.0	7264.0
85㎡~102㎡	8750.0	13678.0	10018.0	8774.0	9296.0	11141.0	9037.0	12073.0	25944.0	9848.0	8861.0	11528.0	7909.0	8276.0	10709.0	9120.0	8391.0
전체	7478.0	12560.0	8659.0	8079.0	9904.0	11771.0	9786.0	11936.0	22610.0	9805.0	10493.0	11257.0	7284.0	7293.0	10785.0	7815.0	7219.0

```
# 연도, 지역명으로 평당분양가격의 평균을 구합니다.
g = df_last.groupby(['연도','지역명'])['평당분양가격'].mean()
g

'''
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
'''

# 지역명, 연도 순서로 바꾸기
# g.unstack()

# 행, 열 바꾸기
# g.unstack().T
# g.unstack().transpose()
```

3-17. 데이터 집계하기 - pivot_table

※ 피벗 테이블은 커다란 표의 데이터를 요약하는 통계표이다. 이 요약에는 합계, 평균, 기타 통계가 포함될 수 있으며 피벗 테이블이 이들을 함께 의미있는 방식으로 묶어준다. 피벗 테이블은 데이터 처리의 한 기법이다. (Wikipedia)

- pivot - 집계(aggregate)를 하지 않음
- pivot_table - 집계를 함

```
# 지역명을 index 로 평당분양가격 을 values 로 구합니다.
# 도움말 확인 > pd.pivot_table?
# pivot_table 기본은 평균값 추출, 별도로 지정할 수 있음

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

	평당분양가격
지역명	
강원	7890.750000
경기	13356.895200
경남	9268.778138
경북	8376.536515
광주	9951.535821
대구	11980.895455
대전	10253.333333
...	

```
# 전용면적을 index 로 평당분양가격 을 values 로 구합니다.
# 둘 이상의 인덱스 적용시 [,] 를 사용
# 단일 인덱스는 '컬럼명' 으로 처리 가능
pd.pivot_table(df_last, index=['전용면적'], values=['평당분양가격'])
```

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

```
# groupby() vs pivot_table() 차이점을 비교해보세요

# 지역명, 전용면적으로 평당분양가격의 평균을 구합니다.
df_last.groupby(["전용면적", index="지역명"])["평당분양가격"].mean().unstack().round()
```

지역명	강원	경기	경남	경북	광주	대구	대전	부산	서울	세종	울산	인천	전남	전북	제주	충남	충북
전용면적																	
102㎡~	8311.0	14772.0	10358.0	9157.0	11042.0	13087.0	14877.0	13208.0	23446.0	10107.0	9974.0	14362.0	8168.0	8194.0	10523.0	8689.0	8195.0
60㎡	7567.0	13252.0	8689.0	7883.0	9431.0	11992.0	9176.0	11354.0	23213.0	9324.0	9202.0	11241.0	7210.0	7610.0	14022.0	7911.0	7103.0
60㎡~85㎡	7486.0	12524.0	8619.0	8061.0	9911.0	11779.0	9711.0	11865.0	22787.0	9775.0	10503.0	11384.0	7269.0	7271.0	10621.0	7819.0	7264.0
85㎡~102㎡	8750.0	13678.0	10018.0	8774.0	9296.0	11141.0	9037.0	12073.0	25944.0	9848.0	8861.0	11528.0	7909.0	8276.0	10709.0	9120.0	8391.0
전체	7478.0	12560.0	8659.0	8079.0	9904.0	11771.0	9786.0	11936.0	22610.0	9805.0	10493.0	11257.0	7284.0	7293.0	10785.0	7815.0	7219.0

```
# 첫번째, 두번째 데이터 셋의 형태가 다름 --> 동일한 형태로 집계
df_last.pivot_table(index='전용면적', columns='지역명', values=['평당분양가격'])
```

	평당분양가격											
지역명	강원	경기	경남	경북	광주	대구	대전	부산	서울	세종	울산	인천
전용면적												
102㎡~	8311.380000	14771.790	10358.363265	9157.302000	11041.532432	13087.338000	14876.871429	13208.250	23446.038	10106.976000	9974.448000	14362.
60㎡	7567.098000	13251.744	8689.175000	7883.172000	9430.666667	11992.068000	9176.475000	11353.782	23212.794	9323.927027	9202.106897	11241.
60㎡~85㎡	7485.588000	12523.566	8618.676000	8061.372000	9910.692000	11778.690000	9711.372000	11864.820	22786.830	9775.458000	10502.531707	11384.
85㎡~102㎡	8749.557143	13677.774	10017.612000	8773.814634	9296.100000	11140.642857	9037.430769	12072.588	25943.874	9847.926000	8861.007692	11527.
전체	7477.536000	12559.602	8658.672000	8078.532000	9903.630000	11771.298000	9786.018000	11936.166	22610.346	9805.422000	10492.712195	11257.

```
# 연도, 지역명으로 평당분양가격의 평균을 구합니다.
# g = df_last.groupby(["연도", "지역명"])["평당분양가격"].mean()

p = pd.pivot_table(df_last, index=['연도','지역명'], values='평당분양가격')

# 인덱스(행) 데이터 추출하기
p.loc[2017]

# 인덱스 번호로 데이터 추출하기
# p.iloc[2]

'''
평당분양가격    8459.22
Name: (2015, 경남), dtype: float64
'''
```

	평당분양가격
지역명	
강원	7273.560000
경기	12304.980000
경남	8786.760000
경북	8280.800000
광주	9613.977551
대구	12206.700000
대전	9957.158491
...	