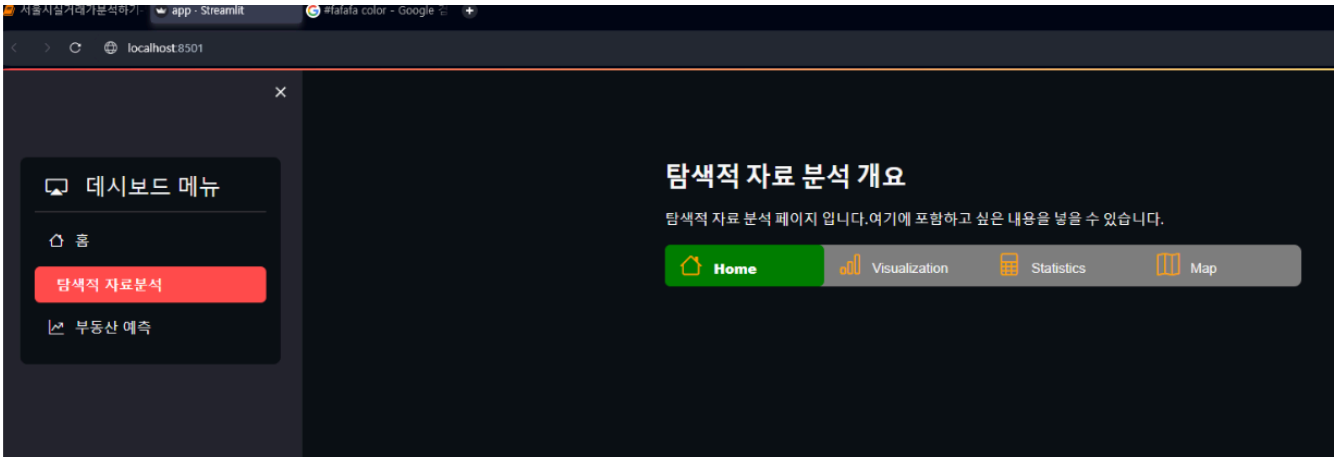
4주차 복습 과제

서울시 부동산 실거래가 데이터를 가져와 대시보드 만들기

> data.seoul.go.kr

> 서울 열린데이터 광장 데이터 가져오기



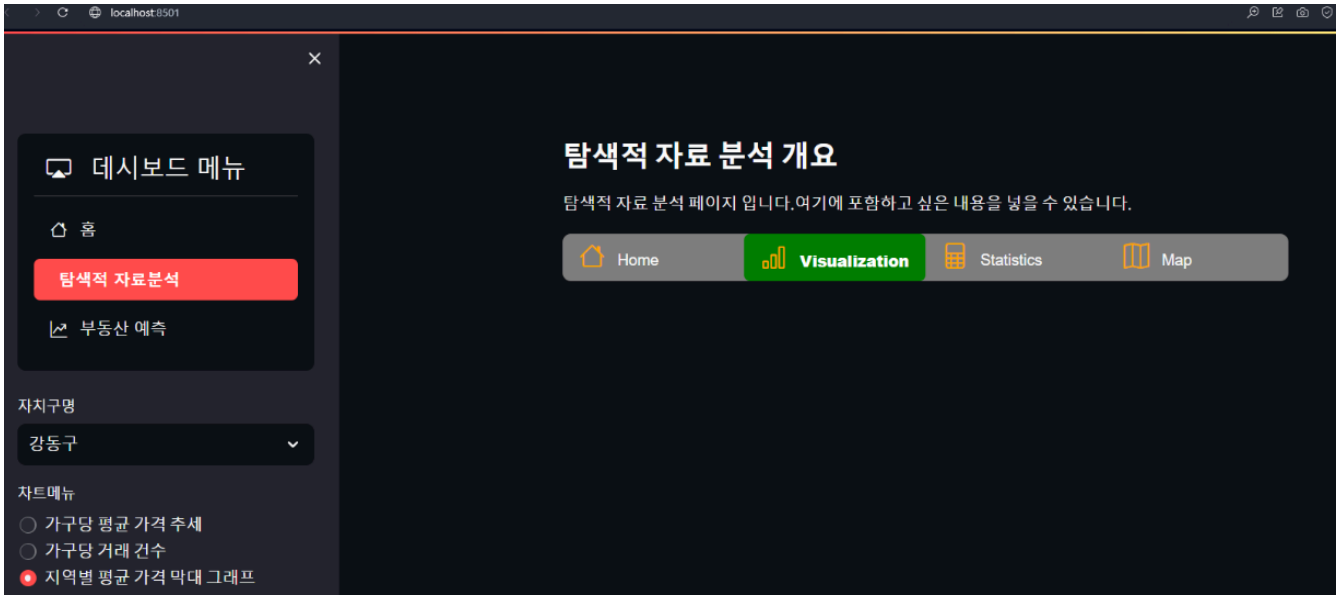
복습. 'Visualization' 화면 만들기

- 왼쪽 sidebar에서 자치구를 선택
- 왼쪽 sidebar에서 '가구당 평균 가격 추이', '가구당 거래 건수 추이', '지역별 평균 가격 막대 그래프' 선택
- 현재는 약 2~3달치 데이터만 수집되어 있어 전반적 흐름을 찾기 어려우나 1년치 데이터를 수집하며 추이 분석 가능
- viz.py 파일에 showViz() 함수로 구현하기

eda.py에서 'Visualization' 버튼을 누르면, viz.py 파일에 showViz() 호출하도록 수정

viz.py 파일에 showViz() 함수 구현 (viz1.py)

- 왼쪽 sidebar에서 자치구를 선택
- 왼쪽 sidebar에서 '가구당 평균 가격 추이', '가구당 거래 건수 추이', '지역별 평균 가격 막대 그래프' 선택



viz.py 파일에 showViz() 함수 구현 (viz2.py)

- '가구당 평균 가격 추이' 버튼 선택시 그래프 그리기 (아파트)

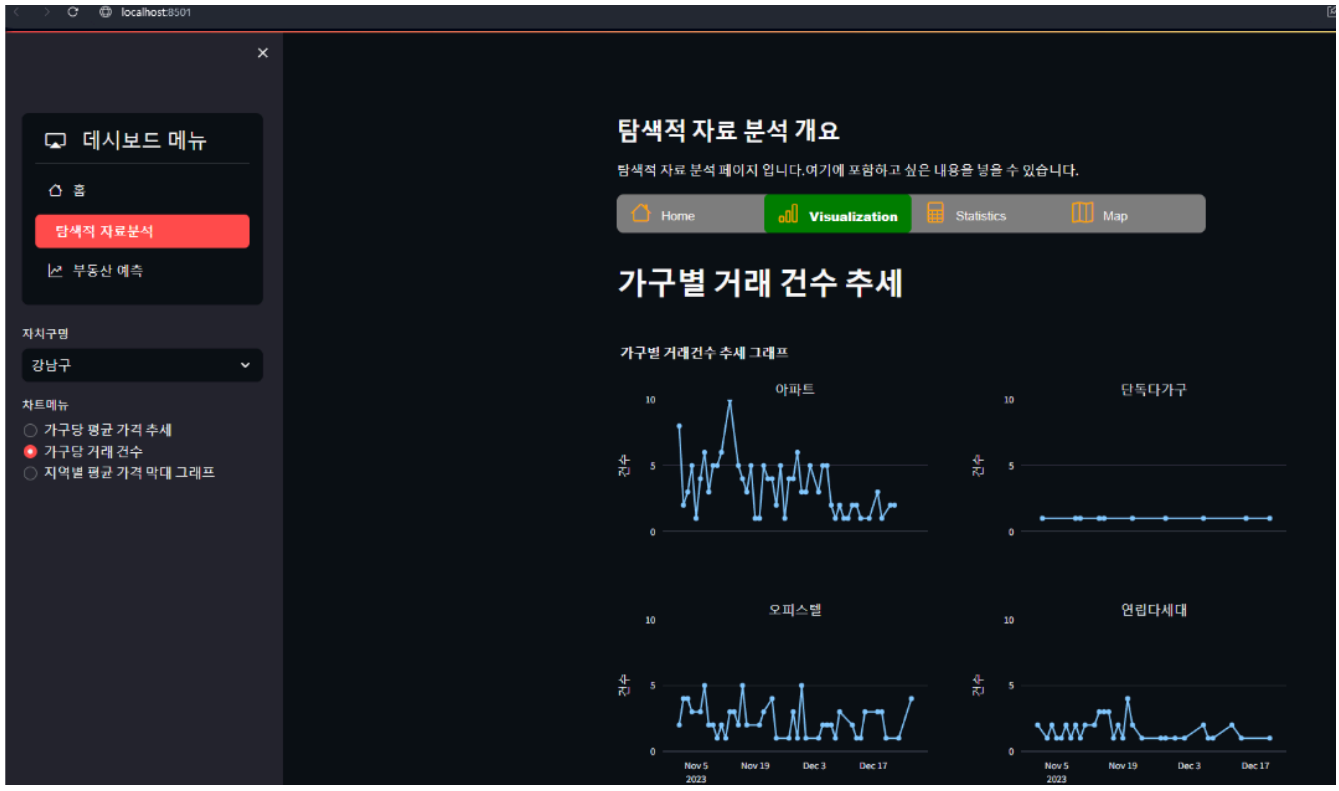
복습과제(1) - viz.py 파일에 showViz() 함수 구현 (viz3.py)

- '가구당 평균 가격 추이' 버튼 선택시 '단독다가구', '오피스텔', '연립다세대' 추가하여 그래프 그리기

복습과제(2) - viz.py 파일에 showViz() 함수 구현 (viz3.py)

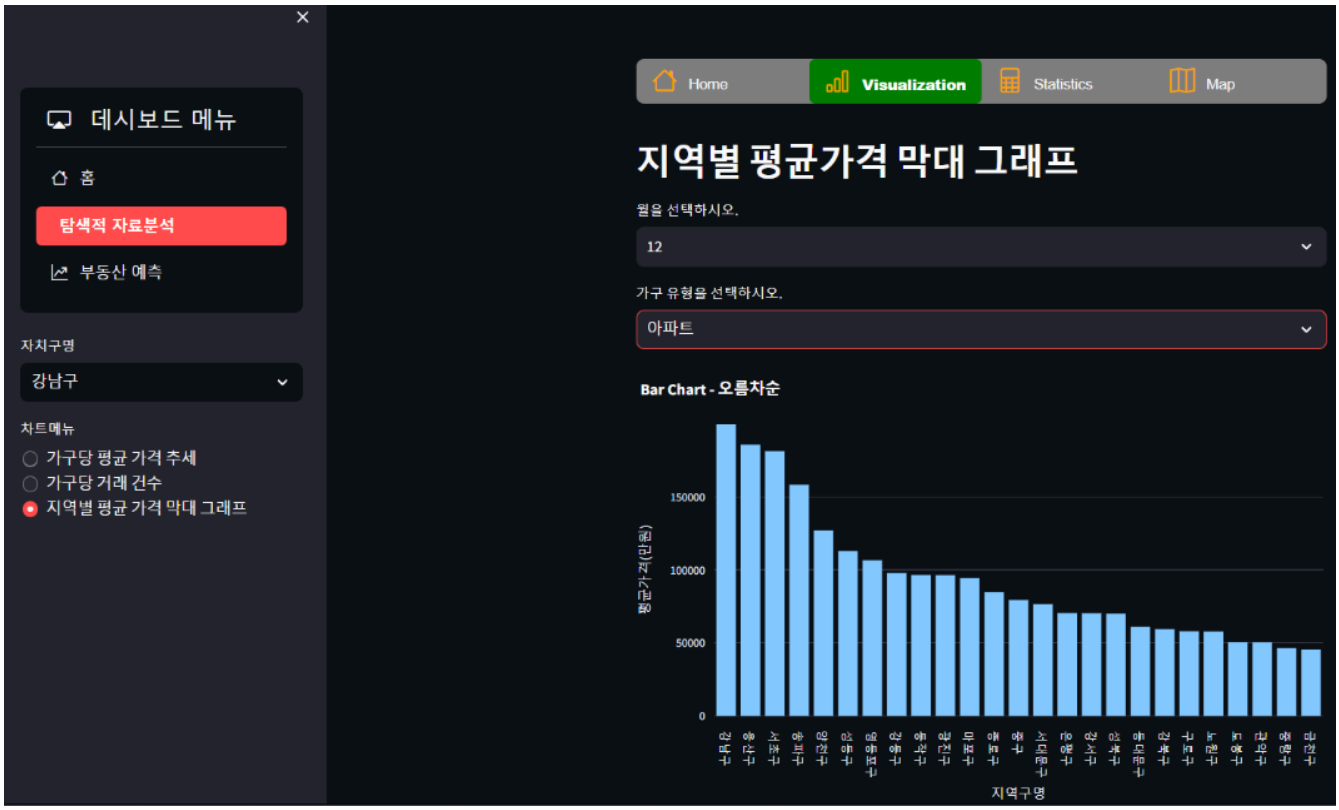
- '가구당 거래건수' 버튼 선택시 '아파트', '단독다가구', '오피스텔', '연립다세대' 거래 건수 그래프 그리기

```
def cntChart(total_df, sgg_nm):
    st.markdown("## 가구별 거래 건수 추세 Wn")

    filtered_df = total_df[total_df['SGG_NM'] == sgg_nm]
    filtered_df = filtered_df[filtered_df['DEAL_YMD'].between('2023-11-01', '2023-12-31')]
    result = filtered_df.groupby(['DEAL_YMD', 'HOUSE_TYPE'])['OBJ_AMT'].count().reset_index().rename(columns = {'OBJ_AMT' : '거래건수'})
    ....코드 추가 하세요...
```



지역별 평균 가격 막대 그래프 (viz5.py)

- 월(month)과 가구유형(아파트, 단독다가구, 오피스텔, 연립다세대)에 따른 구별 평균 거래가격을 막대그래프로 그리기

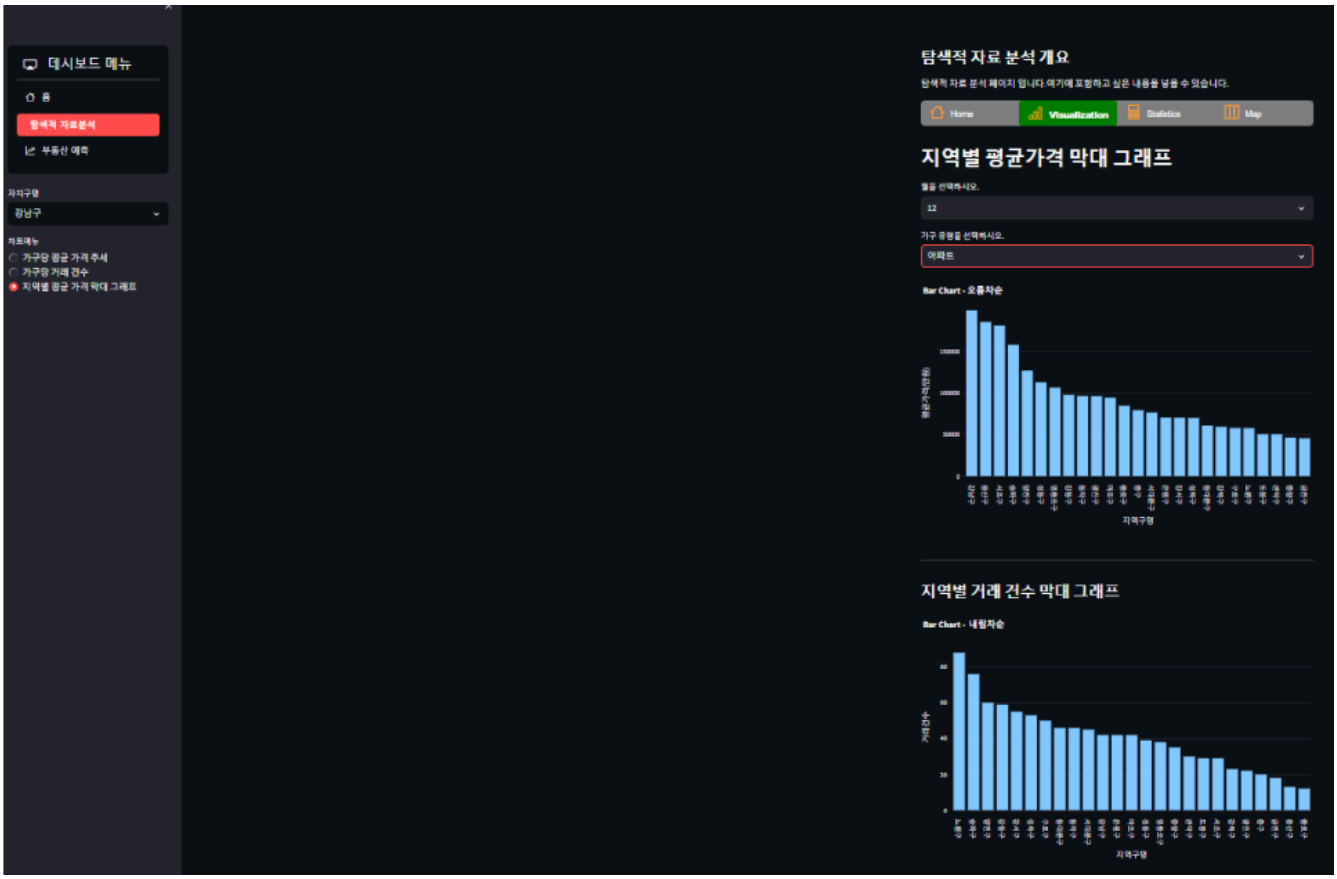


In [ ]:
```
#
#  viz.py (viz5.py)
#
#  > streamlit run app.py
#

'''
import streamlit as st
import pandas as pd
from plotly.subplots import make_subplots
import plotly.express as px

def meanChart(total_df, sgg_nm) :
    st.markdown('## 가구별 평균 가격 추세 Wn')

    filtered_df = total_df[total_df['SGG_NM'] == sgg_nm]
    filtered_df = filtered_df[filtered_df['DEAL_YMD'].between('2023-11-01', '2023-12-31')]
    result = filtered_df.groupby(['DEAL_YMD', 'HOUSE_TYPE'])['OBJ_AMT'].agg('mean').reset_index()

    df1 = result[result['HOUSE_TYPE'] == '아파트']
    df2 = result[result['HOUSE_TYPE'] == '단독다가구']
    df3 = result[result['HOUSE_TYPE'] == '오피스텔']
    df4 = result[result['HOUSE_TYPE'] == '연립다세대']

    fig = make_subplots(rows=2, cols=2,
                        shared_xaxes=True,
                        subplot_titles=('아파트', '단독다가구', '오피스텔', '연립다세대'),
                        horizontal_spacing=0.15)

    fig.add_trace(px.line(df1, x='DEAL_YMD', y='OBJ_AMT',
                          title='아파트 실거래가 평균', markers=True).data[0], row=1, col=1)

    fig.add_trace(px.line(df2, x='DEAL_YMD', y='OBJ_AMT',
                          title='단독다가구 실거래가 평균', markers=True).data[0], row=1, col=2)

    fig.add_trace(px.line(df3, x='DEAL_YMD', y='OBJ_AMT',
                          title='오피스텔 실거래가 평균', markers=True).data[0], row=2, col=1)

    fig.add_trace(px.line(df4, x='DEAL_YMD', y='OBJ_AMT',
                          title='연립다세대 실거래가 평균', markers=True).data[0], row=2, col=2)

    fig.update_yaxes(tickformat='.0f',
                     title_text='물건가격(원)',
                     range=[result['OBJ_AMT'].min(), result['OBJ_AMT'].max()])
    fig.update_layout(
        title = '가구별 평균값 추세 그래프',
        width=800, height=600,
        showlegend=True, template='plotly_white')
    st.plotly_chart(fig)

def cntChart(total_df, sgg_nm) :
    st.markdown('## 가구별 거래 건수 추세 Wn')

    filtered_df = total_df[total_df['SGG_NM'] == sgg_nm]
    filtered_df = filtered_df[filtered_df['DEAL_YMD'].between('2023-11-01', '2023-12-31')]
    result = filtered_df.groupby(['DEAL_YMD', 'HOUSE_TYPE'])['OBJ_AMT'].count().reset_index().rename(columns = {'OBJ_AMT

    df1 = result[result['HOUSE_TYPE'] == '아파트']
    df2 = result[result['HOUSE_TYPE'] == '단독다가구']
    df3 = result[result['HOUSE_TYPE'] == '오피스텔']
    df4 = result[result['HOUSE_TYPE'] == '연립다세대']

    fig = make_subplots(rows=2, cols=2,
                        shared_xaxes=True,
                        subplot_titles=('아파트', '단독다가구', '오피스텔', '연립다세대'),
                        horizontal_spacing=0.15)

    fig.add_trace(px.line(df1, x='DEAL_YMD', y='거래건수',
                          title='아파트 거래건수', markers=True).data[0], row=1, col=1)

    fig.add_trace(px.line(df2, x='DEAL_YMD', y='거래건수',
                          title='단독다가구 거래건수', markers=True).data[0], row=1, col=2)

    fig.add_trace(px.line(df3, x='DEAL_YMD', y='거래건수',
                          title='오피스텔 거래건수', markers=True).data[0], row=2, col=1)

    fig.add_trace(px.line(df4, x='DEAL_YMD', y='거래건수',
                          title='연립다세대 거래건수', markers=True).data[0], row=2, col=2)

    fig.update_yaxes(tickformat='.0f',
                     title_text='건수',
                     range=[0, result['거래건수'].max()])
```

```
    fig.update_layout(
        title = '가구별 거래건수 추세 그래프',
        width=800, height=600,
        showlegend=True, template='plotly_white')
    st.plotly_chart(fig)

def barChart(total_df) :
    st.markdown('## 지역별 평균가격 막대 그래프')

    month_selected = st.selectbox("월을 선택하시오.", [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
    house_selected = st.selectbox("가구 유형을 선택하시오.", total_df['HOUSE_TYPE'].unique())

    total_df['month'] = total_df['DEAL_YMD'].dt.month
    result = total_df[(total_df['month'] == month_selected) & (total_df['HOUSE_TYPE'] == house_selected)]

    bar_df = result.groupby('SGG_NM')['OBJ_AMT'].agg('mean').reset_index()

    df_sorted = bar_df.sort_values('OBJ_AMT', ascending=False)

    # 바 차트 만들기
    fig = px.bar(df_sorted, x = 'SGG_NM', y = 'OBJ_AMT')

    # update Layout
    fig.update_yaxes(tickformat='.0f',
                     title_text = '물건 가격(만원)',
                     range=[0, df_sorted['OBJ_AMT'].max()])
    fig.update_layout(title='Bar Chart - 오름차순',
                      xaxis_title='지역구명',
                      yaxis_title='평균가격(만원)')
    st.plotly_chart(fig)


def showViz(total_df) :
    total_df['DEAL_YMD'] = pd.to_datetime(total_df['DEAL_YMD'], format='%Y-%m-%d')

    sgg_nm = st.sidebar.selectbox('자치구명', sorted(total_df['SGG_NM'].unique()))
    selected = st.sidebar.radio('차트메뉴',
                                ['가구당 평균 가격 추세', '가구당 거래 건수', '지역별 평균 가격 막대 그래프'])

    if selected == '가구당 평균 가격 추세' :
        meanChart(total_df, sgg_nm)
    elif selected == '가구당 거래 건수' :
        cntChart(total_df, sgg_nm)
    elif selected == '지역별 평균 가격 막대 그래프' :
        barChart(total_df)
    else :
        st.warning("Error")

'''
```

복습과제(3) - (viz6.py)

- 월(month)과 가구유형(아파트, 단독다가구, 오피스텔, 연립다세대)에 따른 구별 평균 거래 가격을 막대그래프로 그리고
- 월(month)과 가구유형(아파트, 단독다가구, 오피스텔, 연립다세대)에 따른 구별 거래 건수을 막대그래프를 추가 하시오



In [ ]:

1. '탐색적자료분석' 통계 화면 만들기

- 차이검정
- 상관분석
- 회기분석

2. 두 집단 간의 차이 검정 (T-test)

모집단의 분산이나 표준편차를 알지 못할 때 모집단을 대표하는 표본으로부터 추정된 분산이나 표준편차를 가지고 검정하는 방법으로 “두 모집단의 평균간의 차이는 없다”라는 귀무가설과 “두 모집단의 평균 간에 차이가 있다”라는 대립가설 중에 하나를 선택할 수 있도록 하는 통계적 검정방법이다.

- 귀무가설(H0) : 두 집단 간의 평균 차이는 없을 것이다.
- 대립가설(H1) : 두 집단 간의 평균 차이가 있을 것이다.

예를 들어, 유의 수준(α)을 0.05라고 가정할 때 t값이 커져서 (평균차이가 있을 가능성이 커져서) 기각역에 존재하여 유의확률(p값, p-value)이 0.05보다 작으면 평균 차이가 유의미한 것으로 해석되어 귀무가설을 기각한다. 그 반대의 경우, 평균 차이가 유의미하지 않으므로 귀무가설을 수용한다.

(LAB 1) 11월과 12월 아파트 평균 가격의 차이가 있는가? (차이 검정)

In [3]:
```
!pip install pingouin
```

...

In [5]:
```
from pingouin import ttest
import pandas as pd

seoul = pd.read_csv("seoul_real_estate.csv", parse_dates=['DEAL_YMD'])
seoul['month'] = seoul['DEAL_YMD'].dt.month
apt_df = seoul[(seoul['HOUSE_TYPE'] == '아파트') & (seoul['month'].isin([2, 3]))]

dec_df = apt_df[apt_df['month'] == 2]
nov_df = apt_df[apt_df['month'] == 3]

print('11월 아파트 평균 가격(만원) : ', dec_df['OBJ_AMT'].mean())
print('12월 아파트 평균 가격(만원) : ', nov_df['OBJ_AMT'].mean())


ttest(dec_df['OBJ_AMT'], nov_df['OBJ_AMT'], paired=False)
```

```
11월 아파트 평균 가격(만원) :  104537.82953020134
12월 아파트 평균 가격(만원) :  86951.29898989899
```

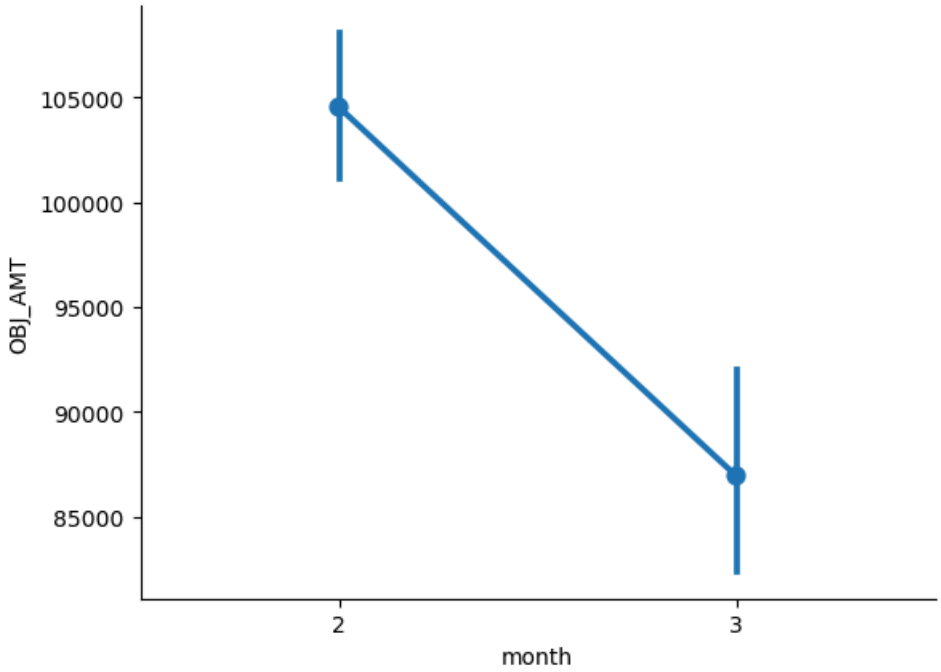
Out[5]:

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	5.868286	1038.958757	two-sided	5.915944e-09	[11705.91, 23467.15]	0.22835	1.299e+06	0.995797

p-val : 5.915944x10-9, p-val 값이 0.05보다 작은 값으로 2월과 3월의 평균 가격은 같지 않다.

(LAB 2) 2월과 3월 아파트 평균 가격의 차이가 있는가? (차이 검정) - 시각화

In [7]:
```
import seaborn as sns
import matplotlib.pyplot as plt

fig, ax = plt.subplots()
sns.pointplot(x = 'month', y = 'OBJ_AMT', data = apt_df)
sns.despine()

plt.savefig('stat01.png', dpi=200)
plt.show()
```



오차 막대는 95% 신뢰구간을 나타냄

3. 상관 분석

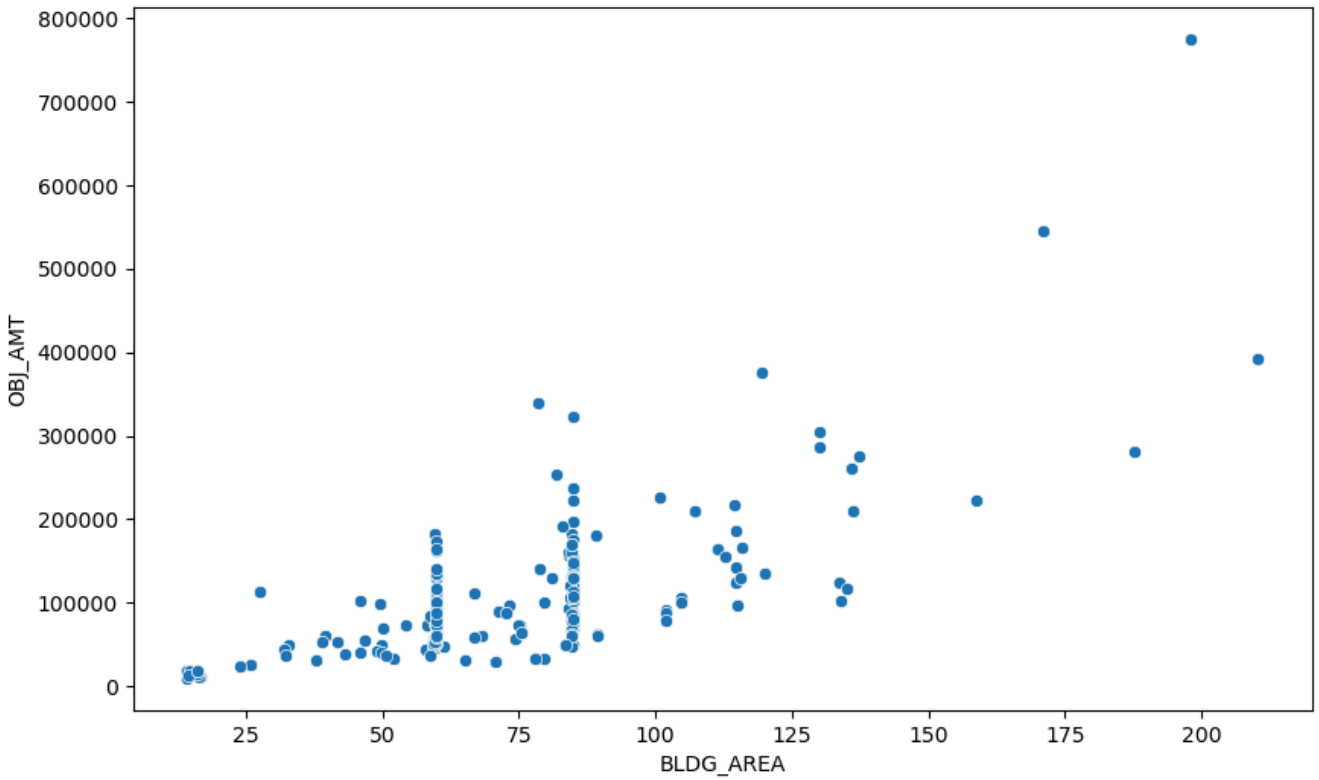
- 변수가 서로 관련이 있는지 여부를 결정
- 두 변수가 연속형 변수 일 경우만 사용 가능
- 변수의 연관성 정도는 상관계수(r)로 표시, 피어슨 상관 계수
- r = +1 : 완벽한 양의 상관 관계
- r = -1 : 완벽한 음의 상관 관계

- r = 0 : 상관 관계가 없음

(LAB 3) 상관 분석 - 건물 면적과 가격의 연관성이 있는가?

(1) 산포도 그리기 (건물 면적과 가격과의 관련성)

In [8]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

seoul = pd.read_csv("seoul_real_estate.csv", parse_dates=['DEAL_YMD'])
seoul['month'] = seoul['DEAL_YMD'].dt.month
apt_df = seoul[(seoul['HOUSE_TYPE'] == '아파트') & (seoul['month'].isin([11, 12]))]

corr_df = apt_df[['DEAL_YMD', 'OBJ_AMT', 'BLDG_AREA', 'SGG_NM', 'month']].reset_index(drop=True)

fig, ax = plt.subplots(figsize=(10, 6))
sns.scatterplot(x = 'BLDG_AREA', y = 'OBJ_AMT', data = corr_df)

plt.savefig('stat02.png', dpi=200)
plt.show()
```



산포도 차트를 그리고, 건물 면적과 가격과의 관련성을 시각적으로 확인한다.

(2) 상관 계수 구하기 (건물 면적과 가격과의 관련성)

In [9]:
```
import pingouin as pg

pg.corr(corr_df['BLDG_AREA'], corr_df['OBJ_AMT']).round(3)
```

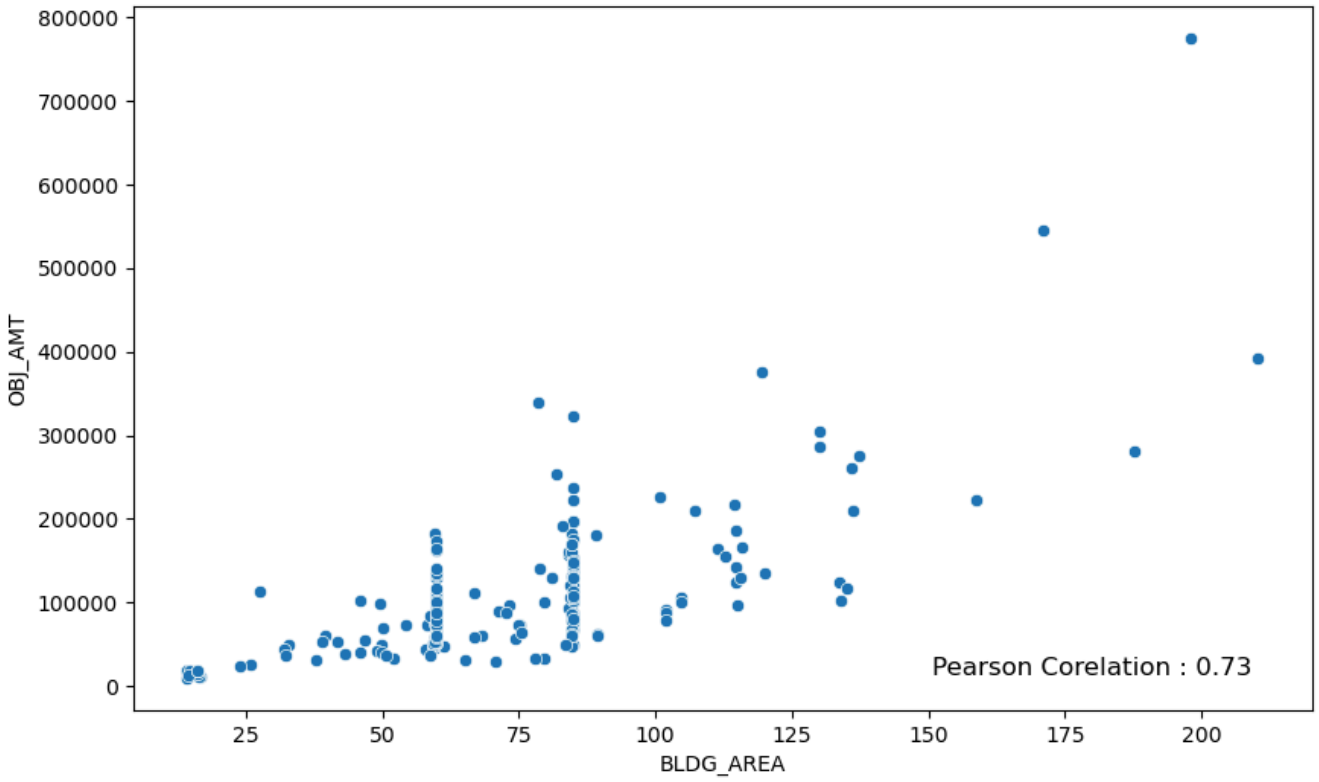
Out[9]:

	n	r	CI95%	p-val	BF10	power
pearson	213	0.731	[0.66, 0.79]	0.0	4.629e+33	1.0

상관계수 r = 0.731로 양의 상관관계를 가짐

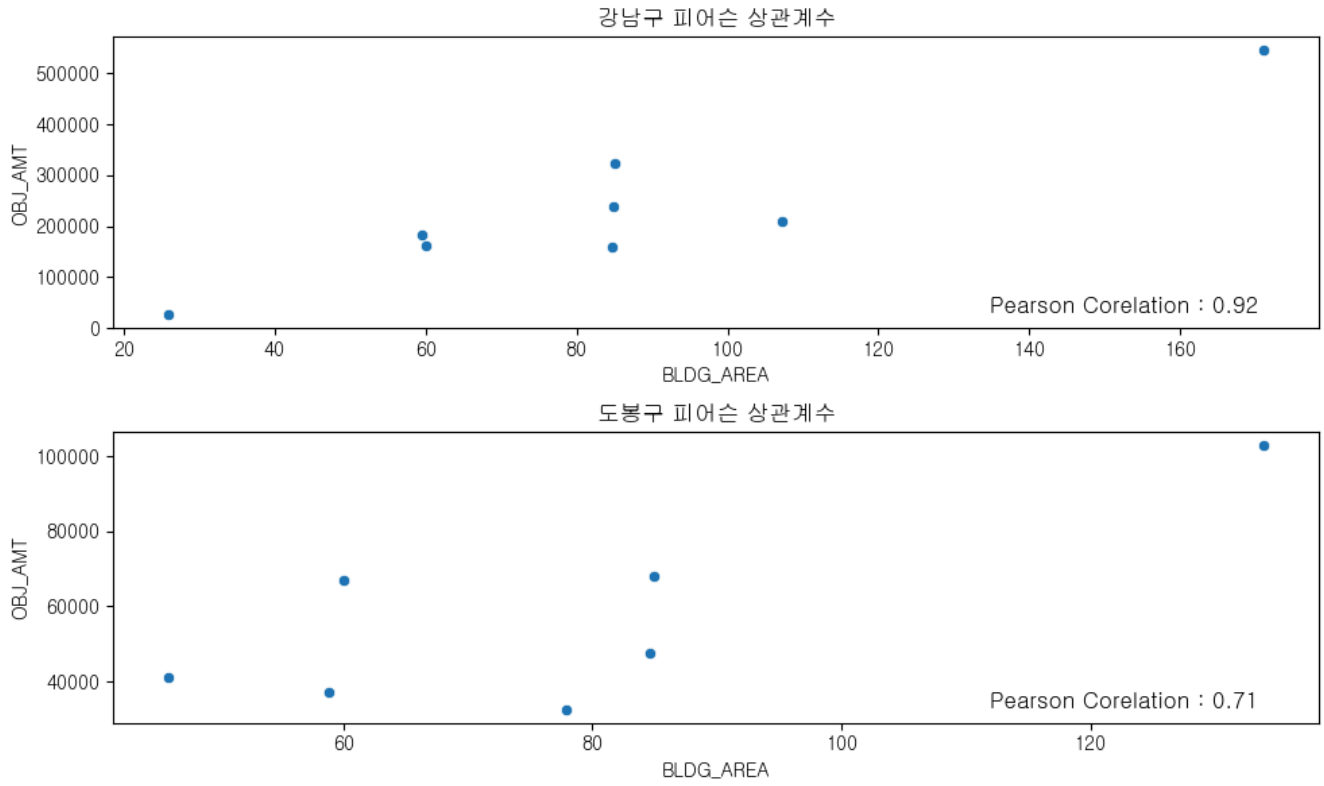
(3) 산포도에 상관 계수 표시하기 (건물 면적과 가격과의 관련성)

In [10]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import pingouin as pg


seoul = pd.read_csv("seoul_real_estate.csv", parse_dates=['DEAL_YMD'])
seoul['month'] = seoul['DEAL_YMD'].dt.month
apt_df = seoul[(seoul['HOUSE_TYPE'] == '아파트') & (seoul['month'].isin([11, 12]))]

corr_df = apt_df[['DEAL_YMD', 'OBJ_AMT', 'BLDG_AREA', 'SGG_NM', 'month']].reset_index(drop=True)

# 상관계수 구하기
corr_coef = pg.corr(corr_df['BLDG_AREA'], corr_df['OBJ_AMT']).round(3)['r'].values[0]

# 산포도 그리기
fig, ax = plt.subplots(figsize=(10, 6))
sns.scatterplot(x = 'BLDG_AREA', y = 'OBJ_AMT', data = corr_df, ax = ax)
ax.text(0.95,0.05, f'Pearson Corelation : {corr_coef:.2f}',
        transform=ax.transAxes, ha='right', fontsize=12)

plt.savefig('stat03png', dpi=200)
plt.show()
```



(4) 구별 건물 면적과 가격과의 관련성 알아보기

In [11]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import pingouin as pg

from matplotlib import font_manager, rc
font_path = "C:/Windows/Fonts/NGULIM.TTF"
font = font_manager.FontProperties(fname=font_path).get_name()
rc('font', family=font)

# 데이터 읽어 오기
seoul = pd.read_csv("seoul_real_estate.csv", parse_dates=['DEAL_YMD'])
seoul['month'] = seoul['DEAL_YMD'].dt.month
apt_df = seoul[(seoul['HOUSE_TYPE'] == '아파트') & (seoul['month'].isin([11, 12]))]

corr_df = apt_df[['DEAL_YMD', 'OBJ_AMT', 'BLDG_AREA', 'SGG_NM', 'month']].reset_index(drop=True)

# 상관 관계를 알아 볼 구 목록
sgg_nm = ['강남구', '도봉구']

# 구별 산포도 그리기
fig, ax = plt.subplots(figsize=(10, 6), nrows = 2)
for i in range(len(sgg_nm)) :
    sgg_df = corr_df[corr_df['SGG_NM'] == sgg_nm[i]]

    # 상관계수 구하기
    corr_coef = pg.corr(sgg_df['BLDG_AREA'], sgg_df['OBJ_AMT']).round(3)['r'].values[0]

    # 산포도 그리기
    sns.scatterplot(x = 'BLDG_AREA', y = 'OBJ_AMT', data = sgg_df, ax = ax[i])
    ax[i].text(0.95,0.05, f'Pearson Corelation : {corr_coef:.2f}',
        transform=ax[i].transAxes, ha='right', fontsize=12)
    ax[i].set_title(f'{sgg_nm[i]} 피어슨 상관계수')
plt.tight_layout()
plt.savefig('stat04.png', dpi=200)
plt.show()
```



강남구의 면적 대비 가격의 관련성이 0.92로 도봉구의 0.71보다 양의 상관 관계가 있음

In [ ]: