

### 3주차 복습 과제

서울시 부동산 실거래가 데이터를 가져와 대시보드 만들기

data.seoul.go.kr

서울 열린데이터 광장 데이터 가져오기

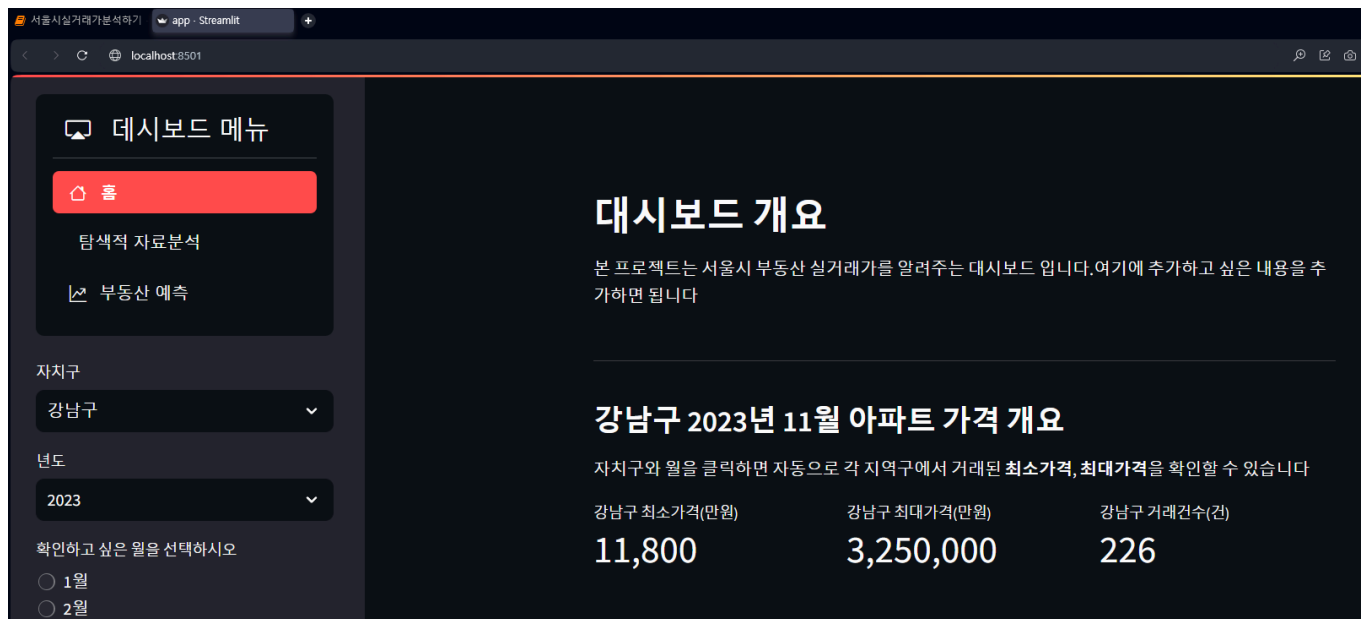
복습과제-1 - 자치구 월별 거래건수를 함께 표시하도록 수정하시오(home2)

기본 화면 : app.py

데이터 읽어오기 : utils.py

홈화면(아파트 값 개요, 거래 건수 등) 만들기 : home.py

streamlit run app.py



복습과제-2 - 자치구 월별 거래 아파트 가격 상위 3개와 하위 3개를 테이블 형태로 보여주시오 (home3)

- 테이블에는 SGG\_NM(자치구명), BJDONG\_NM(법정동명), BLDG\_NM(건물명), BLDG\_AREA(건물면적), OBJ\_AMT(물건금액)을 정렬하여 보여주시오



In [ ]:

## LAB-1. '탐색적자료분석' 기본 화면 만들기

- '탐색적 자료분석' 버튼을 누르면 실행하는 화면을 만든다
- run\_eda\_home() 함수 호출
- run\_eda\_home() 함수는 별도의 파이썬 파일 eda.py로 만들고, app.py에서는 'from eda import run\_eda\_home'로 임포트함

```
In [1]: #
# app.py (app2.py)
#
# > streamlit run app.py
#

'''
import streamlit as st
from streamlit_option_menu import option_menu
from utils import load_data
from home import run_home
from eda import run_eda_home

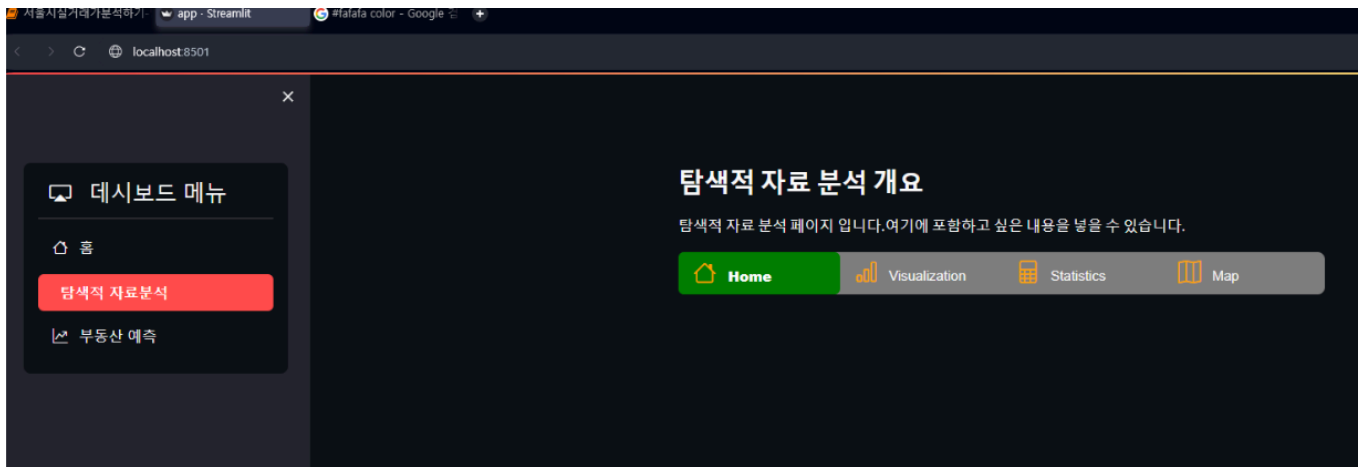
def main() :
    total_df = load_data()

    with st.sidebar:
        selected = option_menu('데시보드 메뉴', ['홈', '탐색적 자료분석', '부동산 예측'],
                               icons=['house', 'file-bar_graph', 'graph-up-arrow'], menu_icon='cast', default_index=0)

    if selected == '홈':
        run_home(total_df)
    elif selected == '탐색적 자료분석' :
        run_eda_home(total_df)
    elif selected == '부동산 예측' :
        pass
    else :
        print('error')

if __name__ == "__main__":
    main()
'''
```

Out[1]: 'Wnimport streamlit as stWnfrom streamlit\_option\_menu import option\_menuWnfrom utils import load\_dataWnfrom home import run\_
\_homeWnfrom eda import run\_eda\_homeWnWnWnWndef main() :Wn total\_df = load\_data()WnWn with st.sidebar:Wn selected
= option\_menu(W'데시보드 메뉴', [W'홈', W'탐색적 자료분석', W'부동산 예측'], Wn icons=
[W'house', W'file-bar\_graph', W'graph-up-arrow'], menu\_icon=W'cast', default\_index=0)Wn Wn if selected ==
W'홈':Wn run\_home(total\_df)Wn elif selected == W'탐색적 자료분석' :Wn run\_eda\_home(total\_df)Wn elif s
elected == W'부동산 예측' :Wn passWn else :Wn print(W'error')Wn Wnif \_\_name\_\_ == "\_\_main\_\_":Wn
main()Wn'



In [3]:

```
#
# eda.py (eda1.py)
#
# > streamlit run app.py
#

...

import streamlit as st
import pandas as pd
from streamlit_option_menu import option_menu

def run_eda_home() :
    st.markdown("### 탐색적 자료 분석 개요 \n"
                "탐색적 자료 분석 페이지 입니다."
                "여기에 포함하고 싶은 내용을 넣을 수 있습니다.")

    selected = option_menu(None, ['Home', 'Visualization', 'Statistics', 'Map'],
                           icons=['house', 'bar-chart', 'file-spreadsheet', 'map'],
                           menu_icon='cast', default_index=0, orientation='horizontal',
                           styles={
                               'container' : {
                                   'padding' : '0!important',
                                   'background-color' : '#808080'},
                               'icon' : {
                                   'color' : 'orange',
                                   'font-size' : '25px'},
                               'nav-link' : {
                                   'font-size' : '15px',
                                   'text-align' : 'left',
                                   'margin' : '0px',
                                   '--hover-color' : '#eee'},
                               'nav-link-selected' : {
                                   'background-color' : 'green'}
                           })

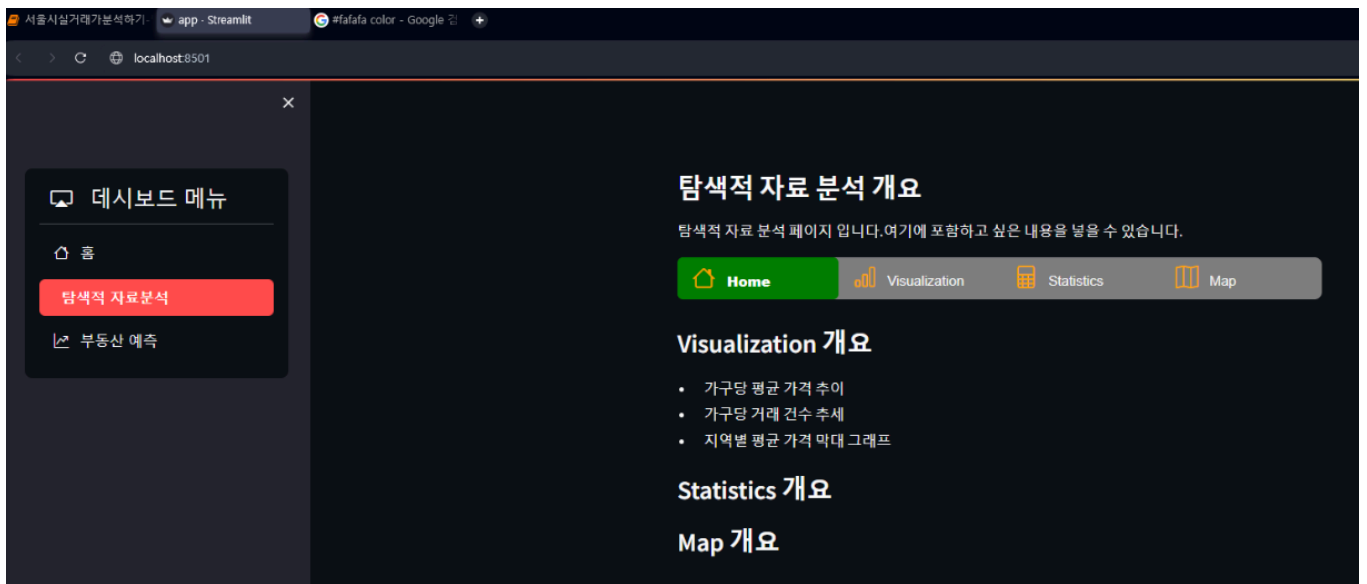
    if selected == 'Home' :
        pass
    elif selected == 'Visualization' :
        pass
    elif selected == 'Statistics' :
        pass
    elif selected == 'Map' :
        pass
    else:
        st.warning('Wrong')
    ...
```

Out[3]:

```
'\nimport streamlit as st\nimport pandas as pd\nfrom streamlit_option_menu import option_menu\n\ndef run_eda_home() :\n    st.markdown("### 탐색적 자료 분석 개요 \n"
                "탐색적 자료 분석 페이지 입니다."
                "여기에 포함하고 싶은 내용을 넣을 수 있습니다.")\n    selected = option_menu(None, ['Home', 'Visualization', 'Statistics', 'Map'],\n                           icons=['house', 'bar-chart', 'file-spreadsheet', 'map'],\n                           menu_icon='cast', default_index=0, orientation='horizontal',\n                           styles={\n                               'container' : {\n                                   'padding' : '0!important',\n                                   'background-color' : '#808080'},\n                               'icon' : {\n                                   'color' : 'orange',\n                                   'font-size' : '25px'},\n                               'nav-link' : {\n                                   'font-size' : '15px',\n                                   'text-align' : 'left',\n                                   'margin' : '0px',\n                                   '--hover-color' : '#eee'},\n                               'nav-link-selected' : {\n                                   'background-color' : 'green'}\n                           })\n    if selected == 'Home' :\n        pass\n    elif selected == 'Visualization' :\n        pass\n    elif selected == 'Statistics' :\n        pass\n    elif selected == 'Map' :\n        pass\n    else:\n        st.warning('Wrong')\n    ...'
```

## 실습과제 1: 홈 버튼을 누르면 다음과 같이 출력되도록 구현하시오

- 함수 이름은 eda.py에 home()을 만들어 구현하시오. (% eda2.py)



## Lab-2. '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

In [4]:

```
#
# eda.py (eda3.py)
#
# > streamlit run app.py
#

...

import streamlit as st
import pandas as pd
from streamlit_option_menu import option_menu

from viz import showViz

def home() :
    st.markdown("### Visualization 개요 \n\n"
                "- 가구당 평균 가격 추이 \n\n"
                "- 가구당 거래 건수 추이 \n\n"
                "- 지역별 평균 가격 막대 그래프 \n\n")
    st.markdown("### Statistics 개요 \n\n")
    st.markdown("### Map 개요 \n\n")

def run_eda_home(total_df) :
    st.markdown("### 탐색적 자료 분석 개요 \n\n"
                "탐색적 자료 분석 페이지 입니다."
                "여기에 포함하고 싶은 내용을 넣을 수 있습니다.")

    selected = option_menu(None, ['Home', 'Visualization', 'Statistics', 'Map'],
                           icons=['house', 'bar-chart', 'file-spreadsheet', 'map'],
                           menu_icon='cast', default_index=0, orientation='horizontal',
                           styles={
                               'container' : {
                                   'padding' : '0!important',
                                   'background-color' : '#808080'},
                               'icon' : {
                                   'color' : 'orange',
                                   'font-size' : '25px'},
                               'nav-link' : {
                                   'font-size' : '15px',
                                   'text-align' : 'left',
                                   'margin' : '0px',
                                   '--hover-color' : '#eee'},
                               'nav-link-selected' : {
                                   'background-color' : 'green'}
                           })

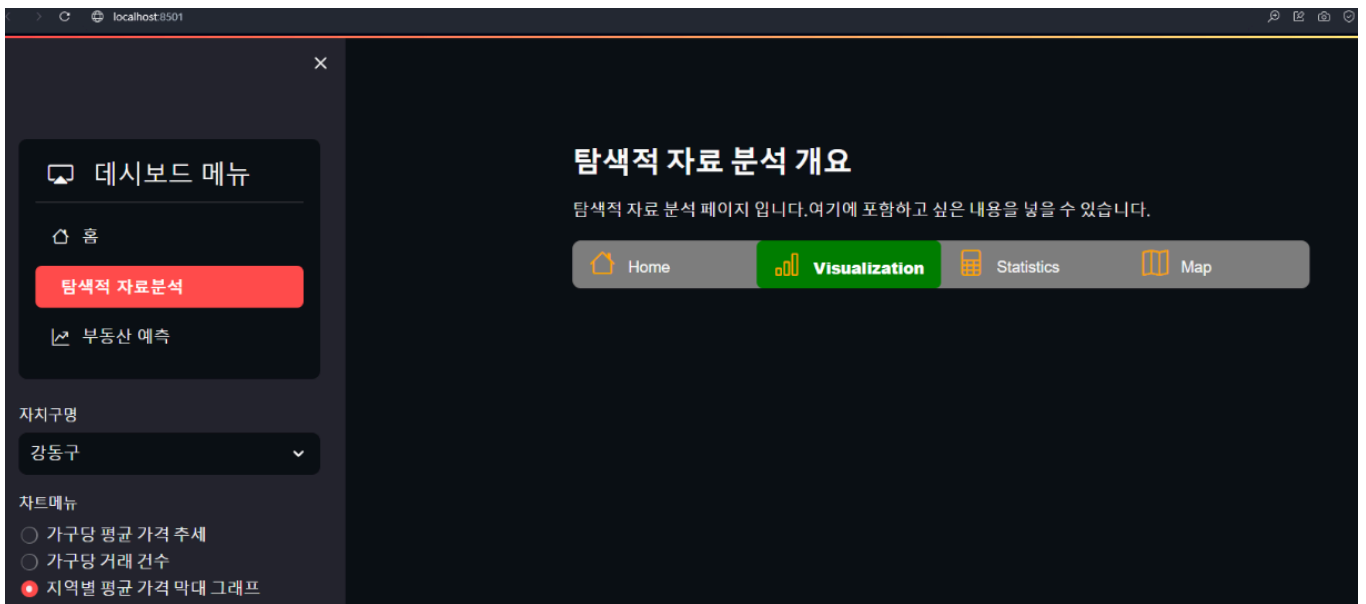
    if selected == 'Home' :
        home()
    elif selected == 'Visualization' :
```

```
Wnimport streamlit as stWnimport pandas as pdWnfrom streamlit_option_menu import option_menuWnWnfrom viz import showVizWn
Wndef home() :Wn    st.markdown("### Visualization 개요 Wn"Wn                                "- 가구당 평균 가격 추이 Wn"Wn
Wn    "- 가구당 거래 건수 추세 Wn"Wn                                "- 지역별 평균 가격 막대 그래프 Wn")Wn    st.markdown("### Statistics 개요
Wn")Wn    st.markdown("### Map 개요 Wn")Wn                                WnWndef run_eda_home(total_df) :Wn    st.markdown("### 탐색적 자
료 분석 개요 Wn"Wn                                "탐색적 자료 분석 페이지 입니다."Wn                                "여기에 포함하고 싶은 내용을 넣을 수
있습니다.")Wn    Wn    selected = option_menu(None, [W'HomeW', W'VisualizationW', W'StatisticsW', W'MapW'],Wn
icons=[W'houseW', W'bar-chartW', W'file-spreadsheetW', W'mapW'],Wn                                menu_icon=W'castW', default_in
dex=0, orientation=W'horizontalW',Wn                                styles={Wn                                W'containerW' : {Wn
W'paddingW' : W'0!importantW',Wn                                W'background-colorW' : W'#808080W'},Wn
W'iconW' : {Wn                                W'colorW' : W'orangeW',Wn
W'font-sizeW' : W'25pxW'},Wn                                W'nav-linkW' : {Wn
W'font-sizeW' : W'15pxW',Wn                                W'text-alignW' : W'leftW',Wn
W'marginW' : W'0pxW',Wn                                W'--hover-colorW' : W'#eeeW'},Wn
W'nav-link-selectedW' : {Wn                                W'background-colorW' : W'greenW'}Wn
})Wn    Wn    if selected == W'HomeW' :Wn                                home()Wn                                elif selected == W'VisualizationW' :Wn                                sho
wViz(total_df)Wn                                elif selected == W'StatisticsW' :Wn                                passWn                                els
e:Wn                                st.warning(W'WrongW')Wn'
```

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

```
Out[5]: 'Wnimport streamlit as stWnimport pandas as pdWnfrom plotly.subplots import make_subplotsWnimport plotly.express as pxWnWn
def showViz(total_df) :Wn    total_df[W'DEAL_YMDW'] = pd.to_datetime(total_df[W'DEAL_YMDW'], format=W'%Y-%m-%dW')Wn    Wn
sgg_nm = st.sidebar.selectbox(W'자지구명W', sorted(total_df[W'SGG_NMW'].unique()))Wn    selected = st.sidebar.radio(W'차트
메뉴W', Wn
[W'가구당 평균 가격 추세W', W'가구당 거래 건수W', W'지역별 평균 가격 막대 그래프
W'])Wn    Wn    if selected == W'가구당 평균 가격 추세W' :Wn        passWn    elif selected == W'가구당 거래 건수W' :Wn
passWn    elif selected == W'지역별 평균 가격 막대 그래프W' :Wn        passWn    else :Wn            st.warning("Error")Wn'
```

```
Out[5]: 'Wnimport streamlit as stWnimport pandas as pdWnfrom plotly.subplots import make_subplotsWnimport plotly.express as pxWnWn
def showViz(total_df) :Wn    total_df[W'DEAL_YMDW'] = pd.to_datetime(total_df[W'DEAL_YMDW'], format=W'%Y-%m-%dW')Wn    Wn
sgg_nm = st.sidebar.selectbox(W'자지구명W', sorted(total_df[W'SGG_NMW'].unique()))Wn    selected = st.sidebar.radio(W'차트
메뉴W', Wn
[W'가구당 평균 가격 추세W', W'가구당 거래 건수W', W'지역별 평균 가격 막대 그래프
W'])Wn    Wn    if selected == W'가구당 평균 가격 추세W' :Wn        passWn    elif selected == W'가구당 거래 건수W' :Wn
passWn    elif selected == W'지역별 평균 가격 막대 그래프W' :Wn        passWn    else :Wn            st.warning("Error")Wn'
```



## viz.py 파일에 showViz() 함수 구현 (viz2.py)

- '가구당 평균 가격 추세' 버튼 선택시 그래프 그리기 (아파트)

```
In [6]: #
# viz.py (viz2.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.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 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' :
        pass
    elif selected == '지역별 평균 가격 막대 그래프' :
        pass
    else :
        st.warning("Error")
    ...
```

```

Out[6]: 'Wnimport streamlit as stWnimport pandas as pdWnfrom plotly.subplots import make_subplotsWnimport plotly.express as pxWnWn
def meanChart(total_df, sgg_nm) :Wn    st.markdown(W'## 가구별 평균 가격 추세 WnW')WnWn    filtered_df = total_df[total_df
[W'SGG_NMW'] == sgg_nm]Wn    filtered_df = filtered_df[filtered_df[W'DEAL_YMDW'].between(W'2023-11-01W', W'2023-12-31W')]Wn
result = filtered_df.groupby([W'DEAL_YMDW', W'HOUSE_TYPEW'])[W'OBJ_AMTW'].agg(W'meanW').reset_index()Wn    Wn    df1 = resu
lt[result[W'HOUSE_TYPEW'] == W'아파트W'] Wn    df2 = result[result[W'HOUSE_TYPEW'] == W'단독다가구W'] Wn    df3 = result[re
sult[W'HOUSE_TYPEW'] == W'오피스텔W'] Wn    df4 = result[result[W'HOUSE_TYPEW'] == W'연립다세대W'] WnWn    fig = make_subpl
ots(rows=2, cols=2, Wn    shared_xaxes=True, Wn    subplot_titles=(W'아파트W', W'단
독다가구W', W'오피스텔W', W'연립다세대W'),Wn    horizontal_spacing=0.15)Wn    fig.add_trace(px.line(df
1, x=W'DEAL_YMDW', y=W'OBJ_AMTW', Wn    title=W'아파트 실거래가 평균W', markers=True).data[0], row=1,
col=1)Wn    Wn    fig.update_yaxes(tickformat=W'.0fW',Wn    title_text=W'물건가격(원)W',Wn
range=[result[W'OBJ_AMTW'].min(), result[W'OBJ_AMTW'].max()])Wn    fig.update_layout(Wn    title = W'가구별 평균값 추세
그래프W',Wn    width=800, height=600,Wn    showlegend=True, template=W'plotly_whiteW')Wn    st.plotly_chart(fig)Wn
Wndef showViz(total_df) :Wn    total_df[W'DEAL_YMDW'] = pd.to_datetime(total_df[W'DEAL_YMDW'], format=W'%Y-%m-%dW')Wn    Wn
sgg_nm = st.sidebar.selectbox(W'자치구명W', sorted(total_df[W'SGG_NMW'].unique()))Wn    selected = st.sidebar.radio(W'차트
메뉴W', Wn    [W'가구당 평균 가격 추세W', W'가구당 거래 건수W', W'지역별 평균 가격 막대 그래프
W'])Wn    Wn    if selected == W'가구당 평균 가격 추세W' :Wn        meanChart(total_df, sgg_nm)Wn    elif selected == W'가
구당 거래 건수W' :Wn        passWn    elif selected == W'지역별 평균 가격 막대 그래프W' :Wn        passWn    else :Wn
st.warning("Error")Wn'

```

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

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



```

In [ ]:

```