```python
import geopandas
import streamlit as st
import pandas     as pd
import numpy      as np
import folium

from streamlit_folium import folium_static
from folium.plugins   import MarkerCluster

st.set_page_config( layout='wide' )

@st.cache( allow_output_mutation=True )
def get_data( path ):
    data = pd.read_csv( path )

    return data


@st.cache( allow_output_mutation=True )
def get_geofile( url ):
    geofile = geopandas.read_file( url )

    return geofile

# get data
path = '../kc_house_data.csv'
data = get_data( path )

# get geofile
url = 'https://opendata.arcgis.com/datasets/
83fc2e72903343aabff6de8cb445b81c_2.geojson'
geofile = get_geofile( url )

# add new features
data['price_m2'] = data['price'] / data['sqft_lot']

# =======================
# Data Overview
# =======================
f_attributes = st.sidebar.multiselect( 'Enter columns',
data.columns )
f_zipcode = st.sidebar.multiselect(
    'Enter zipcode',
    data['zipcode'].unique() )

st.title( 'Data Overview' )

if ( f_zipcode != [] ) & ( f_attributes != [] ):
    data = data.loc[data['zipcode'].isin( f_zipcode ), f_attributes]

elif ( f_zipcode != [] ) & ( f_attributes == [] ):
    data = data.loc[data['zipcode'].isin( f_zipcode ), :]

elif ( f_zipcode == [] ) & ( f_attributes != [] ):
```

```python
        data = data.loc[:, f_attributes]

    else:
        data = data.copy()

    st.dataframe( data )

    c1, c2 = st.beta_columns((1, 1) )

    # Average metrics
    df1 = data[['id',
    'zipcode']].groupby( 'zipcode' ).count().reset_index()
    df2 = data[['price',
    'zipcode']].groupby( 'zipcode').mean().reset_index()
    df3 = data[['sqft_living',
    'zipcode']].groupby( 'zipcode').mean().reset_index()
    df4 = data[['price_m2',
    'zipcode']].groupby( 'zipcode').mean().reset_index()


    # merge
    m1 = pd.merge( df1, df2, on='zipcode', how='inner' )
    m2 = pd.merge( m1, df3, on='zipcode', how='inner' )
    df = pd.merge( m2, df4, on='zipcode', how='inner' )

    df.columns = ['ZIPCODE', 'TOTAL HOUSES', 'PRICE', 'SQRT LIVING',
                  'PRICe/M2']

    c1.header( 'Average Values' )
    c1.dataframe( df, height=600 )

    # Statistic Descriptive
    num_attributes = data.select_dtypes( include=['int64', 'float64'] )
    media = pd.DataFrame( num_attributes.apply( np.mean ) )
    mediana = pd.DataFrame( num_attributes.apply( np.median ) )
    std = pd.DataFrame( num_attributes.apply( np.std ) )

    max_ = pd.DataFrame( num_attributes.apply( np.max ) )
    min_ = pd.DataFrame( num_attributes.apply( np.min ) )

    df1 = pd.concat([max_, min_, media, mediana, std],
    axis=1 ).reset_index()

    df1.columns = ['attributes', 'max', 'min', 'mean', 'median', 'std']

    c2.header( 'Descriptive Analysis' )
    c2.dataframe( df1, height=800 )


    # =======================
    # Densidade de Portfolio
    # =======================
    st.title( 'Region Overview' )
```

```python
c1, c2 = st.beta_columns( ( 1, 1 ) )
c1.header( 'Portfolio Density' )


df = data.sample( 10 )

# Base Map - Folium
density_map = folium.Map( location=[data['lat'].mean(),
                                    data['long'].mean() ],
                                    default_zoom_start=15 )

marker_cluster = MarkerCluster().add_to( density_map )
for name, row in df.iterrows():
    folium.Marker( [row['lat'], row['long'] ],
        popup='Sold R${0} on: {1}. Features: {2} sqft, {3} bedrooms,
{4} bathrooms, year built: {5}'.format( row['price'],
                                        row['date'],
                                        row['sqft_living'],
                                        row['bedrooms'],
                                        row['bathrooms'],

row['yr_built'] ) ).add_to( marker_cluster )


with c1:
    folium_static( density_map )


# Region Price Map
c2.header( 'Price Density' )

df = data[['price',
'zipcode']].groupby( 'zipcode' ).mean().reset_index()
df.columns = ['ZIP', 'PRICE']

#df = df.sample( 10 )

geofile = geofile[geofile['ZIP'].isin( df['ZIP'].tolist() )]

region_price_map = folium.Map( location=[data['lat'].mean(),
                                    data['long'].mean() ],
                                    default_zoom_start=15 )


region_price_map.choropleth( data = df,
                             geo_data = geofile,
                             columns=['ZIP', 'PRICE'],
                             key_on='feature.properties.ZIP',
                             fill_color='YlOrRd',
                             fill_opacity = 0.7,
                             line_opacity = 0.2,
                             legend_name='AVG PRICE' )

with c2:
```

```
folium_static( region_price_map )
```