dashboard

May 24, 2021

1 San Francisco Rental Prices Dashboard

In this notebook, you will compile the visualizations from the previous analysis into functions that can be used for a Panel dashboard.

[19]:
```
# imports
import panel as pn
pn.extension('plotly')
import plotly.express as px
from panel.interact import interact
import pandas as pd
import hvplot.pandas
import matplotlib.pyplot as plt
import os
from pathlib import Path
from dotenv import load_dotenv
import numpy as np
```

[20]:
```
# Read the Mapbox API key
load_dotenv()
map_box_api = os.getenv("mapbox")
px.set_mapbox_access_token(map_box_api)
```

2 Import Data

[21]:
```
# Import the necessary CSVs to Pandas DataFrames
# YOUR CODE HERE!
file_path = Path("Data/sfo_neighborhoods_census_data.csv")
sfo_data = pd.read_csv(file_path, index_col="year")
#print(sfo_data.head())
#print('len = ',len(sfo_data))

file_path2 = Path("Data/neighborhoods_coordinates.csv")
header_list = ["neighborhood", "Lat", "Lon"]
nb_coord_data = pd.read_csv(file_path2, index_col="neighborhood",␣
 ↪names=header_list, skiprows=1)
#nb_coord_data.rename(index={"Neighborhood": "neighborhood"}, inplace=True)
```

```
#sfo_data = pd.read_csv(file_path2, index_col="year")
#print(nb_coord_data.head())
#print('len = ',len(nb_coord_data))
```

SFO Yearly, Neighborhood and Map calculatations - global space outside the functions

[22]:
```
sfo_housing_costs = sfo_data[['sale_price_sqr_foot', 'gross_rent']]
sfo_housing_costs_mean = sfo_housing_costs.
 ↪groupby('year')[['sale_price_sqr_foot', 'gross_rent']].mean()


sfo_nb_housing_costs = sfo_data[['neighborhood', 'sale_price_sqr_foot']]
sfo_nb_housing_costs_mean = sfo_nb_housing_costs.groupby(['neighborhood',␣
 ↪'year']).mean()
#print(sfo_nb_housing_costs_mean.tail())
#sfo_nb_housing_costs_mean.reset_index()
```

Cost Analysis calculatations - global space outside the functions

[23]:
```
sfo_data_exp = sfo_data.groupby(['neighborhood']).mean()
#print(sfo_data_exp)
sfo_data_exp10 = sfo_data_exp.sort_values("sale_price_sqr_foot",␣
 ↪ascending=False).head(10)
top10_compare = sfo_data_exp
sfo_data_exp_sale = sfo_data_exp10['sale_price_sqr_foot']

#print(sfo_data_exp10_reset.head())
sfo_data_exp10_reset = sfo_data_exp10.reset_index()
#print(sfo_data_exp10_reset)
#print(sfo_data.head())
#print(sfo_data.head())
sfo_cost_anal = sfo_data[sfo_data["neighborhood"].
 ↪isin(sfo_data_exp10_reset["neighborhood"])]
sfo_cost_anal.reset_index(inplace=True)
#print(sfo_cost_anal)
```

2.1 Panel Visualizations

In this section, you will copy the code for each plot type from your analysis notebook and place it into separate functions that Panel can use to create panes for the dashboard.

These functions will convert the plot object to a Panel pane.

Be sure to include any DataFrame transformation/manipulation code required along with the plotting code.

Return a Panel pane object from each function that can be used to build the dashboard.

Note: Remove any `.show()` lines from the code. We want to return the plots instead of showing them. The Panel dashboard will then display the plots.

[24]:
```
# Define Panel Visualization Functions
def housing_units_per_year():
    """Housing Units Per Year."""

    # YOUR CODE HERE!
    sfo_housing_units_mean = sfo_data.groupby('year')['housing_units'].mean()
    mean_plot = sfo_housing_units_mean.hvplot.bar(ylim=(365000,387500)).
 ↪opts(yformatter="%.0f", title='SFO Housing Units Mean')
    return mean_plot

def average_gross_rent():
    """Average Gross Rent in San Francisco Per Year."""

    # YOUR CODE HERE!
    #sfo_housing_costs_mean_rent = sfo_housing_costs_mean['gross_rent']
    #sfo_housing_rent_mean_plot = sfo_housing_costs_mean_rent.plot.
 ↪line(title='SFO Housing Average gross rent by Year -  Line plot', color =␣
 ↪"g")

    #sfo_housing_rent_mean_plot = sfo_housing_costs_mean['gross_rent'].plot.
 ↪line(title='SFO Housing Average gross rent by Year -  Line plot', color =␣
 ↪"g")
    #return sfo_housing_rent_mean_plot
    sfo_housing_rent_mean_plot_hv = sfo_housing_costs_mean['gross_rent'].hvplot.
 ↪line().opts(title='SFO Housing Average gross rent by Year  -  HV Line Plot',␣
 ↪color = "g")
    return sfo_housing_rent_mean_plot_hv



def average_sales_price():
    """Average Sales Price Per Year."""

    # YOUR CODE HERE!
#    sfo_housing_sale_mean_plot_hv =␣
 ↪sfo_housing_costs_mean['sale_price_sqr_foot'].hvplot.line().opts( title='SFO␣
 ↪Housing Average Price per sqr ft - HV Line plot', color = "m")

    #sfo_housing_sale_mean_plot = sfo_housing_costs_mean['sale_price_sqr_foot'].
 ↪plot.line( title='SFO Housing Average Price per sqr ft - Line plot', color =␣
 ↪"m")
    #return sfo_housing_sale_mean_plot
```

```
    sfo_housing_sale_mean_plot_hv =␣
↪sfo_housing_costs_mean['sale_price_sqr_foot'].hvplot.line().opts( title='SFO␣
↪Housing Average Price per sqr ft - HV Line plot', color = "m")
    return sfo_housing_sale_mean_plot_hv

def nb_sale(neighborhood):
#     sfo_nb_housing_costs_mean = sfo_data.
↪groupby(['neighborhood'])['sale_price_sqr_foot'].mean()
    sloc = sfo_nb_housing_costs_mean.loc[neighborhood]
#     sfo_nb_housing_sale_mean_plot_hv = sloc.hvplot.line(x='year',␣
↪y='sale_price_sqr_foot').opts( title='SFO NB Housing Average Price per sqr␣
↪ft - HV Line plot', color = "m")
    sfo_nb_housing_sale_mean_plot_hv = sloc.hvplot.line(x='year',␣
↪y='sale_price_sqr_foot').opts( title='SFO NB Housing Average Price per sqr␣
↪ft - HV Line plot', color = "m")

    #sfo_nb_housing_sale_mean_plot_hv = sfo_nb_housing_costs_mean.hvplot.
↪line(x='year', y='sale_price_sqr_foot').opts( title='SFO NB Housing Average␣
↪Price per sqr ft - HV Line plot', color = "m")

    return sfo_nb_housing_sale_mean_plot_hv

def average_price_by_neighborhood():
    """Average Prices by Neighborhood."""

    # YOUR CODE HERE!
#     sfo_nb_housing_costs = sfo_data[['neighborhood', 'sale_price_sqr_foot']]
#     sfo_nb_housing_costs_mean = sfo_nb_housing_costs.groupby(['neighborhood',␣
↪'year']).mean()
#     #print(sfo_nb_housing_costs_mean.tail())
#     sfo_nb_housing_costs_mean.reset_index()
    layout_sale = interact(nb_sale, neighborhood=sfo_data['neighborhood'])
    return pn.Column('**SFO Housing**', pn.Row(layout_sale[1], layout_sale[0]))

def top_most_expensive_neighborhoods():
    """Top 10 Most Expensive Neighborhoods."""

    # YOUR CODE HERE!
    sfo_data_exp = sfo_data.groupby(['neighborhood']).mean()
    #print(sfo_data_exp)
    sfo_data_exp10 = sfo_data_exp.sort_values("sale_price_sqr_foot",␣
↪ascending=False).head(10)
    top10_compare = sfo_data_exp
    sfo_data_exp_sale = sfo_data_exp10['sale_price_sqr_foot']
    sfo_data_exp_sale_plot = sfo_data_exp_sale.hvplot.bar(rot=90, height=400,␣
↪title='Top 10 most expensive neighborhoods')
```

```
    return sfo_data_exp_sale_plot

def most_expensive_neighborhoods_rent_sales():
    """Comparison of Rent and Sales Prices of Most Expensive Neighborhoods."""

    # YOUR CODE HERE!
    sfo_data_mean = sfo_data.groupby(['neighborhood', 'year']).mean()
    sfo_data_mean = sfo_data_mean.reset_index()
    sfo_data_mean = sfo_data_mean[sfo_data_mean["neighborhood"].
 ↪isin(sfo_data_exp10_reset["neighborhood"])]
    sfo_data_mean_plot = sfo_data_mean.hvplot.bar(x='year',␣
 ↪y=['sale_price_sqr_foot', 'gross_rent'], groupby='neighborhood', rot=90,␣
 ↪height=400, title='Compare costs to purchase vs Rental income by␣
 ↪Neighborhoods')
    return sfo_data_mean_plot


def parallel_coordinates():
    """Parallel Coordinates Plot."""

    # YOUR CODE HERE!
    parallel_coord = px.parallel_coordinates(sfo_data_exp10_reset,␣
 ↪color='sale_price_sqr_foot')
    return parallel_coord


def parallel_categories():
    """Parallel Categories Plot."""

    # YOUR CODE HERE!
    parallel_cat = px.parallel_categories(
        sfo_data_exp10_reset,
        dimensions=["neighborhood", "sale_price_sqr_foot", "housing_units",␣
 ↪"gross_rent"],
        color="sale_price_sqr_foot",
        color_continuous_scale=px.colors.sequential.Inferno,
        labels={
            "neighborhood": "Neighborhood",
            "sale_price_sqr_foot": "Sale price/sqft",
            "housing_units": "Housing Units",
            "gross_rent": "Gross Rent",
        },
    )
    return parallel_cat
```

```
def neighborhood_map():
    """Neighborhood Map."""

    # YOUR CODE HERE!
    sfo_data_exp = sfo_data.groupby(['neighborhood']).mean()
    combined_df = pd.concat([sfo_data_exp, nb_coord_data], axis="columns",␣
 ↪join="inner")
    #combined_df.head()
    combined_df.reset_index(inplace=True)
    map_plot = px.scatter_mapbox(
        combined_df,
        lat="Lat",
        lon="Lon",
        size="sale_price_sqr_foot",
        color="gross_rent",
        zoom=10,
        text='neighborhood')
    #pane = pn.pane.Plotly(plot)
    #pane
    return map_plot


def sunburst():
    """Sunburst Plot."""

    # YOUR CODE HERE!
    fig = px.sunburst(sfo_cost_anal, path=['year', 'neighborhood'],␣
 ↪values='sale_price_sqr_foot',
                      color='gross_rent', hover_data=['housing_units'],
                      color_continuous_scale='RdBu',
                      #color_continuous_midpoint=np.
 ↪average(sfo_cost_anal['gross_rent'],␣
 ↪weights=sfo_cost_anal['sale_price_sqr_foot'])
                         )
    return fig
```

2.2 Panel Dashboard

In this section, you will combine all of the plots into a single dashboard view using Panel. Be creative with your dashboard design!

[25]:
```
# Create a Title for the Dashboard
# YOUR CODE HERE!


# Create a tab layout for the dashboard
# YOUR CODE HERE!
```

```
map_plot_welcome = pn.Row(neighborhood_map())

row_of_yearly = pn.Column(
    "## Yearly SFO Plot analysis",
    housing_units_per_year(),
    average_gross_rent(),
#    sfo_housing_costs_mean['gross_rent'].plot.line(title='SFO Housing Average␣
 ↪gross rent by Year -  Line plot', color = "g"),
    average_sales_price()
#    sfo_housing_costs_mean['sale_price_sqr_foot'].plot.line( title='SFO␣
 ↪Housing Average Price per sqr ft - Line plot', color = "m")
)

col_neighborhood = pn.Column(
    "## SFO Neighborhood Plot analysis",
    average_price_by_neighborhood(),
    top_most_expensive_neighborhoods(),
    most_expensive_neighborhoods_rent_sales(),
    neighborhood_map()
)

col_parallel = pn.Column(
    "## SFO Parallel Coordinates and Category Plot analysis",
    parallel_coordinates(),
    parallel_categories()
)

col_sunburst = pn.Column(
    "## SFO Sunbirst Plot analysis",
    sunburst()
)


#row_of_bar = pn.Row(num_foreclosures_plot)
#row_of_bar.append(num_sales_plot)

sfo_analysis_dashboard = pn.Tabs(
    ("Welcome", map_plot_welcome),
    ("Yearly Market Analysis", row_of_yearly),
    ("Neighborhood Analysis", col_neighborhood),
    ("Parallel plots Analysis", col_parallel),
    ("Sunburst plot Analysis", col_sunburst)
)



# Create the dashboard
```

```
# YOUR CODE HERE!
# Create tabs
```

2.3 Serve the Panel Dashboard

[26]:
```
# Serve the# dashboard
# YOUR CODE HERE!
sfo_analysis_dashboard.servable(title="Real estate Analysis of San Francisco␣
↪from 2010 to 2016")
```

[26]:
```
Tabs
    [0] Row
        [0] Plotly(Figure)
    [1] Column
        [0] Markdown(str)
        [1] HoloViews(Bars)
        [2] HoloViews(Curve)
        [3] HoloViews(Curve)
    [2] Column
        [0] Markdown(str)
        [1] Column
            [0] Markdown(str)
            [1] Row
                [0] Row
                    [0] HoloViews(Curve, name='interactive03199')
                [1] Column
                    [0] Select(name='neighborhood', options=['Alamo Square',
…], value='Alamo Square')
        [2] HoloViews(Bars)
        [3] Row
            [0] HoloViews(DynamicMap)
            [1] Column
                [0] WidgetBox
                    [0] Select(margin=(20, 20, 20, 20), name='neighborhood',
options=['Cow Hollow', …], value='Cow Hollow', width=250)
                [1] VSpacer()
        [4] Plotly(Figure)
    [3] Column
        [0] Markdown(str)
        [1] Plotly(Figure)
        [2] Plotly(Figure)
    [4] Column
        [0] Markdown(str)
        [1] Plotly(Figure)
```

3 Debugging

Note: Some of the Plotly express plots may not render in the notebook through the panel functions.

However, you can test each plot by uncommenting the following code

[27]:
```
#housing_units_per_year()
```

[28]:
```
#average_gross_rent()
```

[29]:
```
#average_sales_price()
```

[30]:
```
# average_price_by_neighborhood()
```

[31]:
```
# top_most_expensive_neighborhoods()
```

[32]:
```
# most_expensive_neighborhoods_rent_sales()
```

[33]:
```
# neighborhood_map().show()
```

[34]:
```
# parallel_categories()
```

[35]:
```
# parallel_coordinates()
```

[36]:
```
# sunburst()
```

[ ]:
```

```