rental_analysis

April 10, 2021

1 San Francisco Housing Cost Analysis

In this assignment, you will perform fundamental analysis for the San Francisco housing market to allow potential real estate investors to choose rental investment properties.

[1]:

```
# imports
import panel as pn
import plotly.express as px
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import os
from pathlib import Path
from dotenv import load_dotenv

import warnings
warnings.filterwarnings('ignore')
```

[2]:

```
# Set up Panel Plotly extension
pn.extension('plotly')
```

[3]:

```
# Import hvplot.pandas after pn.extension
# This avoids plotly initialization failure
import hvplot.pandas
```

[4]:

```
# Read the Mapbox API key
load_dotenv('api_keys.env')
map_box_api = os.getenv("MAPBOX_TOKEN")
```

1.1 Load Data

[5]:

```
# Read the census data into a Pandas DataFrame
file_path = Path("Data/sfo_neighborhoods_census_data.csv")
df_costs = pd.read_csv(file_path, index_col="year")
df_costs.head()
```

[5]:

```
          neighborhood  sale_price_sqr_foot  housing_units  gross_rent
year
```

```
2010      Alamo Square          291.182945        372560        1239
2010        Anza Vista          267.932583        372560        1239
2010           Bayview          170.098665        372560        1239
2010  Buena Vista Park          347.394919        372560        1239
2010  Central Richmond          319.027623        372560        1239
```

1.2 Housing Units Per Year

In this section, you will calculate the number of housing units per year and visualize the results as a bar chart using the Pandas plot function.

Hint: Use the Pandas `groupby` function.

Optional challenge: Use the min, max, and std to scale the y limits of the chart.

[6]:
```
df_costs.count()
```

[6]:
```
neighborhood          397
sale_price_sqr_foot   392
housing_units         397
gross_rent            397
dtype: int64
```

[7]:
```
df_costs.dropna(inplace=True)
df_costs.count()
```

[7]:
```
neighborhood          392
sale_price_sqr_foot   392
housing_units         392
gross_rent            392
dtype: int64
```

[8]:
```
# Calculate the mean number of housing units per year (hint: use groupby)
sfo_housing_units_mean = df_costs.groupby("year")["housing_units"].mean()
#sfo_housing_units_mean.describe()
sfo_housing_units_mean.head()
```

[8]:
```
year
2010    372560
2011    374507
2012    376454
2013    378401
2014    380348
Name: housing_units, dtype: int64
```

[9]:
```
# Save the dataframe as a csv file
sfo_housing_units_mean.to_csv('sfo_housing_units_by_year.csv', index=True)
```

[10]:

```
# Use the Pandas plot function to plot the average housing units per year.
# Note: You will need to manually adjust the y limit of the chart using the min␣
 ↪and max values from above.
sfo_housing_units_mean.hvplot.bar(xlabel='Year', ylabel='Housing Units',␣
 ↪height=400).opts(title='Housing Units in San Francisco from 2010 to 2016',␣
 ↪yformatter="%.0f")
```

[10]:

```
:Bars   [year]   (housing_units)
```

[11]:

```
# sfo_housing_units_by_year.hvplot.bar(ylim=(370000, 385000), height=600).
 ↪opts(yformatter="%.0f")
bar_min = sfo_housing_units_mean.describe(include='all').loc['min'] -␣
 ↪(sfo_housing_units_mean.describe(include='all').loc['std']/4)
bar_max = sfo_housing_units_mean.describe(include='all').loc['max'] +␣
 ↪(sfo_housing_units_mean.describe(include='all').loc['std']/4)

# Optional Challenge: Use the min, max, and std to scale the y limits of the␣
 ↪chart
sfo_housing_units_mean.hvplot.bar(xlabel='Year', ylabel='Housing Units',␣
 ↪ylim=(bar_min, bar_max), height=400).opts(title='Housing Units in San␣
 ↪Francisco from 2010 to 2016', yformatter="%.0f")
```

[11]:

```
:Bars   [year]   (housing_units)
```

- - -

1.3 Average Housing Costs in San Francisco Per Year

In this section, you will calculate the average monthly rent and the average price per square foot for each year. An investor may wish to better understand the sales price of the rental property over time. For example, a customer will want to know if they should expect an increase or decrease in the property value over time so they can determine how long to hold the rental property. Plot the results as two line charts.

Optional challenge: Plot each line chart in a different color.

[12]:

```
# Calculate the average sale price per square foot and average gross rent
sfo_yearly_avg = df_costs.groupby(['year']).mean()
```

[13]:

```
sfo_yearly_avg
```

[13]:

```
      sale_price_sqr_foot  housing_units  gross_rent
year
2010           369.344353         372560        1239
2011           341.903429         374507        1530
2012           399.389968         376454        2324
2013           483.600304         378401        2971
2014           556.277273         380348        3528
```

```
2015          632.540352          382295        3739
2016          697.643709          384242        4390
```

[14]:

```
# Create two line charts, one to plot the average sale price per square foot␣
 ↪and another for average montly rent
# Line chart for average sale price per square foot
avg_price_sf = sfo_yearly_avg['sale_price_sqr_foot'].hvplot.
 ↪line(line_color='purple',xlabel='Year',ylabel='USD',width=500,height=400,grid=True,title␣
 ↪= 'Average Price per SqFt by Year')

# Line chart for average montly rent
avg_monthly_rent = sfo_yearly_avg['gross_rent'].hvplot.
 ↪line(line_color='red',xlabel='Year',ylabel='Price per␣
 ↪SqFt',width=500,height=400,grid=True, title='Average Gross Rent by Year')


avg_monthly_rent
```

[14]:

```
:Curve   [year]   (gross_rent)
```

[15]:

```
avg_price_sf
```

[15]:

```
:Curve   [year]   (sale_price_sqr_foot)
```

1.4 Average Prices by Neighborhood

In this section, you will use hvplot to create two interactive visulizations of average prices with a dropdown selector for the neighborhood. The first visualization will be a line plot showing the trend of average price per square foot over time for each neighborhood. The second will be a line plot showing the trend of average montly rent over time for each neighborhood.

Hint: It will be easier to create a new DataFrame from grouping the data and calculating the mean prices for each year and neighborhood

[16]:

```
# Group by year and neighborhood and then create a new dataframe of the mean␣
 ↪values
sfo_neighborhood_avg = df_costs.groupby(['year', 'neighborhood']).mean()
sfo_neighborhood_avg.count()
```

[16]:

```
sale_price_sqr_foot    392
housing_units          392
gross_rent             392
dtype: int64
```

[18]:

```
# Use hvplot to create an interactive line chart of the average price per sq ft.
# The plot should have a dropdown selector for the neighborhood
neighborgood = sfo_neighborhood_avg['sale_price_sqr_foot']
```

```
neighborgood.hvplot.
 ↪line(groupby='neighborhood',line_color='blue',xlabel='Year',ylabel='Avg.␣
 ↪Sale Price per Square Foot',width=600,height=300)
```

[18]:
```
:DynamicMap   [neighborhood]
   :Curve   [year]   (sale_price_sqr_foot)
```

[19]:
```
# Use hvplot to create an interactive line chart of the average monthly rent.
# The plot should have a dropdown selector for the neighborhood
sfo_neighborhood_avg['gross_rent'].hvplot.
 ↪line(groupby='neighborhood',line_color='green',xlabel='Year',ylabel='Average␣
 ↪Gross Rent per Year',width=600,height=300,title='Neighborhood')
```

[19]:
```
:DynamicMap   [neighborhood]
   :Curve   [year]   (gross_rent)
```

1.5 The Top 10 Most Expensive Neighborhoods

In this section, you will need to calculate the mean sale price per square foot for each neighborhood and then sort the values to obtain the top 10 most expensive neighborhoods on average. Plot the results as a bar chart.

[20]:
```
# Getting the data from the top 10 expensive neighborhoods to own
sfo_top_neighborhood = df_costs.groupby(['neighborhood']).mean()
sfo_top_neighborhood.sort_values('sale_price_sqr_foot', ascending=False,␣
 ↪inplace=True)
df_expensive_neighborhoods = sfo_top_neighborhood[:10]
```

[21]:
```
# Plotting the data from the top 10 expensive neighborhoods
df_expensive_neighborhoods.hvplot.
 ↪bar(height=400,x='neighborhood',xlabel='Neighborhood',y='sale_price_sqr_foot',ylabel='Avg.
 ↪ Sale Price per Square Foot',rot=90).opts(title='Top 10 Expensive␣
 ↪Neighborhoods in SFO')
```

[21]:
```
:Bars   [neighborhood]   (sale_price_sqr_foot)
```

1.6 Comparing cost to purchase versus rental income

In this section, you will use `hvplot` to create an interactive visualization with a dropdown selector for the neighborhood. This visualization will feature a side-by-side comparison of average price per square foot versus average montly rent by year.

Hint: Use the `hvplot` parameter, `groupby`, to create a dropdown selector for the neighborhood.

[22]:
```
sfo_neighborhood_avg.head()
```

[22]:
```
                         sale_price_sqr_foot  housing_units  gross_rent
year neighborhood
2010 Alamo Square                 291.182945         372560        1239
     Anza Vista                   267.932583         372560        1239
     Bayview                      170.098665         372560        1239
     Buena Vista Park             347.394919         372560        1239
     Central Richmond             319.027623         372560        1239
```

[23]:
```
# Fetch the previously generated DataFrame that was grouped by year and␣
↪neighborhood
sfo_neighborhood_avg.hvplot.bar(groupby='neighborhood',height=400,x='year',␣
↪xlabel='Year',y=['sale_price_sqr_foot','gross_rent'],ylabel='Price',rot=90,␣
↪title='Comparing Cost to Purchase Versus Rental Income')
```

[23]:
```
:DynamicMap   [neighborhood]
   :Bars   [year,Variable]   (value)
```

1.7 Neighborhood Map

In this section, you will read in neighborhoods location data and build an interactive map with the average house value per neighborhood. Use a scatter_mapbox from Plotly express to create the visualization. Remember, you will need your Mapbox API key for this.

1.7.1 Load Location Data

[24]:
```
# Load neighborhoods coordinates data
file_path2 = Path("Data/neighborhoods_coordinates.csv")
neighborhood_coordinates = pd.read_csv(file_path2)
neighborhood_coordinates.columns=["neighborhood", "lat", "log"]
neighborhood_coordinates.head()
```

[24]:
```
      neighborhood        lat         log
0     Alamo Square  37.791012 -122.402100
1       Anza Vista  37.779598 -122.443451
2          Bayview  37.734670 -122.401060
3  Bayview Heights  37.728740 -122.410980
4   Bernal Heights  37.728630 -122.443050
```

1.7.2 Data Preparation

You will need to join the location data with the mean values per neighborhood.

1. Calculate the mean values for each neighborhood.
2. Join the average values with the neighborhood locations.

[25]:
```
# Calculate the mean values for each neighborhood
neighborhood_avg = df_costs.groupby('neighborhood').mean()
```

```
neighborhood_avg.reset_index(inplace=True)
```

[26]:

```
# Join the average values with the neighborhood locations
neighborhood_map = pd.merge(neighborhood_coordinates, neighborhood_avg,␣
 ↪on='neighborhood')
neighborhood_map.head()
```

[26]:

```
       neighborhood        lat         log  sale_price_sqr_foot  \
0      Alamo Square  37.791012 -122.402100           366.020712
1        Anza Vista  37.779598 -122.443451           373.382198
2           Bayview  37.734670 -122.401060           204.588623
3   Bayview Heights  37.728740 -122.410980           590.792839
4  Buena Vista Park  37.768160 -122.439330           452.680591

   housing_units   gross_rent
0       378401.0  2817.285714
1       379050.0  3031.833333
2       376454.0  2318.400000
3       382295.0  3739.000000
4       378076.5  2698.833333
```

1.7.3 Mapbox Visualization

Plot the average values per neighborhood using a Plotly express `scatter_mapbox` visualization.

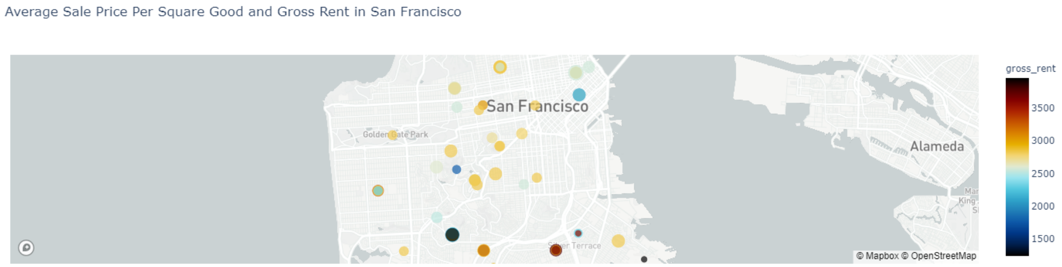
[27]:

```
# Set the mapbox access token

# Set the Mapbox API
px.set_mapbox_access_token(map_box_api)

# Create a scatter mapbox to analyze neighborhood info
# Plot Data
map_plot = px.scatter_mapbox(
    neighborhood_map,
    lat="lat",
    lon="log",
    size="sale_price_sqr_foot",
    color="gross_rent",
    zoom=11,
    color_continuous_scale=px.colors.cyclical.IceFire,
    size_max=15,
    title='Average Sale Price Per Square Good and Gross Rent in San Francisco'
)

# Display the map
map_plot.show()
```



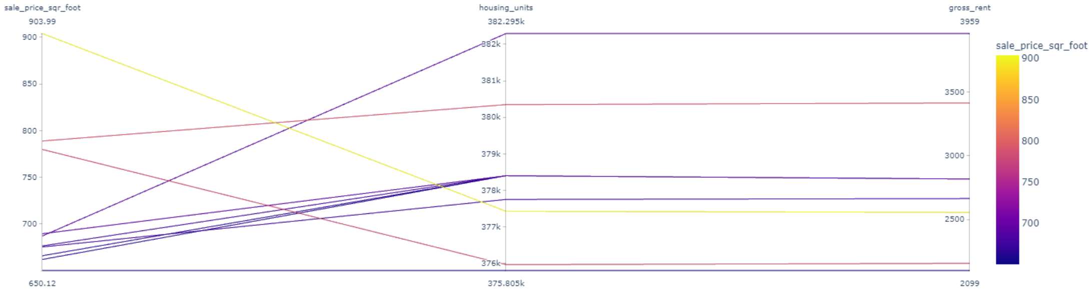
1.8 Cost Analysis - Optional Challenge

In this section, you will use Plotly express to create visualizations that investors can use to interactively filter and explore various factors related to the house value of the San Francisco's neighborhoods.

1.8.1 Create a DataFrame showing the most expensive neighborhoods in San Francisco by year

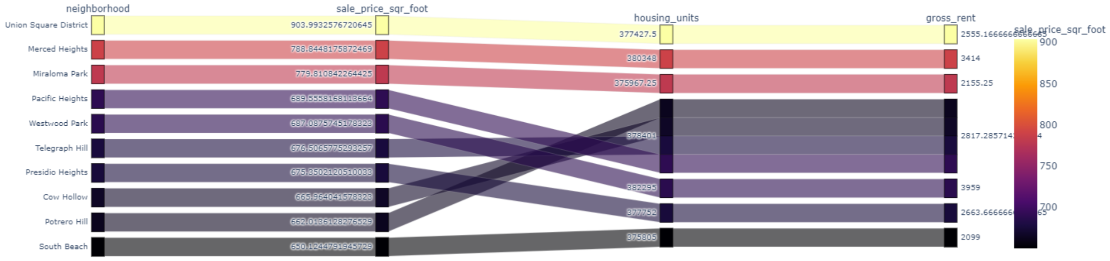
[28]:
```
# Fetch the data from all expensive neighborhoods per year.
df_expensive_neighborhoods.reset_index(inplace=True)
df_expensive_neighborhoods_per_year = df_costs[df_costs["neighborhood"].
 ↪isin(df_expensive_neighborhoods["neighborhood"])]
df_expensive_neighborhoods_per_year.reset_index(inplace=True)
```

[29]:
```
# Parallel Coordinates Plot
px.parallel_coordinates(df_expensive_neighborhoods, color='sale_price_sqr_foot')
```



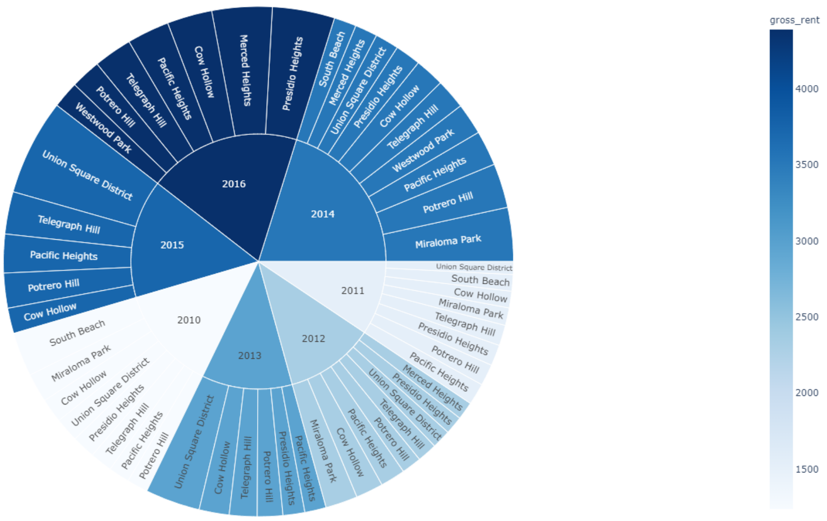
1.8.2 Create a parallel coordinates plot and parallel categories plot of most expensive neighborhoods in San Francisco per year

[30]:
```
# Parallel Categories Plot
px.parallel_categories(
    df_expensive_neighborhoods,
    dimensions=['neighborhood','sale_price_sqr_foot','housing_units',␣
 ↪'gross_rent'],
    color='sale_price_sqr_foot',
    color_continuous_scale=px.colors.sequential.Inferno
)
```



1.8.3 Create a sunburst chart to conduct a costs analysis of most expensive neighborhoods in San Francisco per year

[31]:
```
# Sunburst Plot
px.sunburst(
    df_expensive_neighborhoods_per_year,
    path=['year', 'neighborhood'],
    values='sale_price_sqr_foot',
    color='gross_rent',
    color_continuous_scale='blues',
    width=800,
    height=800
)
```


2016
Presidio Heights
Merced Heights
Cow Hollow
Pacific Heights
Telegraph Hill
Potrero Hill
Westwood Park
2014
South Beach
Merced Heights
Union Square District
Presidio Heights
Cow Hollow
Telegraph Hill
Westwood Park
Pacific Heights
Potrero Hill
Miraloma Park
2015
Union Square District
Telegraph Hill
Pacific Heights
Potrero Hill
Cow Hollow
2011
Union Square District
South Beach
Cow Hollow
Miraloma Park
Telegraph Hill
Presidio Heights
Potrero Hill
Pacific Heights
2012
Merced Heights
Presidio Heights
Union Square District
Telegraph Hill
Potrero Hill
Pacific Heights
Cow Hollow
Miraloma Park
2013
Pacific Heights
Presidio Heights
Potrero Hill
Telegraph Hill
Cow Hollow
Union Square District
2010
South Beach
Miraloma Park
Cow Hollow
Union Square District
Presidio Heights
Telegraph Hill
Pacific Heights
Potrero Hill
gross_rent
4000
3500
3000
2500
2000
1500

```
[ ]:
```