rental_analysis

May 24, 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
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 numpy as np
import os
from pathlib import Path
from dotenv import load_dotenv

import warnings
warnings.filterwarnings('ignore')
import numpy as np
```

```
Bad key "text.kerning_factor" on line 4 in
C:\Users\chakravartiraghavan\anaconda3\envs\pyvizenv\lib\site-
packages\matplotlib\mpl-data\stylelib\_classic_test_patch.mplstyle.
You probably need to get an updated matplotlibrc file from
http://github.com/matplotlib/matplotlib/blob/master/matplotlibrc.template
or from the matplotlib source distribution
```

[2]:
```
#pn.extension()
```

[3]:
```
# Read the Mapbox API key
load_dotenv()
map_box_api = os.getenv("mapbox")
```

[4]:
```
#print(map_box_api)      # check Mapbox Api key but dont leave it in code
```

1.1 Load Data

[5]:
```
# Read the census data into a Pandas DataFrame
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))
```

```
          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
len =  397
```

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]:
```
# Calculate the mean number of housing units per year (hint: use groupby)
# YOUR CODE HERE!
sfo_housing_units_mean = sfo_data.groupby('year')['housing_units'].mean()
print('Mean')
print(sfo_housing_units_mean)
sfo_housing_units_min = sfo_data.groupby('year')['housing_units'].min()
print('Min')
print(sfo_housing_units_min)
sfo_housing_units_max = sfo_data.groupby('year')['housing_units'].max()
sfo_housing_units_std = sfo_data.groupby('year')['housing_units'].std()
print('STD')
print(sfo_housing_units_std)
```

```
Mean
year
2010    372560
2011    374507
2012    376454
2013    378401
2014    380348
2015    382295
2016    384242
Name: housing_units, dtype: int64
```

```
Min
year
2010    372560
2011    374507
2012    376454
2013    378401
2014    380348
2015    382295
2016    384242
Name: housing_units, dtype: int64
STD
year
2010    0.0
2011    0.0
2012    0.0
2013    0.0
2014    0.0
2015    0.0
2016    0.0
Name: housing_units, dtype: float64
```

[7]:
```
# Save the dataframe as a csv file
# YOUR CODE HERE!
sfo_housing_units_mean.to_csv(r'DATA/sfo_housing_units_mean.csv')
```

[8]:
```
# 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.
# YOUR CODE HERE!
mean_plot = sfo_housing_units_mean.hvplot.bar(ylim=(365000,387500)).
 ↪opts(yformatter="%.0f", title='SFO Housing Units Mean')
#sfo_housing_units_mean.hvplot.bar()
# Optional Challenge: Use the min, max, and std to scale the y limits of the␣
 ↪chart
# YOUR CODE HERE!
std_plot = sfo_housing_units_std.hvplot.bar(ylim=(365000,387500)).
 ↪opts(yformatter="%.0f", title='SFO Housing Units STD')
min_plot = sfo_housing_units_min.hvplot.bar(ylim=(365000,387500)).
 ↪opts(yformatter="%.0f", title='SFO Housing Units MIN')
max_plot = sfo_housing_units_max.hvplot.bar(ylim=(365000,387500)).
 ↪opts(yformatter="%.0f", title='SFO Housing Units MAX')

mean_plot + std_plot + min_plot + max_plot
```

[8]:
```
:Layout
   .Bars.Housing_units.I   :Bars   [year]   (housing_units)
   .Bars.Housing_units.II  :Bars   [year]   (housing_units)
```

```
.Bars.Housing_units.III :Bars   [year]   (housing_units)
.Bars.Housing_units.IV  :Bars   [year]   (housing_units)
```

data is same for each year for the Housing Units so mean, max, min are the same. and STD is 0

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.

[9]:
```
# Calculate the average sale price per square foot and average gross rent
# YOUR CODE HERE!
#print(sfo_data.head())
# First slice needed columns
sfo_housing_costs = sfo_data[['sale_price_sqr_foot', 'gross_rent']]
#print(sfo_housing_costs.head())    # print the sliced data
sfo_housing_costs_mean = sfo_housing_costs.
 ↪groupby('year')[['sale_price_sqr_foot', 'gross_rent']].mean()
print('Housing costs Mean')
print(sfo_housing_costs_mean)
```

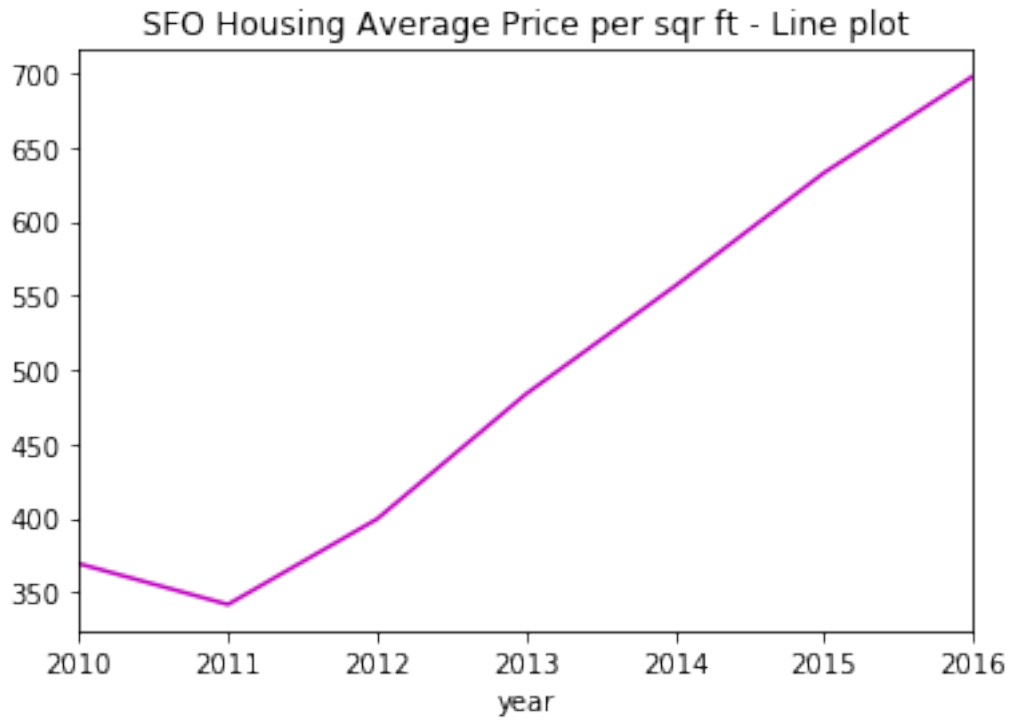
```
Housing costs Mean
      sale_price_sqr_foot  gross_rent
year
2010           369.344353        1239
2011           341.903429        1530
2012           399.389968        2324
2013           483.600304        2971
2014           556.277273        3528
2015           632.540352        3739
2016           697.643709        4390
```

[10]:
```
# Create two line charts, one to plot the average sale price per square foot␣
 ↪and another for average montly rent
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')
# Line chart for average sale price per square foot
# YOUR CODE HERE!
```

```
# Line chart for average montly rent
# YOUR CODE HERE!
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")
#sfo_housing_rent_mean_plot = sfo_housing_costs_mean['gross_rent'].plot.
 ↪line(title='SFO Housing Average gross rent by Year -  Line plot')
```

[11]:
```
print(sfo_housing_costs_mean)
sfo_housing_costs_mean_sale = sfo_housing_costs_mean['sale_price_sqr_foot']
#sfo_housing_costs_mean_sale.plot.line( title='SFO Housing Average Price per␣
 ↪sqr ft - Line plot', color = "m")
sfo_housing_sale_mean_plot = sfo_housing_costs_mean_sale.plot.line( title='SFO␣
 ↪Housing Average Price per sqr ft - Line plot', color = "m")
```

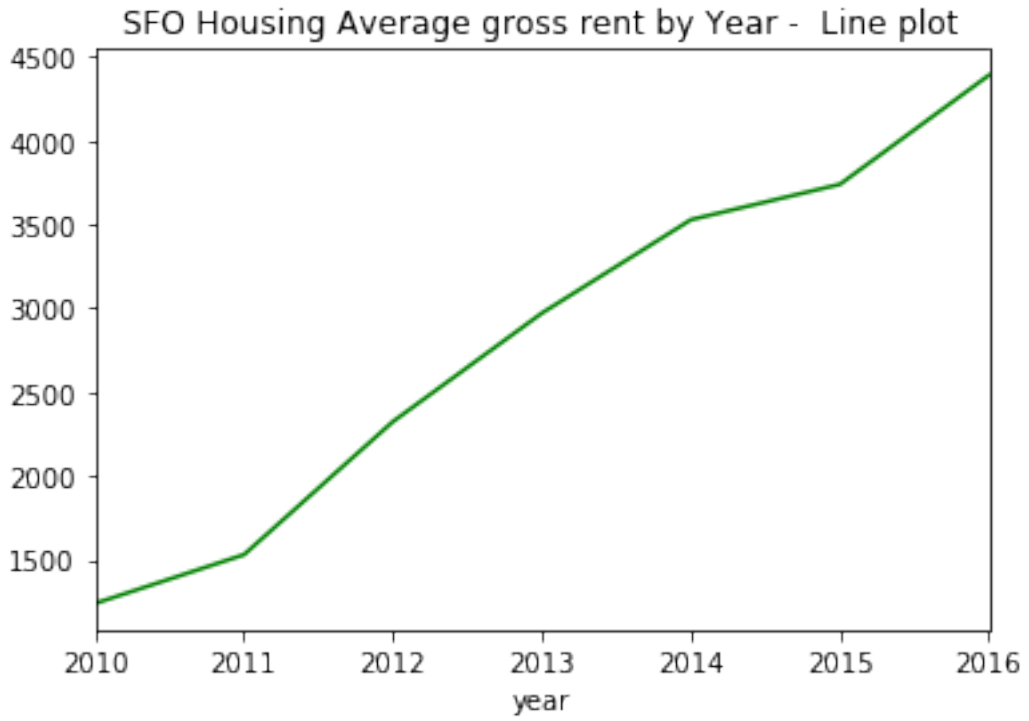
```
      sale_price_sqr_foot  gross_rent
year
2010           369.344353        1239
2011           341.903429        1530
2012           399.389968        2324
2013           483.600304        2971
2014           556.277273        3528
2015           632.540352        3739
2016           697.643709        4390
```



[12]:
```
sfo_housing_sale_mean_plot_hv # also printing hvplot
```

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

[13]:
```
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")
```



[14]:
```
sfo_housing_rent_mean_plot_hv # also printing hvplot
```

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

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

[15]:
```
# Group by year and neighborhood and then create a new dataframe of the mean␣
↪values
# YOUR CODE HERE!
#print(sfo_data.head())
sfo_nb_housing_costs = sfo_data[['neighborhood', 'sale_price_sqr_foot']]
#print(sfo_nb_housing_costs)
#sfo_nb_housing_costs = sfo_nb_housing_costs.groupby(['neighborhood'])
#sfo_nb_housing_costs.head()
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()
#yb = sfo_nb_housing_costs_mean.loc['Yerba Beuna':]
#yb.reset_index()


sfo_nb_housing_rents = sfo_data[['neighborhood', 'gross_rent']]
sfo_nb_housing_rents_mean = sfo_nb_housing_rents.groupby(['neighborhood',␣
↪'year']).mean()
sfo_nb_housing_rents_mean.reset_index()
```

[15]:

	neighborhood	year	gross_rent
0	Alamo Square	2010	1239
1	Alamo Square	2011	1530
2	Alamo Square	2012	2324
3	Alamo Square	2013	2971
4	Alamo Square	2014	3528
5	Alamo Square	2015	3739
6	Alamo Square	2016	4390
7	Anza Vista	2010	1239
8	Anza Vista	2012	2324
9	Anza Vista	2013	2971
10	Anza Vista	2014	3528
11	Anza Vista	2015	3739
12	Anza Vista	2016	4390
13	Bayview	2010	1239
14	Bayview	2011	1530
15	Bayview	2012	2324
16	Bayview	2013	2971
17	Bayview	2014	3528
18	Bayview Heights	2015	3739
19	Bernal Heights	2011	1530
20	Bernal Heights	2012	2324
21	Bernal Heights	2013	2971
22	Bernal Heights	2014	3528

```
23          Bernal Heights   2015        3739
24          Bernal Heights   2016        4390
25        Buena Vista Park   2010        1239
26        Buena Vista Park   2011        1530
27        Buena Vista Park   2012        2324
28        Buena Vista Park   2013        2971
29        Buena Vista Park   2015        3739
..                     …   …         …
367  Van Ness/ Civic Center  2011        1530
368  Van Ness/ Civic Center  2012        2324
369  Van Ness/ Civic Center  2013        2971
370  Van Ness/ Civic Center  2014        3528
371  Van Ness/ Civic Center  2015        3739
372  Van Ness/ Civic Center  2016        4390
373      Visitacion Valley   2013        2971
374      Visitacion Valley   2014        3528
375      Visitacion Valley   2015        3739
376      Visitacion Valley   2016        4390
377            West Portal   2010        1239
378            West Portal   2012        2324
379            West Portal   2013        2971
380            West Portal   2014        3528
381       Western Addition   2010        1239
382       Western Addition   2011        1530
383       Western Addition   2012        2324
384       Western Addition   2013        2971
385       Western Addition   2014        3528
386       Western Addition   2015        3739
387     Westwood Highlands   2011        1530
388     Westwood Highlands   2013        2971
389          Westwood Park   2014        3528
390          Westwood Park   2016        4390
391            Yerba Buena   2010        1239
392            Yerba Buena   2011        1530
393            Yerba Buena   2012        2324
394            Yerba Buena   2013        2971
395            Yerba Buena   2014        3528
396            Yerba Buena   2015        3739

[397 rows x 3 columns]
```

[16]:
```
#yb.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_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")
```

[17]:
```
# 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
# YOUR CODE HERE!
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

#sfo_nb_housing_sale_mean_plot_hv = sfo_nb_housing_costs_mean.hvplot.line().
 ↪opts( title='SFO Housing Average Price per sqr ft per Neighborhood- HV Line␣
 ↪plot', color = "m")
#sfo_nb_housing_sale_mean_plot_hv
#interact(nb_sale, neighborhood=sfo_data['neighborhood'])
layout_sale = interact(nb_sale, neighborhood=sfo_data['neighborhood'])
pn.Column('**SFO Housing**', pn.Row(layout_sale[1], layout_sale[0]))
#x="year",  y="sale_price_sqr_foot",c="neighborhood"
```

[17]:
```
Column
    [0] Markdown(str)
    [1] Row
        [0] Row
            [0] HoloViews(Curve, name='interactive02810')
        [1] Column
            [0] Select(name='neighborhood', options=['Alamo Square', …],
value='Alamo Square')
```

[ ]:
```
```

[18]:
```
# Use hvplot to create an interactive line chart of the average monthly rent.
# The plot should have a dropdown selector for the neighborhood
# YOUR CODE HERE!
def nb_rent(neighborhood):
#    sfo_nb_housing_costs_mean = sfo_data.
 ↪groupby(['neighborhood'])['sale_price_sqr_foot'].mean()
    sloc = sfo_nb_housing_rents_mean.loc[neighborhood]
```

```
    sfo_nb_housing_rents_mean_plot_hv = sloc.hvplot.line(x='year',␣
↪y='gross_rent').opts( title='SFO NB Housing Average Rent per sqr ft - HV␣
↪Line plot', color = "m")

    return sfo_nb_housing_rents_mean_plot_hv

layout_rent = interact(nb_rent, neighborhood=sfo_data['neighborhood'])
pn.Column('**SFO Housing**', pn.Row(layout_rent[1], layout_rent[0]))
```

[18]:
```
Column
    [0] Markdown(str)
    [1] Row
        [0] Row
            [0] HoloViews(Curve, name='interactive03039')
        [1] Column
            [0] Select(name='neighborhood', options=['Alamo Square', …],
value='Alamo Square')
```

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.

[19]:
```
# Getting the data from the top 10 expensive neighborhoods to own
# 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']
```

[20]:
```
# Plotting the data from the top 10 expensive neighborhoods
# YOUR CODE HERE!
sfo_data_exp_sale.hvplot.bar(rot=90, height=400, title='Top 10 most expensive␣
↪neighborhoods')
```

[20]:
```
: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.

[21]:
```
# Fetch the previously generated DataFrame that was grouped by year and␣
↪neighborhood
# YOUR CODE HERE!
sfo_data_mean = sfo_data.groupby(['neighborhood', 'year']).mean()
sfo_data_mean = sfo_data_mean.reset_index()
sfo_data_exp10_reset = sfo_data_exp10.reset_index()
#print(sfo_data_exp10_reset.head())
```

[22]:
```
#top10_compare.reset_index
sfo_data_mean = sfo_data_mean[sfo_data_mean["neighborhood"].
↪isin(sfo_data_exp10_reset["neighborhood"])]
```

[23]:
```
# Plotting the data from the top 10 expensive neighborhoods
# YOUR CODE HERE!
print(sfo_data_mean.head())

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

#sfo_data_mean.hvplot(x='year', y=['sale_price_sqr_foot', 'gross_rent'],␣
↪value_label ='housing_units', groupby('neighborhood'))
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')
#top10_compare.hvplot.bar(x='year', y=['sale_price_sqr_foot', 'gross_rent'],␣
↪groupby='neighborhood')



#value_label ='housing_units'
```

```
   neighborhood  year  sale_price_sqr_foot  housing_units  gross_rent
52   Cow Hollow  2010           569.379968         372560        1239
53   Cow Hollow  2011           390.595653         374507        1530
54   Cow Hollow  2012           644.818307         376454        2324
55   Cow Hollow  2013           707.402809         378401        2971
56   Cow Hollow  2014           691.865411         380348        3528
```

[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
# YOUR CODE HERE!
# Read the census data into a Pandas DataFrame
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))
```

```
                      Lat         Lon
neighborhood
Alamo Square     37.791012 -122.402100
Anza Vista       37.779598 -122.443451
Bayview          37.734670 -122.401060
Bayview Heights  37.728740 -122.410980
Bernal Heights   37.728630 -122.443050
len =  73
```

[25]:
```
nb_coord_data.head()
```

[25]:
```
                      Lat         Lon
neighborhood
Alamo Square     37.791012 -122.402100
Anza Vista       37.779598 -122.443451
Bayview          37.734670 -122.401060
Bayview Heights  37.728740 -122.410980
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.

[26]:
```
# Calculate the mean values for each neighborhood
# YOUR CODE HERE!
print(sfo_data_exp.head())
```

```
                 sale_price_sqr_foot  housing_units   gross_rent
neighborhood
Alamo Square              366.020712       378401.0  2817.285714
```

```
Anza Vista                373.382198       379050.0  3031.833333
Bayview                   204.588623       376454.0  2318.400000
Bayview Heights           590.792839       382295.0  3739.000000
Bernal Heights            576.746488       379374.5  3080.333333
```

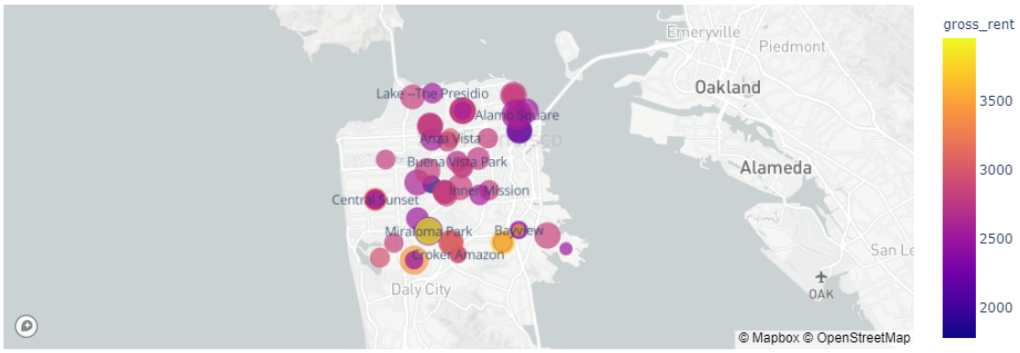
[27]:
```
# Join the average values with the neighborhood locations
# YOUR CODE HERE!
combined_df = pd.concat([sfo_data_exp, nb_coord_data], axis="columns",␣
 ↪join="inner")
combined_df.head()
combined_df.reset_index(inplace=True)
```

1.7.3 Mapbox Visualization

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

[28]:
```
# Set the mapbox access token
# YOUR CODE HERE!
px.set_mapbox_access_token(map_box_api)

# Create a scatter mapbox to analyze neighborhood info
# YOUR CODE HERE!
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
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

[29]:
```
# Fetch the data from all expensive neighborhoods per year.
#df_expensive_neighborhoods_per_year = df_costs[df_costs["neighborhood"].
 ↪isin(df_expensive_neighborhoods["neighborhood"])]
#df_expensive_neighborhoods_per_year.head()
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)
```

```
            neighborhood  sale_price_sqr_foot  housing_units   gross_rent
0  Union Square District           903.993258      377427.50  2555.166667
1         Merced Heights           788.844818      380348.00  3414.000000
2          Miraloma Park           779.810842      375967.25  2155.250000
3       Pacific Heights           689.555817      378401.00  2817.285714
4          Westwood Park           687.087575      382295.00  3959.000000
```

```
          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
    year           neighborhood  sale_price_sqr_foot  housing_units  \
0   2010             Cow Hollow           569.379968         372560
1   2010          Miraloma Park           680.608729         372560
2   2010        Pacific Heights           496.516014         372560
3   2010           Potrero Hill           491.450004         372560
4   2010       Presidio Heights           549.417931         372560
5   2010            South Beach          1037.099789         372560
6   2010         Telegraph Hill           524.793509         372560
7   2010  Union Square District           569.193448         372560
8   2011             Cow Hollow           390.595653         374507
9   2011          Miraloma Park           414.676065         374507
10  2011        Pacific Heights           509.021480         374507
11  2011           Potrero Hill           503.138505         374507
12  2011       Presidio Heights           493.814917         374507
13  2011            South Beach           388.644337         374507
14  2011         Telegraph Hill           483.405773         374507
15  2011  Union Square District           302.123253         374507
16  2012             Cow Hollow           644.818307         376454
17  2012         Merced Heights           421.141245         376454
18  2012          Miraloma Park           756.192373         376454
19  2012        Pacific Heights           586.218215         376454
20  2012           Potrero Hill           466.647840         376454
21  2012       Presidio Heights           426.608592         376454
22  2012         Telegraph Hill           465.802122         376454
23  2012  Union Square District           445.196788         376454
24  2013             Cow Hollow           707.402809         378401
25  2013        Pacific Heights           503.899261         378401
26  2013           Potrero Hill           598.704795         378401
27  2013       Presidio Heights           512.840248         378401
28  2013         Telegraph Hill           653.311617         378401
29  2013  Union Square District          1290.472107         378401
30  2014             Cow Hollow           691.865411         380348
31  2014         Merced Heights           528.726541         380348
32  2014          Miraloma Park          1267.766203         380348
33  2014        Pacific Heights           824.658694         380348
34  2014           Potrero Hill          1045.201546         380348
35  2014       Presidio Heights           603.450997         380348
36  2014            South Beach           524.629312         380348
37  2014         Telegraph Hill           708.193032         380348
38  2014  Union Square District           558.271119         380348
39  2014          Westwood Park           742.979723         380348
```

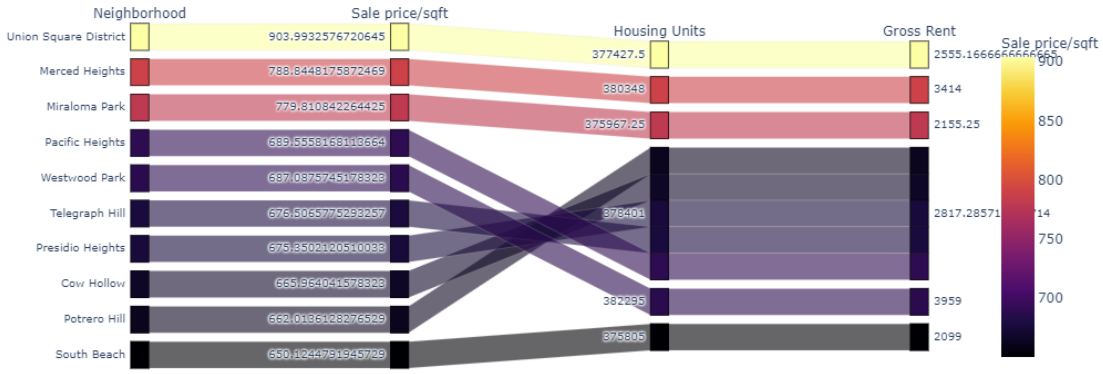
40	2015	Cow Hollow	598.620541	382295
41	2015	Pacific Heights	910.187690	382295
42	2015	Potrero Hill	821.300990	382295
43	2015	Presidio Heights	NaN	382295
44	2015	Telegraph Hill	996.990220	382295
45	2015	Union Square District	2258.702832	382295
46	2016	Cow Hollow	1059.065602	384242
47	2016	Merced Heights	1416.666667	384242
48	2016	Pacific Heights	996.389364	384242
49	2016	Potrero Hill	707.651609	384242
50	2016	Presidio Heights	1465.968586	384242
51	2016	Telegraph Hill	903.049771	384242
52	2016	Westwood Park	631.195426	384242

	gross_rent
0	1239
1	1239
2	1239
3	1239
4	1239
5	1239
6	1239
7	1239
8	1530
9	1530
10	1530
11	1530
12	1530
13	1530
14	1530
15	1530
16	2324
17	2324
18	2324
19	2324
20	2324
21	2324
22	2324
23	2324
24	2971
25	2971
26	2971
27	2971
28	2971
29	2971
30	3528
31	3528
32	3528

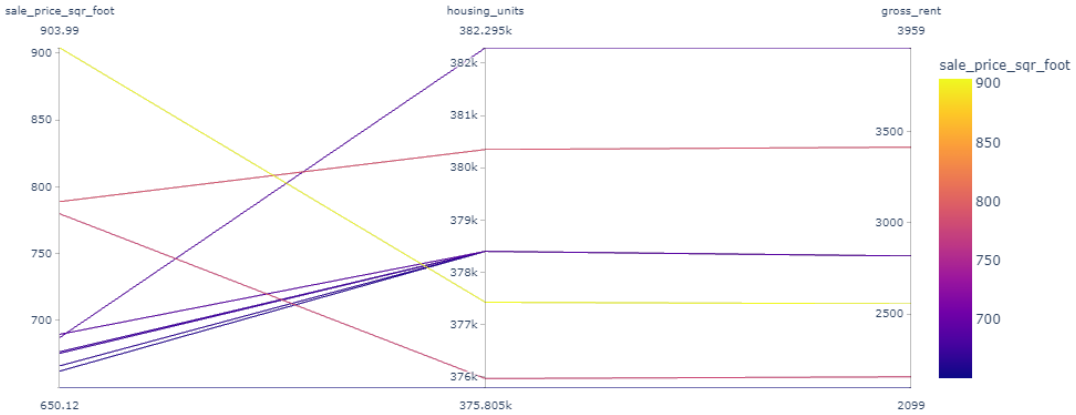
```
33        3528
34        3528
35        3528
36        3528
37        3528
38        3528
39        3528
40        3739
41        3739
42        3739
43        3739
44        3739
45        3739
46        4390
47        4390
48        4390
49        4390
50        4390
51        4390
52        4390
```

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
# YOUR CODE HERE!
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",
    },
)
```



[31]:
```
# Parallel Coordinates Plot
# YOUR CODE HERE!
px.parallel_coordinates(sfo_data_exp10_reset, color='sale_price_sqr_foot')
```

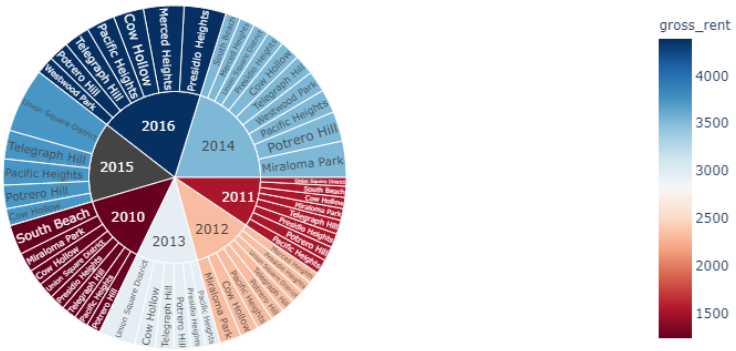


1.8.3 Create a sunburst chart to conduct a costs analysis of most expensive neighborhoods in San Francisco per year

[32]:
```
# 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']))
fig.show()
```



[ ]: