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.

In [2]:

```
# imports
import panel as pn
pn.extension('plotly')
import plotly.express as px
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')
```

```
Bad key savefig.frameon in file /Users/dallasgold/opt/anaconda3/envs/pyvizenv/
lib/python3.7/site-packages/matplotlib/mpl-data/stylelib/_classic_test.mplstyl
e, line 421 ('savefig.frameon : True')
You probably need to get an updated matplotlibrc file from
https://github.com/matplotlib/matplotlib/blob/v3.3.4/matplotlibrc.template
or from the matplotlib source distribution

Bad key verbose.level in file /Users/dallasgold/opt/anaconda3/envs/pyvizenv/li
b/python3.7/site-packages/matplotlib/mpl-data/stylelib/_classic_test.mplstyle,
line 472 ('verbose.level  : silent      # one of silent, helpful, debug, debug
-annoying')
You probably need to get an updated matplotlibrc file from
https://github.com/matplotlib/matplotlib/blob/v3.3.4/matplotlibrc.template
or from the matplotlib source distribution

Bad key verbose.fileo in file /Users/dallasgold/opt/anaconda3/envs/pyvizenv/li
b/python3.7/site-packages/matplotlib/mpl-data/stylelib/_classic_test.mplstyle,
line 473 ('verbose.fileo  : sys.stdout  # a log filename, sys.stdout or sys.st
derr')
You probably need to get an updated matplotlibrc file from
https://github.com/matplotlib/matplotlib/blob/v3.3.4/matplotlibrc.template
or from the matplotlib source distribution
In /Users/dallasgold/opt/anaconda3/envs/pyvizenv/lib/python3.7/site-packages/m
atplotlib/mpl-data/stylelib/_classic_test.mplstyle:
The text.latex.preview rcparam was deprecated in Matplotlib 3.3 and will be re
moved two minor releases later.
In /Users/dallasgold/opt/anaconda3/envs/pyvizenv/lib/python3.7/site-packages/m
atplotlib/mpl-data/stylelib/_classic_test.mplstyle:
The mathtext.fallback_to_cm rcparam was deprecated in Matplotlib 3.3 and will
be removed two minor releases later.
In /Users/dallasgold/opt/anaconda3/envs/pyvizenv/lib/python3.7/site-packages/m
atplotlib/mpl-data/stylelib/_classic_test.mplstyle: Support for setting the 'm
athtext.fallback_to_cm' rcParam is deprecated since 3.3 and will be removed tw
o minor releases later; use 'mathtext.fallback : 'cm' instead.
In /Users/dallasgold/opt/anaconda3/envs/pyvizenv/lib/python3.7/site-packages/m
atplotlib/mpl-data/stylelib/_classic_test.mplstyle:
The validate_bool_maybe_none function was deprecated in Matplotlib 3.3 and wil
l be removed two minor releases later.
In /Users/dallasgold/opt/anaconda3/envs/pyvizenv/lib/python3.7/site-packages/m
atplotlib/mpl-data/stylelib/_classic_test.mplstyle:
The savefig.jpeg_quality rcparam was deprecated in Matplotlib 3.3 and will be
removed two minor releases later.
In /Users/dallasgold/opt/anaconda3/envs/pyvizenv/lib/python3.7/site-packages/m
atplotlib/mpl-data/stylelib/_classic_test.mplstyle:
The keymap.all_axes rcparam was deprecated in Matplotlib 3.3 and will be remov
ed two minor releases later.
In /Users/dallasgold/opt/anaconda3/envs/pyvizenv/lib/python3.7/site-packages/m
atplotlib/mpl-data/stylelib/_classic_test.mplstyle:
The animation.avconv_path rcparam was deprecated in Matplotlib 3.3 and will be
removed two minor releases later.
In /Users/dallasgold/opt/anaconda3/envs/pyvizenv/lib/python3.7/site-packages/m
atplotlib/mpl-data/stylelib/_classic_test.mplstyle:
The animation.avconv_args rcparam was deprecated in Matplotlib 3.3 and will be
removed two minor releases later.
```

In [28]:

```
# Read the Mapbox API key
load_dotenv()
map_box_api = os.getenv("mapbox")
px.set_mapbox_access_token('map_box_api')
```

Load Data

In [4]:

```
# 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")
sfo_data.head()
```

Out[4]:

	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

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.

In [5]:

```
houses = sfo_data['housing_units']
# type(houses)
units_per_yr_df = houses.groupby(by = 'year').mean()

units_per_yr_df
```

```
Out[5]: year
        2010    372560
        2011    374507
        2012    376454
        2013    378401
        2014    380348
        2015    382295
        2016    384242
        Name: housing_units, dtype: int64
```

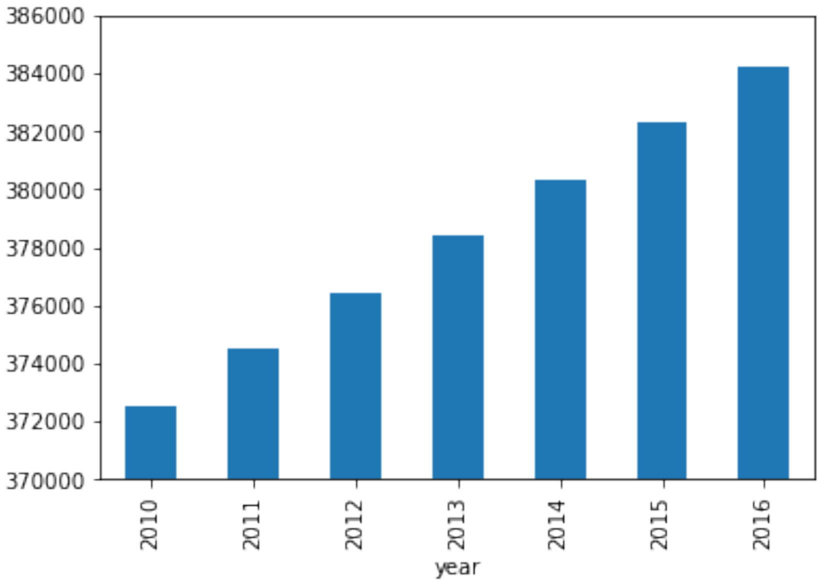
```
In [6]:  # Calculate the mean number of housing units per year (hint: use groupby)
         # YOUR CODE HERE!
```

```
In [7]:  # Save the dataframe as a csv file
         units_per_yr_df.to_csv()
```

```
Out[7]: 'year,housing_units\n2010,372560\n2011,374507\n2012,376454\n2013,378401\n2014,
        380348\n2015,382295\n2016,384242\n'
```

```
In [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 m
         unit_graph = units_per_yr_df.plot(kind='bar')

         # Optional Challenge: Use the min, max, and std to scale the y limits of the
         unit_graph.set_ylim(370000,386000)
         unit_graph.set_xlabel = 'Year'
         unit_graph.set_ylabel = 'Avg Housing Availability'
```



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.

In [9]:

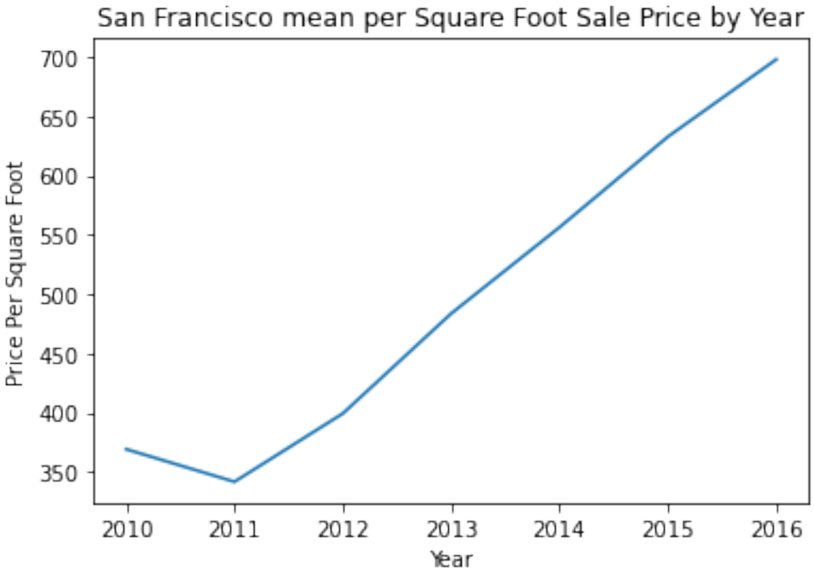
```
# Calculate the average sale price per square foot and average gross rent
rent_sq_ft = sfo_data[['sale_price_sqr_foot','gross_rent']]
sale_df = rent_sq_ft.groupby(by = 'year').mean()
sale_df

# gross rent
axis = sale_df['sale_price_sqr_foot'].plot(title = 'San Francisco mean per Sq

# axis labels
axis.set_xlabel('Year')
axis.set_ylabel('Price Per Square Foot')
```

Out[9]:

```
Text(0, 0.5, 'Price Per Square Foot')
```



In [10]:

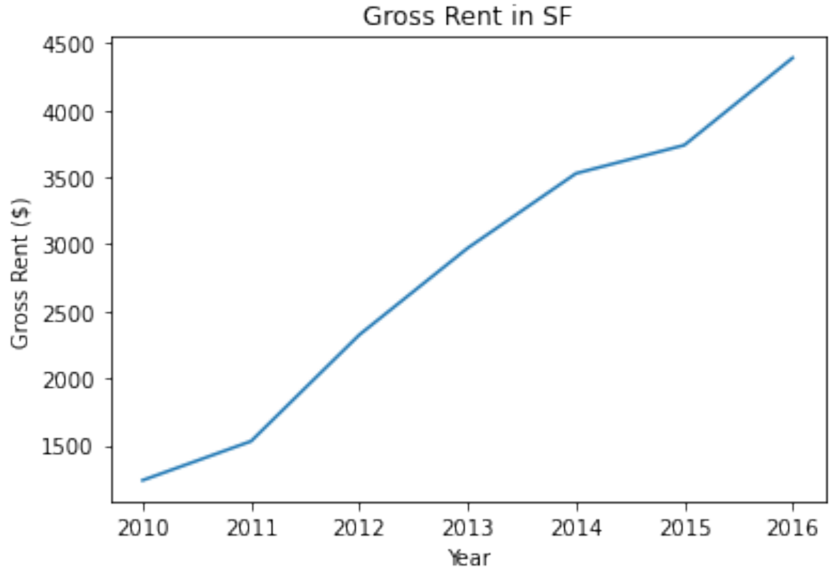
```
# Create two line charts, one to plot the average sale price per square foot

# Line chart for average sale price per square foot
gross_rent = sale_df['gross_rent'].plot(title = 'Gross Rent in SF')

# Line chart for average montly rent
gross_rent.set_xlabel('Year')
gross_rent.set_ylabel('Gross Rent ($)')
```

Out[10]:

```
Text(0, 0.5, 'Gross Rent ($)')
```



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

In [11]:

```
# Group by year and neighborhood and then create a new dataframe of the mean
hood_df = sfo_data.groupby(['year','neighborhood']).mean().reset_index()
hood_df.head(10)
```

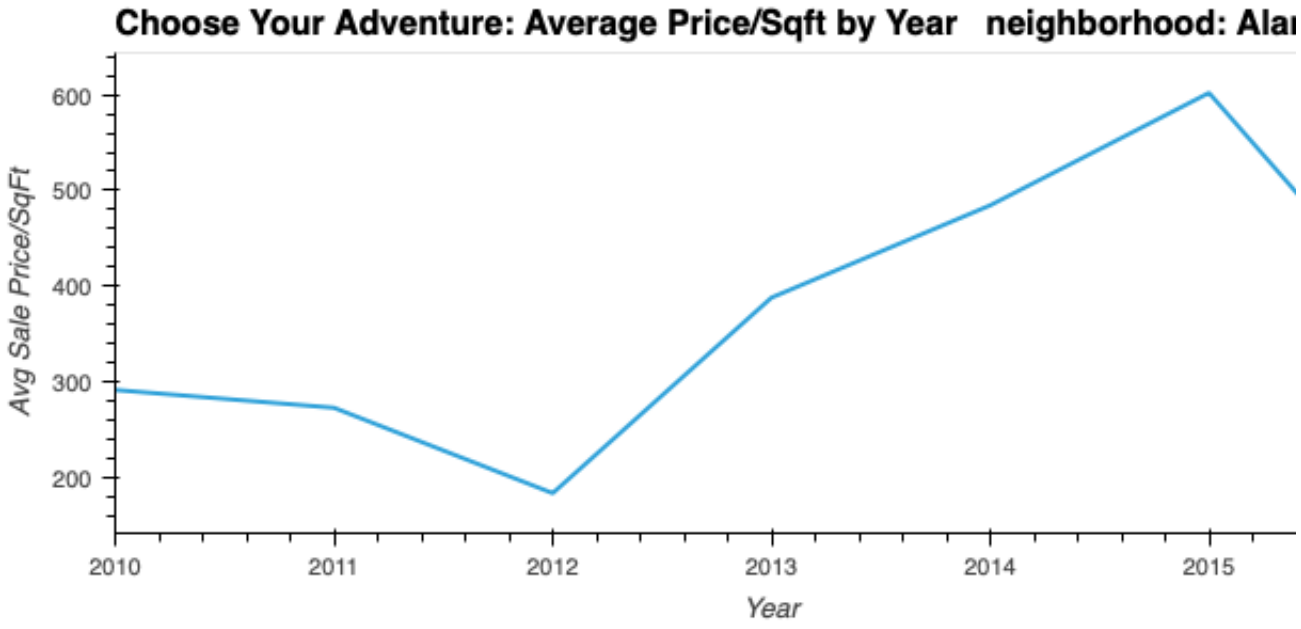
Out[11]:

	year	neighborhood	sale_price_sqr_foot	housing_units	gross_rent
0	2010	Alamo Square	291.182945	372560	1239
1	2010	Anza Vista	267.932583	372560	1239
2	2010	Bayview	170.098665	372560	1239
3	2010	Buena Vista Park	347.394919	372560	1239
4	2010	Central Richmond	319.027623	372560	1239
5	2010	Central Sunset	418.172493	372560	1239
6	2010	Corona Heights	369.359338	372560	1239
7	2010	Cow Hollow	569.379968	372560	1239
8	2010	Croker Amazon	165.645730	372560	1239
9	2010	Diamond Heights	456.930822	372560	1239

In [12]:

```
# Use hvplot to create an interactive line chart of the average price per sq
# The plot should have a dropdown selector for the neighborhood
hood_df.hvplot(label='Choose Your Adventure: Average Price/Sqft by Year ',
                      x='year',
                      y='sale_price_sqr_foot',
                      xlabel='Year',
                      ylabel='Avg Sale Price/SqFt',
                      groupby='neighborhood')
```

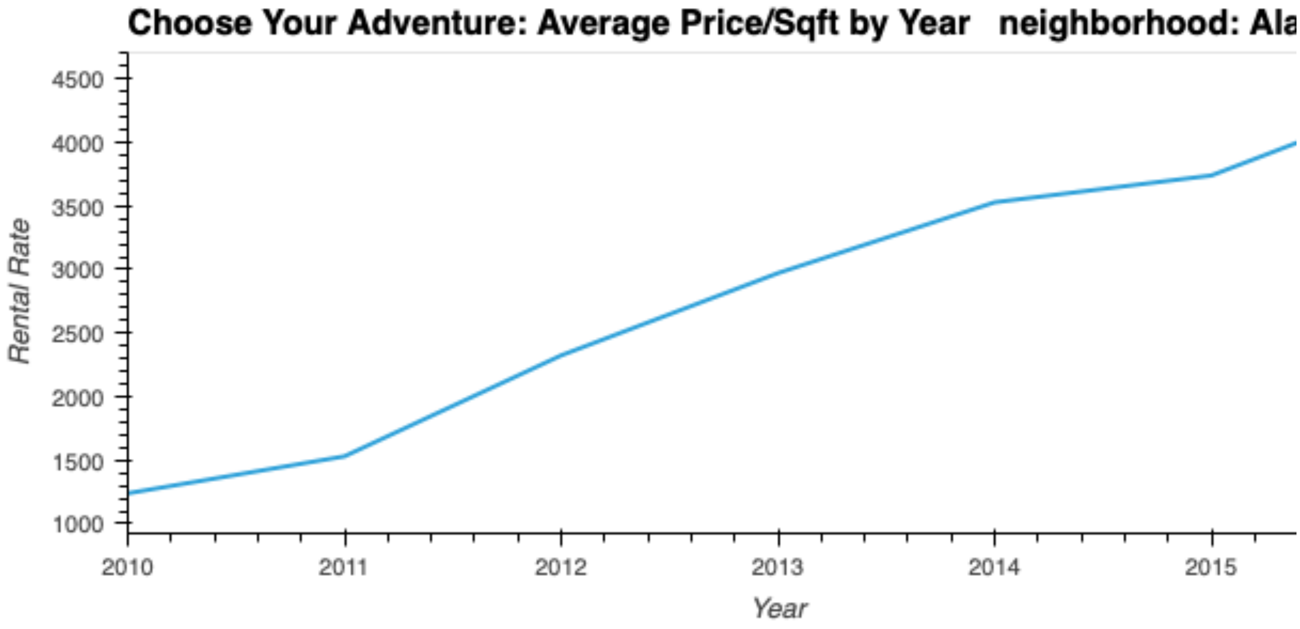
Out[12]:



In [13]:

```
# Use hvplot to create an interactive line chart of the average monthly rent.
# The plot should have a dropdown selector for the neighborhood
hood_df.hvplot(label='Choose Your Adventure: Average Price/Sqft by Year ',
                       x='year',
                       y='gross_rent',
                       xlabel='Year',
                       ylabel='Rental Rate',
                       groupby='neighborhood')
```

Out[13]:



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.

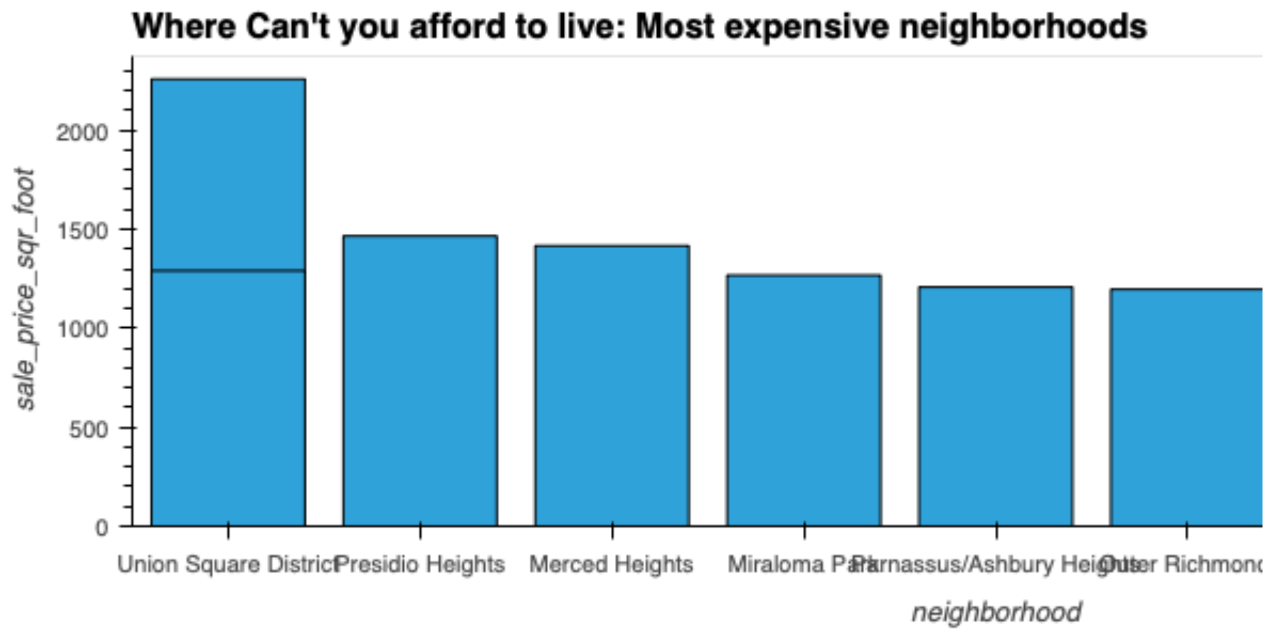
In [14]:

```
# Getting the data from the top 10 expensive neighborhoods to own
most_expensive = hood_df.sort_values(by='sale_price_sqr_foot', ascending=Fals
most_expensive_nbhd = most_expensive[['neighborhood','sale_price_sqr_foot']]
```

In [15]:

```
# Plotting the data from the top 10 expensive neighborhoods
most_expensive_nbhd.hvplot.bar(label='Where Can\'t you afford to live: Most e
                       x='neighborhood',
                       y='sale_price_sqr_foot', width=900
                       )
```

Out[15]:



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.

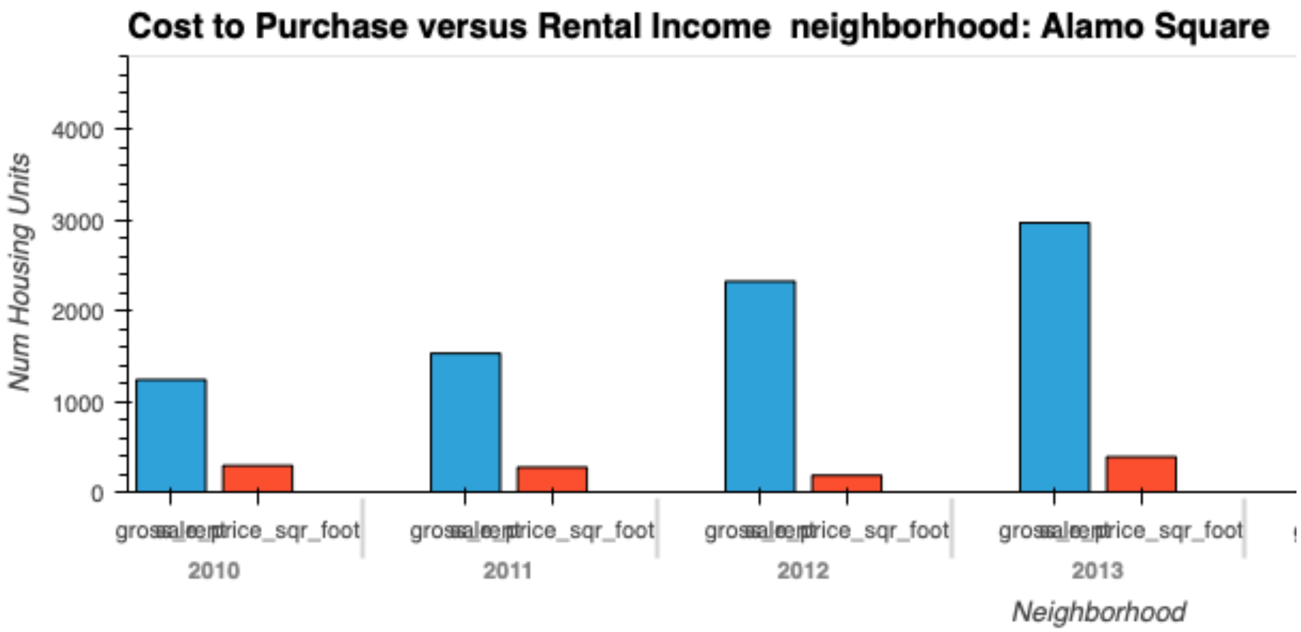
In [26]:

```
# Fetch the previously generated DataFrame that was grouped by year and neigh
most_expensive_both =most_expensive[['year','neighborhood','sale_price_sqr_fo
```

In [27]:

```
# Plotting the data from the top 10 expensive neighborhoods
hood_df.hvplot.bar(label='Cost to Purchase versus Rental Income',
                       x='year',
                       y=['gross_rent','sale_price_sqr_foot'],
                       xlabel='Neighborhood',
                       ylabel='Num Housing Units',
                       groupby='neighborhood',
                       width=1000
                       )
```

Out[27]:



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.

Load Location Data

In [17]:

```
location_data = Path("Data/neighborhoods_coordinates.csv")
location_data_df = pd.read_csv(location_data)
location_data_df.head(5)
```

Out[17]:

	Neighborhood	Lat	Lon
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

In [16]:

```
# Load neighborhoods coordinates data
```

Out[16]:

	Neighborhood	Lat	Lon
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

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.

In [21]:

```
neighborhood_avg = sfo_data.groupby('neighborhood',as_index=False).mean()
neighborhood_avg.head(5)
```

Out[21]:

	neighborhood	sale_price_sqr_foot	housing_units	gross_rent
0	Alamo Square	366.020712	378401.0	2817.285714
1	Anza Vista	373.382198	379050.0	3031.833333
2	Bayview	204.588623	376454.0	2318.400000
3	Bayview Heights	590.792839	382295.0	3739.000000
4	Bernal Heights	576.746488	379374.5	3080.333333

In [17]:

```
# Calculate the mean values for each neighborhood
# For Reference
```

Out[17]:

	Neighborhood	sale_price_sqr_foot	housing_units	gross_rent
0	Alamo Square	366.020712	378401.0	2817.285714
1	Anza Vista	373.382198	379050.0	3031.833333
2	Bayview	204.588623	376454.0	2318.400000
3	Bayview Heights	590.792839	382295.0	3739.000000
4	Bernal Heights	576.746488	379374.5	3080.333333

In [23]:

```
neighborhood_combined = pd.concat(
    [location_data_df, neighborhood_avg],
    axis="columns",
    join="inner"
).drop(columns='neighborhood')

neighborhood_combined.head()
```

Out[23]:

	Neighborhood	Lat	Lon	sale_price_sqr_foot	housing_units	gross_rent
0	Alamo Square	37.791012	-122.402100	366.020712	378401.0	2817.285714
1	Anza Vista	37.779598	-122.443451	373.382198	379050.0	3031.833333
2	Bayview	37.734670	-122.401060	204.588623	376454.0	2318.400000
3	Bayview Heights	37.728740	-122.410980	590.792839	382295.0	3739.000000
4	Bernal Heights	37.728630	-122.443050	576.746488	379374.5	3080.333333

In [18]:

```
# Join the average values with the neighborhood locations
# For Reference
```

Out[18]:

	Neighborhood	Lat	Lon	sale_price_sqr_foot	housing_units	gross_rent
0	Alamo Square	37.791012	-122.402100	366.020712	378401.0	2817.285714
1	Anza Vista	37.779598	-122.443451	373.382198	379050.0	3031.833333
2	Bayview	37.734670	-122.401060	204.588623	376454.0	2318.400000
3	Bayview Heights	37.728740	-122.410980	590.792839	382295.0	3739.000000
4	Buena Vista Park	37.768160	-122.439330	452.680591	378076.5	2698.833333

Mapbox Visualization

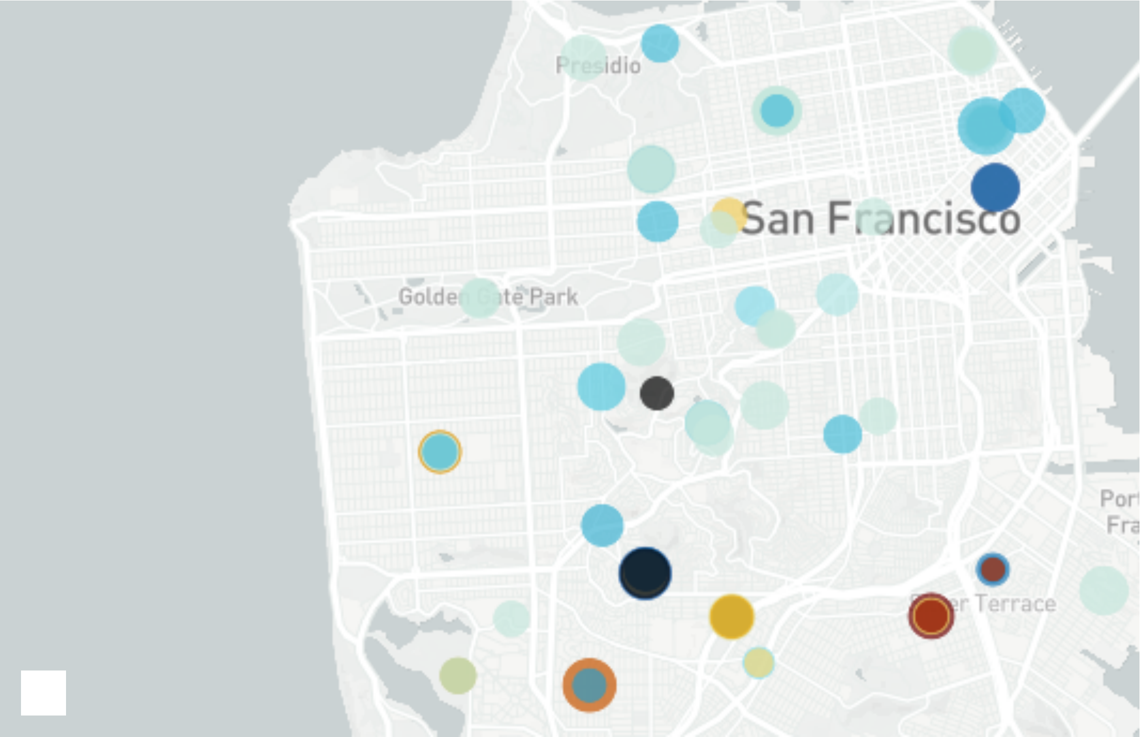
Plot the average values per neighborhood using a Plotly express `scatter_mapbox` visualization.

In [37]:

```
map_title='Avg Sale Value/SqFt and Rental Value in San Francisco Neighborhood
px.set_mapbox_access_token(')

px.scatter_mapbox(
    neighborhood_combined,
    lat='Lat',
    lon='Lon',
    size='sale_price_sqr_foot',
    color='gross_rent',
    color_continuous_scale=px.colors.cyclical.IceFire,
    zoom=11,
    title= map_title,
    width=900
)
```

Avg Sale Value/SqFt and Rental Value in San Francisco Neighborhc



In [36]:

```
# # Set the mapbox access token
# map_box =
# # Create a scatter mapbox to analyze neighborhood info
# # Keeping this for reference
# map_title='Average Values per neighborhood in San Francisco'
# px.scatter_mapbox(
#     neighborhood_stats_coord,
#     lat='Lat',
#     lon='Lon',
#     size='sale_price_sqr_foot',
#     color='gross_rent',
#     color_continuous_scale=px.colors.cyclical.IceFire,
#     zoom=10,
#     title=map_title,
#     width=900
# )
```

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.

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

In [20]:

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

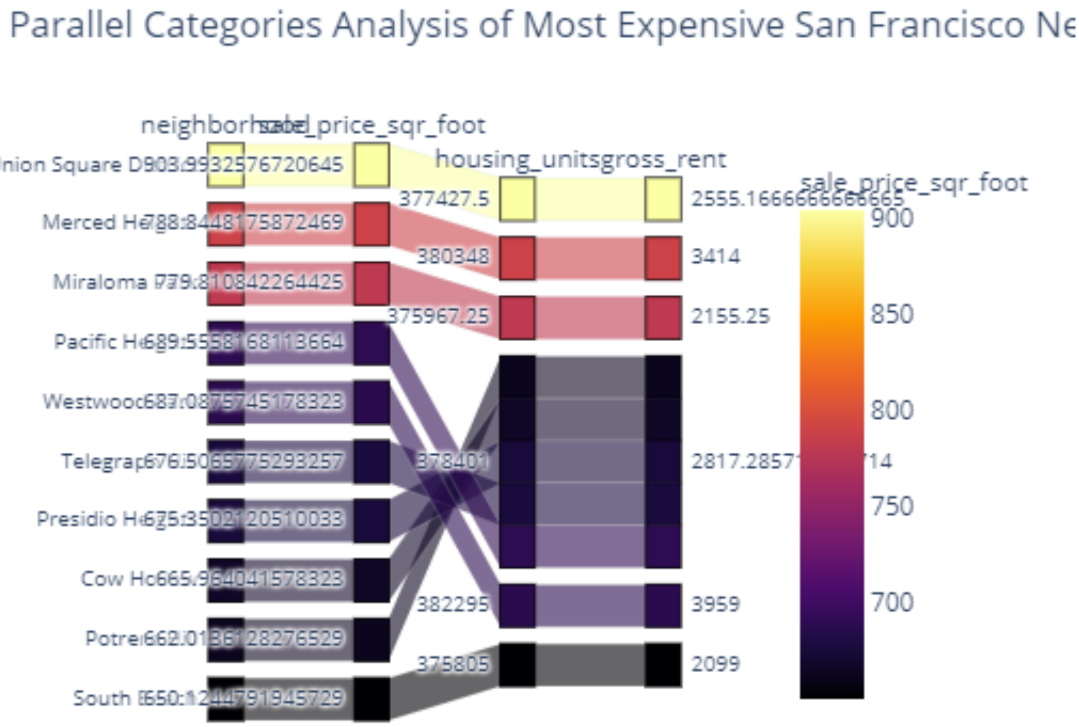
Out[20]:

	year	neighborhood	sale_price_sqr_foot	housing_units	gross_rent
7	2010	Cow Hollow	569.379968	372560	1239
31	2010	Miraloma Park	680.608729	372560	1239
41	2010	Pacific Heights	496.516014	372560	1239
46	2010	Potrero Hill	491.450004	372560	1239
47	2010	Presidio Heights	549.417931	372560	1239

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

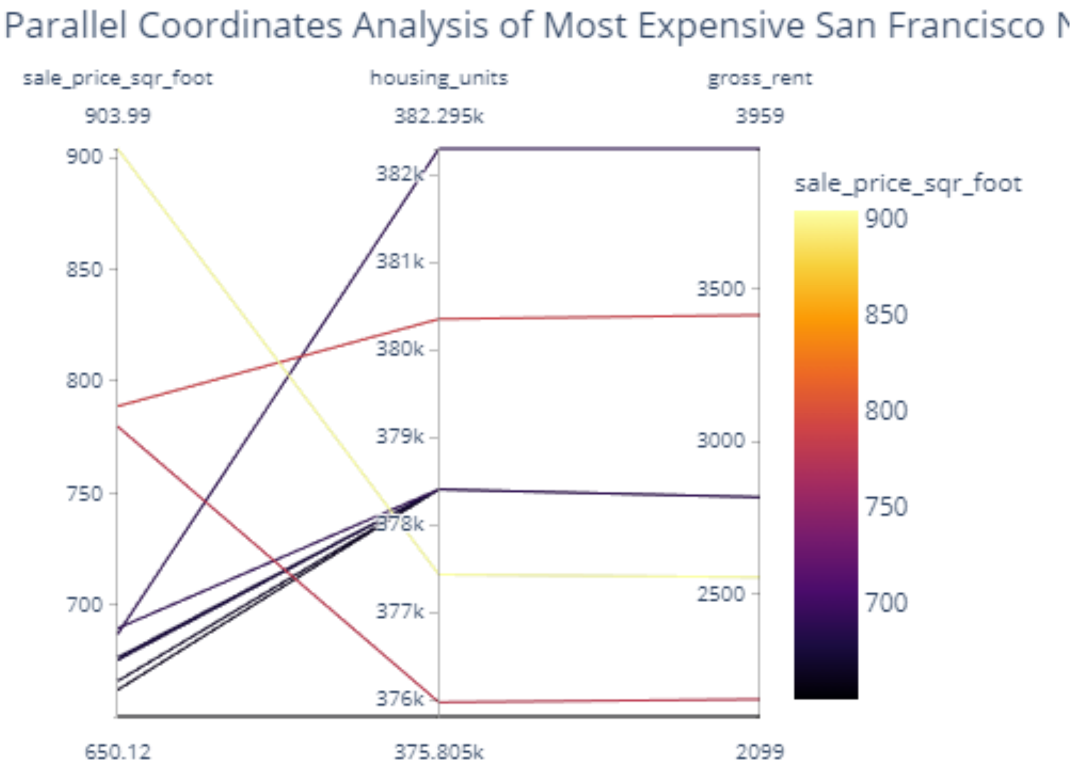
In [21]:

```
# Parallel Categories Plot
# YOUR CODE HERE!
```



In [22]:

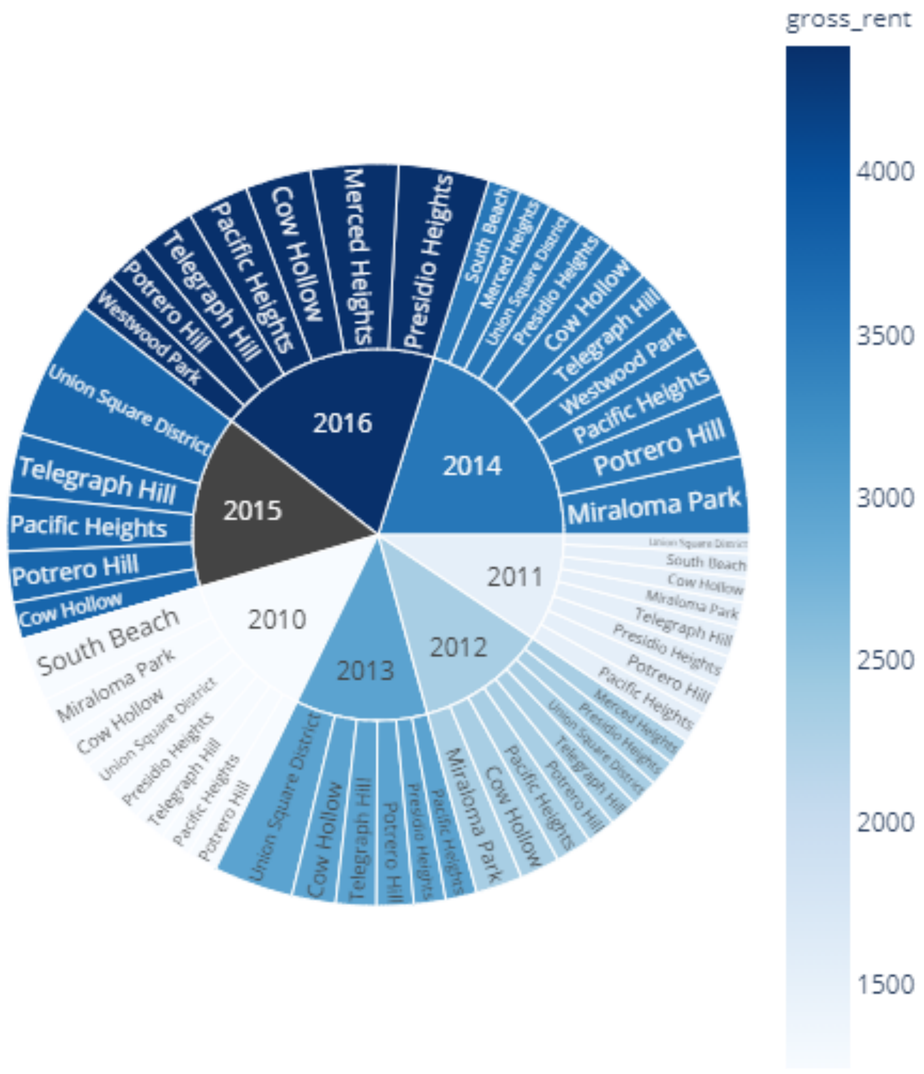
```
# Parallel Coordinates Plot
# YOUR CODE HERE!
```



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

In [23]:

```
# Sunburst Plot
# YOUR CODE HERE!
```



Costs Analysis of Most Expensive neighborhoods in San Francis

```
In [ ]:
```