San Francisco Housing Rental Analysis

In this assignment, you will perform basic analysis for the San Francisco Housing Market to allow potential real estate investors to choose rental investment properties.

In [13]:
```
# initial imports
import os
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import hvplot.pandas
from pathlib import Path
from dotenv import load_dotenv

%matplotlib inline
```

In [14]:
```
# Read the Mapbox API key
load_dotenv()
mapbox_token = os.getenv("MAPBOX_API_KEY")
```

Load Data

In [15]:
```
# 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[15]:

year	neighborhood	sale_price_sqr_foot	housing_units	gross_rent
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

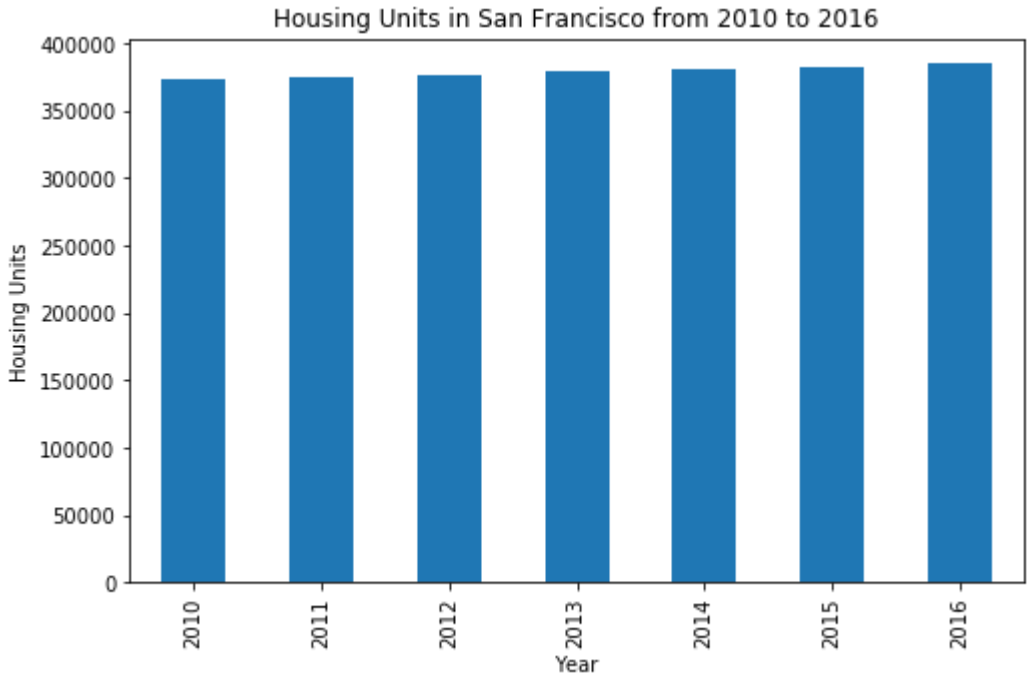
Optional challenge: Use the min, max, and std to scale the y limits of the chart.

In [16]:
```
# Calculate the mean number of housing units per year (hint: use groupby)
housing_units = sfo_data.groupby("year").mean()["housing_units"].sort_values()
housing_units
```

Out[16]:
```
year
2010    372560
2011    374507
2012    376454
2013    378401
2014    380348
2015    382295
2016    384242
Name: housing_units, dtype: int64
```

In [17]:
```
# Use the Pandas plot function to plot the average housing units per year "Default Bar Chart".
fig_housing_units = plt.figure()

ax = housing_units.plot.bar(x="year", y="housing_units", title="Housing Units in San Francisco from 2010 to 2016",figsize=(8,5))
ax.set(xlabel="Year", ylabel="Housing Units")
plt.show()
plt.close(fig_housing_units)
```

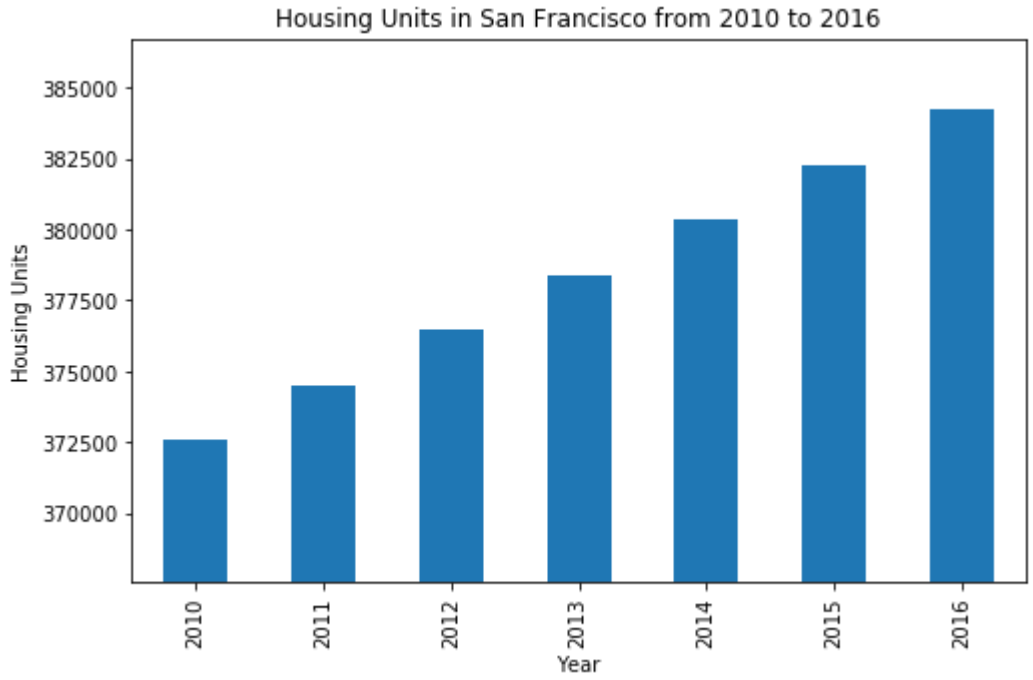


In [18]:
```
# Use the Pandas plot function to plot the average housing units per year "Y-axis limits adjusted".
# Optional Challenge: Use the min, max, and std to scale the y limits of the chart
fig_housing_units = plt.figure()

minimum = sfo_data["housing_units"].min()
maximum = sfo_data["housing_units"].max()
ax = housing_units.plot.bar(x="year", y="housing_units", title="Housing Units in San Francisco from 2010 to 2016",figsize=(8,5))
ax.set(xlabel="Year", ylabel="Housing Units")

plt.ylim((minimum-5000, maximum+2500))

plt.show()
plt.close(fig_housing_units)
```



Average Prices per Square Foot

In this section, you will calculate the average gross rent and average sales price for each year. Plot the results as a line chart.

Average Gross Rent in San Francisco Per Year

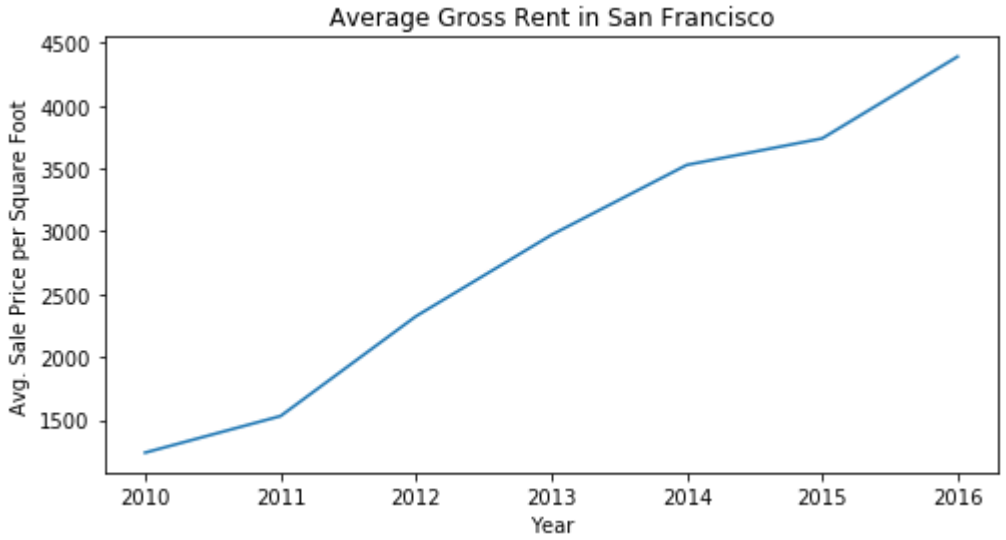
In [19]:
```
# Calculate the average gross rent and average sale price per square foot
sfo_data_grp = sfo_data.drop(columns=["neighborhood","housing_units"])
sfo_data_df = sfo_data_grp.groupby("year").agg({"sale_price_sqr_foot":"mean","gross_rent": "mean"})
sfo_data_df
```

Out[19]:

year	sale_price_sqr_foot	gross_rent
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

In [20]:
```
# Plot the Average Gross Rent per Year as a Line Chart
avg_gross_rent = sfo_data.groupby("year").mean()["gross_rent"]
ax2 = avg_gross_rent.plot.line(figsize=(8,4), title=("Average Gross Rent in San Francisco"))
ax2.set(xlabel="Year", ylabel="Avg. Sale Price per Square Foot")
```

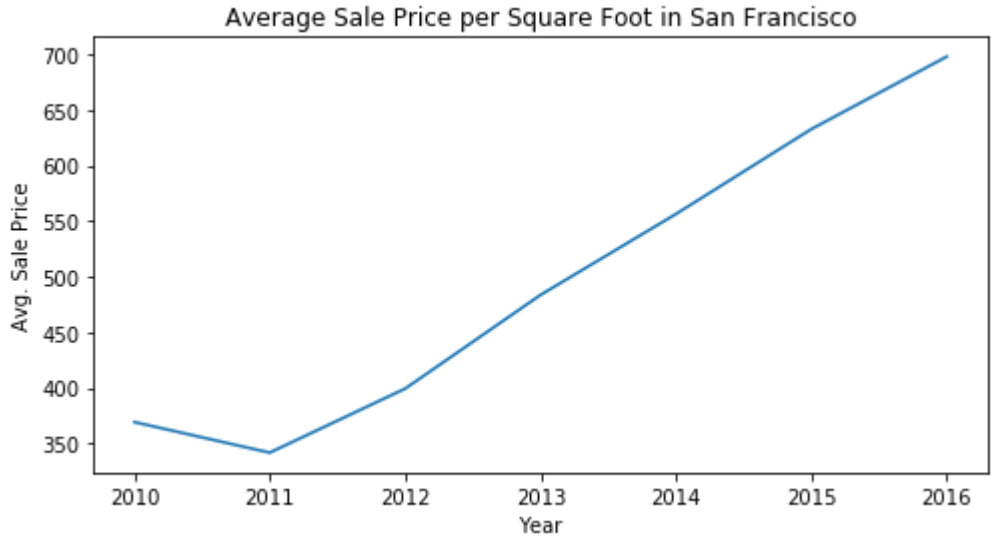
Out[20]:
```
[Text(0, 0.5, 'Avg. Sale Price per Square Foot'), Text(0.5, 0, 'Year')]
```



Average Sales Price per Year

In [21]:
```
# Plot the Average Sales Price per Year as a line chart
avg_sale_price_sqr_foot = sfo_data.groupby("year").mean()["sale_price_sqr_foot"]
ax3 = avg_sale_price_sqr_foot.plot.line(figsize=(8, 4),title=("Average Sale Price per Square Foot in San Francisco"))
ax3.set(xlabel="Year", ylabel="Avg. Sale Price")
```

Out[21]:
```
[Text(0, 0.5, 'Avg. Sale Price'), Text(0.5, 0, 'Year')]
```



Average Prices by Neighborhood

In this section, you will use hvplot to create an interactive visulization of the Average Prices with a dropdown selector for the 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 [22]:
```
# Group by year and neighborhood and then create a new dataframe of the mean values
sfo_data_new_df = pd.read_csv(file_path)
sfo_data_new_df.head(10)
```

Out[22]:

	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 [23]:
```
# Group by year and neighborhood and then create a new dataframe of the mean values
sfo_data_grp_neighborhood = sfo_data_new_df.groupby(["year","neighborhood"]).mean()
sfo_data_grp_neighborhood.head(10)
```

Out[23]:

year	neighborhood	sale_price_sqr_foot	housing_units	gross_rent
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
	Central Sunset	418.172493	372560	1239
	Corona Heights	369.359338	372560	1239
	Cow Hollow	569.379968	372560	1239
	Croker Amazon	165.645730	372560	1239
	Diamond Heights	456.930822	372560	1239

In [24]:
```
# 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
sfo_data_grp_neighborhood.hvplot(kind="line", x="year", y="sale_price_sqr_foot", xlabel="Year", ylabel="Avg.
 Sale Price per Square Foot", groupby="neighborhood")
```

Out[24]:

The Top 10 Most Expensive Neighborhoods

In this section, you will need to calculate the mean sale price 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 [25]:
```
# Getting the data from the top 10 expensive neighborhoods
top_10_neighborhoods = sfo_data_new_df.groupby(["neighborhood"]).mean()
top_10_neighborhoods = top_10_neighborhoods.drop(columns=["year"])
top_10_neighborhoods = top_10_neighborhoods.sort_values(by="sale_price_sqr_foot", ascending=False)
top_10_neighborhoods = top_10_neighborhoods.reset_index()

top_10_neighborhoods.head(10)
```

Out[25]:

	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
5	Telegraph Hill	676.506578	378401.00	2817.285714
6	Presidio Heights	675.350212	378401.00	2817.285714
7	Cow Hollow	665.964042	378401.00	2817.285714
8	Potrero Hill	662.013613	378401.00	2817.285714
9	South Beach	650.124479	375805.00	2099.000000

In [26]:
```
# Plotting the data from the top 10 expensive neighborhoods
top_10_neighborhoods.head(10).sort_values(["sale_price_sqr_foot"], ascending=False).hvplot(kind="bar", x="neighborhood", y="sale_price_sqr_foot", xlabel="Neighborhood", ylabel="Avg. Sale Price per Square Foot", height=400, rot=90, title="Top 10 Expensive Neighborhoods in SFO")
```

Out[26]:

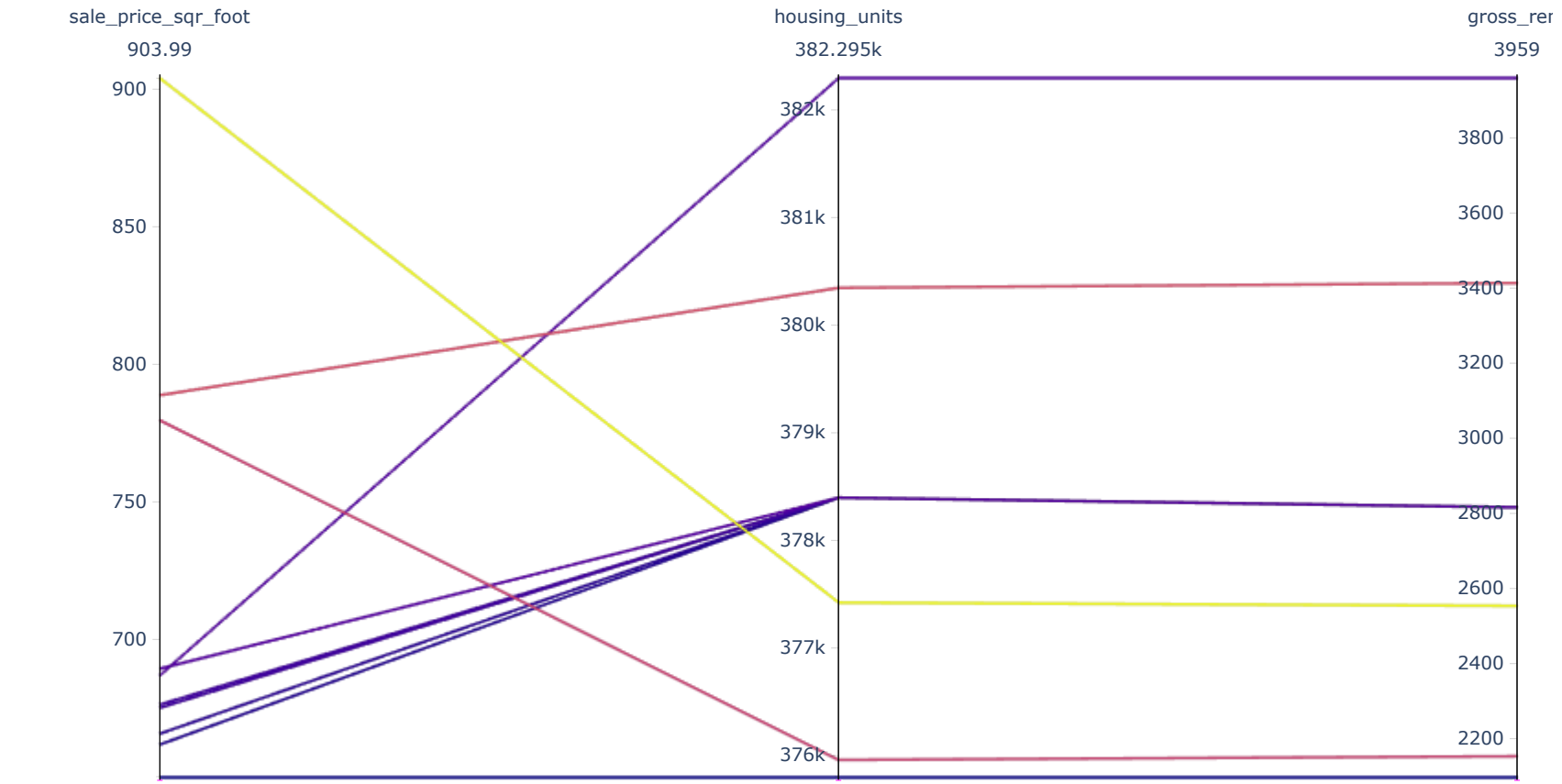
Parallel Coordinates and Parallel Categories Analysis

In this section, you will use plotly express to create parallel coordinates and parallel categories visualizations so that investors can interactively filter and explore various factors related to the sales price of the neighborhoods.

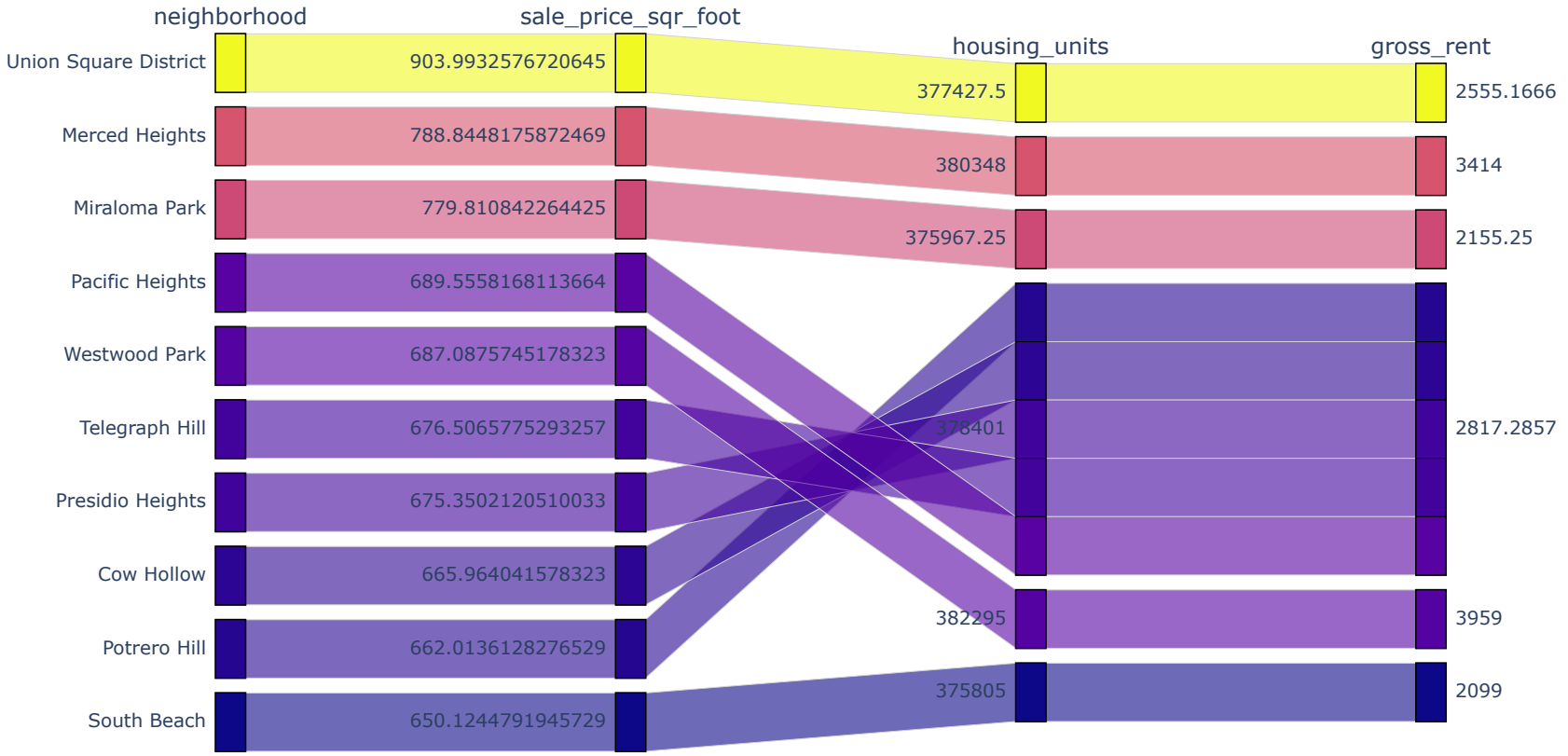
Using the DataFrame of Average values per neighborhood (calculated above), create the following visualizations:

1. Create a Parallel Coordinates Plot
2. Create a Parallel Categories Plot

In [27]:
```
# Parallel Coordinates Plot
px.parallel_coordinates(top_10_neighborhoods.head(10), color='sale_price_sqr_foot')
```



In [28]:
```
# Parallel Categories Plot
px.parallel_categories(top_10_neighborhoods.head(10), color='sale_price_sqr_foot')
```



Neighborhood Map

In this section, you will read in neighboor location data and build an interactive map with the average prices 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 [29]:
```
# Load neighborhoods coordinates data
file_path = Path("Data/neighborhoods_coordinates.csv")
df_neighborhood_locations = pd.read_csv(file_path)
df_neighborhood_locations.head()
```

Out[29]:

	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 prices per neighborhood

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

In [30]:
```
# Calculate the mean values for each neighborhood
avg_prices_neighborhood = sfo_data_new_df.groupby(["neighborhood"]).mean()
avg_prices_neighborhood = avg_prices_neighborhood.drop(columns=["year"])
avg_prices_neighborhood = avg_prices_neighborhood.reset_index(drop=False)
avg_prices_neighborhood.head()
```

Out[30]:

	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 [31]:
```
# Join the average values with the neighborhood locations
df_concat_neighborhood = pd.concat([df_neighborhood_locations, avg_prices_neighborhood], axis="columns", join
="inner")
df_concat_neighborhood = df_concat_neighborhood.drop(columns=["neighborhood"])
df_concat_neighborhood.head()
```

Out[31]:

	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

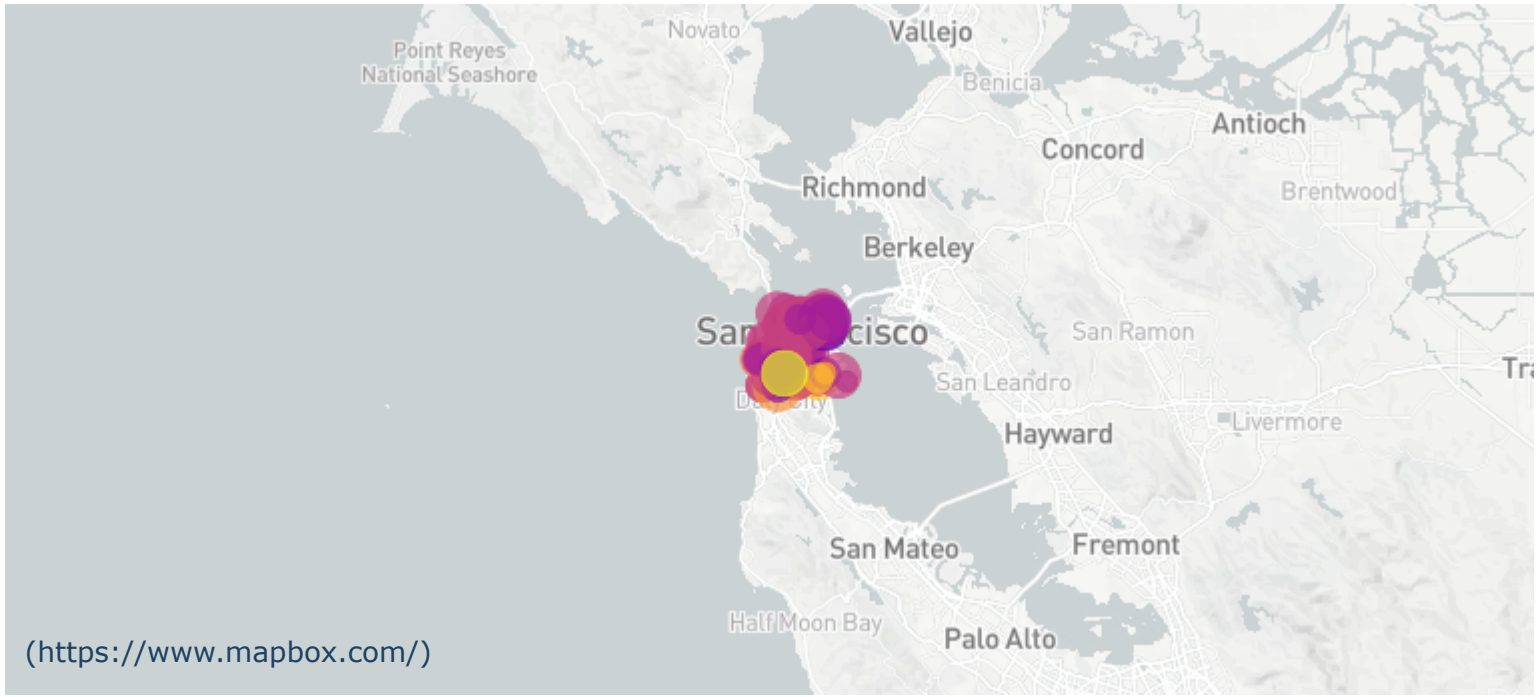
Mapbox Visualization

Plot the aveage values per neighborhood with a plotly express scatter_mapbox visualization.

In [32]:
```
# Set the Mapbox API
px.set_mapbox_access_token(mapbox_token)

# Create a scatter mapbox to analyze neighborhood info
map_1 = px.scatter_mapbox(
    df_concat_neighborhood,
    lat="Lat",
    lon="Lon",
    size="sale_price_sqr_foot",
    color="gross_rent",
    title="Average Sale Price per Square Foot and Gross Rent in San Francisco"
)
map_1.show()
```

Average Sale Price per Square Foot and Gross Rent in San Francisco



In [ ]: