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.

In [1]:

```
# 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 os
from pathlib import Path
from dotenv import load_dotenv
```

In [2]:

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

Import Data

In [3]:

```
# Import the necessary CSVs to Pandas DataFrames
sfo_data = pd.read_csv(Path("../Dashboard_Analysis/Resources/sfo_neighborhoods_census_d
neighborhoods_coordinates = pd.read_csv(Path("../Dashboard_Analysis/Resources/neighborh
```

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.

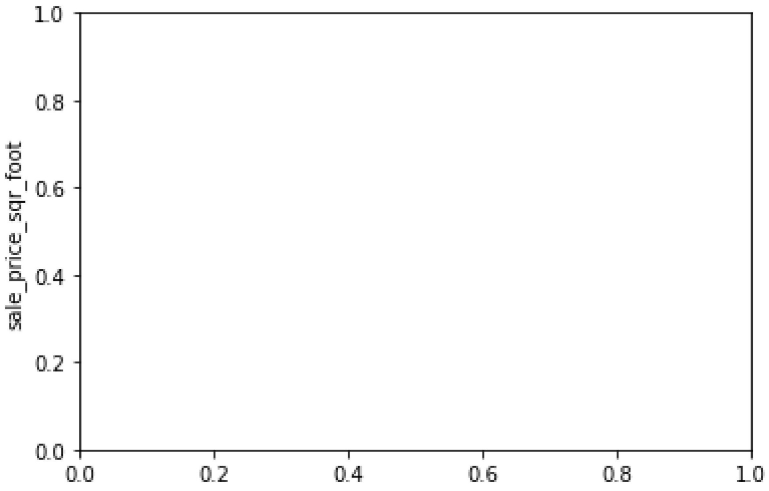
Be sure to include any DataFrame transformation/manipulation code required along with the plotting code.

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.

In [4]:

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

    sfo_data.groupby('year')['housing_units'].mean()
    sfo_dataa = sfo_data.groupby('year')['housing_units'].mean()
    sfo_dataa.to_csv('../Dashboard_Analysis/Resources/housing_costs_mean.csv')
    sfo_dataa.plot(kind='bar', x= 'year', y='housing_units')
    sfo_min = sfo_dataa.min()
    sfo_max = sfo_dataa.max()
    sfo_std = sfo_dataa.std()
    sfo_dataa.plot(kind='bar', figsize= (10,5), title=' San Francisco Housing Units')
    plt.ylabel('Housing Units')
    plt.ylim(sfo_min-sfo_std, sfo_max+sfo_std )

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

    sfo_data_average = sfo_data[['sale_price_sqr_foot','gross_rent']]
    sfo_data_average_output= sfo_data_average.groupby(sfo_data.index).mean()
    sfo_data_average_output

    sfo_monthly_rent_average = sfo_data.groupby('year')['gross_rent'].mean()
    sfo_monthly_rent_average.plot.line(xlabel='Year', ylabel='price per sqr feet', colo



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

    # YOUR CODE HERE!
    sfo_data_average = sfo_data[['sale_price_sqr_foot','gross_rent']]
    sfo_data_average_output= sfo_data_average.groupby(sfo_data.index).mean()
    sfo_data_average_output

    sfo_sales_price_average = sfo_data.groupby('year')['sale_price_sqr_foot'].mean().pl
plt.ylabel('sale_price_sqr_foot')


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

    # YOUR CODE HERE!
    sfo_data_yn = sfo_data.groupby([sfo_data.index, 'neighborhood']).mean().reset_index
    sfo_data_yn.head(10)
    sfo_data_yn.hvplot.line(
    "year",
    "sale_price_sqr_foot",
    xlabel="Year",
    ylabel="Average Sale Price per Square Foot",
    groupby="neighborhood",
)



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

    # YOUR CODE HERE!
    expensive_neighborhoods = sfo_data.groupby('neighborhood').mean()
    expensive_neighborhoods = expensive_neighborhoods.sort_values('sale_price_sqr_foot'
    top_ten_expensive_neighborhoods = expensive_neighborhoods.head(10)
    top_ten_expensive_neighborhoods
    top_ten_expensive_neighborhoods.hvplot.bar(
    "neighborhood",
    "sale_price_sqr_foot",
    title="Top 10 Expensive Neighborhoods in SFO",
    xlabel="Neighborhood",
    ylabel="Avg. Sale Price per Square Foot",
    height=400,
    rot=90
)

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

    # YOUR CODE HERE!
    sfo_data_yn = sfo_data.groupby([sfo_data.index, 'neighborhood']).mean().reset_index
    sfo_data_yn.head(10)
    sfo_data_yn.hvplot.bar(
    "year",
    ["gross_rent", "sale_price_sqr_foot"],
    title="Top 10 Expensive Neighborhoods in SFO",
    xlabel="Neighborhood",
    ylabel="neighborhood",
    height=600,
    rot=90
)

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

    # YOUR CODE HERE!
    sfo_data_yn = sfo_data.groupby([sfo_data.index, 'neighborhood']).mean().reset_index
    expensive_neighborhoods = sfo_data.groupby('neighborhood').mean()
    expensive_neighborhoods = expensive_neighborhoods.sort_values('sale_price_sqr_foot'
    expensive_neighborhoods_per_year =sfo_data_yn[sfo_data_yn["neighborhood"].isin(expe
    expensive_neighborhoods_per_year.head()
    px.parallel_coordinates(
    expensive_neighborhoods_per_year ,
    color="sale_price_sqr_foot",
    title="Parallel Coordinates Analysis of Most Expensive San Francisco Neighborhoods"
)
def parallel_categories():
    """Parallel Categories Plot."""

    # YOUR CODE HERE!
    sfo_data_yn = sfo_data.groupby([sfo_data.index, 'neighborhood']).mean().reset_index
    expensive_neighborhoods = sfo_data.groupby('neighborhood').mean()
    expensive_neighborhoods = expensive_neighborhoods.sort_values('sale_price_sqr_foot'
    expensive_neighborhoods_per_year =sfo_data_yn[sfo_data_yn["neighborhood"].isin(expe
    expensive_neighborhoods_per_year.head()
    px.parallel_categories(
    expensive_neighborhoods_per_year ,
    color="sale_price_sqr_foot",
    title="Parallel Categories Analysis of Most Expensive San Francisco Neighborhoods"
)


def neighborhood_map():
    """Neighborhood Map."""
    # YOUR CODE HERE!
    mean_of_neighborhoods = sfo_data.groupby(by="neighborhood").mean().reset_index()
    mean_of_neighborhoods.rename(columns={"neighborhood": "Neighborhood"}, inplace=True
    mean_of_neighborhoods
    neighborhood_data = pd.concat([
    neighborhoods_coordinates,
    mean_of_neighborhoods['sale_price_sqr_foot'],
    mean_of_neighborhoods['housing_units'],
    mean_of_neighborhoods['gross_rent']
    ], axis=1).dropna()

    neighborhood_data.head()

    px.set_mapbox_access_token(map_box_api)
    Neighborhoods_map = px.scatter_mapbox(
    neighborhood_data,
    lat="Lat",
    lon="Lon",
    size="sale_price_sqr_foot",
    color="gross_rent",
    size_max=20,
    zoom=11,
    width=2000,
    hover_name="Neighborhood",
    title="Averange Sale Price Per Square Foot and Gross Rent in San Francisco",
)

    Neighborhoods_map.show()

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

    # YOUR CODE HERE!
    sfo_data_yn = sfo_data.groupby([sfo_data.index, 'neighborhood']).mean().reset_index
    expensive_neighborhoods = sfo_data.groupby('neighborhood').mean()
    expensive_neighborhoods = expensive_neighborhoods.sort_values('sale_price_sqr_foot'
    expensive_neighborhoods_per_year =sfo_data_yn[sfo_data_yn["neighborhood"].isin(expe
    expensive_neighborhoods_per_year.head()
    sunburst = px.sunburst(
    expensive_neighborhoods_per_year,
    path=["year", "neighborhood"],
    values='sale_price_sqr_foot',
    color='gross_rent',
    color_continuous_scale='Blues',
    title="Costs Analysis of Most Expensive neighborhoods in San Francisco per Year",
    height=800
)


    sunburst.show()
```

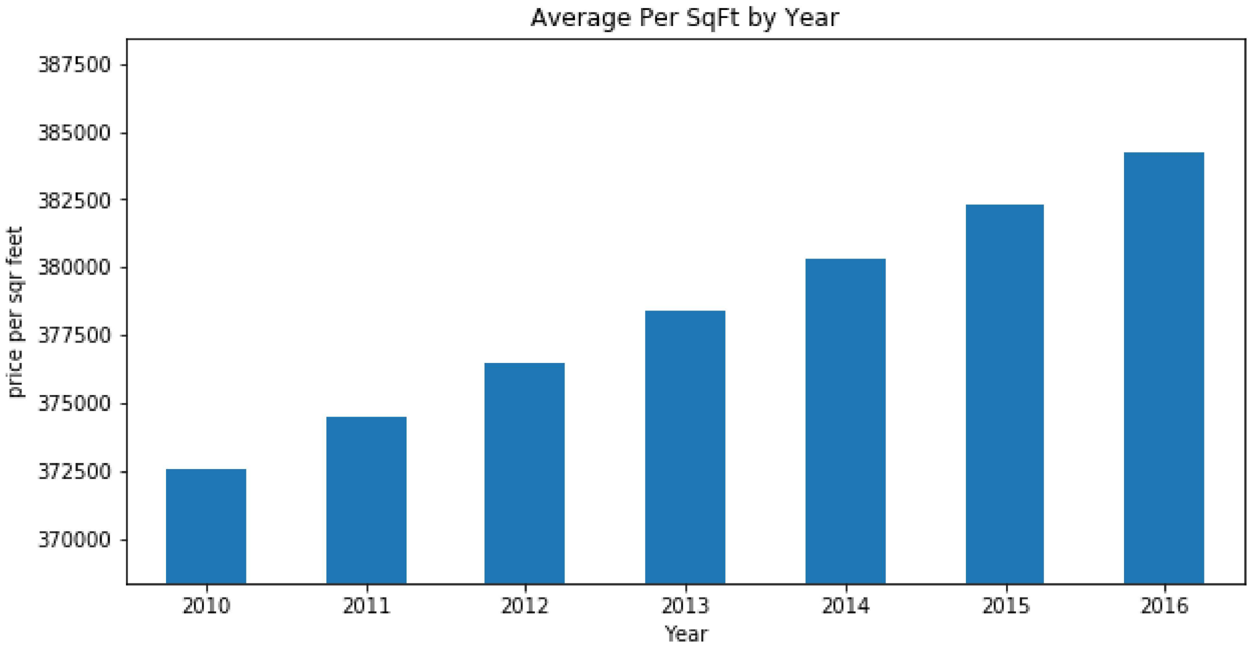


Panel Dashboard

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

In [5]:

```
# Create a Title for the Dashboard
# YOUR CODE HERE!
title = pn.pane.Markdown(
    """
# Real Estate Analysis of San Francisco from 2010 to 2016
""",
    width=800,
)

# Create a tab layout for the dashboard
# YOUR CODE HERE!
layout = pn.Tabs(
('Yearly Market Analysis' , pn.Row(housing_units_per_year(), average_gross_rent(), aver


('Neighborhood Analysis' , pn.Column(average_price_by_neighborhood(), top_most_expensiv

('Parallel Plot Analysis' , pn.Column(parallel_categories(), parallel_coordinates(),wid

('Sunburst Plot Analysis' , pn.Column( sunburst())),
)
# Create the dashboard
dashboard = pn.Column(pn.Row(title), layout, width=500)
```


Average Per SqFt by Year
387500
385000
382500
380000
377500
375000
372500
370000
price per sqr feet
2010
2011
2012
2013
2014
2015
2016
Year

Serve the Panel Dashboard

In [6]:

```
# Serve the dashboard
dashboard.servable()
```

Out[6]:

Debugging

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

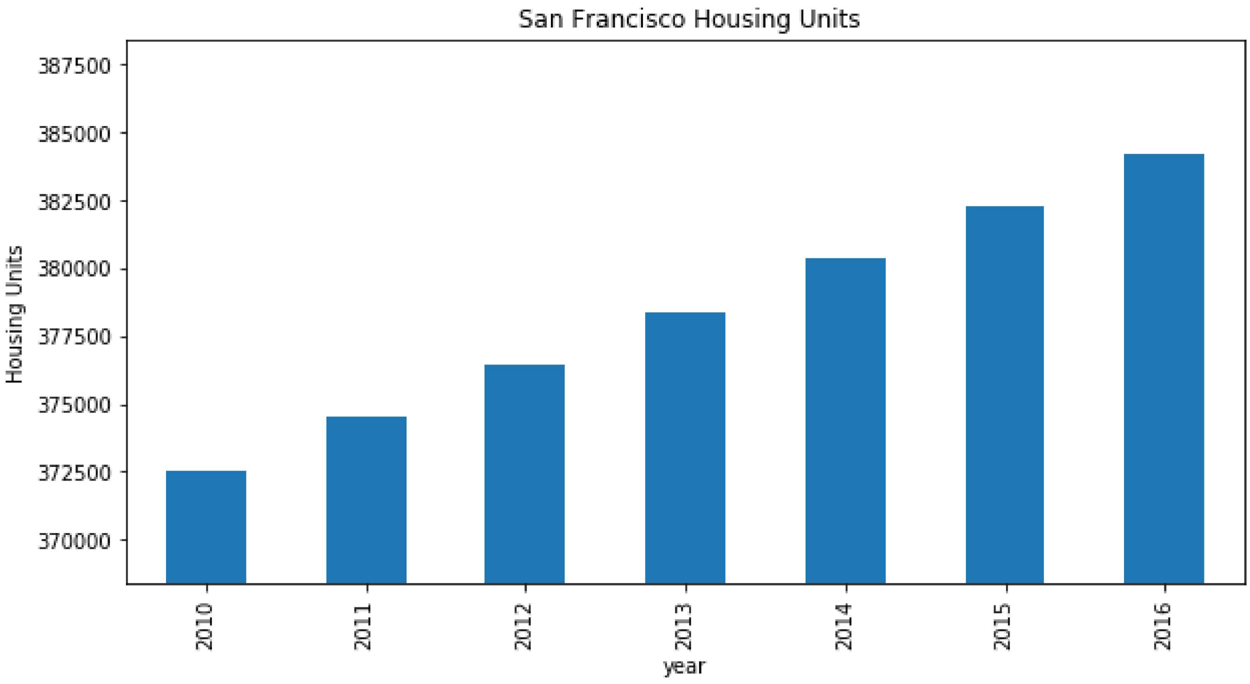
However, you can test each plot by uncommenting the following code

In [7]:

```
# housing_units_per_year()
sfo_data.groupby('year')['housing_units'].mean()
sfo_dataa = sfo_data.groupby('year')['housing_units'].mean()
sfo_dataa.to_csv('../Dashboard_Analysis/Resources/housing_costs_mean.csv')
sfo_dataa.plot(kind='bar', x= 'year', y='housing_units')
sfo_min = sfo_dataa.min()
sfo_max = sfo_dataa.max()
sfo_std = sfo_dataa.std()
sfo_dataa.plot(kind='bar', figsize= (10,5), title=' San Francisco Housing Units')
plt.ylabel('Housing Units')
plt.ylim(sfo_min-sfo_std, sfo_max+sfo_std )
```

Out[7]:

```
(368353.9992867333, 388448.0007132667)
```

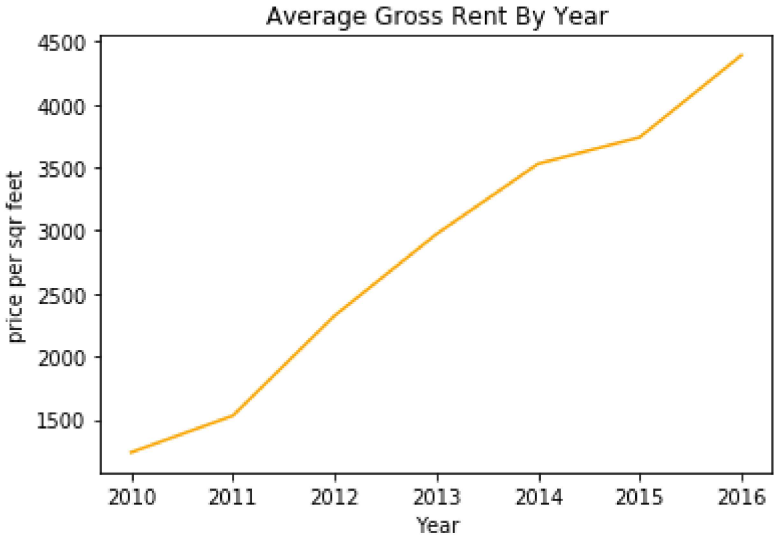


In [8]:

```
# average_gross_rent()
sfo_data_average = sfo_data[['sale_price_sqr_foot','gross_rent']]
sfo_data_average_output= sfo_data_average.groupby(sfo_data.index).mean()
sfo_data_average_output
```

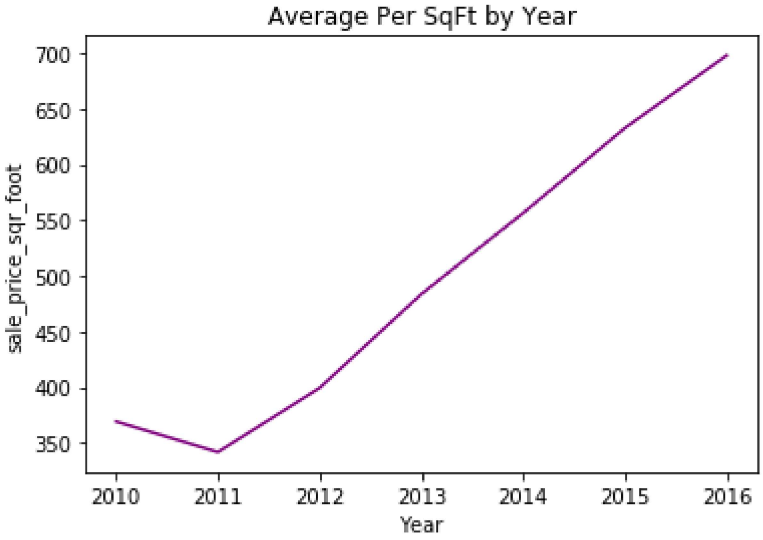
```
sfo_monthly_rent_average = sfo_data.groupby('year')['gross_rent'].mean()
sfo_monthly_rent_average.plot.line(xlabel='Year', ylabel='price per sqr feet', color='o
```

```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x24ee260d108>
```



```
In [9]:
# average_sales_price()
sfo_data_average = sfo_data[['sale_price_sqr_foot','gross_rent']]
sfo_data_average_output= sfo_data_average.groupby(sfo_data.index).mean()
sfo_data_average_output
sfo_sales_price_average = sfo_data.groupby('year')['sale_price_sqr_foot'].mean().plot.l
plt.ylabel('sale_price_sqr_foot')
```

```
Out[9]: Text(0, 0.5, 'sale_price_sqr_foot')
```



```
In [10]:
# average_price_by_neighborhood()
sfo_data_yn = sfo_data.groupby([sfo_data.index, 'neighborhood']).mean().reset_index()
sfo_data_yn.head(10)
sfo_data_yn.hvplot.line(
    "year",
    "sale_price_sqr_foot",
    xlabel="Year",
    ylabel="Average Sale Price per Square Foot",
    groupby="neighborhood",
)
```

Out[10]:

In [11]:

```
# top_most_expensive_neighborhoods()
expensive_neighborhoods = sfo_data.groupby('neighborhood').mean()
expensive_neighborhoods = expensive_neighborhoods.sort_values('sale_price_sqr_foot', as
top_ten_expensive_neighborhoods = expensive_neighborhoods.head(10)
top_ten_expensive_neighborhoods
top_ten_expensive_neighborhoods.hvplot.bar(
    "neighborhood",
    "sale_price_sqr_foot",
    title="Top 10 Expensive Neighborhoods in SFO",
    xlabel="Neighborhood",
    ylabel="Avg. Sale Price per Square Foot",
    height=400,
    rot=90
)
```

Out[11]:

In [12]:

```
# most_expensive_neighborhoods_rent_sales()
sfo_data_yn = sfo_data.groupby([sfo_data.index, 'neighborhood']).mean().reset_index()
sfo_data_yn.head(10)
sfo_data_yn.hvplot.bar(
    "year",
    ["gross_rent", "sale_price_sqr_foot"],
    title="Top 10 Expensive Neighborhoods in SFO",
    xlabel="Neighborhood",
    ylabel="neighborhood",
    height=600,
    rot=90
)
```

Out[12]:

In [14]:

```
# neighborhood_map().show()
mean_of_neighborhoods = sfo_data.groupby(by="neighborhood").mean().reset_index()
mean_of_neighborhoods.rename(columns={"neighborhood": "Neighborhood"}, inplace=True)
mean_of_neighborhoods
neighborhood_data = pd.concat([
    neighborhoods_coordinates,
    mean_of_neighborhoods['sale_price_sqr_foot'],
    mean_of_neighborhoods['housing_units'],
    mean_of_neighborhoods['gross_rent']
    ], axis=1).dropna()
neighborhood_data.head()

px.set_mapbox_access_token(map_box_api)
Neighborhoods_map = px.scatter_mapbox(
    neighborhood_data,
    lat="Lat",
    lon="Lon",
    size="sale_price_sqr_foot",
    color="gross_rent",
    size_max=20,
    zoom=11,
    width=2000,
    hover_name="Neighborhood",
    title="Averange Sale Price Per Square Foot and Gross Rent in San Francisco",
)

Neighborhoods_map.show()
```

In [15]:

```
# parallel_categories()
sfo_data_yn = sfo_data.groupby([sfo_data.index, 'neighborhood']).mean().reset_index()
expensive_neighborhoods = sfo_data.groupby('neighborhood').mean()
expensive_neighborhoods = expensive_neighborhoods.sort_values('sale_price_sqr_foot', as
expensive_neighborhoods_per_year =sfo_data_yn[sfo_data_yn["neighborhood"].isin(expensiv
expensive_neighborhoods_per_year.head()
px.parallel_categories(
    expensive_neighborhoods_per_year ,
    color="sale_price_sqr_foot",
    title="Parallel Categories Analysis of Most Expensive San Francisco Neighborhoods"
)
```

In [16]:

```
# parallel_coordinates()
sfo_data_yn = sfo_data.groupby([sfo_data.index, 'neighborhood']).mean().reset_index()
expensive_neighborhoods = sfo_data.groupby('neighborhood').mean()
expensive_neighborhoods = expensive_neighborhoods.sort_values('sale_price_sqr_foot', as
expensive_neighborhoods_per_year =sfo_data_yn[sfo_data_yn["neighborhood"].isin(expensiv
expensive_neighborhoods_per_year.head()
px.parallel_coordinates(
    expensive_neighborhoods_per_year ,
    color="sale_price_sqr_foot",
    title="Parallel Coordinates Analysis of Most Expensive San Francisco Neighborhoods"
)
```

In [17]:

```
# sunburst()
sfo_data_yn = sfo_data.groupby([sfo_data.index, 'neighborhood']).mean().reset_index()
expensive_neighborhoods = sfo_data.groupby('neighborhood').mean()
expensive_neighborhoods = expensive_neighborhoods.sort_values('sale_price_sqr_foot', as
expensive_neighborhoods_per_year =sfo_data_yn[sfo_data_yn["neighborhood"].isin(expensiv
expensive_neighborhoods_per_year.head()
sunburst = px.sunburst(
    expensive_neighborhoods_per_year,
    path=["year", "neighborhood"],
    values='sale_price_sqr_foot',
    color='gross_rent',
    color_continuous_scale='Blues',
    title="Costs Analysis of Most Expensive neighborhoods in San Francisco per Year",
    height=800
)


sunburst.show()
```

```
In [ ]:
```