

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 [185]:

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

# Initialize the Panel Extensions (for Plotly)
pn.extension('plotly')

from panel.interact import interact
from panel import widgets
```

In [2]:

```
# Read the Mapbox API key
load_dotenv()
mapbox_token = os.getenv("MAPBOX_TOKEN")
```

Import Data

In [3]:

```
# Import the CSVs to Pandas DataFrames
file_path = Path("Data/sfo_neighborhoods_census_data.csv")
sfo_data = pd.read_csv(file_path, index_col="year")

file_path = Path("Data/neighborhoods_coordinates.csv")
df_neighborhood_locations = pd.read_csv(file_path)
```

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 [179]:

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

    # YOUR CODE HERE!
    units_per_year= sfo_data.groupby('year').mean()["housing_units"]
    std_y=units_per_year.std()
    min_y=units_per_year.min()-std_y
    max_y=units_per_year.max()+std_y
    ax = units_per_year.plot(kind="bar", title='Housing Units in San Francisco from 2010 t
o 2016')
    ax.set_xlabel("Year")
    ax.set_ylabel("Housing Units")
    plt.ylim(min_y,max_y)
    plt.show()
    return ax

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

    # YOUR CODE HERE!
    average_per_year= sfo_data.groupby('year').mean()
    ax = average_per_year['gross_rent'].plot(kind="line",title="Average Gross Rent in San
Francisco")
    ax.set_xlabel("Year")
    ax.set_ylabel("Gross Rent")
    return ax

def average_sales_price():
    """Average Sales Price Per Year."""
    average_per_year= sfo_data.groupby('year').mean()
    # YOUR CODE HERE!
    ax = average_per_year['sale_price_sqr_foot'].plot(kind="line",title="Average Sale Pric
e per Square Foot in San Francisco")
    ax.set_xlabel("Year")
    ax.set_ylabel("Avg. Sale Price")
    return ax

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

    # YOUR CODE HERE!
    neighborhood_data_per_year= sfo_data.groupby(['year','neighborhood']).mean()
    neighborhood_data_per_year.reset_index(inplace=True)
    df_sliced=neighborhood_data_per_year[['year','neighborhood','sale_price_sqr_foot']]
    neighborhood_df=df_sliced.loc[df_sliced["neighborhood"] == neighborhood]
    ax = neighborhood_df['sale_price_sqr_foot'].plot(kind="line",title="Average Sale Price
per Square Foot")

    return ax

def top_most_expensive_neighborhoods():
```

```

"""Top 10 Most Expensive Neighborhoods."""

# YOUR CODE HERE!
averages_by_neighborhood_top10=sfo_data.groupby('neighborhood').mean().sort_values(by=
['sale_price_sqr_foot'], ascending=False).head(10)
#averages_by_neighborhood["sale_price_sqr_foot"].plot(kind="bar")
std_y=averages_by_neighborhood_top10["sale_price_sqr_foot"].std()
min_y=averages_by_neighborhood_top10["sale_price_sqr_foot"].min()-std_y
max_y=averages_by_neighborhood_top10["sale_price_sqr_foot"].max()+std_y

ax = averages_by_neighborhood_top10["sale_price_sqr_foot"].plot(kind="bar", title='Top
10 expensive neighborhoods')
ax.set_xlabel("Neighborhood")
ax.set_ylabel("Sale Price Per Sqr Foot")
plt.ylim(min_y,max_y)
return ax

def parallel_coordinates():
    """Parallel Coordinates Plot."""
    averages_by_neighborhood=sfo_data.groupby('neighborhood').mean().sort_values(by=['sale
_price_sqr_foot'], ascending=False)
    # YOUR CODE HERE!
    averages_by_neighborhood.reset_index(inplace=True)
    ax=px.parallel_coordinates(averages_by_neighborhood, color='sale_price_sqr_foot', width
h=1000,height=800)

    return ax

def parallel_categories():
    """Parallel Categories Plot."""
    averages_by_neighborhood=sfo_data.groupby('neighborhood').mean().sort_values(by=['sale
_price_sqr_foot'], ascending=False)
    averages_by_neighborhood.reset_index(inplace=True)
    # YOUR CODE HERE!
    ax=px.parallel_categories(
        averages_by_neighborhood,
        dimensions=["neighborhood", "housing_units", "gross_rent"],
        color="sale_price_sqr_foot",
        color_continuous_scale=px.colors.sequential.Inferno,
        labels={
            "neighborhood": "Neighborhood",
            "housing_units": "Uousing Units",
            "gross_rent": "Gross Rent",
        },
        width=1000,
        height=800
    )
    return ax

def neighborhood_map():
    """Neighborhood Map"""
    # YOUR CODE HERE!
    df_neighborhood=sfo_data.groupby('neighborhood').mean()
    df_neighborhood_locations.columns=["lat", "lon"]
    df_joined=pd.concat([df_neighborhood,df_neighborhood_locations], axis='columns', join=

```

```
'inner')
df_joined.reset_index(inplace=True)
px.set_mapbox_access_token(mapbox_token)

myplot= px.scatter_mapbox(
    df_joined,
    lat="lat",
    lon="lon",
    size="sale_price_sqr_foot",
    color="neighborhood",
    color_continuous_scale=px.colors.cyclical.IceFire,
    title="Neighborhood Info",
    zoom=10,
    width=1200,
    height=700
)
return myplot
```

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 [202]:

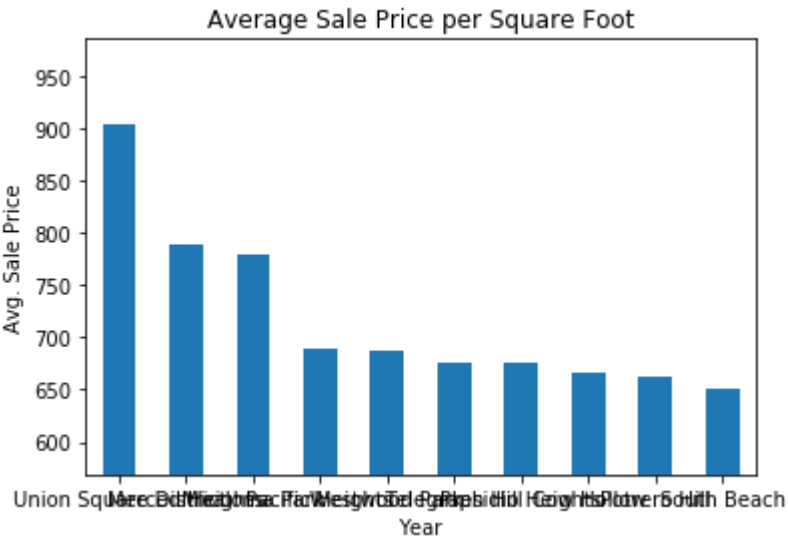
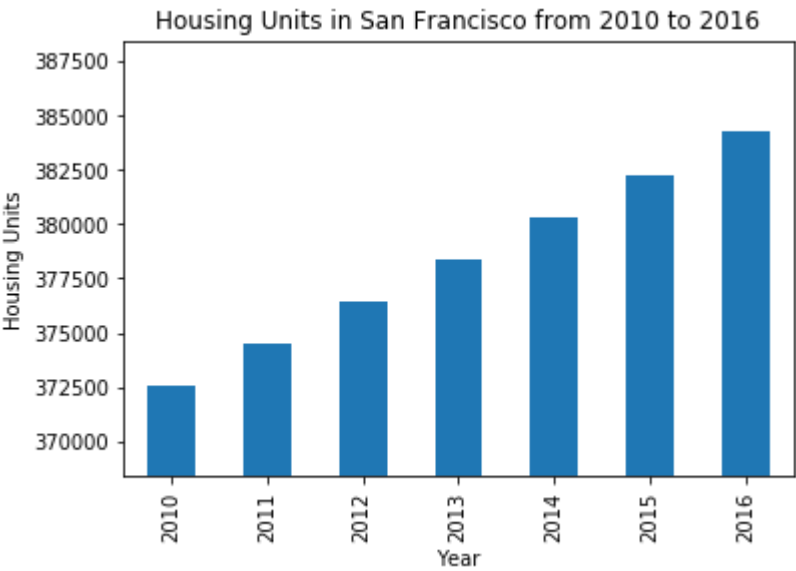
```
# YOUR CODE HERE!
#average_price_by_neighborhood()
Row1 = pn.Row("Housing Units and Most Expensive Neighborhoods", housing_units_per_year(), top_most_expensive_neighborhoods())

Row2 = pn.Row("Averages - Gross Rent, Sale Price", average_gross_rent(), average_sales_price(), interact(average_price_by_neighborhood, neighborhood=sfo_data['neighborhood']))

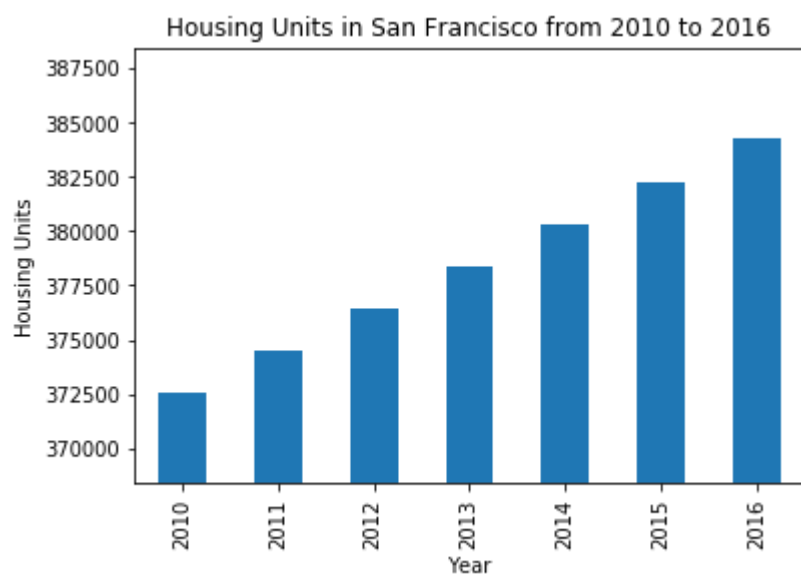
column1=pn.Column("Parallels", parallel_coordinates(), parallel_categories())

column2=pn.Column("Sanfrancisco", neighborhood_map())

# Create tabs
sanFrancisco_picture = pn.Tabs(
    ("Overview", Row1), ("Averages", Row2), ("Parallels", column1) , ("View Map", column2)
)
```



In [200]:



Out[200]:

[]

Serve the Panel Dashboard

In [203]:

```
sanFrancisco_picture.servable()
```

Out[203]:

In [168]:

In []:

In []: