

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]:

```
# 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

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

In [2]:

```
# Read the Mapbox API key
mapbox_token = os.getenv("MAPBOX_API_KEY")
px.set_mapbox_access_token(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 [34]:

```
def housing_units_per_year():

    df_sfo_mean_max = df_sfo_mean.max()
    df_sfo_mean_min = df_sfo_mean.min()

    annual_houses_plot = df_sfo_mean.housing_units.plot.bar(
        x='Year',
        y= 'Housing Units',
        title = 'Housing Units in San Francisco from 2010 to 2016',
        ylim = (df_sfo_mean_min.housing_units - 2500, df_sfo_mean_max.housing_units + 2500
    )
    )
    return annual_houses_plot
```

In [51]:

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

    # YOUR CODE HERE!
    df_sfo_mean = sfo_data.groupby(['year']).mean()
    df_sfo_mean_max = df_sfo_mean.max()
    df_sfo_mean_min = df_sfo_mean.min()

    avg_gross_rent_plot = df_sfo_mean.gross_rent.plot.line(
        x='Year',
        y= 'Gross Rent',
        title = 'Average Gross Rent in San Francisco',
        ylim = (df_sfo_mean_min.gross_rent - 500, df_sfo_mean_max.gross_rent + 500)
    )
    return avg_gross_rent_plot
```

In [52]:

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

    # YOUR CODE HERE!
    df_sfo_mean = sfo_data.groupby(['year']).mean()
    df_sfo_mean_max = df_sfo_mean.max()
    df_sfo_mean_min = df_sfo_mean.min()

    avg_sales_price_plot = df_sfo_mean.sale_price_sqr_foot.plot.line(
        x='Year',
        y='Avg. Sale Price',
        title='Average Sale Price per Square Foot in San Francisco',
        ylim=(df_sfo_mean_min.sale_price_sqr_foot - 50, df_sfo_mean_max.sale_price_sqr_f
oot + 50)
    )
    return avg_sales_price_plot
```

In [53]:

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

    # YOUR CODE HERE!
    sfo_data_neighborhood = sfo_data.groupby(['year', 'neighborhood']).mean()

    avg_price_neighborhood_plot = sfo_data_neighborhood.sale_price_sqr_foot.hvplot.line(gr
oupby='neighborhood')

    return avg_price_neighborhood_plot
```

In [54]:

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

    # YOUR CODE HERE!
    sfo_group_neighborhoods = sfo_data.groupby(['neighborhood']).mean()
    top10_neighborhoods = sfo_group_neighborhoods.sort_values(['sale_price_sqr_foot'], asc
ending=False)
    df_top10_neighborhoods = top10_neighborhoods.head(10)

    top10_cost_neighborhood_plot = df_top10_neighborhoods.sale_price_sqr_foot.hvplot.bar(r
ot=90)

    return top10_cost_neighborhood_plot
```

In [55]:

```
def parallel_coordinates():  
    """Parallel Coordinates Plot."""  
  
    # YOUR CODE HERE!  
    sfo_group_neighborhoods = sfo_data.groupby(['neighborhood']).mean()  
    top10_neighborhoods = sfo_group_neighborhoods.sort_values(['sale_price_sqr_foot'], ascending=False)  
    df_top10_neighborhoods = top10_neighborhoods.head(10)  
  
    parrallel_coordinates_plot = px.parallel_coordinates(  
        df_top10_neighborhoods,  
        color= 'sale_price_sqr_foot',  
        color_continuous_scale=px.colors.sequential.Inferno,  
    )  
    return parrallel_coordinates_plot
```

In [56]:

```
def parallel_categories():  
    """Parallel Categories Plot."""  
  
    # YOUR CODE HERE!  
    sfo_group_neighborhoods = sfo_data.groupby(['neighborhood']).mean()  
    top10_neighborhoods = sfo_group_neighborhoods.sort_values(['sale_price_sqr_foot'], ascending=False)  
    df_top10_neighborhoods = top10_neighborhoods.head(10)  
  
    parallel_categories_plot = px.parallel_categories(  
        df_top10_neighborhoods,  
        color= 'sale_price_sqr_foot',  
        color_continuous_scale=px.colors.sequential.Inferno,  
    )  
    return parallel_categories_plot
```

In [57]:

```
def neighborhood_map():  
    """Neighborhood Map"""  
  
    # YOUR CODE HERE!  
    file_path = Path("Data/neighborhoods_coordinates.csv")  
    df_neighborhood_locations = pd.read_csv(file_path)  
    df_neighborhood_locations.rename(columns={'Neighborhood': 'neighborhood'}, inplace=True)  
  
    df_sfo_all_neighbor = sfo_data.groupby(['neighborhood'], as_index=False).mean()  
  
    df_neighborhood_locations.set_index(['neighborhood'], inplace=True)  
    df_sfo_all_neighbor.set_index(['neighborhood'], inplace=True)  
    df_sfo_locations_value = pd.concat([df_neighborhood_locations, df_sfo_all_neighbor], axis=1, sort=True)  
  
    neighborhood_map_plot = px.scatter_mapbox(data_frame=df_sfo_locations_value, lat="Lat", lon="Lon", color="sale_price_sqr_foot")  
    return neighborhood_map_plot
```

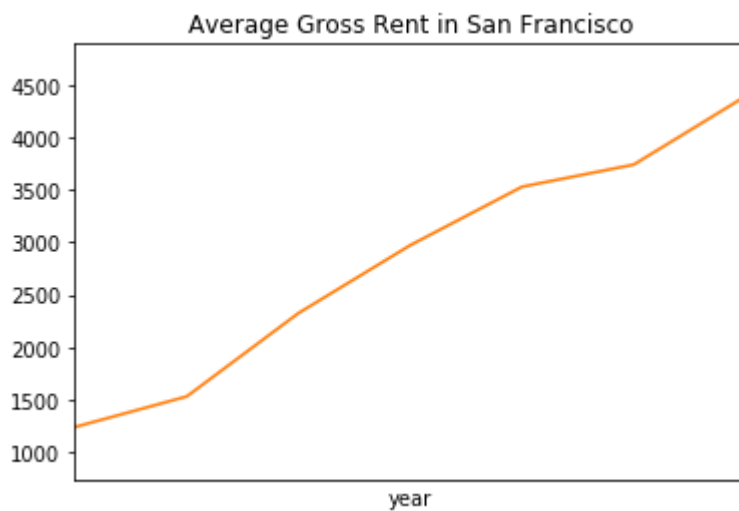
Panel Dashboard

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

In [78]:

```
visual_neighborhood_book = pn.Tabs(  
    ("Map - San Francisco Neighborhood Values", neighborhood_map),  
    ("Sales Price SQRFT by Neighborhood", average_price_by_neighborhood),  
    ("Most Expensive Neighborhoods", top_most_expensive_neighborhoods),  
    ("Parallel Coordinates", parallel_coordinates),  
    ("Parallel Categories", parallel_categories),  
    ("Housing Units Per Year", housing_units_per_year),  
    ("Average Sales Price", average_sales_price),  
    ("Average Gross Rent", average_gross_rent)  
)  
visual_neighborhood_book
```

Out[78]:



Serve the Panel Dashboard

In [79]:

```
visual_neighborhood_book.servable()
```

Out[79]:

In []: