dashboard

February 3, 2021

1 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.

[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 os
from pathlib import Path
from dotenv import load_dotenv
from panel.interact import interact
```

[3]:
```
# Read the Mapbox API key
load_dotenv()
map_box_api = os.getenv("MAPBOX_API_KEY")
px.set_mapbox_access_token(map_box_api)
```

2 Import Data

[4]:
```
# Import the necessary CSVs to Pandas DataFrames
file_path = Path("Data/sfo_neighborhoods_census_data.csv")
sfo_data = pd.read_csv(file_path, index_col="year")
coord_path = Path("Data/neighborhoods_coordinates.csv")
coord_data = pd.read_csv(coord_path,index_col="Neighborhood")
```

[5]:
```
# Concat data
# neighborhod prices data
neighborhood_df = sfo_data.groupby("neighborhood").mean()
neighborhood_df.reset_index(inplace=True)
neighborhood_df.rename(columns={"neighborhood":"Neighborhood"},inplace=True)
neighborhood_df.set_index("Neighborhood",inplace=True)
#neighborhood coordinates
```

```
neighborhood_coords = pd.concat([neighborhood_df,coord_data], axis="columns",␣
↪join="inner")
neighborhood_coords.reset_index(inplace=True)
```

[6]:
```
top_10_df = sfo_data.groupby("neighborhood").mean()
top_10_df.sort_values(by=["sale_price_sqr_foot"],ascending=False,inplace=True)
top_10_neighborhoods = top_10_df.nlargest(10,"sale_price_sqr_foot")
top_10_neighborhoods.reset_index(inplace=True)
```

2.1 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.

[7]:
```
# Define Panel Visualization Functions
def housing_units_per_year():
    """Housing Units Per Year."""

    yearly_housing_units = sfo_data["housing_units"].groupby("year").mean()
    yearly_housing_min = yearly_housing_units.min()
    yearly_housing_max = yearly_housing_units.max()
    yearly_housing_std = yearly_housing_units.std()
    y_min = yearly_housing_min - yearly_housing_std
    y_max = yearly_housing_max + yearly_housing_std
    plot = px.bar(yearly_housing_units,title="Average Housing Units by Year",␣
↪y="housing_units", range_y=(y_min,y_max),width=800)

    return plot

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

    yearly_prices = sfo_data.groupby("year").mean()
    yearly_prices.drop(columns=["housing_units"],inplace=True)
    plot = px.line(yearly_prices,title="Average Rent per␣
↪Year",y="gross_rent",width=800)
```

```
    return plot

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

    yearly_prices = sfo_data.groupby("year").mean()
    yearly_prices.drop(columns=["housing_units"],inplace=True)
    plot = px.line(yearly_prices,title="Average Sales Price per SqFt by␣
 ↪Year",y="sale_price_sqr_foot",width=800)

    return plot

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

    yearly_neighborhood_mean = sfo_data.groupby(["year","neighborhood"]).mean()
    plot = yearly_neighborhood_mean["sale_price_sqr_foot"].hvplot.
 ↪line(groupby="neighborhood",title="Average Yearly Sale Price per SqFt by␣
 ↪Neighborhood",width=800)
    plot2 = yearly_neighborhood_mean["gross_rent"].hvplot.
 ↪line(groupby="neighborhood",title="Average Yearly Rent by␣
 ↪Neighborhood",width=800)

    visual = pn.Column(plot,plot2)


    return visual

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

    df = top_10_neighborhoods.set_index("neighborhood")
    plot = df["sale_price_sqr_foot"].hvplot.bar(
        title="Top 10 Most Expensive Neighborhoods in San Francisco",
        ylabel="Avg. Sale Price per SqFt",
        xlabel="Neighborhood",
        rot=90,
        width=800,
        height=400,
    )

    return plot

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

    yearly_neighborhood_mean = sfo_data.groupby(["year","neighborhood"]).mean()
```

```
    plot = yearly_neighborhood_mean.hvplot.bar(
        y=["gross_rent","sale_price_sqr_foot"],
        x="year",
        ylabel="Price USD",
        xlabel="Year",
        title="Average Rent vs Sales Price per SqFt by Year",
        rot=90,
        groupby="neighborhood",
        height=400,
        width=800
    )
    return plot


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

    df_expensive_neighborhoods_per_year = sfo_data[sfo_data["neighborhood"].
 ↪isin(top_10_neighborhoods["neighborhood"])]
    df_expensive_neighborhoods_per_year.reset_index(inplace=True)
    plot = px.parallel_coordinates(
        top_10_neighborhoods,
        color="sale_price_sqr_foot",
        color_continuous_scale=px.colors.cyclical.IceFire,
        labels={
            "sale_price_sqr_foot":"Sales Price Per SqFt",
            "housing_units":"Housing Units",
            "gross_rent":"Gross Rent"
        }
    )
    return plot


def parallel_categories():
    """Parallel Categories Plot."""

    df_expensive_neighborhoods_per_year = sfo_data[sfo_data["neighborhood"].
 ↪isin(top_10_neighborhoods["neighborhood"])]
    df_expensive_neighborhoods_per_year.reset_index(inplace=True)
    plot = px.parallel_categories(
        top_10_neighborhoods,
        color="sale_price_sqr_foot",
        color_continuous_scale=px.colors.cyclical.IceFire,
        labels={
            "neighborhood":"Neighborhood",
            "sale_price_sqr_foot":"Sales Price per SqFt",
            "housing_units":"Housing Units",
```

```
            "gross_rent":"Gross Rent"
        }
    )
    return plot

def neighborhood_map():
    """Neighborhood Map."""

    map_1 = px.scatter_mapbox(
        neighborhood_coords,
        lat="Lat",
        lon="Lon",
        hover_name="Neighborhood",
        color="gross_rent",
        size="sale_price_sqr_foot",
        color_continuous_scale=px.colors.cyclical.IceFire,
        mapbox_style="streets",
        zoom=11,
    )

    return map_1


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

    df_expensive_neighborhoods_per_year = sfo_data[sfo_data["neighborhood"].
↪isin(top_10_neighborhoods["neighborhood"])]
    df_expensive_neighborhoods_per_year.reset_index(inplace=True)
    plot = px.sunburst(
        df_expensive_neighborhoods_per_year,
        path=["year","neighborhood"],
        values="sale_price_sqr_foot",
        color="gross_rent",
        color_continuous_scale=px.colors.sequential.Blues,
        height=600,
        title="Cost Analysis of Most Expensive Neighborhoods in San Francisco␣
↪per Year"
    )
    return plot
```

2.2 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!

[8]:
```
# Create a Title for the Dashboard
title = "# San Francisco Real Estate Analysis: 2010 to 2016"


# Create a tab layout for the dashboard
welcome = pn.panel(neighborhood_map())
yearly = pn.
 ↪Row(housing_units_per_year(),average_sales_price(),average_gross_rent())
neighborhood = pn.
 ↪Column(average_price_by_neighborhood(),most_expensive_neighborhoods_rent_sales(),top_most_e
parallel = pn.Column(parallel_categories(),parallel_coordinates())
sunburstpanel = pn.panel(sunburst())

tabs = pn.Tabs(
    ("Welcome",welcome),
    ("Yearly Analysis",yearly),
    ("Neighborhood Analysis",neighborhood),
    ("Parallel Plots Analysis",parallel),
    ("Sunburst Plot Analysis",sunburstpanel)
)

# Create the dashboard
dashboard = pn.Column(title,tabs)
```

2.3 Serve the Panel Dashboard

[19]:
```
# Serve the# dashboard
dashboard.servable()
```

[19]:
```
Column
    [0] Markdown(str)
    [1] Tabs
        [0] Plotly(Figure)
        [1] Row
            [0] Plotly(Figure, viewport={'xaxis.range': [2009.5, …})
            [1] Plotly(Figure, viewport={'xaxis.range': [2010, …})
            [2] Plotly(Figure, viewport={'xaxis.range': [2010, …})
        [2] Column
            [0] Column
                [0] Row
                    [0] HoloViews(DynamicMap)
                    [1] Column
                        [0] WidgetBox
                            [0] Select(margin=(20, 20, 20, 20),
name='neighborhood', options=['Alamo Square', …], value='Alamo Square',
width=250)
                        [1] VSpacer()
```

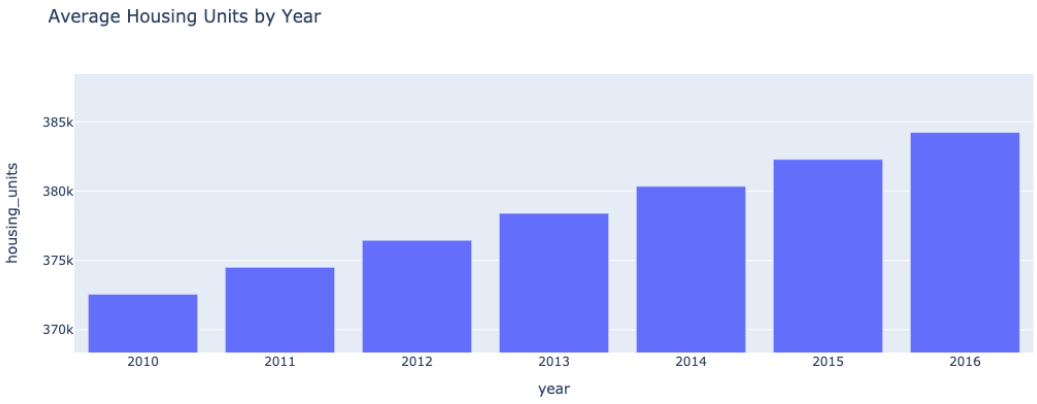
```
                [1] Row
                    [0] HoloViews(DynamicMap)
                    [1] Column
                        [0] WidgetBox
                            [0] Select(margin=(20, 20, 20, 20),
name='neighborhood', options=['Alamo Square', …], value='Alamo Square',
width=250)
                        [1] VSpacer()
            [1] Row
                [0] HoloViews(DynamicMap)
                [1] Column
                    [0] WidgetBox
                        [0] Select(margin=(20, 20, 20, 20), name='neighborhood',
options=['Alamo Square', …], value='Alamo Square', width=250)
                    [1] VSpacer()
            [2] HoloViews(Bars)
        [3] Column
            [0] Plotly(Figure)
            [1] Plotly(Figure)
        [4] Plotly(Figure)
```

3 Debugging

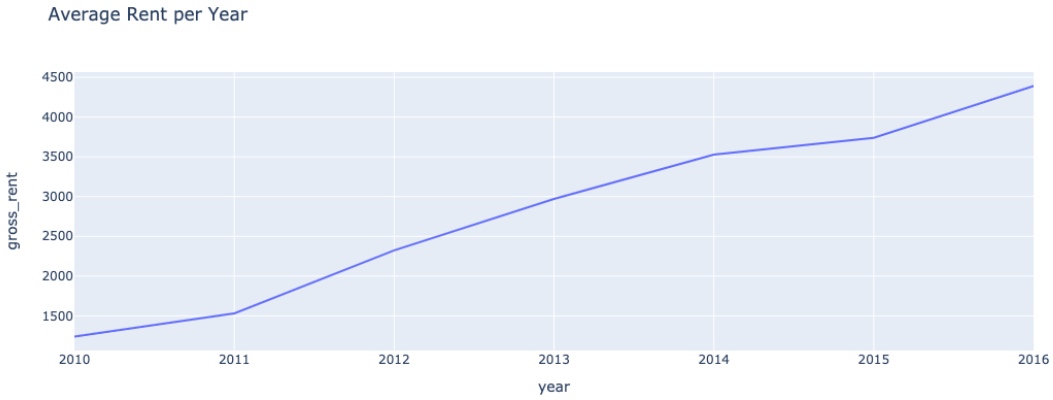
Note: Some of the Plotly express plots may not render in the notebook through the panel functions.

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

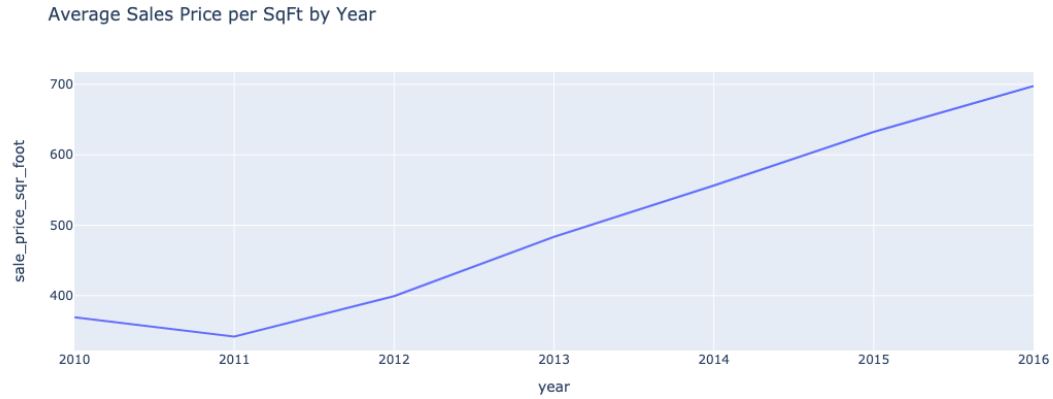
[11]:
```
housing_units_per_year()
```



[13]:
```
average_gross_rent()
```



[14]:
```
average_sales_price()
```



[15]:
```
average_price_by_neighborhood()
```

[15]:
```
Column
    [0] Row
        [0] HoloViews(DynamicMap)
        [1] Column
            [0] WidgetBox
                [0] Select(margin=(20, 20, 20, 20), name='neighborhood',
options=['Alamo Square', …], value='Alamo Square', width=250)
            [1] VSpacer()
    [1] Row
        [0] HoloViews(DynamicMap)
        [1] Column
```

```
            [0] WidgetBox
                [0] Select(margin=(20, 20, 20, 20), name='neighborhood',
options=['Alamo Square', …], value='Alamo Square', width=250)
            [1] VSpacer()
```

[16]:
```
top_most_expensive_neighborhoods()
```

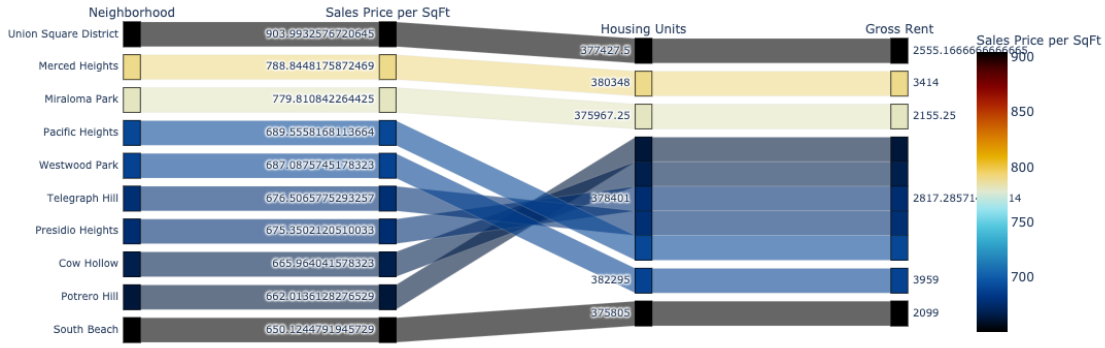
[16]:
```
:Bars   [neighborhood]   (sale_price_sqr_foot)
```

[17]:
```
most_expensive_neighborhoods_rent_sales()
```

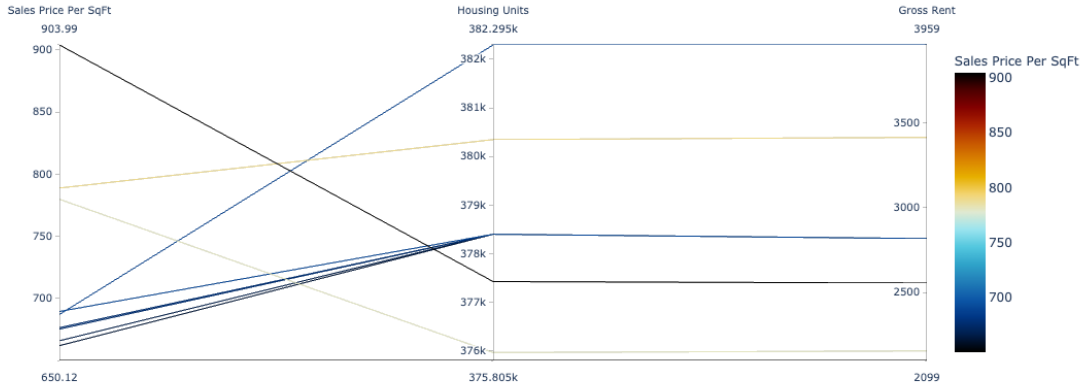
[17]:
```
:DynamicMap   [neighborhood]
   :Bars   [year,Variable]   (value)
```

[18]:
```
neighborhood_map().show()
```

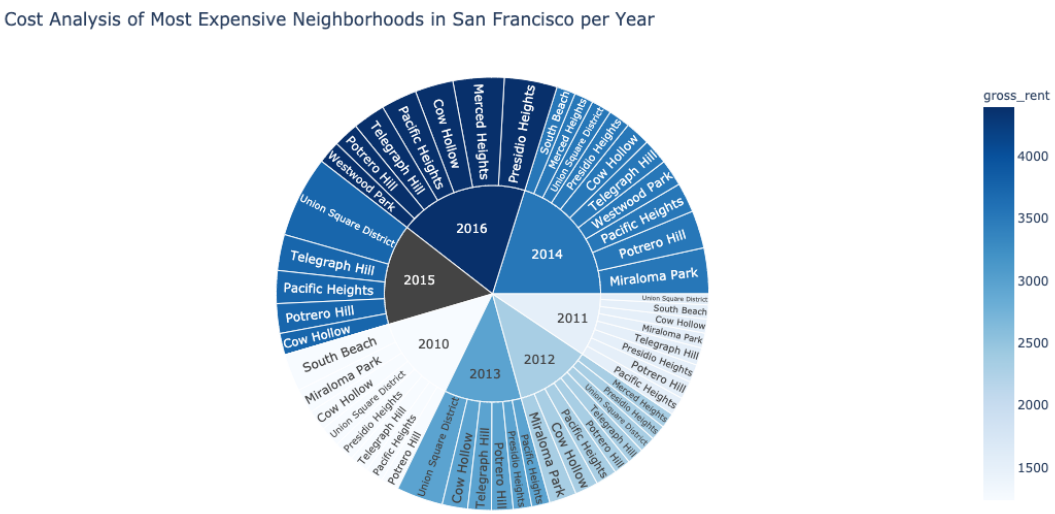
[19]:
```
parallel_categories()
```



[20]:
```
parallel_coordinates()
```


Sales Price Per SqFt
903.99
900
850
800
750
700
650.12
Housing Units
382.295k
382k
381k
380k
379k
378k
377k
376k
375.805k
Gross Rent
3959
3500
3000
2500
2099
Sales Price Per SqFt
900
850
800
750
700

[21]:
```
sunburst()
```


Cost Analysis of Most Expensive Neighborhoods in San Francisco per Year
2016
2015
2014
2011
2012
2013
2010
Merced Heights
Presidio Heights
Cow Hollow
Pacific Heights
Telegraph Hill
Potrero Hill
Westwood Park
Union Square District
South Beach
Miraloma Park
gross_rent
4000
3500
3000
2500
2000
1500

[ ]:
```

```