

0. Imports:

```
In [6]: import numpy as np
import pandas as pd
import seaborn as sns

from matplotlib import pyplot as plt
import plotly.express as px
from IPython.core.display import HTML
```

```
In [7]: def jupyter_settings():
    %matplotlib inline
    %pylab inline
    plt.style.use('bmh')
    plt.rcParams['figure.figsize'] = [10, 6]
    plt.rcParams['font.size'] = 24
    display(HTML('<style>.container {width:100% !important;}</style>'))
    pd.options.display.max_columns = None
    pd.options.display.max_rows = None
    pd.set_option('display.expand_frame_repr', False)
    sns.set()

jupyter_settings()
```

Populating the interactive namespace from numpy and matplotlib

```
In [8]: data = pd.read_csv('~/.PycharmProjects/kc_house_data.csv')
```

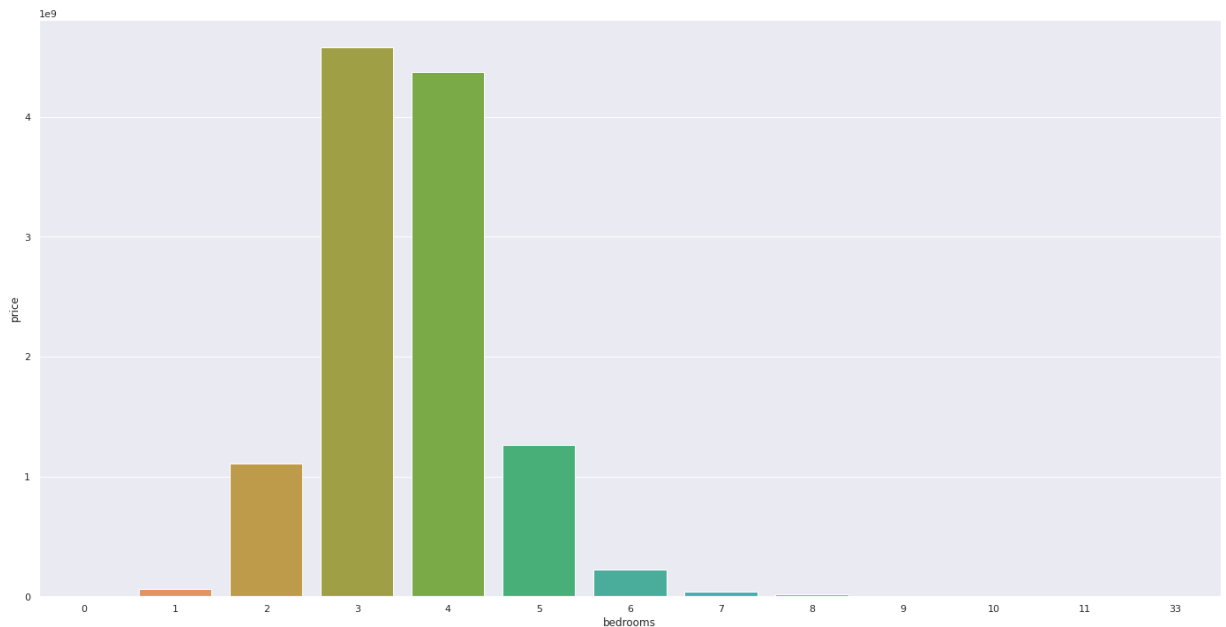
1. Crie uma nova coluna chamada: “dormitory_type”

- Se o valor da coluna “bedrooms” for igual à 1 => ‘studio’
- Se o valor da coluna “bedrooms” for igual a 2 => ‘apartment’
- Se o valor da coluna “bedrooms” for maior que 2 => ‘house’

```
In [9]: data['dormitory_type'] = data['bedrooms'].apply(lambda x: 'studio' if x == 1
                                                    'apartment' if x
                                                    'house' if x > 2)
```

2. Faça um gráfico de barras que represente a soma dos preços pelo número de quartos.

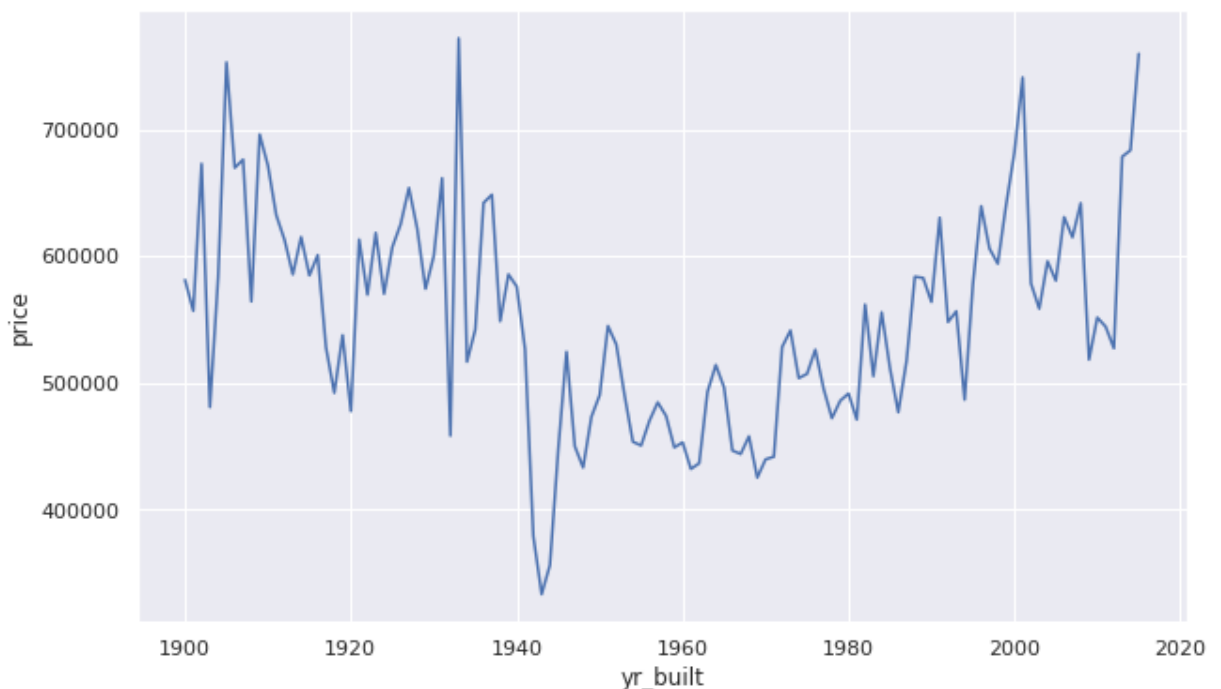
```
In [10]: plt.figure(figsize=(24, 12))
df = data[['price', 'bedrooms']].groupby('bedrooms').sum().reset_index()
sns.barplot(x='bedrooms', y='price', data=df);
```



3. Faça um gráfico de linhas que represente a média dos preços pelo ano construção dos imóveis.

In [11]:

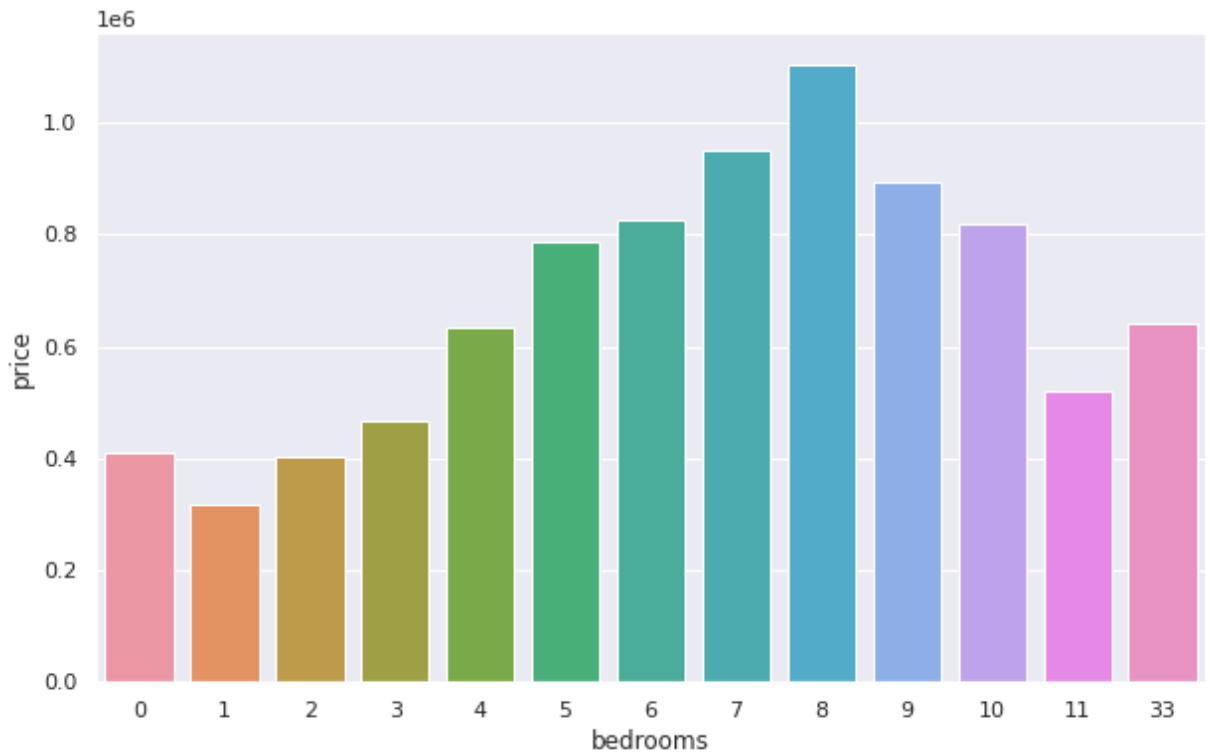
```
df = data[['price', 'yr_built']].groupby('yr_built').mean().reset_index()
sns.lineplot(x='yr_built', y='price', data=df);
```



4. Faça um gráfico de barras que represente a média dos preços pelo tipo dos dormitórios.

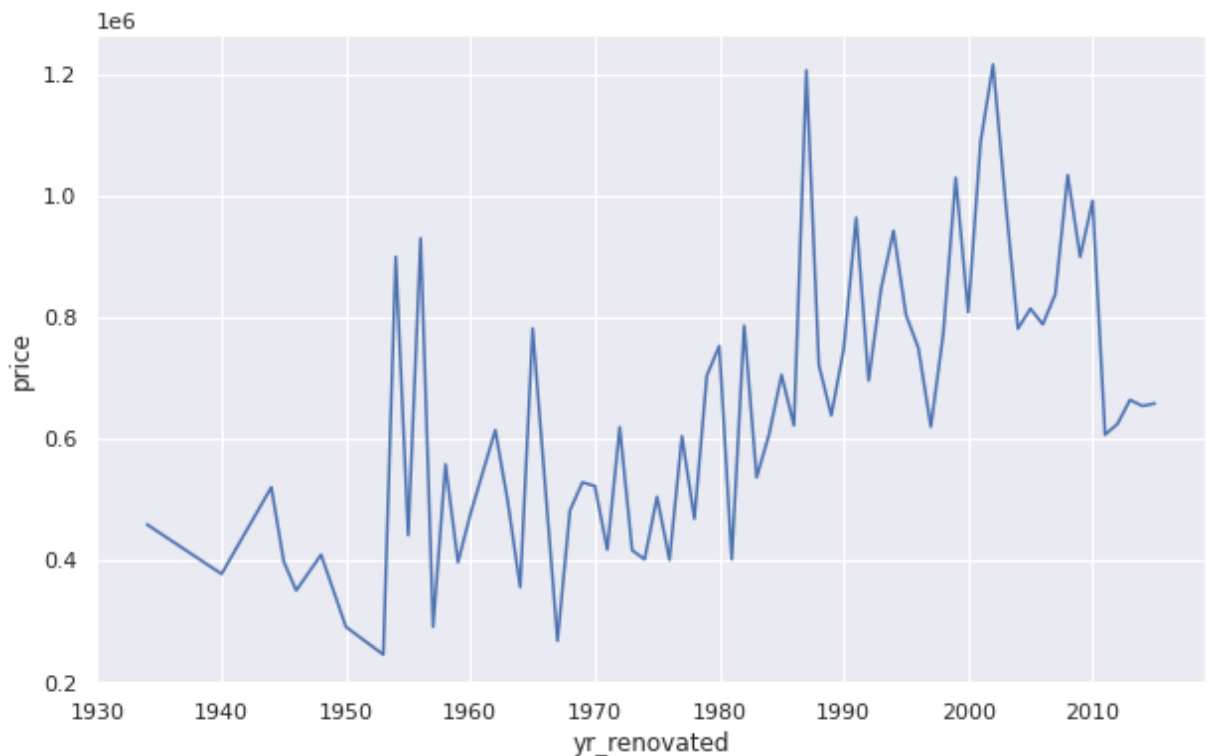
In [12]:

```
df = data[['price', 'bedrooms']].groupby('bedrooms').mean().reset_index()
#plt.bar(df['bedrooms'], df['price'])
sns.barplot(x='bedrooms', y='price', data=df);
```



5. Faça um gráfico de linha que mostre a evolução da média dos preços pelo ano da reforma dos imóveis, a partir do ano de 1930.

```
In [13]: df = data.loc[data['yr_renovated'] > 1930, ['price', 'yr_renovated']].groupby
sns.lineplot(x='yr_renovated', y='price', data=df);
```



6. Faça um tabela que mostre a média dos

preços por ano de construção e tipo de dormitórios dos imóveis.

```
In [14]: df = data[['price', 'dormitory_type']].groupby('dormitory_type').mean().reset_index().head()
```

```
Out[14]:
```

	dormitory_type	price
0	NA	409503.846154
1	apartamento	401372.681884
2	house	563092.211899
3	studio	317642.884422

7. Crie um Dashboard com os gráficos das questões 02, 03, 04 (Dashboard: 1 Linha e 2 colunas)

```
In [15]: from matplotlib import gridspec

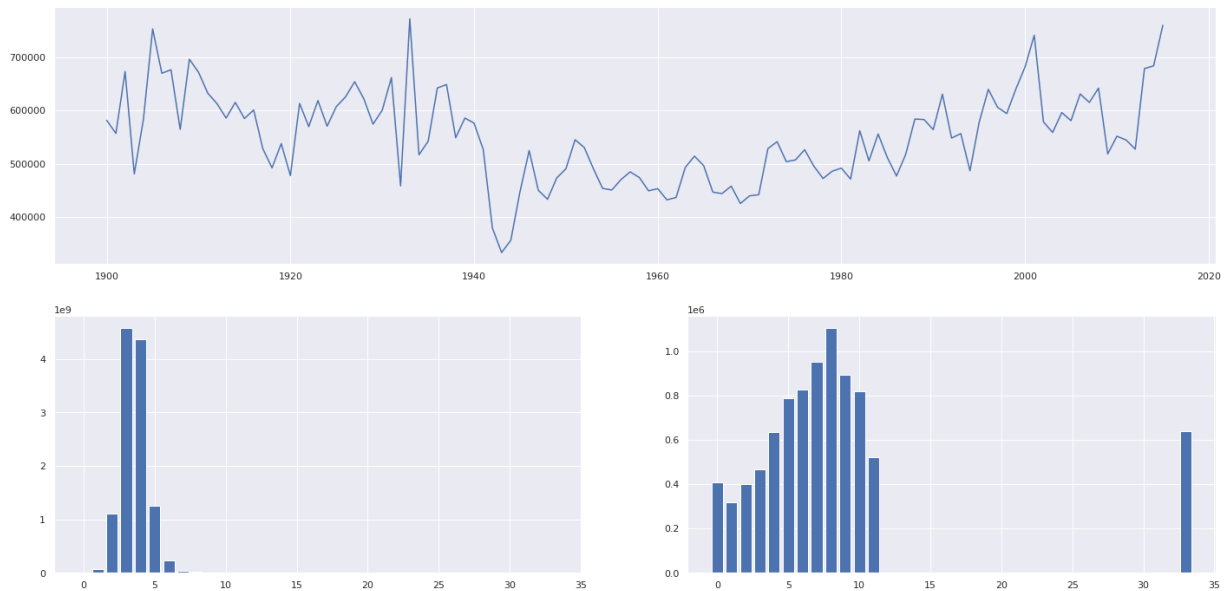
fig = plt.figure(figsize=(24, 12))
specs = gridspec.GridSpec(ncols=2, nrows=2, figure=fig)

ax1 = fig.add_subplot( specs[0, :]) # primeira linha e as 2 colunas
ax2 = fig.add_subplot( specs[1, 0]) # segunda linha e a primeira coluna
ax3 = fig.add_subplot( specs[1, 1]) # segunda linha e a segunda coluna

# questao 03:
df = data[['price', 'yr_built']].groupby('yr_built').mean().reset_index()
ax1.plot(df['yr_built'], df['price']);

# questao 02:
df = data[['price', 'bedrooms']].groupby('bedrooms').sum().reset_index()
ax2.bar(df['bedrooms'], df['price']);

# questao 04:
df = data[['price', 'bedrooms']].groupby('bedrooms').mean().reset_index()
ax3.bar(df['bedrooms'], df['price'], data=df);
```



8. Crie um Dashboard com os gráficos das perguntas 02, 04 (Dashboard: 2 colunas)

In [16]:

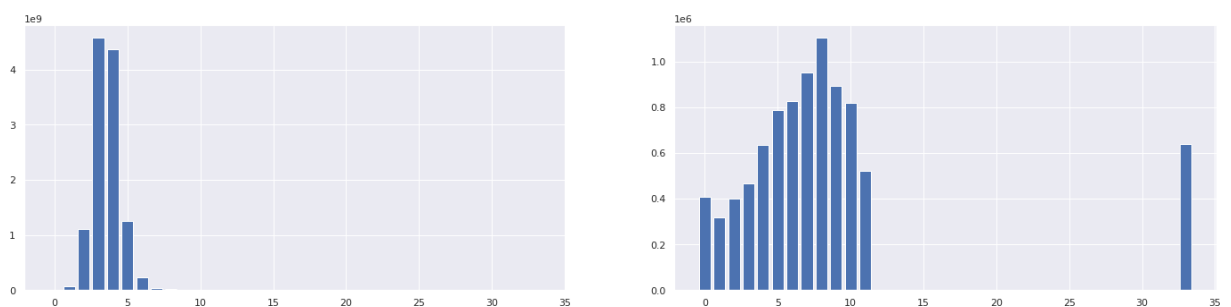
```
from matplotlib import gridspec

fig = plt.figure(figsize=(24,12))
specs = gridspec.GridSpec(ncols=2, nrows=2, figure=fig)

ax1 = fig.add_subplot(specs[0, 0]) # primeira linha e primeira coluna
ax2 = fig.add_subplot(specs[0, 1]) # primeira linha e segunda coluna

# questao 02:
df = data[['price', 'bedrooms']].groupby('bedrooms').sum().reset_index()
ax1.bar(df['bedrooms'], df['price']);

# questao 04:
df = data[['price', 'bedrooms']].groupby('bedrooms').mean().reset_index()
ax2.bar(df['bedrooms'], df['price'], data=df);
```



9. Crie um Dashboard com os gráficos das perguntas 03, 05 (Dashboard: 2 Linhas)

In [17]:

```
from matplotlib import gridspec

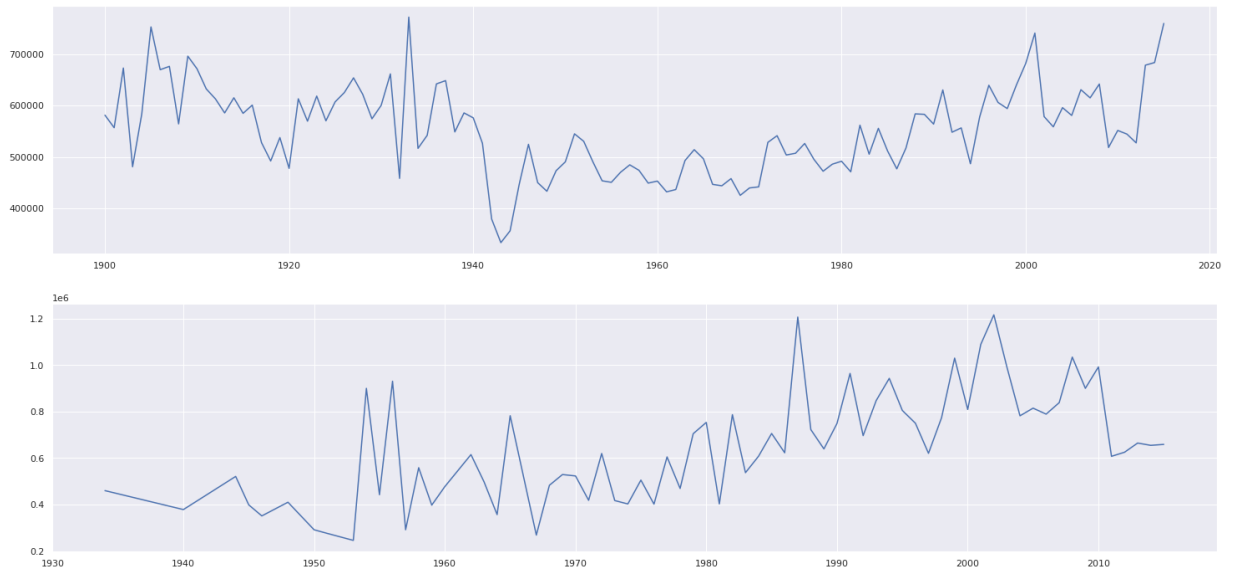
fig = plt.figure(figsize=(25,12))
specs = gridspec.GridSpec(ncols=1, nrows=2, figure=fig)

ax1 = fig.add_subplot(specs[0, 0]) # primeira linha e primeira coluna
```

```
ax2 = fig.add_subplot(specs[1, 0]) # primeira linha e primeira coluna

# questao 03:
df = data[['price', 'yr_built']].groupby('yr_built').mean().reset_index()
ax1.plot(df['yr_built'], df['price']);

# questao 05:
df = data.loc[data['yr_renovated'] > 1930, ['price', 'yr_renovated']].groupby
ax2.plot(df['yr_renovated'], df['price']);
```



10. Faça um gráfico com o tamanho dos pontos sendo igual ao tamanho da sala de estar

```
In [18]: import plotly.express as px

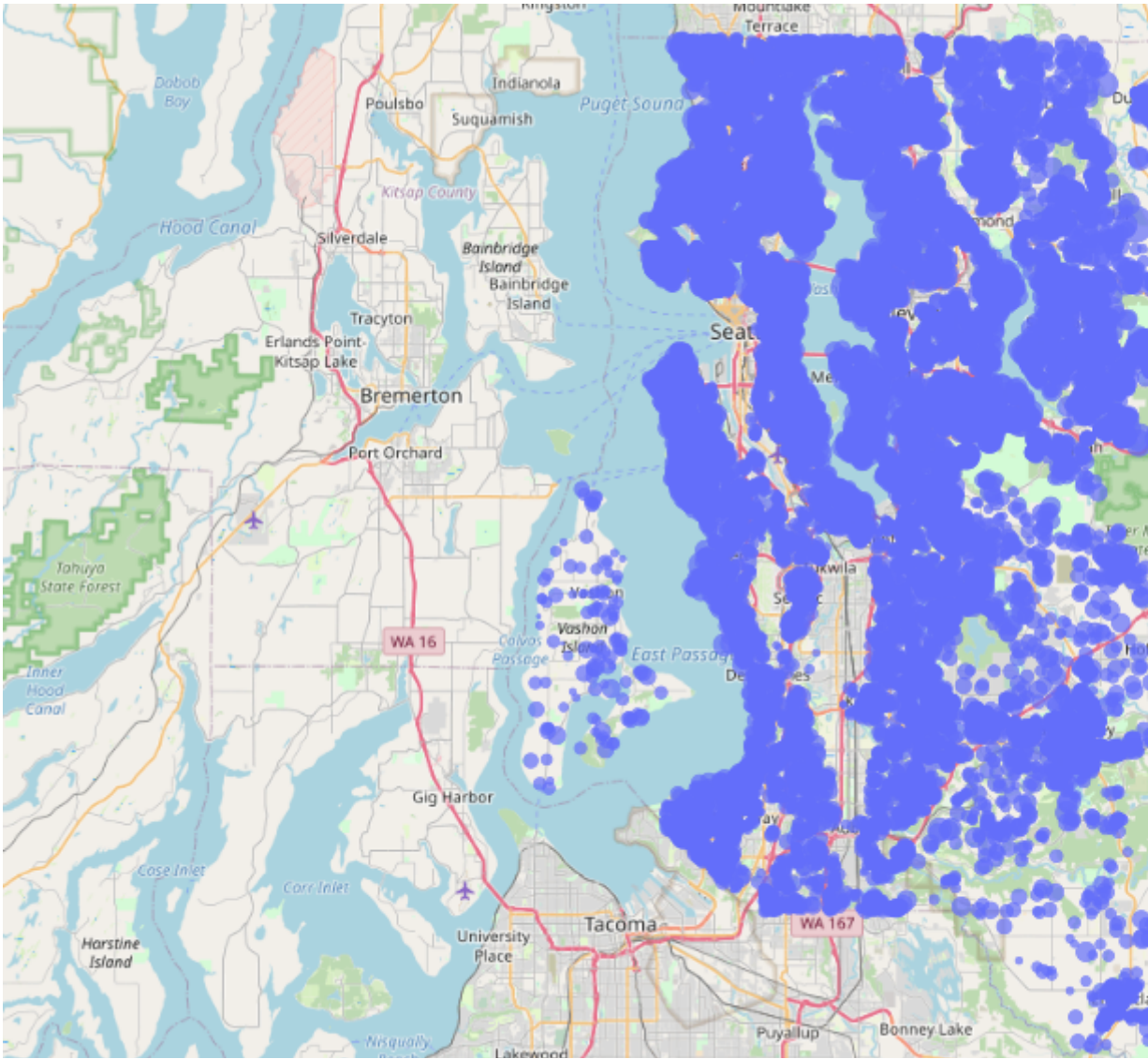
houses = data[['id', 'lat', 'long', 'price', 'sqft_living']].copy()

# # define level of prices
# # houses['level'] = houses['price'].apply(lambda x: 0 if x <= 321950 else
# #                                         #1 if (x > 321950) & (x <
# #                                         #2 if (x > 450000) & (x <

# # convert level to astype
# # houses['level'] = houses['level'].astype(int)

fig = px.scatter_mapbox(houses,
                        lat='lat',
                        lon='long',
                        size='sqft_living',
                        color_continuous_scale=px.colors.cyclical.IceFire,
                        size_max=15,
                        zoom=10)

fig.update_layout(mapbox_style='open-street-map')
fig.update_layout(height=600, margin={'r':0, 'l':0, 't':0, 'b':0})
fig.show()
```



In []: