Jupyter Notebook
Bengaluru_House_Price_Analysis Last Checkpoint: a few seconds ago (autosaved) Current Kernel Logout
Menu

Python 3 (ipykernel)

Not Trusted

- File
  - New NotebookDropdown
    - Python 3 (ipykernel)
  - Open...
  - Make a Copy...
  - Save as...
  - Rename...
  - Save and CheckpointCtrl-S
  - Revert to CheckpointDropdown
    - Monday, June 27, 2022 10:12 AM
  - Print Preview
  - Download asDropdown
    - AsciiDoc (.asciidoc)
    - HTML (.html)
    - LaTeX (.tex)
    - Markdown (.md)
    - Notebook (.ipynb)
    - PDF via LaTeX (.pdf)
    - reST (.rst)
    - Python (.py)
    - Reveal.js slides (.slides.html)
    - PDF via HTML (.html)
  - Deploy as
  - Trust Notebook
  - Close and Halt
- Edit
  - Cut CellsX
  - Copy CellsC
  - Paste Cells AboveShift-V
  - Paste Cells BelowV
  - Paste Cells & Replace
  - Delete CellsD,D
  - Undo Delete CellsZ
  - Split CellCtrl-Shift-Minus
  - Merge Cell Above
  - Merge Cell Below
  - Move Cell Up
  - Move Cell Down
  - Edit Notebook Metadata
  - Find and Replace
  - Cut Cell Attachments
  - Copy Cell Attachments
  - Paste Cell Attachments
  - Insert Image
- View
  - Toggle Header
  - Toggle Toolbar
  - Toggle Line NumbersShift-L
  - Cell Toolbar
    - None
    - Edit Metadata
    - Raw Cell Format
    - Slideshow
    - Attachments
    - Tags
- Insert
  - Insert Cell AboveA
  - Insert Cell BelowB
- Cell
  - Run CellsCtrl-Enter
  - Run Cells and Select BelowShift-Enter
  - Run Cells and Insert BelowAlt-Enter
  - Run All
  - Run All Above
  - Run All Below
  - Cell Type
    - CodeY
    - MarkdownM
    - Raw NBConvertR
  - Current Outputs
    - ToggleO
    - Toggle ScrollingShift-O
    - Clear
  - All Output
    - Toggle
    - Toggle Scrolling
    - Clear
- Kernel
  - InterruptI,I
  - Restart0,0
  - Restart & Clear Output
  - Restart & Run All
  - Reconnect
  - Shutdown
  - Change kernel
    - Python 3 (ipykernel)
- Widgets
  - Save Notebook Widget State
  - Clear Notebook Widget State
  - Download Widget State
  - Embed Widgets
- Help
  - User Interface Tour
  - Keyboard ShortcutsH
  - Edit Keyboard Shortcuts
  - Notebook Help
  - Markdown
  - Python Reference
  - IPython Reference
  - NumPy Reference
  - SciPy Reference
  - Matplotlib Reference
  - SymPy Reference
  - pandas Reference
  - About

Run
Code

In [7]:

```
import pandas as pd

import numpy as np

from matplotlib import pyplot as plt

%matplotlib inline

import matplotlib

matplotlib.rcParams["figure.figsize"] = (20,10)
```

In [8]:

```
df1 = pd.read_csv(r'C:\Users\jadut3\project-work\Bengaluru_House_Data.csv')

df1.head()
```

Out[8]:

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.00

In [9]:

```
df1.groupby('area_type')['area_type'].agg('count')
```

Out[9]:

```
area_type
Built-up  Area          2418
Carpet  Area              87
Plot  Area              2025
Super built-up  Area    8790
Name: area_type, dtype: int64
```

In [10]:

```
df2 = df1.drop(['area_type','society','balcony','availability'],axis='columns')

df2.head()
```

Out[10]:

	location	size	total_sqft	bath	price
0	Electronic City Phase II	2 BHK	1056	2.0	39.07
1	Chikka Tirupathi	4 Bedroom	2600	5.0	120.00
2	Uttarahalli	3 BHK	1440	2.0	62.00
3	Lingadheeranahalli	3 BHK	1521	3.0	95.00
4	Kothanur	2 BHK	1200	2.0	51.00

In [25]:

```
df2.isnull().sum()
```

Out[25]:

```
location       1
size          16
total_sqft     0
bath          73
price          0
dtype: int64
```

In [11]:

```
df3 = df2.dropna()

df3.isnull().sum()
```

Out[11]:

```
location      0
size          0
total_sqft    0
bath          0
price         0
dtype: int64
```

In [27]:

```
df3.shape
```

Out[27]:

```
(13246, 5)
```

In [28]:

```
df3['size'].unique()
```

Out[28]:

```
array(['2 BHK', '4 Bedroom', '3 BHK', '4 BHK', '6 Bedroom', '3 Bedroom',
       '1 BHK', '1 RK', '1 Bedroom', '8 Bedroom', '2 Bedroom',
       '7 Bedroom', '5 BHK', '7 BHK', '6 BHK', '5 Bedroom', '11 BHK',
       '9 BHK', '9 Bedroom', '27 BHK', '10 Bedroom', '11 Bedroom',
       '10 BHK', '19 BHK', '16 BHK', '43 Bedroom', '14 BHK', '8 BHK',
       '12 Bedroom', '13 BHK', '18 Bedroom'], dtype=object)
```

In [14]:

```
x

df3['bhk'] = df3['size'].apply(lambda x: int(x.split(' ')[0]))
```

```
C:\Users\jadut3\AppData\Local\Temp\ipykernel_6836\2222900254.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df3['bhk'] = df3['size'].apply(lambda x: int(x.split(' ')[0]))
```

In [15]:

```
df3.head(10)
```

Out[15]:

	location	size	total_sqft	bath	price	bhk
0	Electronic City Phase II	2 BHK	1056	2.0	39.07	2
1	Chikka Tirupathi	4 Bedroom	2600	5.0	120.00	4
2	Uttarahalli	3 BHK	1440	2.0	62.00	3
3	Lingadheeranahalli	3 BHK	1521	3.0	95.00	3
4	Kothanur	2 BHK	1200	2.0	51.00	2
5	Whitefield	2 BHK	1170	2.0	38.00	2
6	Old Airport Road	4 BHK	2732	4.0	204.00	4
7	Rajaji Nagar	4 BHK	3300	4.0	600.00	4
8	Marathahalli	3 BHK	1310	3.0	63.25	3
9	Gandhi Bazar	6 Bedroom	1020	6.0	370.00	6

In [16]:

```
df3['bhk'].unique()
```

Out[16]:

```
array([ 2,  4,  3,  6,  1,  8,  7,  5, 11,  9, 27, 10, 19, 16, 43, 14, 12,
       13, 18], dtype=int64)
```

In [17]:

```
df3[df3.bhk>20]
```

Out[17]:

	location	size	total_sqft	bath	price	bhk
1718	2Electronic City Phase II	27 BHK	8000	27.0	230.0	27
4684	Munnekollal	43 Bedroom	2400	40.0	660.0	43

In [18]:

```
df3.total_sqft.unique()
```

Out[18]:

```
array(['1056', '2600', '1440', ..., '1133 - 1384', '774', '4689'],
      dtype=object)
```

In [27]:

```
def is_float(x):

    try:

        float(x)

    except:

        return False

    return True
```

In [28]:

```
df3[~df3['total_sqft'].apply(is_float)].head(10)
```

Out[28]:

	location	size	total_sqft	bath	price	bhk
30	Yelahanka	4 BHK	2100 - 2850	4.0	186.000	4
122	Hebbal	4 BHK	3067 - 8156	4.0	477.000	4
137	8th Phase JP Nagar	2 BHK	1042 - 1105	2.0	54.005	2
165	Sarjapur	2 BHK	1145 - 1340	2.0	43.490	2
188	KR Puram	2 BHK	1015 - 1540	2.0	56.800	2
410	Kengeri	1 BHK	34.46Sq. Meter	1.0	18.500	1
549	Hennur Road	2 BHK	1195 - 1440	2.0	63.770	2
648	Arekere	9 Bedroom	4125Perch	9.0	265.000	9
661	Yelahanka	2 BHK	1120 - 1145	2.0	48.130	2
672	Bettahalsoor	4 Bedroom	3090 - 5002	4.0	445.000	4

In [29]:

```
x

def convert_sqft_to_num(x):

    tokens = x.split('-')

    if len(tokens) == 2:

        return (float(tokens[0])+float(tokens[1]))/2

    try:

        return float(x)

    except:

        return None
```

In [30]:

```
xxxxxxxxxx


df4=df3.copy()

df4['total_sqft']=df4['total_sqft'].apply(convert_sqft_to_num)

df4.head(3)
```

Out[30]:

	location	size	total_sqft	bath	price	bhk
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3

In [31]:

```
xxxxxxxxxx


df4.loc[30]
```

Out[31]:

```
location      Yelahanka
size              4 BHK
total_sqft       2475.0
bath                4.0
price             186.0
bhk                   4
Name: 30, dtype: object
```

In [32]:

```
xxxxxxxxxx


df5=df4.copy()

df5['price_per_sqft']=df5['price']*100000/df5['total_sqft']

df5.head()
```

Out[32]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	3699.810606
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615.384615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305.555556
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245.890861
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250.000000

In [36]:

```
len(df5.location.unique())
```

Out[36]:

```
1304
```

In [40]:

```
df5.location = df5.location.apply(lambda x: x.strip())



location_stats = df5.groupby('location')['location'].agg('count').sort_values(ascending=False)

location_stats
```

Out[40]:

```
location
Whitefield               535
Sarjapur  Road           392
Electronic City          304
Kanakpura Road           266
Thanisandra              236
                        ...
1 Giri Nagar               1
Kanakapura Road,           1
Kanakapura main  Road      1
Karnataka Shabarimala      1
whitefiled                 1
Name: location, Length: 1293, dtype: int64
```

In [41]:

```
x

len(location_stats[location_stats<=10])
```

Out[41]:

```
1052
```

In [45]:

```
x

location_stats_less_than_10 = location_stats[location_stats<=10]

location_stats_less_than_10
```

Out[45]:

```
location
Basapura                 10
1st Block Koramangala    10
Gunjur Palya             10
Kalkere                  10
Sector 1 HSR Layout      10
                         ..
1 Giri Nagar              1
Kanakapura Road,          1
Kanakapura main  Road     1
Karnataka Shabarimala     1
whitefiled                1
Name: location, Length: 1052, dtype: int64
```

In [43]:

```
len(df5.location.unique())
```

Out[43]:

```
1293
```

In [46]:

```
xxxxxxxxxx


df5.location = df5.location.apply(lambda x: 'other' if x in location_stats_less_than_10 else x)

len(df5.location.unique())
```

Out[46]:

```
242
```

In [47]:

```
x

df5[df5.total_sqft/df5.bhk<300].head()
```

Out[47]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
9	other	6 Bedroom	1020.0	6.0	370.0	6	36274.509804
45	HSR Layout	8 Bedroom	600.0	9.0	200.0	8	33333.333333
58	Murugeshpalya	6 Bedroom	1407.0	4.0	150.0	6	10660.980810
68	Devarachikkanahalli	8 Bedroom	1350.0	7.0	85.0	8	6296.296296
70	other	3 Bedroom	500.0	3.0	100.0	3	20000.000000

In [48]:

```
df5.shape
```

Out[48]:

```
(13246, 7)
```

In [49]:

```
df6=df5[~(df5.total_sqft/df5.bhk<300)]

df6.shape
```

Out[49]:

```
(12502, 7)
```

In [50]:

```
xxxxxxxxxx


df6.price_per_sqft.describe()
```

Out[50]:

```
count     12456.000000
mean       6308.502826
std        4168.127339
min         267.829813
25%        4210.526316
50%        5294.117647
75%        6916.666667
max      176470.588235
Name: price_per_sqft, dtype: float64
```

In [59]:

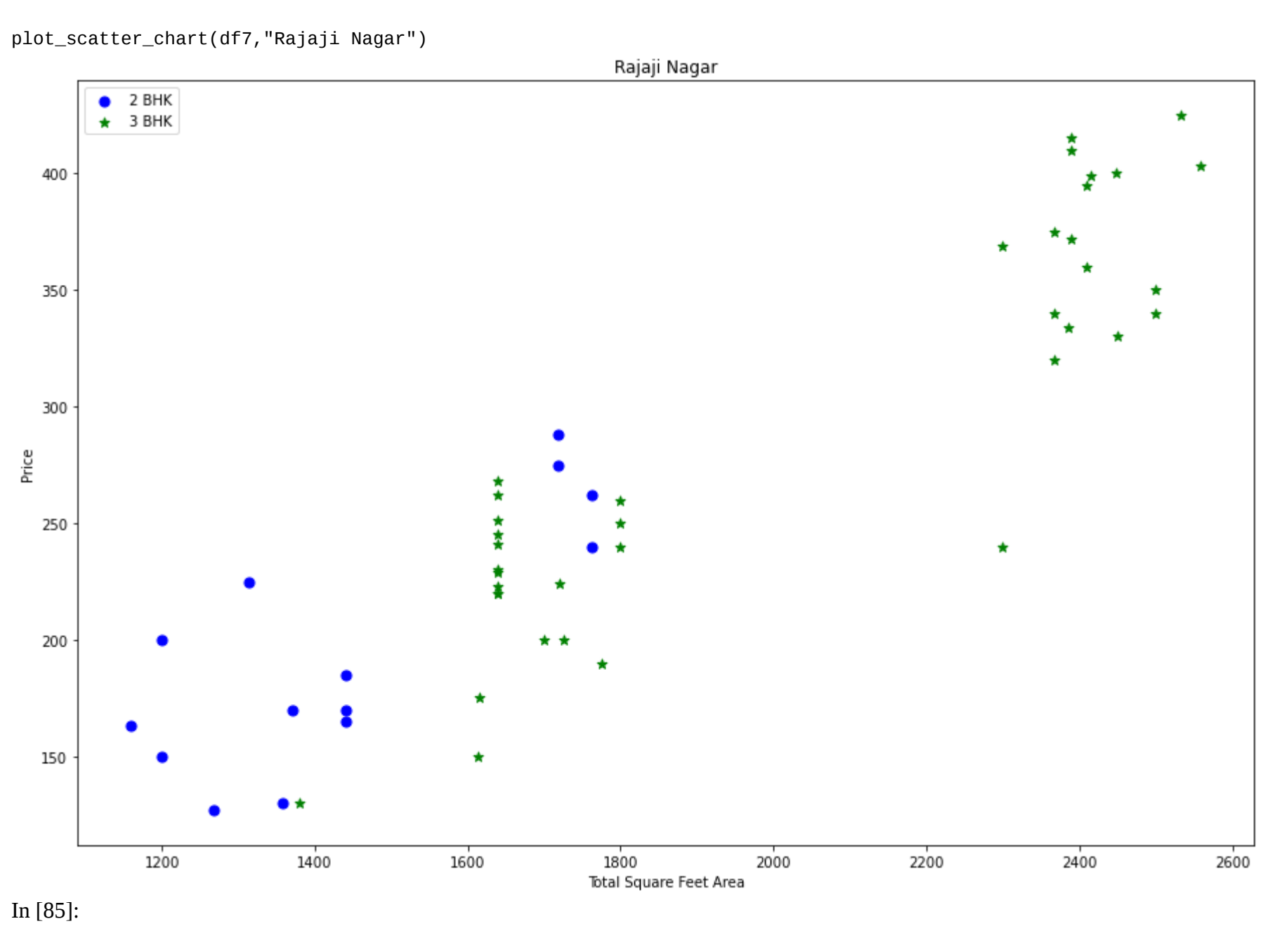
```
def remove_pps_outliers(df):

    df_out = pd.DataFrame()

    for key,subdf in df.groupby('location'):

        m = np.mean(subdf.price_per_sqft)

        st = np.std(subdf.price_per_sqft)

        reduced_df = subdf[(subdf.price_per_sqft>(m-st)) & (subdf.price_per_sqft<=(m+st))]

        df_out = pd.concat([df_out,reduced_df],ignore_index=True)

    return df_out


df7 = remove_pps_outliers(df6)

df7.shape
```

Out[59]:

```
(10241, 7)
```

In [82]:

```
def plot_scatter_chart(df,location):

    bhk2 = df[(df.location==location) & (df.bhk==2)]

    bhk3 = df[(df.location==location) & (df.bhk==3)]

    matplotlib.rcParams['figure.figsize'] = (15,10)

    plt.scatter(bhk2.total_sqft,bhk2.price,color='blue',label='2 BHK', s=50)

    plt.scatter(bhk3.total_sqft,bhk3.price,marker='*', color='green',label='3 BHK', s=50)

    plt.xlabel('Total Square Feet Area')

    plt.ylabel('Price')

    plt.title(location)

    plt.legend()


plot_scatter_chart(df7,"Rajaji Nagar")
```

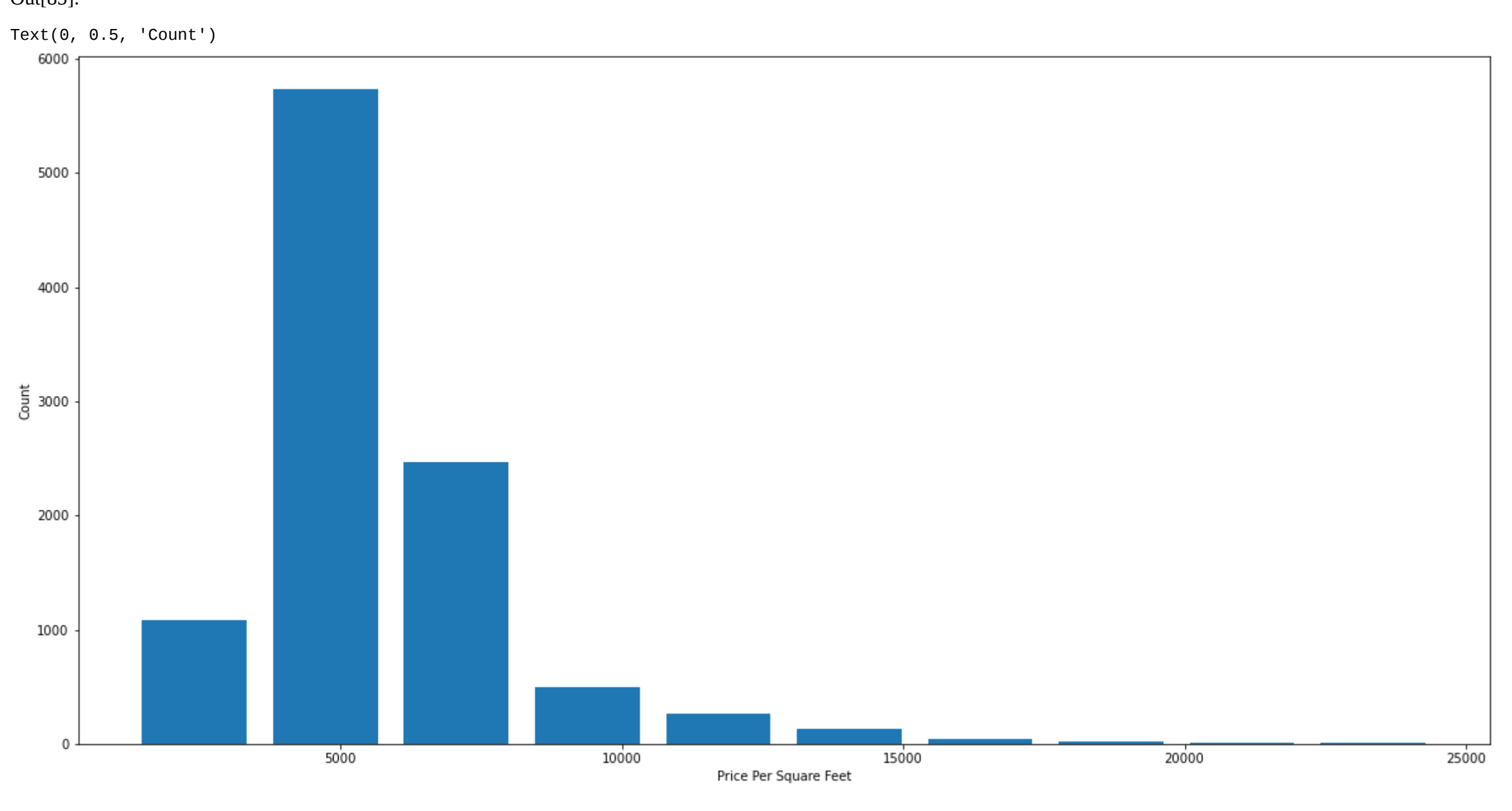


In [85]:

```
import matplotlib

matplotlib.rcParams["figure.figsize"] = (20,10)

plt.hist(df7.price_per_sqft,rwidth=0.8)

plt.xlabel("Price Per Square Feet")

plt.ylabel("Count")
```

Out[85]:

```
Text(0, 0.5, 'Count')
```



In [ ]: