IMPORT ALL REQUIRED LIBRARIES

In [ ]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

LOAD DATASET

In [3]:
```python
df = pd.read_csv('Bengaluru_House_Data.csv')
df.head()
```

Out[3]:

	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 [4]:
```python
df.shape
```

Out[4]:
```
(13320, 9)
```

In [5]:
```python
df.isnull().mean()*100
```

Out[5]:
```
area_type        0.000000
availability     0.000000
location         0.007508
size             0.120120
society         41.306306
total_sqft       0.000000
bath             0.548048
balcony          4.572072
price            0.000000
dtype: float64
```

In [6]:
```python
df['area_type'].value_counts()
```

Out[6]:
```
Super built-up  Area    8790
Built-up  Area          2418
Plot  Area              2025
Carpet  Area              87
Name: area_type, dtype: int64
```

In [7]:
```python
df.drop(columns=["availability","area_type","society","balcony"],axis=1,inplace=True)
```

In [8]:
```python
df.head()
```

Out[8]:

	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 [9]:
```python
df.isnull().sum()
```

Out[9]:
```
location       1
size          16
total_sqft     0
bath          73
price          0
dtype: int64
```

In [10]:
```python
df.dropna(inplace=True)
df.isnull().sum()
```

Out[10]:
```
location      0
size          0
total_sqft    0
bath          0
price         0
dtype: int64
```

In [11]:
```python
df.shape
```

Out[11]:
```
(13246, 5)
```

In [12]:
```python
df.head()
```

Out[12]:

	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 [13]:
```python
df['size'].unique()
```

Out[13]:
```
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]:
```python
df['bhk'] = df['size'].apply(lambda x: int(x.split(' ')[0]))
```

In [15]:
```python
df.head()
```

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

In [ ]:
```python
df.drop(columns=["size"],axis=1,inplace=True)
```

In [17]:
```python
df.shape
```

Out[17]:
```
(13246, 5)
```

In [21]:
```python
df[df.bhk>22]
```

Out[21]:

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

In [22]:
```python
df.total_sqft.unique()
```

Out[22]:
```
array(['1056', '2600', '1440', ..., '1133 - 1384', '774', '4689'],
      dtype=object)
```

In [23]:
```python
def is_float(x):
    try:
        float(x)
    except:
        return False
    return True
```

In [24]:
```python
df[~df['total_sqft'].apply(is_float)].head(10)
```

Out[24]:

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

In [25]:
```python
def convert_sqft_into_number(x):
    token = x.split('-')
    if len(token) == 2:
        return (float(token[0]) + float(token[1])) / 2
    try:
        return float(x)
    except:
        return None
```

In [26]:
```python
df1 = df.copy()

df1['total_sqft'] = df1['total_sqft'].apply(convert_sqft_into_number)
```

In [27]:
```python
df1.loc[30]
```

Out[27]:
```
location      Yelahanka
total_sqft       2475.0
bath                4.0
price             186.0
bhk                   4
Name: 30, dtype: object
```

In [28]:
```python
df2 = df1.copy()
```

In [29]:
```python
df2['price_per_sqft'] = df2['price']*100000 / df2['total_sqft']
df2.head()
```

Out[29]:

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

In [30]:
```python
df2['location'].value_counts()
```

Out[30]:
```
Whitefield           534
Sarjapur  Road       392
Electronic City      302
Kanakpura Road       266
Thanisandra          233
                    ...
Vidyapeeta             1
Maruthi Extension      1
Okalipura              1
Old Town               1
Abshot Layout          1
Name: location, Length: 1304, dtype: int64
```

In [31]:
```python
df2['location'] = df2['location'].apply(lambda x: x.strip())
df2.location.value_counts()
```

Out[31]:
```
Whitefield                        535
Sarjapur  Road                    392
Electronic City                   304
Kanakpura Road                    266
Thanisandra                       236
                                 ...
Vasantapura main road               1
Bapuji Layout                       1
1st Stage Radha Krishna Layout      1
BEML Layout 5th stage               1
Abshot Layout                       1
Name: location, Length: 1293, dtype: int64
```

In [ ]:
```python
location[location<=10]
```

In [ ]:
```python
loc_less_than_10 = loc_stats[loc_stats<=10]
loc_less_than_10
```

In [ ]:
```python
df2.location = df2.location.apply(lambda x: 'other' if x in loc_less_than_10 else x)
df2.head()
```

In [36]:
```python
len(df2.location.unique())
```

Out[36]:
```
1293
```

In [37]:
```python
df2[ (df2.total_sqft / df2.bhk < 300) ].head()
```

Out[37]:

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

In [38]:
```python
df3 = df2[ ~(df2.total_sqft / df2.bhk < 300) ]
df3.shape
```

Out[38]:
```
(12502, 6)
```

In [39]:
```python
df3.price_per_sqft.describe()
```

Out[39]:
```
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
```

CREATE A FUN WHICH REMOVES THE OUTLIERS FROM PRICE_PER_SQFT USING STANDARD DEVIATION TECHNIQUE

In [40]:
```python
def remove_outlier_from_price_per_sqft(df):
    df_out = pd.DataFrame()
    for key,sub in df.groupby('location'):
        m = np.mean( sub.price_per_sqft )
        st = np.std( sub.price_per_sqft )
        reduce_df = sub[( sub.price_per_sqft>(m-st) ) & ( sub.price_per_sqft<=(m+st) ) ]
        df_out = pd.concat( [df_out, reduce_df],ignore_index=True )
    return df_out
```

In [41]:
```python
df4 = remove_outlier_from_price_per_sqft(df3)
df4.shape
```

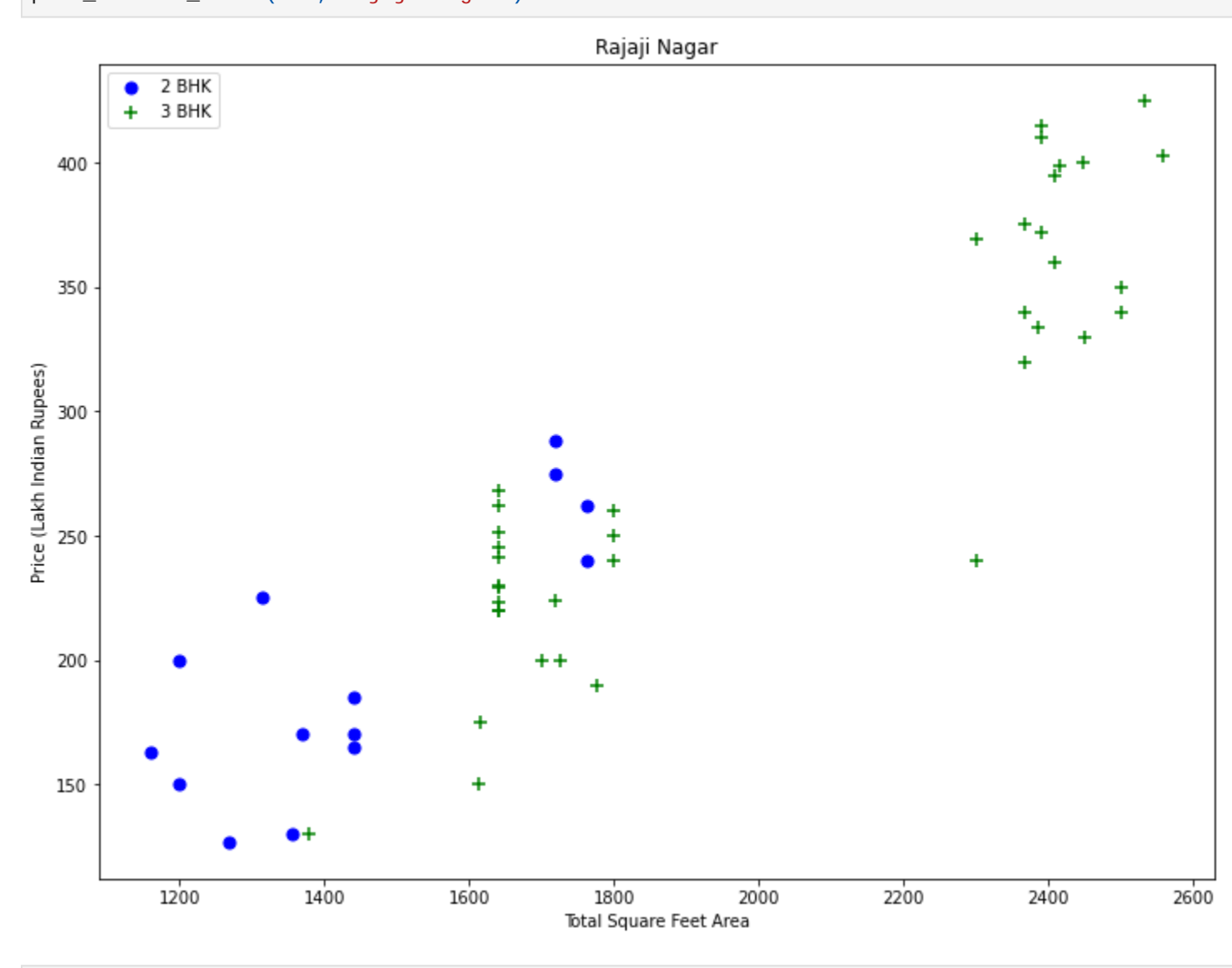
Out[41]:
```
(9267, 6)
```

In [42]:
```python
df4.describe()
```

Out[42]:

	total_sqft	bath	price	bhk	price_per_sqft
count	9267.000000	9267.000000	9267.000000	9267.000000	9267.000000
mean	1504.603273	2.463149	94.140067	2.556707	5724.437494
std	894.614947	0.952020	110.642802	0.847092	2535.271814
min	300.000000	1.000000	10.000000	1.000000	1250.000000
25%	1109.000000	2.000000	49.000000	2.000000	4259.259259
50%	1282.000000	2.000000	67.000000	2.000000	5185.185185
75%	1650.000000	3.000000	100.000000	3.000000	6404.356285
max	30400.000000	14.000000	2912.000000	10.000000	35000.000000

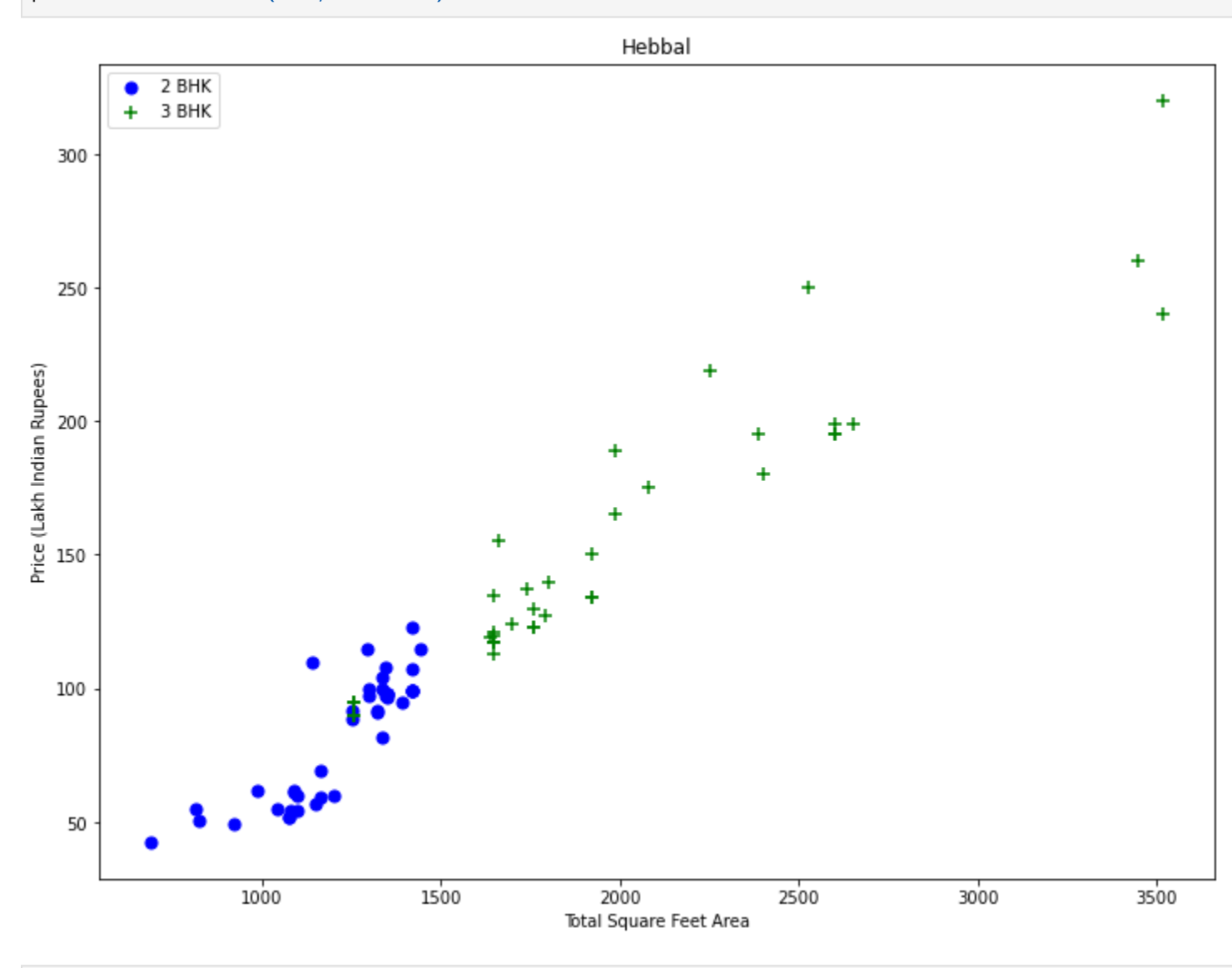
In [43]:
```python
def plot_scatter_chart(df,location):
    bhk2 = df[(df.location==location) & (df.bhk==2)]
    bhk3 = df[(df.location==location) & (df.bhk==3)]
    plt.rcParams['figure.figsize'] = (12,9)
    plt.scatter(bhk2.total_sqft,bhk2.price,color='blue',label='2 BHK', s=50)
    plt.scatter(bhk3.total_sqft,bhk3.price,marker='+', color='green',label='3 BHK', s=50)
    plt.xlabel("Total Square Feet Area")
    plt.ylabel("Price (Lakh Indian Rupees)")
    plt.title(location)
    plt.legend()

plot_scatter_chart(df4,"Rajaji Nagar")
```



In [44]:
```python
def remove_bhk_outliers(df):
    exclude_indices = np.array([])
    for location, location_df in df.groupby('location'):
        bhk_stats = {}
        for bhk, bhk_df in location_df.groupby('bhk'):
            bhk_stats[bhk] = {
                'mean': np.mean(bhk_df.price_per_sqft),
                'std': np.std(bhk_df.price_per_sqft),
                'count': bhk_df.shape[0]
            }
        for bhk, bhk_df in location_df.groupby('bhk'):
            stats = bhk_stats.get(bhk-1)
            if stats and stats['count']>5:
                exclude_indices = np.append(exclude_indices, bhk_df[bhk_df.price_per_sqft<(stats['mean'])].index.values)
    return df.drop(exclude_indices,axis='index')
df5 = remove_bhk_outliers(df4)
df5.shape
```

Out[44]:
```
(7504, 6)
```

In [45]:
```python
plot_scatter_chart(df5,"Hebbal")
```



In [46]:
```python
df5.bath.unique()
```

Out[46]:
```
array([ 3.,  1.,  4.,  2.,  5.,  8.,  9.,  6., 14.,  7., 12.])
```

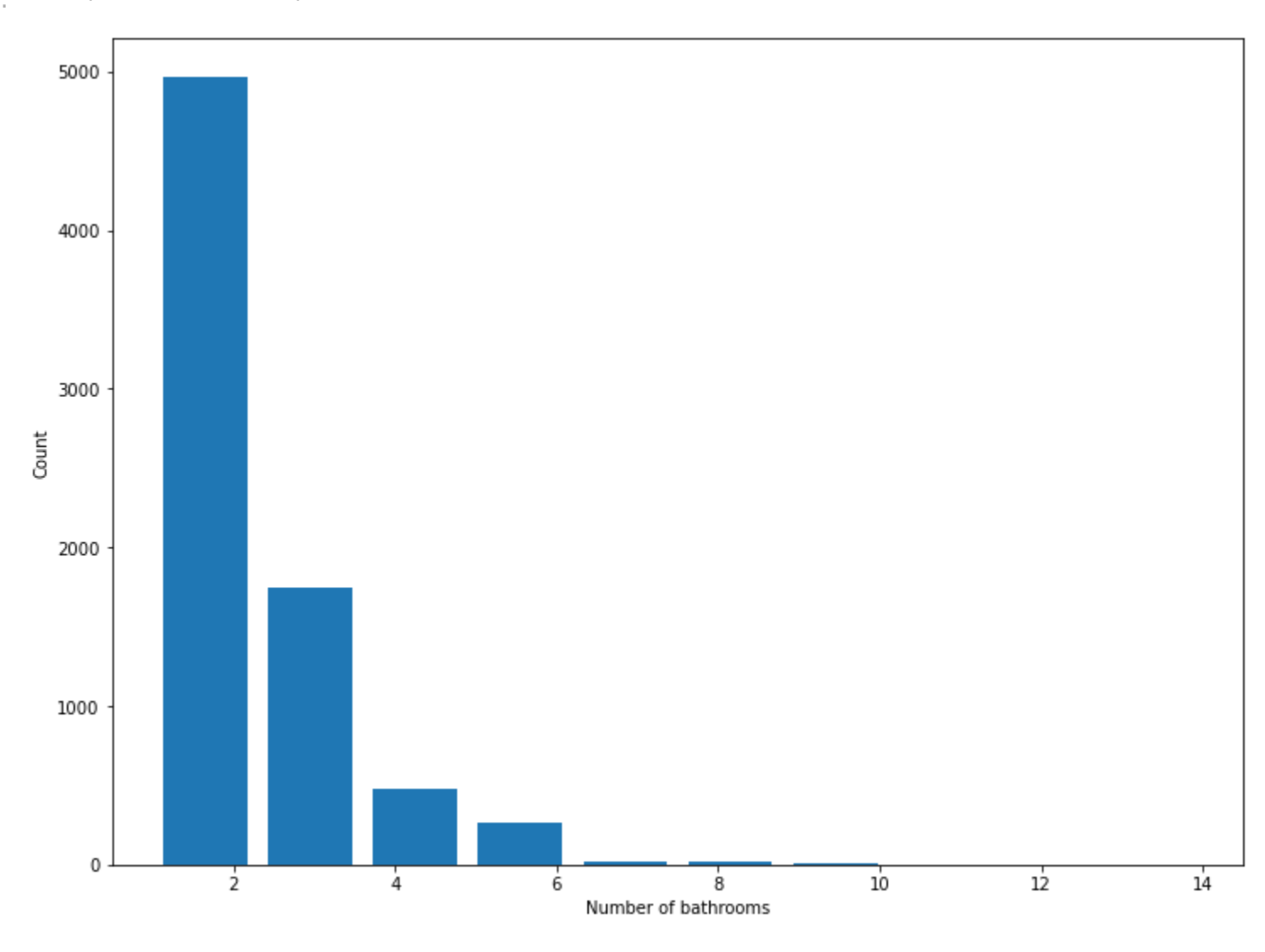
In [47]:
```python
df5[df5.bath>10]
```

Out[47]:

	location	total_sqft	bath	price	bhk	price_per_sqft
757	BTM 1st Stage	3300.0	14.0	500.0	9	15151.515152
6183	Neeladri Nagar	4000.0	12.0	160.0	10	4000.000000

In [48]:
```python
plt.hist(df5.bath,rwidth=0.8)
plt.xlabel("Number of bathrooms")
plt.ylabel("Count")
```

Out[48]:
```
Text(0, 0.5, 'Count')
```



In [49]:
```python
df5[(df5.bath > df5.bhk+2)]
```

Out[49]:

	location	total_sqft	bath	price	bhk	price_per_sqft
757	BTM 1st Stage	3300.0	14.0	500.0	9	15151.515152
1951	Chikkabanavar	2460.0	7.0	80.0	4	3252.032520
6117	Nagasandra	7000.0	8.0	450.0	4	6428.571429
7431	Sathya Sai Layout	11338.0	9.0	1000.0	6	8819.897689
7914	Thanisandra	1806.0	6.0	116.0	3	6423.034330

In [50]:
```python
df6 = df5[~(df5.bath > df5.bhk+2)]
df6.head()
```

Out[50]:

	location	total_sqft	bath	price	bhk	price_per_sqft
0	1st Block BEL Layout	1540.0	3.0	85.0	3	5519.480519
1	1st Block HBR Layout	600.0	1.0	45.0	1	7500.000000
2	1st Block HBR Layout	3150.0	4.0	150.0	4	4761.904762
3	1st Block HRBR Layout	2300.0	3.0	80.0	3	3478.260870
4	1st Block HRBR Layout	1250.0	2.0	67.0	2	5360.000000

In [51]:
```python
df6.shape
```

Out[51]:
```
(7499, 6)
```

In [52]:
```python
df7 = df6.drop(['price_per_sqft'],axis='columns')
df7.head()
```

Out[52]:

	location	total_sqft	bath	price	bhk
0	1st Block BEL Layout	1540.0	3.0	85.0	3
1	1st Block HBR Layout	600.0	1.0	45.0	1
2	1st Block HBR Layout	3150.0	4.0	150.0	4
3	1st Block HRBR Layout	2300.0	3.0	80.0	3
4	1st Block HRBR Layout	1250.0	2.0	67.0	2

In [53]:
```python
dummies = pd.get_dummies(df7.location)
dummies.head()
```

Out[53]:

	1st Block BEL Layout	1st Block HBR Layout	1st Block HRBR Layout	1st Block Jayanagar	1st Block Koramangala	1st Phase JP Nagar	1st Stage Indira Nagar	2nd Block Hrbr Layout	2nd Block Jayanagar	2nd Phase JP Nagar	...	Yelahanka	Yelahanka New Town	Yelenahalli	Yemlur	Yeshwanthpur	Yeshwanthpur Industrial Suburb	frazertown	manyata park	south	tc.palya
0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 761 columns

In [56]:
```python
df8 = pd.concat([df7,dummies.drop('south',axis='columns')],axis='columns')
df8.head()
```

Out[56]:

	location	total_sqft	bath	price	bhk	1st Block BEL Layout	1st Block HBR Layout	1st Block HRBR Layout	1st Block Jayanagar	1st Block Koramangala	...	Yelachenahalli	Yelahanka	Yelahanka New Town	Yelenahalli	Yemlur	Yeshwanthpur	Yeshwanthpur Industrial Suburb	frazertown	manyata park	tc.palya
0	1st Block BEL Layout	1540.0	3.0	85.0	3	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	1st Block HBR Layout	600.0	1.0	45.0	1	0	1	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	1st Block HBR Layout	3150.0	4.0	150.0	4	0	1	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	1st Block HRBR Layout	2300.0	3.0	80.0	3	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0	0
4	1st Block HRBR Layout	1250.0	2.0	67.0	2	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 765 columns

In [57]:
```python
df8.drop('location',axis='columns',inplace=True)
df8.head()
```

Out[57]:

	total_sqft	bath	price	bhk	1st Block BEL Layout	1st Block HBR Layout	1st Block HRBR Layout	1st Block Jayanagar	1st Block Koramangala	1st Phase JP Nagar	...	Yelachenahalli	Yelahanka	Yelahanka New Town	Yelenahalli	Yemlur	Yeshwanthpur	Yeshwanthpur Industrial Suburb	frazertown	manyata park	tc.palya
0	1540.0	3.0	85.0	3	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	600.0	1.0	45.0	1	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	3150.0	4.0	150.0	4	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	2300.0	3.0	80.0	3	0	0	1	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	1250.0	2.0	67.0	2	0	0	1	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 764 columns

In [58]:
```python
df8.shape
```

Out[58]:
```
(7499, 764)
```

In [59]:
```python
x = df8.drop('price',axis=1)
y = df8['price']
```

In [60]:
```python
x.shape
```

Out[60]:
```
(7499, 763)
```

In [61]:
```python
y.shape
```

Out[61]:
```
(7499,)
```

In [62]:
```python
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x,y,test_size=0.2,random_state=101)
```

In [63]:
```python
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

Out[63]:
```
((5999, 763), (1500, 763), (5999,), (1500,))
```

In [64]:
```python
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train,y_train)
lr.score(X_test,y_test)
```

Out[64]:
```
-3.731396464478837e+16
```

In [65]:
```python
pred = lr.predict(X_test)
pred
```

Out[65]:
```
array([618.82104038,  56.24883686,  37.98836656, ...,  58.07789129,
       382.94811557,  41.1765158 ])
```

In [66]:
```python
y_test
```

Out[66]:
```
6450    600.00
5073     52.00
4683     37.00
4721     50.00
5673     86.58
         ...
3516    118.00
1985     59.00
340      65.00
4359     95.00
3813     46.00
Name: price, Length: 1500, dtype: float64
```

In [67]:
```python
def predict_price(location,sqft,bath,bhk):
    loc_index = np.where(x.columns==location)[0][0]

    X = np.zeros(len(x.columns))
    X[0] = sqft
    X[1] = bath
    X[2] = bhk
    if loc_index >= 0:
        X[loc_index] = 1

    return lr.predict([X])[0]
```

In [68]:
```python
predict_price('1st Phase JP Nagar',1000, 2, 2)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[68]:
```
86.16924039955907
```

In [69]:
```python
predict_price('1st Phase JP Nagar',1000, 2, 3)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[69]:
```
78.39152829457444
```

In [70]:
```python
predict_price('Indira Nagar',1400, 2, 3)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[70]:
```
197.2135035555047
```

In [ ]: