```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import matplotlib
matplotlib.rcParams["figure.figsize"] = (20,10)
```

```
df1 = pd.read_csv("bengaluru_house_prices.csv")
df1.head()
```

```
              area_type   availability                  location
size  \
0  Super built-up  Area         19-Dec  Electronic City Phase II
2 BHK
1            Plot  Area  Ready To Move          Chikka Tirupathi  4
Bedroom
2        Built-up  Area  Ready To Move               Uttarahalli
3 BHK
3  Super built-up  Area  Ready To Move        Lingadheeranahalli
3 BHK
4  Super built-up  Area  Ready To Move                  Kothanur
2 BHK

  society total_sqft  bath  balcony   price
0  Coomee       1056   2.0      1.0   39.07
1  Theanmp      2600   5.0      3.0  120.00
2      NaN      1440   2.0      3.0   62.00
3  Soiewre      1521   3.0      1.0   95.00
4      NaN      1200   2.0      1.0   51.00
```

```
df1.shape
```

```
(13320, 9)
```

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

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

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

```
                   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
```

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

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

```
df3 = df2.dropna()
df3.isnull().sum()
```

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

```
df3.shape
```

```
(13246, 5)
```

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

```
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)
```

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

```
C:\Users\melvi\AppData\Local\Temp\ipykernel_15124\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]))
```

```
df3.head()
```

```
                   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
```

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

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

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

```
                       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
```

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

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

```
def is_float(x):
    try:
        float(x)
    except:
        return False
    return True
```

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

```
              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
```

```
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

df4 = df3.copy()
df4['total_sqft'] = df4['total_sqft'].apply(convert_sqft_to_num)
df4.head(3)
```

```
                   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
```

```
df4.loc[30]
```

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

```
(2100+2850)/2
```

```
2475.0
```

```
df4.head(3)
```

```
                   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
```

```
df5 = df4.copy()
df5['price_per_sqft'] = df5['price']*100000/df5['total_sqft']
df5.head()
```

```
                   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

3        Lingadheeranahalli      3 BHK      1521.0   3.0   95.00    3

4                  Kothanur      2 BHK      1200.0   2.0   51.00    2
```

```
   price_per_sqft
0     3699.810606
1     4615.384615
2     4305.555556
3     6245.890861
4     4250.000000
```

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

```
1304
```

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

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

```
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
```

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

```
1052
```

```
location_stats_less_than_10 = location_stats[location_stats<=10]
location_stats_less_than_10
```

```
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
```

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

```
1293
```

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

```
242
```

```
df5.head(10)
```

```
                   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

3        Lingadheeranahalli      3 BHK      1521.0   3.0   95.00    3

4                  Kothanur      2 BHK      1200.0   2.0   51.00    2

5                Whitefield      2 BHK      1170.0   2.0   38.00    2

6          Old Airport Road      4 BHK      2732.0   4.0  204.00    4

7              Rajaji Nagar      4 BHK      3300.0   4.0  600.00    4

8              Marathahalli      3 BHK      1310.0   3.0   63.25    3

9                     other  6 Bedroom      1020.0   6.0  370.00    6


   price_per_sqft
0     3699.810606
1     4615.384615
2     4305.555556
3     6245.890861
4     4250.000000
5     3247.863248
6     7467.057101
7    18181.818182
8     4828.244275
9    36274.509804
```

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

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

    price_per_sqft
9     36274.509804
45    33333.333333
58    10660.980810
68     6296.296296
70    20000.000000
```

```
df5.shape
```

```
(13246, 7)
```

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

```
(12502, 7)
```

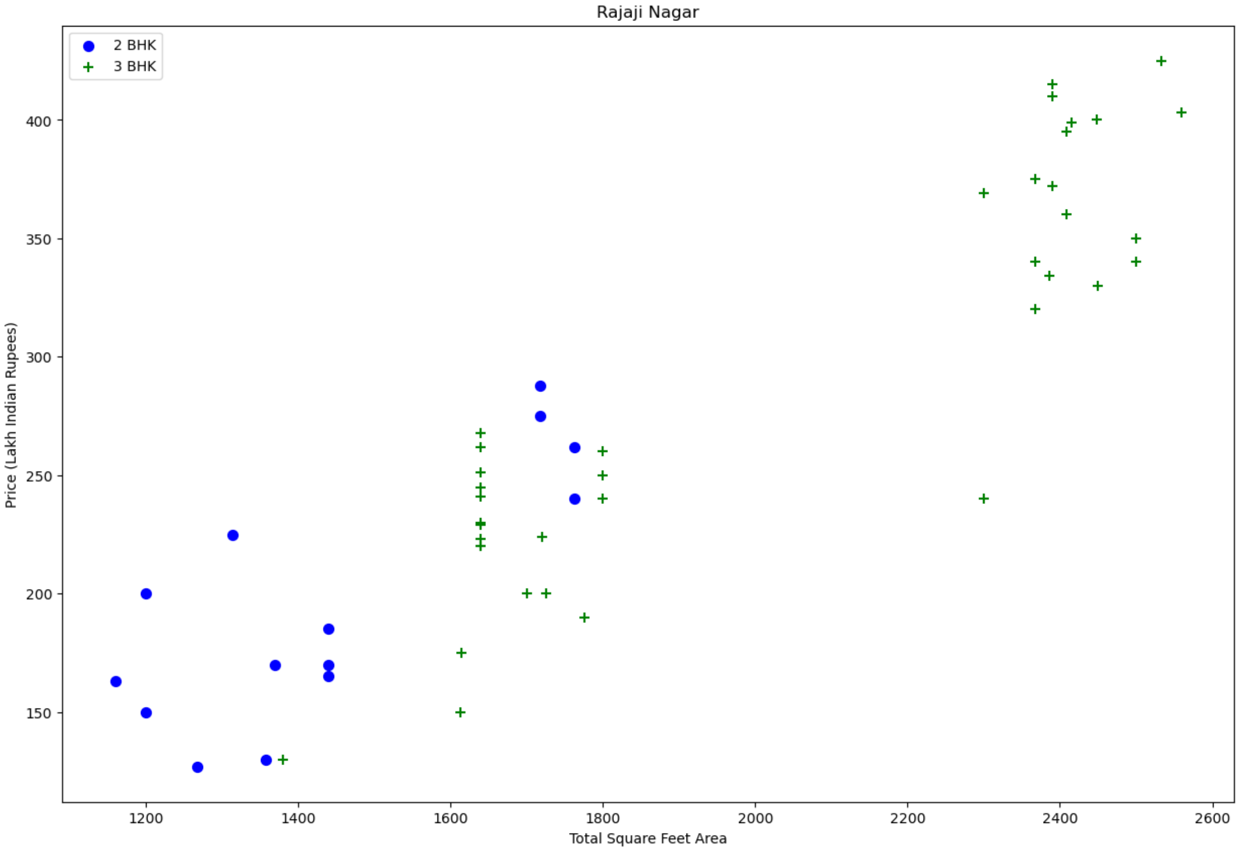
```
df6.price_per_sqft.describe()
```

```
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
```

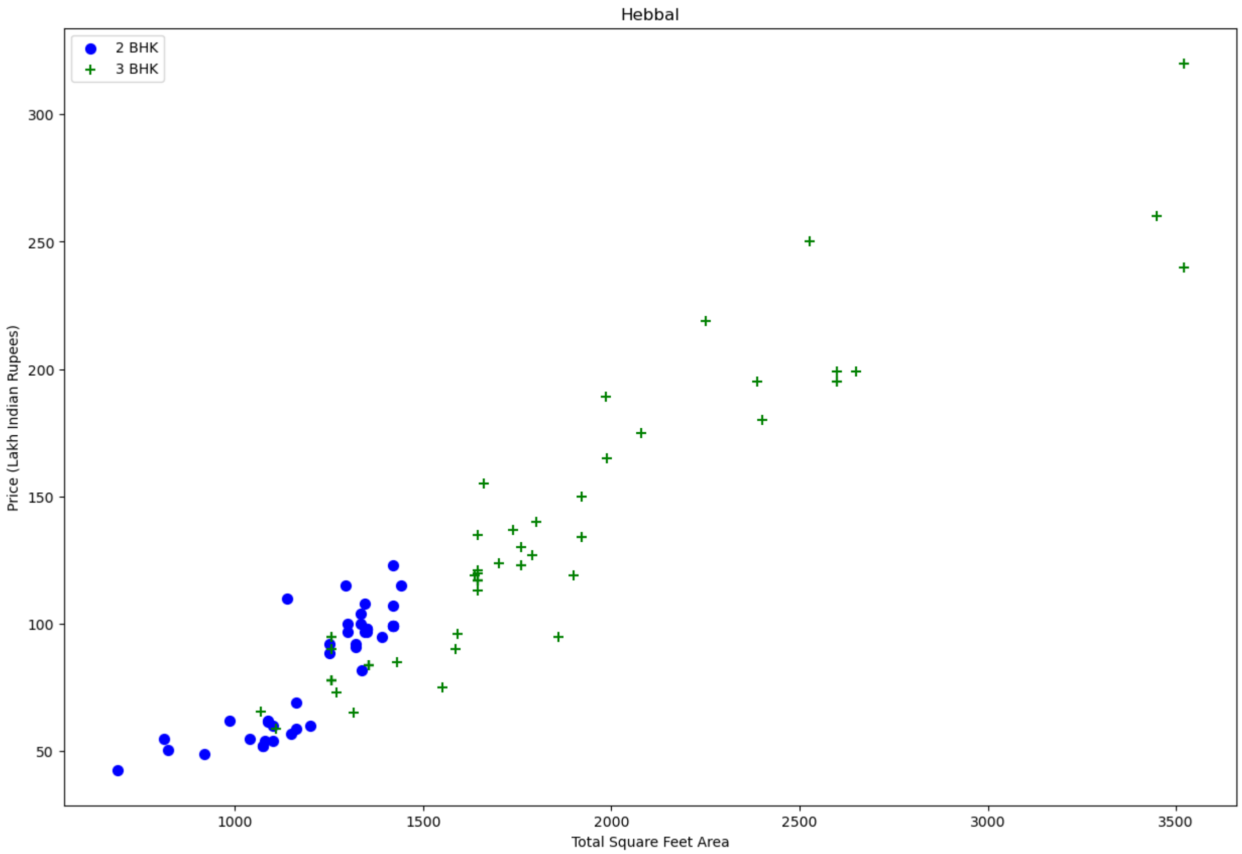
```
def remove_pps_outliers(df):
    df_out = pd.DataFrame()
    for key, subdf in df.groupby('location'):
        m = np.mean(subdf.price_per_sqft)
        st = np.std(subdf.price_per_sqft)
        reduced_df = subdf[(subdf.price_per_sqft>(m-st)) & 
(subdf.price_per_sqft<=(m+st))]
        df_out = pd.concat([df_out,reduced_df],ignore_index=True)
    return df_out
df7 = remove_pps_outliers(df6)
df7.shape
```

```
(10241, 7)
```

```
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 (Lakh Indian Rupees)")
    plt.title(location)
    plt.legend()
plot_scatter_chart(df7,"Rajaji Nagar")
```



```
plot_scatter_chart(df7,"Hebbal")
```



We should also remove properties where for same location, the price of (for example) 3 bedroom apartment is less than 2 bedroom apartment (with same square ft area). What we will do is for a given location, we will build a dictionary of stats per bhk, i.e.

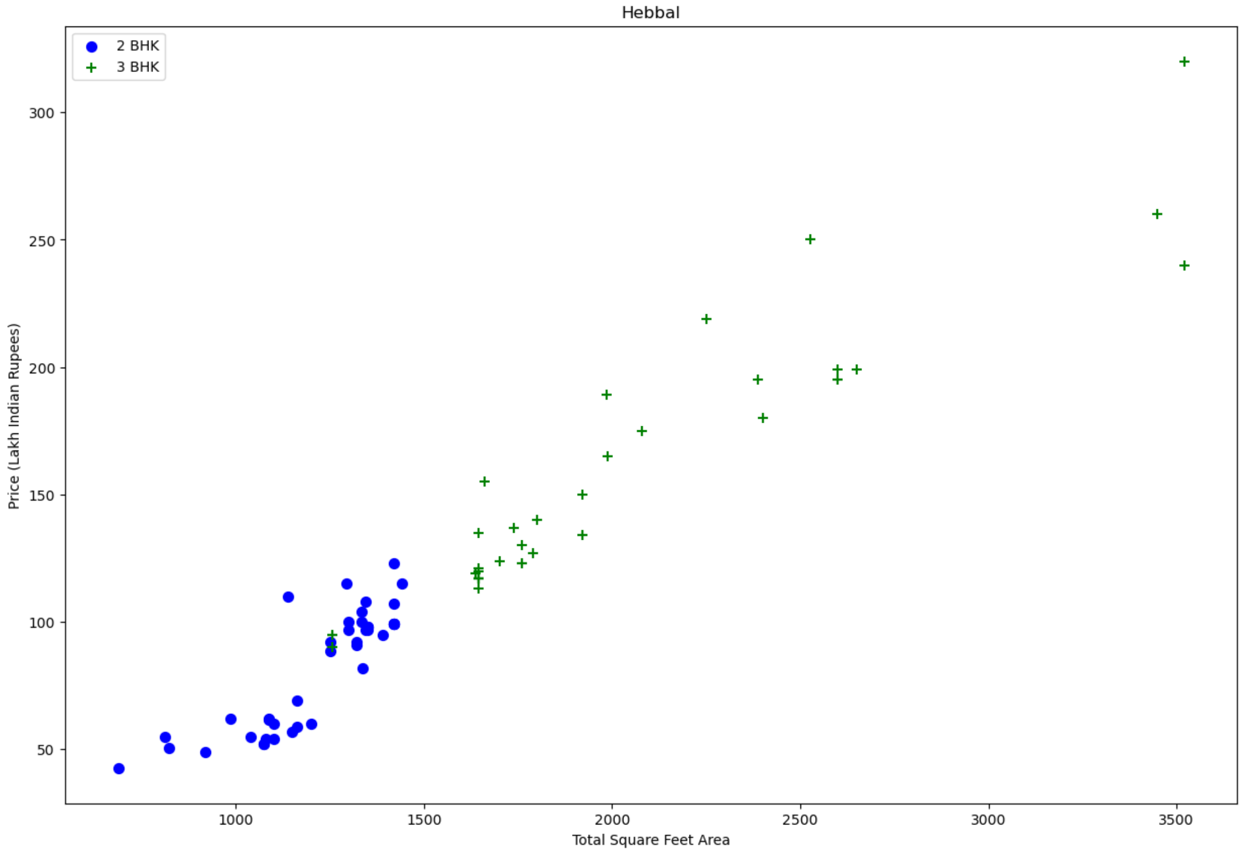
{ '1' : { 'mean': 4000, 'std: 2000, 'count': 34 }, '2' : { 'mean': 4300, 'std: 2300, 'count': 22 }, } Now we can remove those 2 BHK apartments whose price_per_sqft is less than mean price_per_sqft of 1 BHK apartment

```
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')
```

```
df8 = remove_bhk_outliers(df7)
df8.shape
```

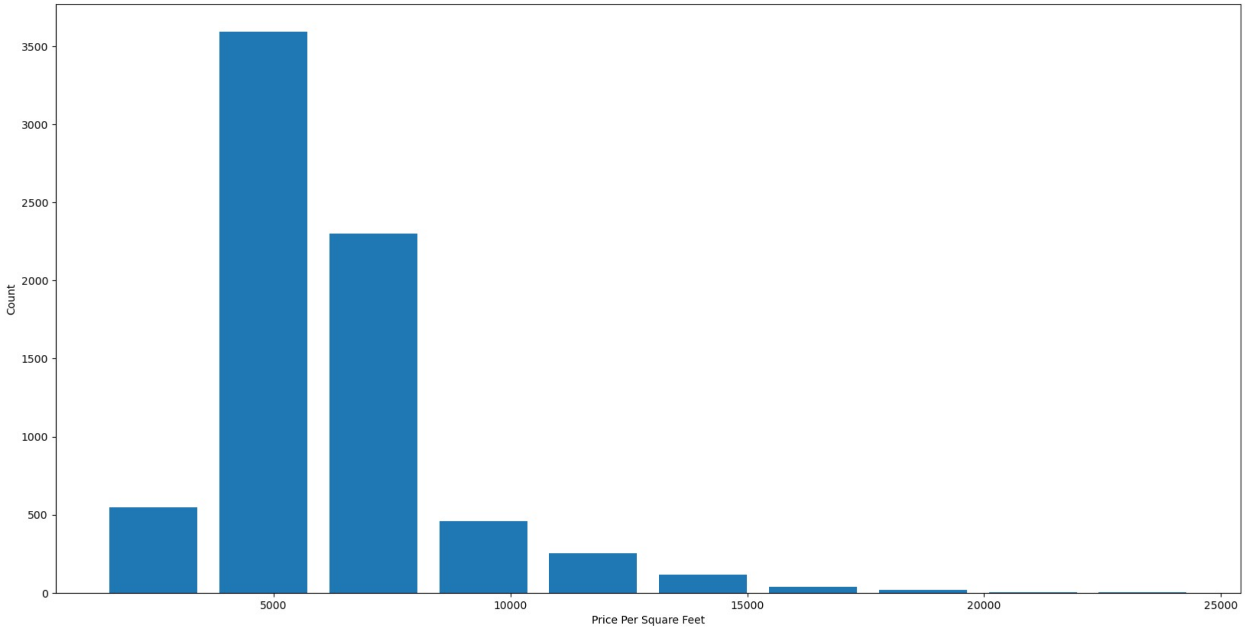
```
(7329, 7)
```

```
plot_scatter_chart(df8,"Hebbal")
```



```
import matplotlib
matplotlib.rcParams["figure.figsize"] = (20,10)
plt.hist(df8.price_per_sqft,rwidth=0.8)
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count")
```

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



```
df8.bath.unique()
```

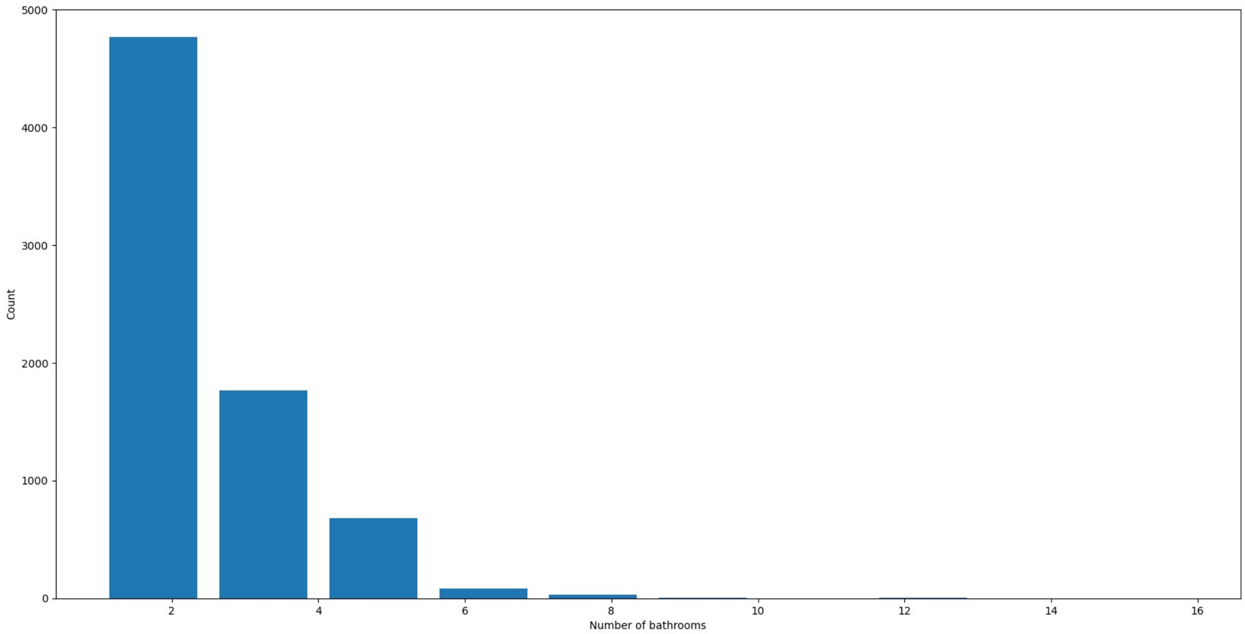
```
array([ 4.,  3.,  2.,  5.,  8.,  1.,  6.,  7.,  9., 12., 16., 13.])
```

```
df8[df8.bath>10]
```

```
            location    size  total_sqft  bath  price  bhk
price_per_sqft
5277  Neeladri Nagar  10 BHK      4000.0  12.0  160.0   10
4000.000000
8486           other  10 BHK     12000.0  12.0  525.0   10
4375.000000
8575           other  16 BHK     10000.0  16.0  550.0   16
5500.000000
9308           other  11 BHK      6000.0  12.0  150.0   11
2500.000000
9639           other  13 BHK      5425.0  13.0  275.0   13
5069.124424
```

```
plt.hist(df8.bath,rwidth=0.8)
plt.xlabel("Number of bathrooms")
plt.ylabel("Count")
```

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



```
df8[df8.bath>df8.bhk+2]
```

```
            location       size  total_sqft  bath   price  bhk  
price_per_sqft
1626  Chikkabanavar  4 Bedroom      2460.0   7.0    80.0    4     
3252.032520
5238     Nagasandra  4 Bedroom      7000.0   8.0   450.0    4     
6428.571429
6711    Thanisandra      3 BHK      1806.0   6.0   116.0    3     
6423.034330
8411          other      6 BHK     11338.0   9.0  1000.0    6     
8819.897689
```

```
df9 = df8[df8.bath<df8.bhk+2]
df9.shape
```

```
(7251, 7)
```

```
df10 = df9.drop(['size','price_per_sqft'],axis='columns')
df10.head(3)
```

```
               location  total_sqft  bath  price  bhk
0  1st Block Jayanagar      2850.0   4.0  428.0    4
1  1st Block Jayanagar      1630.0   3.0  194.0    3
2  1st Block Jayanagar      1875.0   2.0  235.0    3
```

```
dummies = pd.get_dummies(df10.location)
dummies.head(3)
```

```
   1st Block Jayanagar  1st Phase JP Nagar  2nd Phase Judicial Layout  
\
0                    1                   0                          0

1                    1                   0                          0

2                    1                   0                          0


   2nd Stage Nagarbhavi  5th Block Hbr Layout  5th Phase JP Nagar  \
0                     0                     0                   0
1                     0                     0                   0
2                     0                     0                   0

   6th Phase JP Nagar  7th Phase JP Nagar  8th Phase JP Nagar  \
0                   0                   0                   0
1                   0                   0                   0
2                   0                   0                   0

   9th Phase JP Nagar  ...  Vishveshwarya Layout  Vishwapriya
Layout  \
0                   0  ...                     0                   0

1                   0  ...                     0                   0

2                   0  ...                     0                   0


   Vittasandra  Whitefield  Yelachenahalli  Yelahanka  Yelahanka New
Town  \
0            0           0               0          0
0
1            0           0               0          0
0
2            0           0               0          0
0

   Yelenahalli  Yeshwanthpur  other
0            0             0      0
1            0             0      0
2            0             0      0

[3 rows x 242 columns]
```

```
df11 =
pd.concat([df10,dummies.drop('other',axis='columns')],axis='columns')
df11.head()
```

```
              location  total_sqft  bath  price  bhk  1st Block
Jayanagar  \
0  1st Block Jayanagar      2850.0   4.0  428.0    4
1
1  1st Block Jayanagar      1630.0   3.0  194.0    3
```

```
1
2  1st Block Jayanagar      1875.0   2.0  235.0    3
1
3  1st Block Jayanagar      1200.0   2.0  130.0    3
1
4  1st Block Jayanagar      1235.0   2.0  148.0    2
1

   1st Phase JP Nagar  2nd Phase Judicial Layout  2nd Stage Nagarbhavi
\
0                   0                          0                     0

1                   0                          0                     0

2                   0                          0                     0

3                   0                          0                     0

4                   0                          0                     0


   5th Block Hbr Layout  ...  Vijayanagar  Vishveshwarya Layout  \
0                     0  ...            0                     0
1                     0  ...            0                     0
2                     0  ...            0                     0
3                     0  ...            0                     0
4                     0  ...            0                     0

   Vishwapriya Layout  Vittasandra  Whitefield  Yelachenahalli
Yelahanka  \
0                   0            0           0               0
0
1                   0            0           0               0
0
2                   0            0           0               0
0
3                   0            0           0               0
0
4                   0            0           0               0
0

   Yelahanka New Town  Yelenahalli  Yeshwanthpur
0                   0            0             0
1                   0            0             0
2                   0            0             0
3                   0            0             0
4                   0            0             0

[5 rows x 246 columns]
```

```
df12 = df11.drop('location',axis='columns')
df12.head(2)
```

```
   total_sqft  bath  price  bhk  1st Block Jayanagar  1st Phase JP
Nagar  \
0      2850.0   4.0  428.0    4                    1
0
1      1630.0   3.0  194.0    3                    1
0

   2nd Phase Judicial Layout  2nd Stage Nagarbhavi  5th Block Hbr
Layout  \
0                          0                     0
0
1                          0                     0
0

   5th Phase JP Nagar  ...  Vijayanagar  Vishveshwarya Layout  \
0                   0  ...            0                     0
1                   0  ...            0                     0

   Vishwapriya Layout  Vittasandra  Whitefield  Yelachenahalli
Yelahanka  \
0                   0            0           0               0
0
1                   0            0           0               0
0

   Yelahanka New Town  Yelenahalli  Yeshwanthpur
0                   0            0             0
1                   0            0             0

[2 rows x 245 columns]
```

```
df12.shape
```

```
(7251, 245)
```

```
x = df12.drop('price',axis='columns')
x.head()
```

```
   total_sqft  bath  bhk  1st Block Jayanagar  1st Phase JP Nagar  \
0      2850.0   4.0    4                    1                   0
1      1630.0   3.0    3                    1                   0
2      1875.0   2.0    3                    1                   0
3      1200.0   2.0    3                    1                   0
4      1235.0   2.0    2                    1                   0

   2nd Phase Judicial Layout  2nd Stage Nagarbhavi  5th Block Hbr
Layout  \
0                          0                     0
```

```
0
1                          0                     0
0
2                          0                     0
0
3                          0                     0
0
4                          0                     0
0

   5th Phase JP Nagar  6th Phase JP Nagar  ...  Vijayanagar  \
0                   0                   0  ...            0
1                   0                   0  ...            0
2                   0                   0  ...            0
3                   0                   0  ...            0
4                   0                   0  ...            0

   Vishveshwarya Layout  Vishwapriya Layout  Vittasandra
Whitefield  \
0                     0                   0            0           0

1                     0                   0            0           0

2                     0                   0            0           0

3                     0                   0            0           0

4                     0                   0            0           0


   Yelachenahalli  Yelahanka  Yelahanka New Town  Yelenahalli
Yeshwanthpur
0               0          0                   0            0
0
1               0          0                   0            0
0
2               0          0                   0            0
0
3               0          0                   0            0
0
4               0          0                   0            0
0

[5 rows x 244 columns]
```

```
y = df12.price
y.head()
```

```
0    428.0
1    194.0
2    235.0
```

```
3    130.0
4    148.0
Name: price, dtype: float64
```

```
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=10)
```

```
from sklearn.linear_model import LinearRegression
lr_clf = LinearRegression()
lr_clf.fit(x_train,y_train)
lr_clf.score(x_train,y_train)
```

```
0.8541850010771193
```

```
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import cross_val_score
cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)
cross_val_score(LinearRegression(),x, y, cv=cv)
```

```
array([0.82430186, 0.77166234, 0.85089567, 0.80837764, 0.83653286])
```

```
from sklearn.model_selection import GridSearchCV

from sklearn.linear_model import Lasso
from sklearn.tree import DecisionTreeRegressor

def find_best_model_using_gridsearchcv(x,y):
    algos = {
        'linear_regression' : {
            'model': LinearRegression(),
            'params': {
                'fit_intercept': [True, False],
                'copy_X': [True, False],
                'n_jobs': [None],
                'positive': [False]
            }
        },
        'lasso': {
            'model': Lasso(),
            'params': {
                'alpha': [1,2],
                'selection': ['random', 'cyclic']
            }
        },
        'decision_tree': {
            'model': DecisionTreeRegressor(),
            'params': {
                'criterion' : ['mse','friedman_mse'],
                'splitter': ['best','random']
            }
        }
    }
    scores = []
    cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)
    for algo_name, config in algos.items():
        gs =  GridSearchCV(config['model'], config['params'], cv=cv, 
return_train_score=False)
        gs.fit(x,y)
        scores.append({
            'model': algo_name,
            'best_score': gs.best_score_,
            'best_params': gs.best_params_
        })

    return 
pd.DataFrame(scores,columns=['model','best_score','best_params'])

find_best_model_using_gridsearchcv(x,y)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\
_validation.py:425: FitFailedWarning: 
10 fits failed out of a total of 20.
The score on these train-test partitions for these parameters will be 
set to nan.
If these failures are not expected, you can try to debug them by 
setting error_score='raise'.

Below are more details about the failures:
----------------------------------------------------------------------
----------
10 fits failed with the following error:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\
model_selection\_validation.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", 
line 1144, in wrapper
    estimator._validate_params()
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", 
line 637, in _validate_params
    validate_parameter_constraints(
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\
_param_validation.py", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'criterion' 
parameter of DecisionTreeRegressor must be a str among 
{'squared_error', 'poisson', 'friedman_mse', 'absolute_error'}. Got 
'mse' instead.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\
_search.py:976: UserWarning: One or more of the test scores are non-
finite: [       nan        nan 0.7105889  0.74419856]
  warnings.warn(
```

```
               model  best_score  \
0  linear_regression    0.819001
1              lasso    0.687429
2      decision_tree    0.744199

                                         best_params
0  {'copy_X': True, 'fit_intercept': False, 'n_jo...
1                {'alpha': 1, 'selection': 'cyclic'}
2  {'criterion': 'friedman_mse', 'splitter': 'ran...
```

```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_clf.predict([x])[0]
```

```python
predict_price('1st Phase JP Nagar',1000, 2, 2)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:464:
UserWarning: X does not have valid feature names, but LinearRegression
was fitted with feature names
  warnings.warn(
```

```
83.49904677179237
```

```python
predict_price('1st Phase JP Nagar',1000, 3, 3)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:464:
UserWarning: X does not have valid feature names, but LinearRegression
was fitted with feature names
  warnings.warn(
```

```
86.80519395205847
```

```python
predict_price('Indira Nagar',1000, 2, 2)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:464:
UserWarning: X does not have valid feature names, but LinearRegression
was fitted with feature names
  warnings.warn(
181.2781548400685
```

```python
predict_price('Indira Nagar',1000, 3, 3)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:464: 
UserWarning: X does not have valid feature names, but LinearRegression 
was fitted with feature names
  warnings.warn(
```

```
184.58430202033463
```

```python
import pickle
with open('banglore_home_prices_model.pickle','wb') as f:
    pickle.dump(lr_clf,f)
```

```python
import json
columns = {
    'data_columns' : [col.lower() for col in X.columns]
}
with open("columns.json","w") as f:
    f.write(json.dumps(columns))
```