In [1]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
df=pd.read_csv("C:/Users/FamiAmal/Downloads/house_price (1).csv")
```

In [3]:
```
df
```

Out[3]:

	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
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250
...	...	...	...	...	...	...	...
13195	Whitefield	5 Bedroom	3453.0	4.0	231.00	5	6689
13196	other	4 BHK	3600.0	5.0	400.00	4	11111
13197	Raja Rajeshwari Nagar	2 BHK	1141.0	2.0	60.00	2	5258
13198	Padmanabhanagar	4 BHK	4689.0	4.0	488.00	4	10407
13199	Doddathoguru	1 BHK	550.0	1.0	17.00	1	3090

13200 rows × 7 columns

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13200 entries, 0 to 13199
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   location        13200 non-null  object 
 1   size            13200 non-null  object 
 2   total_sqft      13200 non-null  float64
 3   bath            13200 non-null  float64
 4   price           13200 non-null  float64
 5   bhk             13200 non-null  int64  
 6   price_per_sqft  13200 non-null  int64  
dtypes: float64(3), int64(2), object(2)
memory usage: 722.0+ KB
```

In [5]:
```
df.shape
```

Out[5]:
```
(13200, 7)
```

In [6]:

```
df.isnull().sum()
```

Out[6]:

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

In [7]:

```
df.duplicated().sum()
```

Out[7]:

```
1049
```

In [8]:

```
df.duplicated()
```

Out[8]:

```
0        False
1        False
2        False
3        False
4        False
         ...  
13195    False
13196    False
13197    False
13198    False
13199     True
Length: 13200, dtype: bool
```

In [9]:

```
df1=df.drop_duplicates()
```

In [10]:

```
df1
```

Out[10]:

	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
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250
...	...	...	...	...	...	...	...
13194	Green Glen Layout	3 BHK	1715.0	3.0	112.00	3	6530
13195	Whitefield	5 Bedroom	3453.0	4.0	231.00	5	6689
13196	other	4 BHK	3600.0	5.0	400.00	4	11111
13197	Raja Rajeshwari Nagar	2 BHK	1141.0	2.0	60.00	2	5258
13198	Padmanabhanagar	4 BHK	4689.0	4.0	488.00	4	10407

12151 rows × 7 columns

In [11]:
```
df1.shape[0]
```

Out[11]:
```
12151
```

In [12]:
```
df1.describe()
```

Out[12]:

	total_sqft	bath	price	bhk	price_per_sqft
count	12151.000000	12151.000000	12151.000000	12151.000000	1.215100e+04
mean	1574.846013	2.719941	115.471328	2.827504	8.132642e+03
std	1277.328354	1.372210	154.094133	1.326540	1.112329e+05
min	1.000000	1.000000	8.000000	1.000000	2.670000e+02
25%	1100.000000	2.000000	50.000000	2.000000	4.312000e+03
50%	1290.000000	2.000000	74.000000	3.000000	5.500000e+03
75%	1700.000000	3.000000	123.500000	3.000000	7.461000e+03
max	52272.000000	40.000000	3600.000000	43.000000	1.200000e+07

In [13]:
```
s=(df1.isnull().sum()/df1.shape[0])*100
round(s,2)
```

Out[13]:
```
location          0.0
size              0.0
total_sqft        0.0
bath              0.0
price             0.0
bhk               0.0
price_per_sqft    0.0
dtype: float64
```

In [14]:
```
df1.round()
```

Out[14]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.0	2	3699
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.0	4	4615
2	Uttarahalli	3 BHK	1440.0	2.0	62.0	3	4305
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.0	3	6245
4	Kothanur	2 BHK	1200.0	2.0	51.0	2	4250
...	...	...	...	...	...	...	...
13194	Green Glen Layout	3 BHK	1715.0	3.0	112.0	3	6530
13195	Whitefield	5 Bedroom	3453.0	4.0	231.0	5	6689
13196	other	4 BHK	3600.0	5.0	400.0	4	11111
13197	Raja Rajeshwari Nagar	2 BHK	1141.0	2.0	60.0	2	5258
13198	Padmanabhanagar	4 BHK	4689.0	4.0	488.0	4	10407

12151 rows × 7 columns

In [15]:

```
df1
```

Out[15]:

	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
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250
...	...	...	...	...	...	...	...
13194	Green Glen Layout	3 BHK	1715.0	3.0	112.00	3	6530
13195	Whitefield	5 Bedroom	3453.0	4.0	231.00	5	6689
13196	other	4 BHK	3600.0	5.0	400.00	4	11111
13197	Raja Rajeshwari Nagar	2 BHK	1141.0	2.0	60.00	2	5258
13198	Padmanabhanagar	4 BHK	4689.0	4.0	488.00	4	10407

12151 rows × 7 columns

OUTLIERS

In [16]:

```
df1.skew(numeric_only=True)
```

Out[16]:

```
total_sqft         15.112123
bath                4.214944
price               7.915103
bhk                 4.838129
price_per_sqft    103.902032
dtype: float64
```

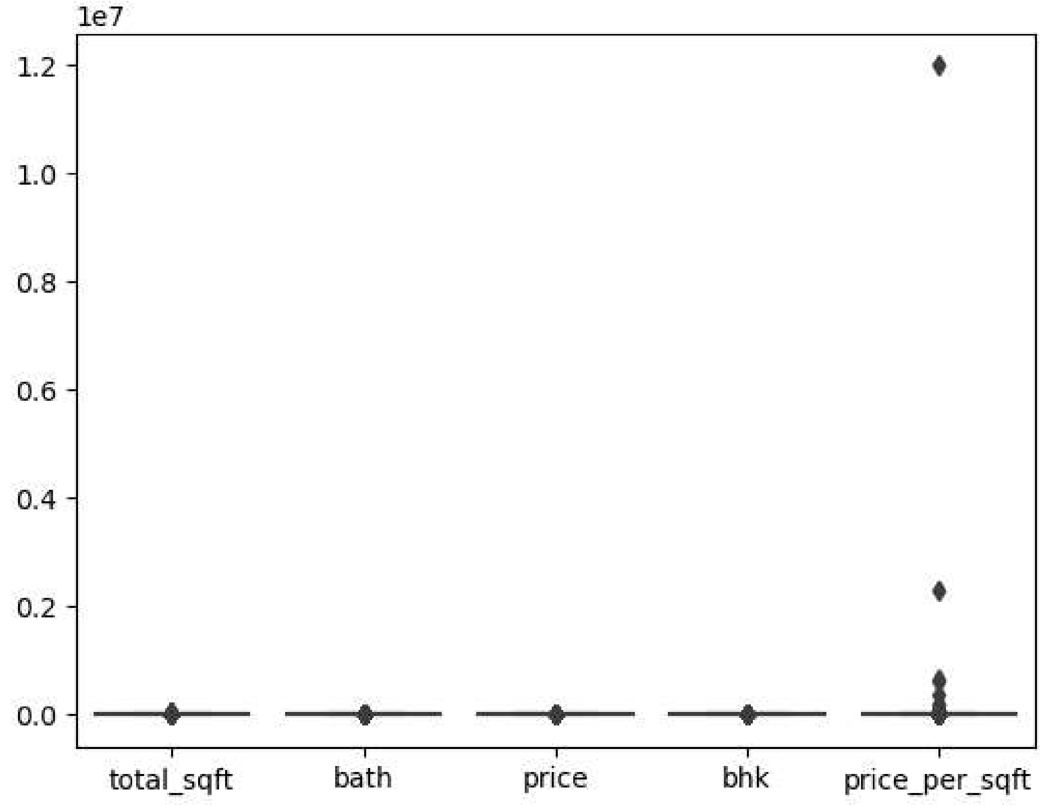
In [17]:

```
numerical_col=df1.select_dtypes(include="number").columns
numerical_col
```

Out[17]:

```
Index(['total_sqft', 'bath', 'price', 'bhk', 'price_per_sqft'], dtype='object')
```

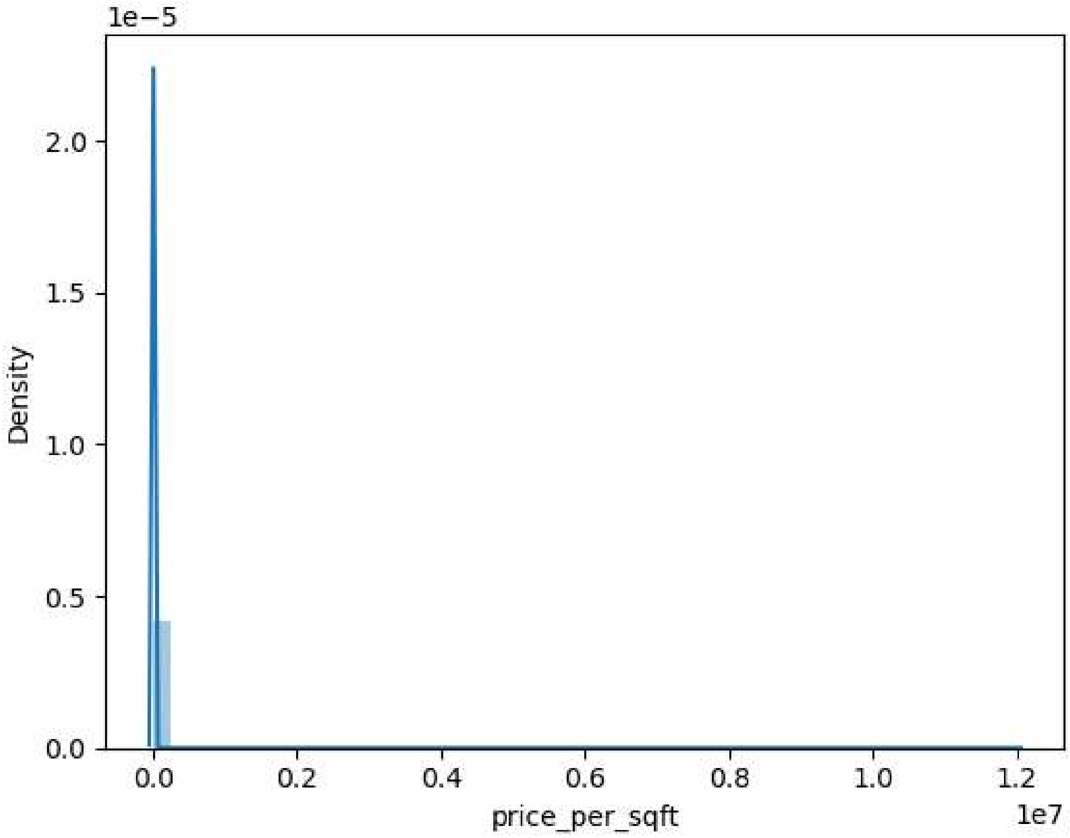
In [18]:
```
sns.boxplot(df1)
plt.show()
```



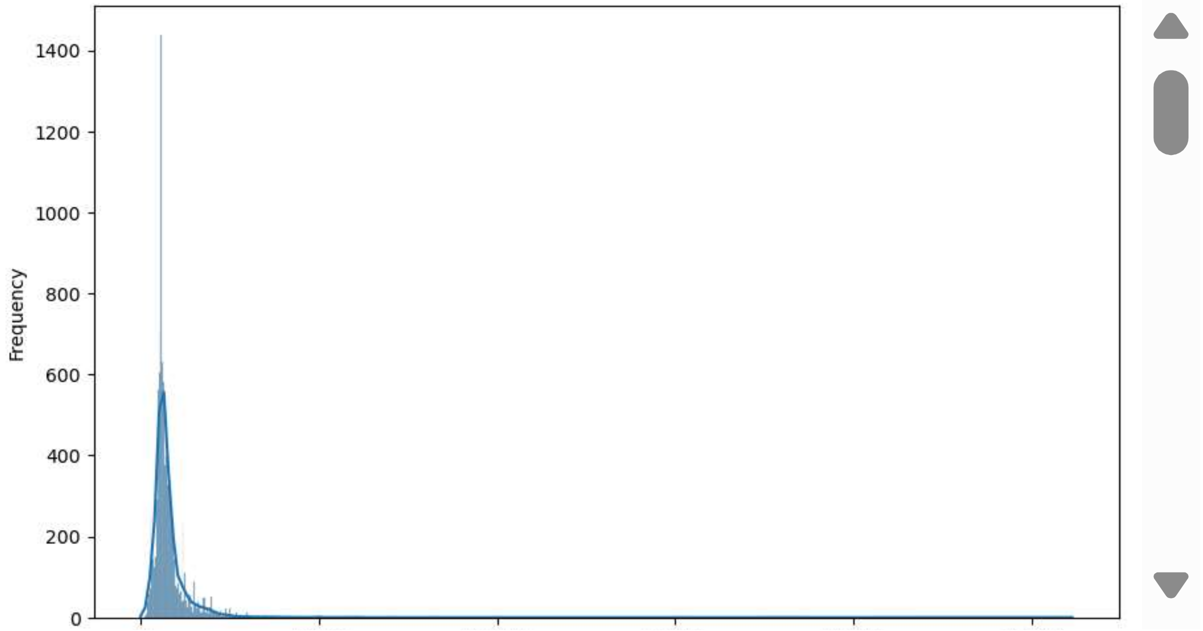
In [19]:
```
sns.distplot(df1["price_per_sqft"])
```

```
C:\Users\FamiAmal\AppData\Local\Temp\ipykernel_12620\1594035437.py:1: UserWarnin
g:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.g
ithub.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df1["price_per_sqft"])
```

Out[19]:
```
<Axes: xlabel='price_per_sqft', ylabel='Density'>
```



In [20]:
```
for i in numerical_col:
    plt.figure(figsize=(10,6))
    sns.histplot(df1[i].dropna(),kde=True)
    plt.xlabel("Column")
    plt.ylabel("Frequency")
    plt.show()
```



METHOD 1 IQR

In [21]:
```
#PRICE_PER_SQFT COLUMN
```

In [22]:
```
df1["price_per_sqft"].skew()
```

Out[22]:
```
103.90203228991889
```

In [26]:
```
q1=df1.price_per_sqft.quantile(0.25)
print("Q1=",q1)
q3=df1.price_per_sqft.quantile(0.75)
print("Q3=",q3)
IQR= q3-q1
print("IQR=",IQR)
```

```
Q1= 4312.0
Q3= 7461.0
IQR= 3149.0
```

In [27]:
```
df1["price_per_sqft"].describe()
```

Out[27]:
```
count    1.215100e+04
mean     8.132642e+03
std      1.112329e+05
min      2.670000e+02
25%      4.312000e+03
50%      5.500000e+03
75%      7.461000e+03
max      1.200000e+07
Name: price_per_sqft, dtype: float64
```

In [30]:
```
lower_whisker=q1-1.5*IQR
upper_whisker=q3+1.5*IQR
lower_whisker,upper_whisker
```

Out[30]:
```
(-411.5, 12184.5)
```

In [31]:
```
 remove_out=df1[(df1.price_per_sqft<lower_whisker)  |(df1.price_per_sqft>upper_wh
```

In [33]:
```
remove_out.skew(numeric_only=True)
```

Out[33]:
```
total_sqft         2.727672
bath               4.828343
price              3.764396
bhk                5.706391
price_per_sqft    31.934933
dtype: float64
```

In [34]:
```
column=df1.select_dtypes(include="number").columns
column
```

Out[34]:
```
Index(['total_sqft', 'bath', 'price', 'bhk', 'price_per_sqft'], dtype='object')
```

In [35]:
```
import pandas as pd
def remove_outliers(df,numerical):
    filterd=df.copy()

    for col in numerical:
        q1=df[col].quantile(0.25)
        q3=df[col].quantile(0.75)
        iqr=q3-q1

        lower_whisker=q1-1.5*iqr
        upper_whisker=q3+1.5*iqr

    filterd = filterd[(filterd[col] >= lower_whisker) & (filterd[col] <= upper_wh


    return filterd
```

In [36]:

```
dff = remove_outliers(df1, ["total_sqft", "bath", "price", "bhk", "price_per_sqft
```

In [37]:

```
dff.skew(numeric_only=True)
```

Out[37]:

```
total_sqft        17.696706
bath               3.166508
price              5.432619
bhk                3.192892
price_per_sqft     0.977840
dtype: float64
```

In [38]:

```
q1=dff.price.quantile(0.25)
q3=dff.price.quantile(0.75)
iqr=q3-q1

lower_whisker=q1-1.5*iqr/100
upper_whisker=q3+1.5*iqr
lower_whisker,upper_whisker

remove_out=dff[(dff.price<lower_whisker)  & (dff.price>upper_whisker)]

remove_out.skew(numeric_only=True)
```

Out[38]:

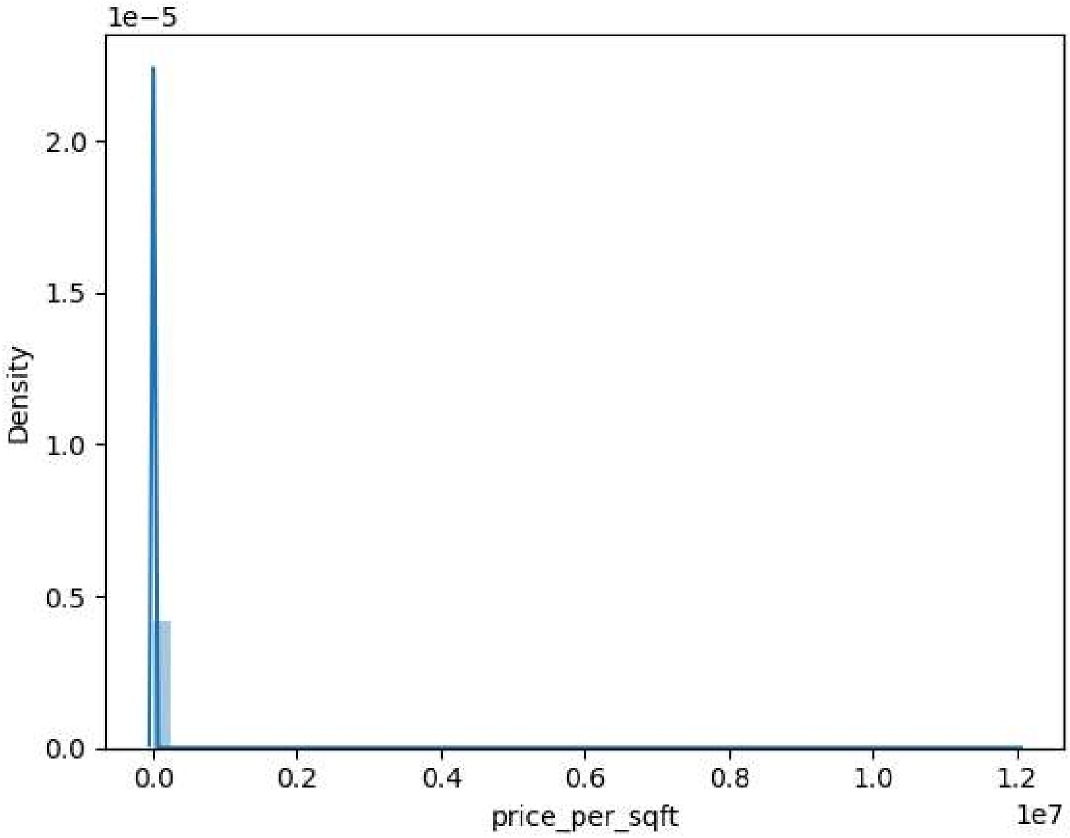
```
total_sqft       NaN
bath             NaN
price            NaN
bhk              NaN
price_per_sqft   NaN
dtype: float64
```

Z- Score Method

```
In [40]: sns.distplot(df1["price_per_sqft"])
```

```
C:\Users\FamiAmal\AppData\Local\Temp\ipykernel_12620\1594035437.py:1: UserWarnin
g:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.g
ithub.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df1["price_per_sqft"])
```

```
Out[40]: <Axes: xlabel='price_per_sqft', ylabel='Density'>
```



```
In [41]: df1["price_per_sqft"].skew()
```

```
Out[41]: 103.90203228991889
```

```
In [47]: upper_limit = df1["price_per_sqft"].mean() + 3 * df1["price_per_sqft"].std()
         lower_limit = df1["price_per_sqft"].mean() - 3 * df1["price_per_sqft"].std()
         print("Upper Limit : " ,upper_limit)
         print("Lower Limit : ",lower_limit)
```

```
Upper Limit :  341831.3445273039
Lower Limit :  -325566.06084694836
```

In [48]:
```
df1.loc[(df1["price_per_sqft"]>upper_limit) |(df1["price_per_sqft"]<lower_limit)]
```

Out[48]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
345	other	3 Bedroom	11.0	3.0	74.0	3	672727
1106	other	5 Bedroom	24.0	2.0	150.0	5	625000
4044	Sarjapur Road	4 Bedroom	1.0	4.0	120.0	4	12000000
4924	other	7 BHK	5.0	7.0	115.0	7	2300000
11447	Whitefield	4 Bedroom	60.0	4.0	218.0	4	363333

In [49]:
```
df1
```

Out[49]:

	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
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250
...	...	...	...	...	...	...	...
13194	Green Glen Layout	3 BHK	1715.0	3.0	112.00	3	6530
13195	Whitefield	5 Bedroom	3453.0	4.0	231.00	5	6689
13196	other	4 BHK	3600.0	5.0	400.00	4	11111
13197	Raja Rajeshwari Nagar	2 BHK	1141.0	2.0	60.00	2	5258
13198	Padmanabhanagar	4 BHK	4689.0	4.0	488.00	4	10407

12151 rows × 7 columns

In [77]:
```
new_df = df1.loc[(df1["price_per_sqft"] <= upper_limit) &(df1["price_per_sqft"] >

print("Old data : ",len(df1))
print("New data : ",len(new_df))
```

```
Old data :  12151
New data :  12146
```

In [78]:
```
new_df=df1.copy()
```

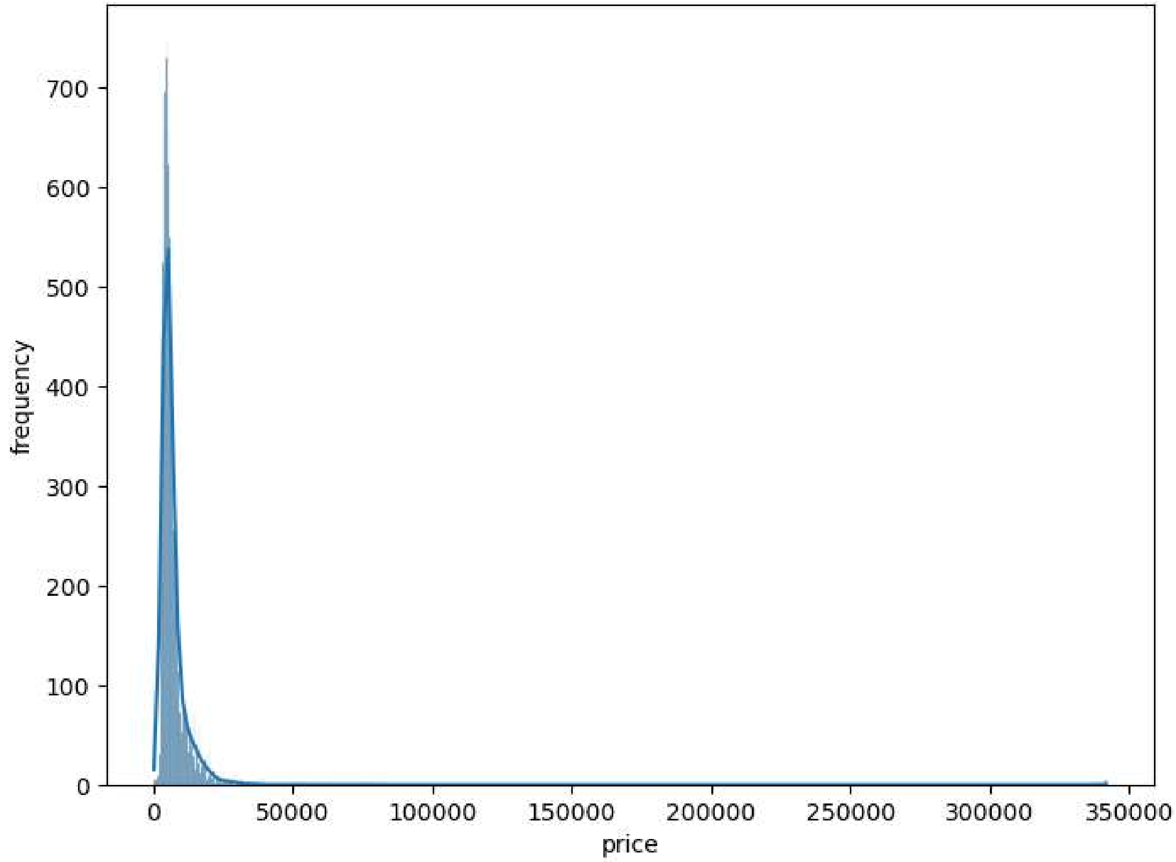
In [79]:
```
new_df.loc[(new_df["price_per_sqft"]>=upper_limit),"price_per_sqft"]=upper_limit
new_df.loc[(new_df["price_per_sqft"]<=lower_limit),"price_per_sqft"]=lower_limit
```

In [80]:
```
len(new_df)
```

Out[80]:
```
12151
```

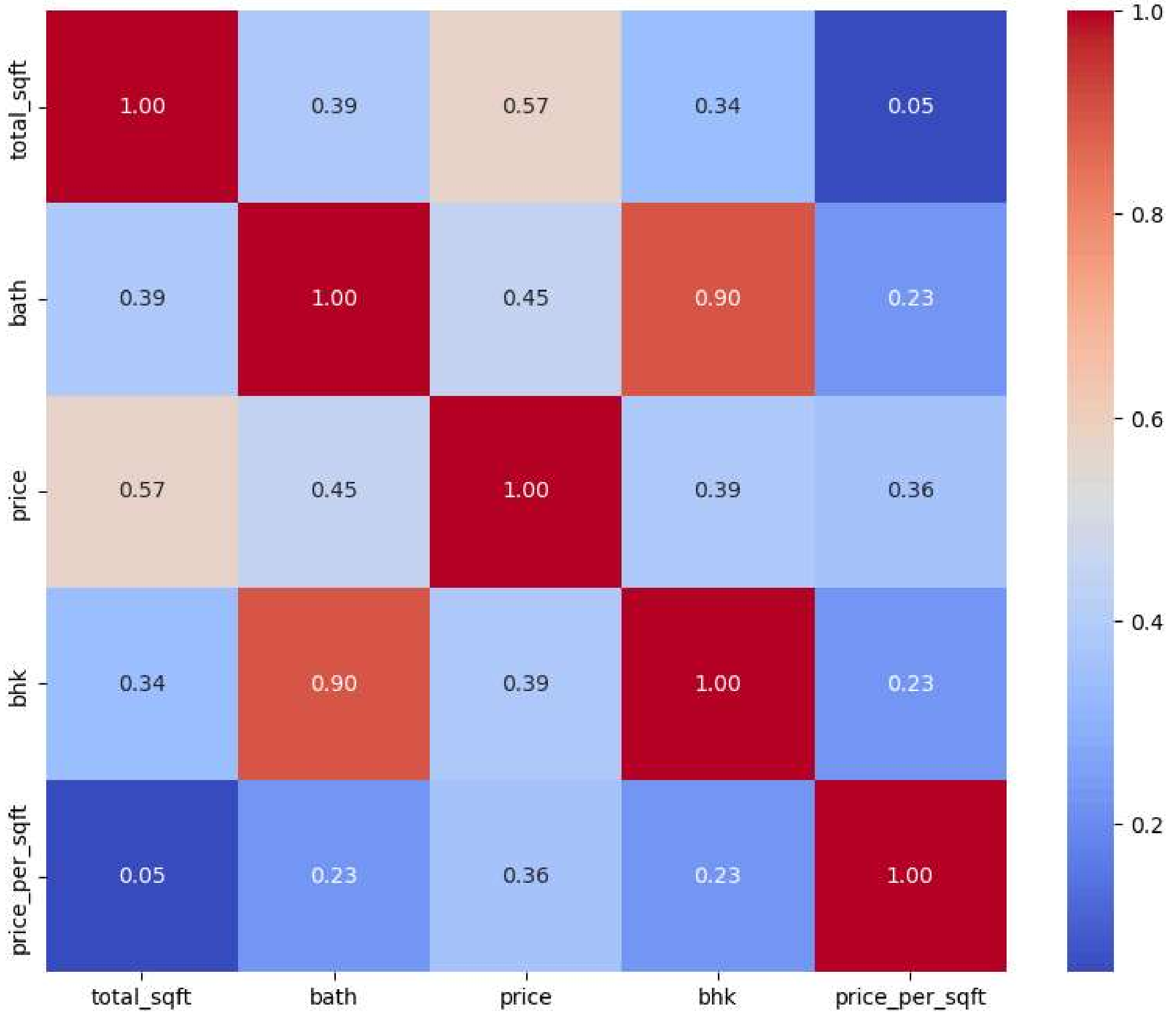
In [81]:

```
plt.figure(figsize=(8,6))
sns.histplot(new_df["price_per_sqft"],kde=True)

plt.xlabel("price")
plt.ylabel("frequency")
plt.show()
```



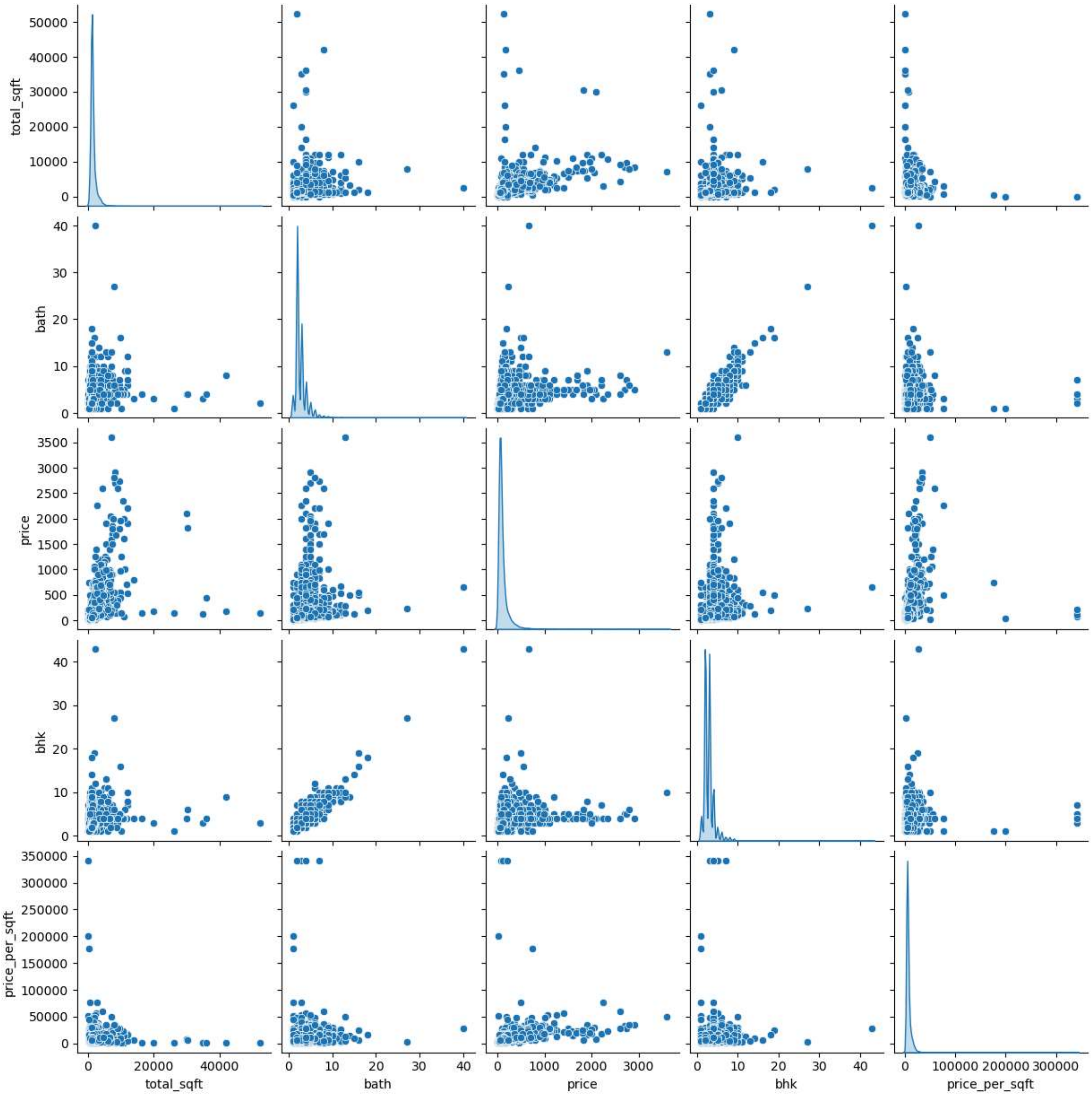
In [82]:

```
numeric_columns=new_df.select_dtypes(include=["number"])
correlation_metrix=numeric_columns.corr()
plt.figure(figsize=(10,8))
sns.heatmap(correlation_metrix,annot=True,cmap="coolwarm",fmt=".2f", square=True)
plt.show()
```



In [83]:
```
sns.pairplot(new_df,diag_kind="kde")
plt.show()
```

```
C:\Users\FamiAmal\.anaconda\annaconda for me\Lib\site-packages\seaborn\axisgrid.
py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



In [ ]:
```

```