Formative Assessment (Statistics)

You are given house_price.csv which contains property prices in the city of Bangalore. You need to examine price per square feet do the following:

Q1. Perform basic EDA
Q2. Detect the outliers using following methods and remove it using methods like trimming / capping/ imputation using mean or median
a) Mean and Standard deviation
b)Percentile method
c) IQR(Inter quartile range method)
d) Z Score method

Q3. Create a box plot and use this to determine which method seems to work best to remove outliers for this data?

Q4. Draw histplot to check the normality of the column(price per sqft column) and perform transformations if needed. Check the skewness and kurtosis before and after the transformation.
Q5. Check the correlation between all the numerical columns and plot heatmap.
Q6. Draw Scatter plot between the variables to check the correlation between them.

Formative Assessment (Statistics)

[2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
from scipy.stats import skew, kurtosis
```

EDA

[3]:
```
df = pd.read_csv('house_price.csv')
```

[3]:
```
print(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)
```

[59]:
```
print(df.describe())
```

```
         total_sqft          bath         price           bhk  price_per_sqft
count  13200.000000  13200.000000  13200.000000  13200.000000    1.320000e+04
mean    1555.302783      2.691136    112.276178      2.800833    7.920337e+03
std     1237.323445      1.338915    149.175995      1.292843    1.067272e+05
min        1.000000      1.000000      8.000000      1.000000    2.670000e+02
25%     1100.000000      2.000000     50.000000      2.000000    4.267000e+03
50%     1275.000000      2.000000     71.850000      3.000000    5.438000e+03
75%     1672.000000      3.000000    120.000000      3.000000    7.317000e+03
max    52272.000000     40.000000   3600.000000     43.000000    1.200000e+07
```

[60]:
```
print(df.isnull().sum())
```

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

[10]:
```
print(df.shape)
```

```
(13200, 7)
```

[11]:
```
print(df.columns)
```

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

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


[11]:
```
print(df.columns)
```

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

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


[12]:

	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

[13]:
```
df.tail()
```

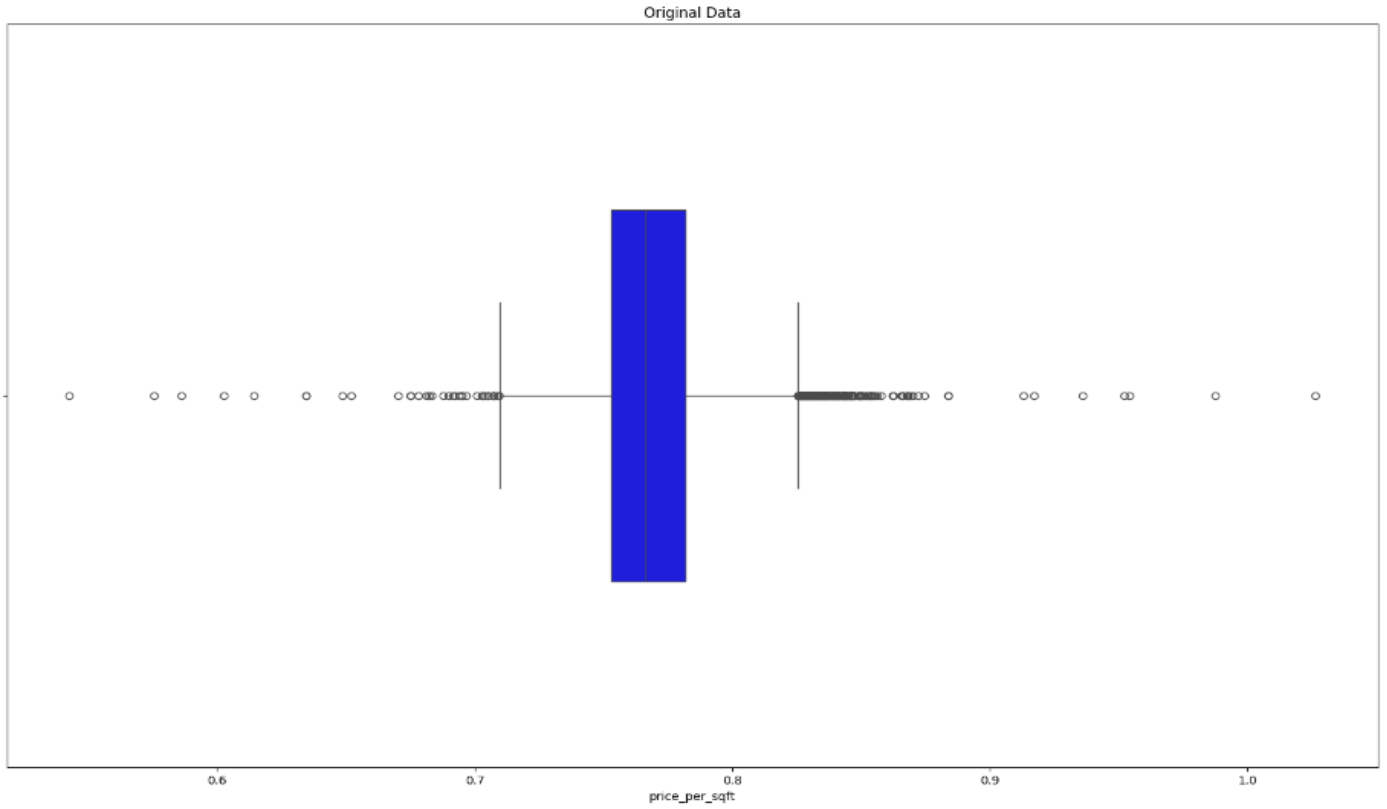
[13]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
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
13199	Doddathoguru	1 BHK	550.0	1.0	17.0	1	3090

Detect the outliers

[43]:
```
plt.figure(figsize=(20, 12))
sns.boxplot(x=df['price_per_sqft'], color='blue', width=0.5)
plt.title('Original Data')
```

[43]:
```
Text(0.5, 1.0, 'Original Data')
```



a)Mean And Standard Deviation Method

[61]:
```
mean = df['price_per_sqft'].mean()
std = df['price_per_sqft'].std()
low_trim = mean - 3 * std
up_trim = mean + 3 * std
print('lower limit :',low_trim)
print('upper limit:',up_trim)
```

```
lower limit : -312261.14424190175
upper limit: 328101.8177267502
```

[62]:
```
df_trim = df[(df['price_per_sqft'] > low_trim) & (df['price_per_sqft'] < up_trim)]
print("Data Length Original:", len(df))
print("Data Length after Trimming:", len(df_trim))
print("Outliers detected using Mean and Standard Deviation Method:")
print(df_trim)
```

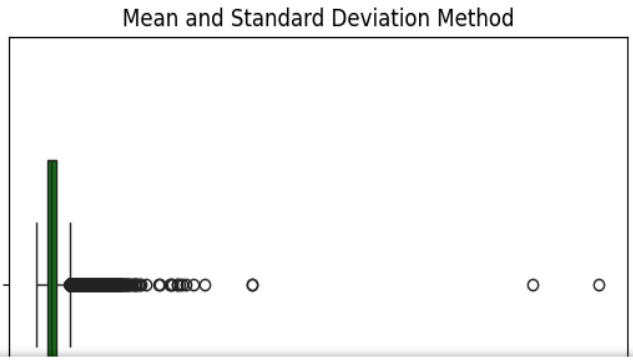
```
Data Length Original: 13200
Data Length after Trimming: 13195
Outliers detected using Mean and Standard Deviation Method:
                     location       size  total_sqft  bath   price  bhk  \
0      Electronic City Phase II      2 BHK      1056.0   2.0   39.07    2
1              Chikka Tirupathi  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
...                        ...        ...         ...   ...     ...  ...
13195               Whitefield  5 Bedroom      3453.0   4.0  231.00    5
13196                    other      4 BHK      3600.0   5.0  400.00    4
13197     Raja Rajeshwari Nagar      2 BHK      1141.0   2.0   60.00    2
13198           Padmanabhanagar      4 BHK      4689.0   4.0  488.00    4
13199              Doddathoguru      1 BHK       550.0   1.0   17.00    1

       price_per_sqft
0                3699
1                4615
2                4305
3                6245
4                4250
...               ...
13195            6689
13196           11111
```

```
[13195 rows x 7 columns]
```

[63]:
```
sns.boxplot(x=df_trim['price_per_sqft'] ,color='green', width=0.5)
plt.title('Mean and Standard Deviation Method')
```

[63]:
```
Text(0.5, 1.0, 'Mean and Standard Deviation Method')
```



b)Percentile Method

[64]:
```
upper_lim = df['price_per_sqft'].quantile(0.95)
lower_lim = df['price_per_sqft'].quantile(0.05)
print('upper limit:', upper_lim)
print('lower limit:', lower_lim)
```

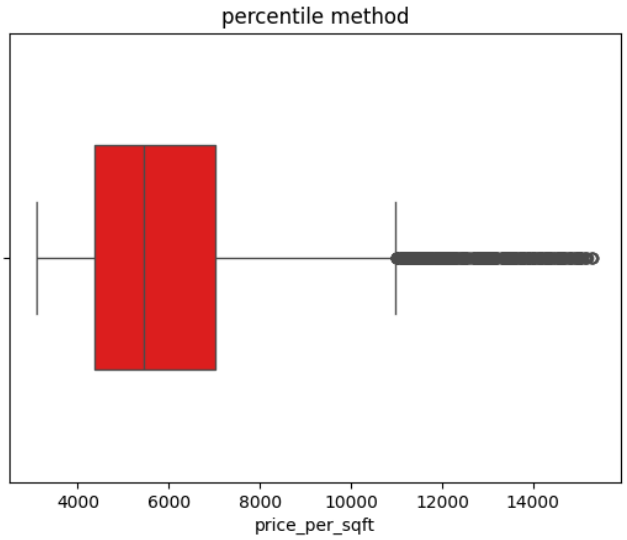
```
upper limit: 15312.099999999984
lower limit: 3107.8500000000004
```

[65]:
```
df_perc = df.loc[(df['price_per_sqft'] <= upper_lim) & (df['price_per_sqft'] >= lower_lim)]
print('before removing outliers:', len(df))
print('after removing outliers:',len(df_perc))
print('outliers:', len(df)-len(df_perc))
```

```
before removing outliers: 13200
after removing outliers: 11880
outliers: 1320
```

[66]:
```
sns.boxplot(x=df_perc['price_per_sqft'],color='red', width=0.5)
plt.title('percentile method')
```

[66]:
```
Text(0.5, 1.0, 'percentile method')
```



c)IQR (Inter quartile range) Method

[67]:
```
q1 = df['price_per_sqft'].quantile(0.25)
q3 = df['price_per_sqft'].quantile(0.75)
iqr = q3-q1
```

[68]:
```
q1, q3, iqr
```

[68]:
```
(4267.0, 7317.0, 3050.0)
```


c)IQR (Inter quartile range) Method

[67]:
```
q1 = df['price_per_sqft'].quantile(0.25)
q3 = df['price_per_sqft'].quantile(0.75)
iqr = q3-q1
```

[68]:
```
q1, q3, iqr
```

[68]:
```
(4267.0, 7317.0, 3050.0)
```


[69]:
```
upper_limit = q3 + (1.5 * iqr)
lower_limit = q1 - (1.5 * iqr)
lower_limit, upper_limit
```

[69]:
```
(-308.0, 11892.0)
```

[70]:
```
df.loc[(df['price_per_sqft'] > upper_limit) | (df['price_per_sqft'] < lower_limit)]
```

[70]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
7	Rajaji Nagar	4 BHK	3300.0	4.0	600.0	4	18181
9	other	6 Bedroom	1020.0	6.0	370.0	6	36274
22	Thanisandra	4 Bedroom	2800.0	5.0	380.0	4	13571
45	HSR Layout	8 Bedroom	600.0	9.0	200.0	8	33333
48	KR Puram	2 Bedroom	800.0	1.0	130.0	2	16250
...	...	...	...	...	...	...	...
13142	other	2 BHK	1140.0	1.0	185.0	2	16228
13157	other	7 Bedroom	1400.0	7.0	218.0	7	15571
13185	Hulimavu	1 BHK	500.0	1.0	220.0	1	44000
13186	other	4 Bedroom	1200.0	5.0	325.0	4	27083
13191	Ramamurthy Nagar	7 Bedroom	1500.0	9.0	250.0	7	16666

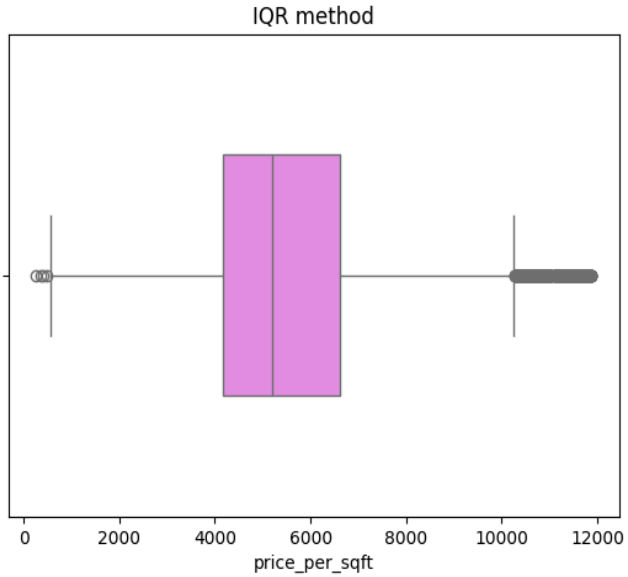
1265 rows × 7 columns

[71]:
```
# trimming - delete the outlier data
df_iqr = df.loc[(df['price_per_sqft'] <= upper_limit) & (df['price_per_sqft'] >= lower_limit)]
print('before removing outliers:', len(df))
print('after removing outliers:',len(df_iqr))
print('outliers:', len(df)-len(df_iqr))
```

```
before removing outliers: 13200
after removing outliers: 11935
outliers: 1265
```

[76]:
```
sns.boxplot(x=df_iqr['price_per_sqft'],color='violet', width=0.5)
plt.title('IQR method')
```

[76]:
```
Text(0.5, 1.0, 'IQR method')
```



d)Z Score Method

d)Z Score Method

[4]:
```
# find the limits
upper_limit = df['price_per_sqft'].mean() + 3*df['price_per_sqft'].std()
lower_limit = df['price_per_sqft'].mean() - 3*df['price_per_sqft'].std()
print('upper limit:', upper_limit)
print('lower limit:', lower_limit)
```

```
upper limit: 328101.8177267502
lower limit: -312261.14424190175
```

[5]:
```
# find the outliers
df.loc[(df['price_per_sqft'] > upper_limit) | (df['price_per_sqft'] < lower_limit)]
```

[5]:

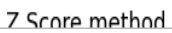
	location	size	total_sqft	bath	price	bhk	price_per_sqft
345	other	3 Bedroom	11.0	3.0	74.0	3	672727
1106	other	5 Bedroom	24.0	2.0	150.0	5	625000
4044	Sarjapur Road	4 Bedroom	1.0	4.0	120.0	4	12000000
4924	other	7 BHK	5.0	7.0	115.0	7	2300000
11447	Whitefield	4 Bedroom	60.0	4.0	218.0	4	363333

[6]:
```
# trimming - delete the outlier data
df_z = df.loc[(df['price_per_sqft'] <= upper_limit) & (df['price_per_sqft'] >= lower_limit)]
print('before removing outliers:', len(df))
print('after removing outliers:',len(df_z))
print('outliers:', len(df)-len(df_z))
```

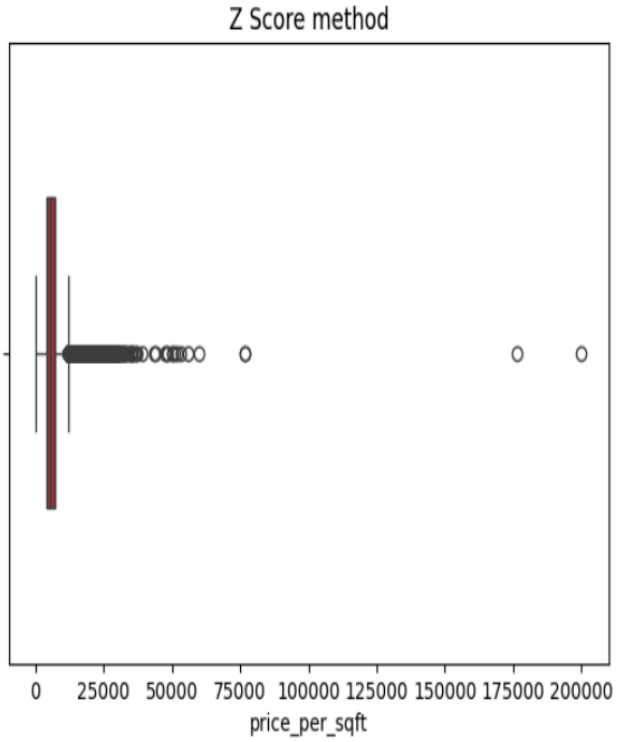
```
before removing outliers: 13200
after removing outliers: 13195
outliers: 5
```

[7]:
```
sns.boxplot(x=df_z['price_per_sqft'],color='brown', width=0.5)
plt.title('Z Score method')
```

[7]:
```
Text(0.5, 1.0, 'Z Score method')
```



[7]: Text(0.5, 1.0, 'Z Score method')



Using percentiles for the box plot is better for outlier detection because it provides a more stable and resistant representation,

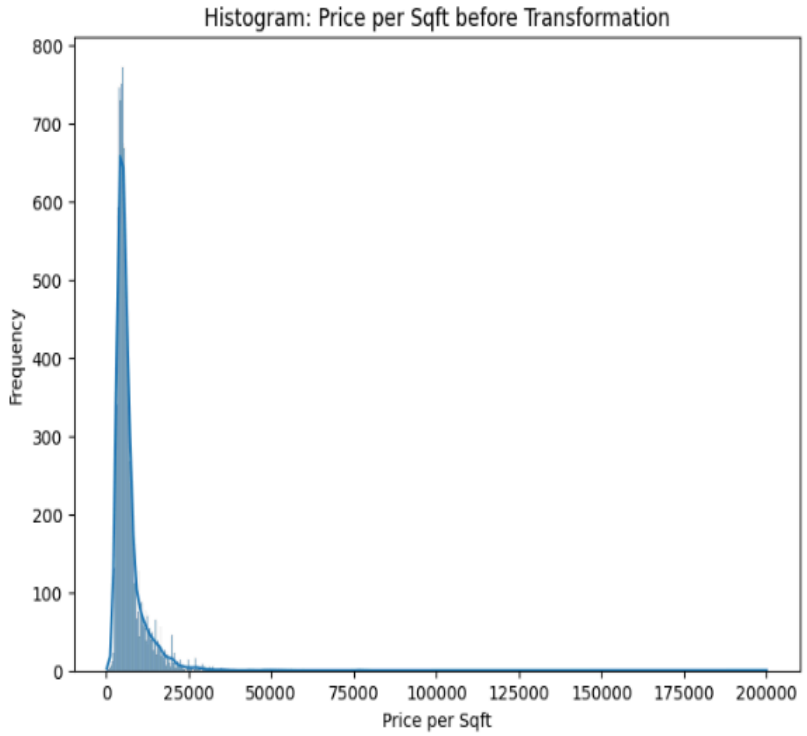
clearly separating genuine outliers from the main data distribution, without being distorted by extreme values.

skewness and kurtosis (histplot)

[8]:
```
# Create a histogram for the price per sqft column
plt.figure(figsize=(8, 6))
sns.histplot(df_z["price_per_sqft"], kde=True)
plt.title("Histogram: Price per Sqft before Transformation")
plt.xlabel("Price per Sqft")
plt.ylabel("Frequency")
plt.show()

# Check skewness and kurtosis
skewness_before = df_z["price_per_sqft"].skew()
kurtosis_before = df_z["price_per_sqft"].kurtosis()

print(f"Skewness (before transformation): {skewness_before:.2f}")
print(f"Kurtosis (before transformation): {kurtosis_before:.2f}")
```




```
Skewness (before transformation): 10.48
Kurtosis (before transformation): 313.65
```

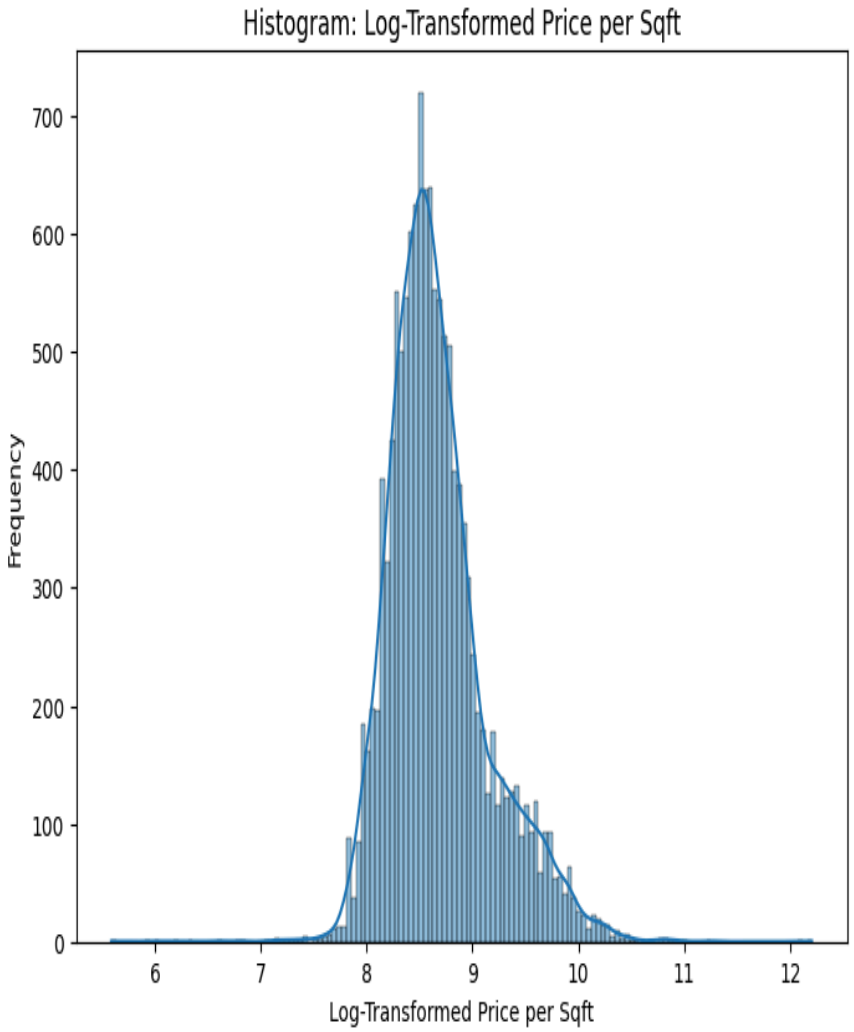
[9]:
```
# Apply the natural logarithm transformation
```

```
# Apply the natural logarithm transformation
df_z["price_per_sqft_log"] = np.log(df_z["price_per_sqft"])

# Create a histogram for the transformed column
plt.figure(figsize=(8, 6))
sns.histplot(df_z["price_per_sqft_log"], kde=True)
plt.title("Histogram: Log-Transformed Price per Sqft")
plt.xlabel("Log-Transformed Price per Sqft")
plt.ylabel("Frequency")
plt.show()

# Check skewness and kurtosis after transformation
skewness_after = df_z["price_per_sqft_log"].skew()
kurtosis_after = df_z["price_per_sqft_log"].kurtosis()

print(f"Skewness (after transformation): {skewness_after:.2f}")
print(f"Kurtosis (after transformation): {kurtosis_after:.2f}")
```

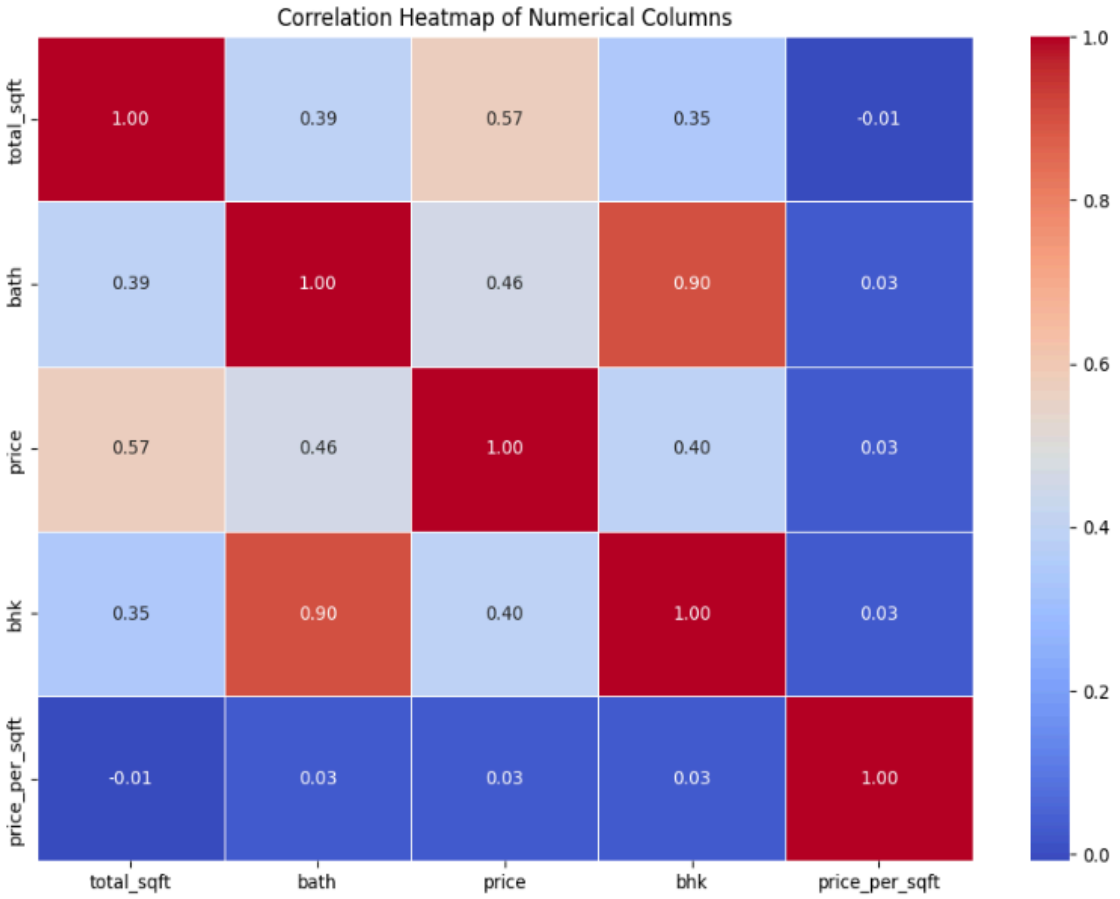



```
Skewness (after transformation): 0.87
Kurtosis (after transformation): 1.91
```

Kurtosis (after transformation): 1.91

Correlation Heatmap

[25]:
```
#  numerical columns
num_df = df.select_dtypes(include=[np.number])
# heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(num_df.corr(), annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('Correlation Heatmap of Numerical Columns')
plt.show()
```



Correlation Scatter Plots

[27]:
```
sns.pairplot(num_df)
plt.suptitle('Scatter Plots between Numerical Variables', y=1.02)
plt.show()
```