In [49]:
```
#Problem Statement: Data Wrangling on Real Estate Market
# Tasks to Perform:
# 1.  Import the "RealEstate_Prices.csv" dataset. Clean column names by removing sp
# special characters, or renaming them for clarity.
# 2. Handle missing values in the dataset, deciding on an appropriate strategy (e.g
# imputation or removal).
# 3. Perform data merging if additional datasets with relevant information are avai
# (e.g., neighborhood demographics or nearby amenities).
# 4. Filter and subset the data based on specific criteria, such as a particular ti
# property type, or location.
# 5. Handle categorical variables by encoding them appropriately (e.g., one-hot enc
# label encoding) for further analysis.
# 6.  Aggregate the data to calculate summary statistics or derived metrics such as
# sale prices by neighborhood or property type.
# 7. Identify and handle outliers or extreme values in the data that may affect the
# or modeling process.

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [50]:
```
real_estate = pd.read_csv('Bengaluru_House_Data.csv')
real_estate.head()
```

Out[50]:

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

In [51]:
```
print("Number of Rows:",real_estate.shape[0])
print("Number of Columns:",real_estate.shape[1])
```

```
Number of Rows: 13320
Number of Columns: 9
```

In [52]:
```
# Step 2: Clean Column Names (remove spaces, etc.)
real_estate.columns = real_estate.columns.str.strip().str.lower().str.replace(' ',
real_estate.columns
```

Out[52]:
```
Index(['area_type', 'availability', 'location', 'size', 'society',
       'total_sqft', 'bath', 'balcony', 'price'],
      dtype='object')
```

In [53]:
```
real_estate.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13320 entries, 0 to 13319
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   area_type     13320 non-null  object 
 1   availability  13320 non-null  object 
 2   location      13319 non-null  object 
 3   size          13304 non-null  object 
 4   society       7818 non-null   object 
 5   total_sqft    13320 non-null  object 
 6   bath          13247 non-null  float64
 7   balcony       12711 non-null  float64
 8   price         13320 non-null  float64
dtypes: float64(3), object(6)
memory usage: 936.7+ KB
```

In [54]:
```
#Step 3: Handle Missing Values
print(real_estate.isnull().sum())
```

```
area_type          0
availability       0
location           1
size              16
society         5502
total_sqft         0
bath              73
balcony          609
price              0
dtype: int64
```

In [55]:
```
real_estate['size'].fillna(real_estate['size'].mode()[0], inplace=True)
real_estate['bath'].fillna(real_estate['bath'].median(), inplace=True)
real_estate['balcony'].fillna(real_estate['balcony'].mode()[0], inplace=True)
real_estate['location'].fillna(real_estate['location'].mode()[0], inplace=True)

real_estate.drop(columns=['society'], inplace=True)

real_estate.isnull().sum()
```

Out[55]:
```
area_type       0
availability    0
location        0
size            0
total_sqft      0
bath            0
balcony         0
price           0
dtype: int64
```

In [56]:
```
real_estate.dtypes
```

Out[56]:
```
area_type        object
availability     object
location         object
size             object
total_sqft       object
bath            float64
balcony         float64
price           float64
dtype: object
```

In [57]:
```
#Step 4: Filter and Subset the Data
available = real_estate[real_estate['availability'] != 'Ready To Move']
print("Properties available for sale", available)
```

```
Properties available for sale                  area_type availability
location       size  \
0      Super built-up  Area       19-Dec  Electronic City Phase II      2 BHK
6      Super built-up  Area       18-May          Old Airport Road      4 BHK
10     Super built-up  Area       18-Feb                Whitefield      3 BHK
21     Super built-up  Area       19-Dec                Binny Pete      3 BHK
24     Super built-up  Area       18-Nov               Thanisandra       1 RK
...                      ...          ...                       ...        ...
13291           Plot  Area       18-Jan            Weavers Colony  1 Bedroom
13292  Super built-up  Area       18-Jul          Udayapur Village      3 BHK
13295  Super built-up  Area       18-Feb              Haralur Road      3 BHK
13299  Super built-up  Area       18-Dec                Whitefield      4 BHK
13318  Super built-up  Area       18-Jun           Padmanabhanagar      4 BHK

        total_sqft  bath  balcony   price
0             1056   2.0      1.0   39.07
6             2732   4.0      2.0  204.00
10            1800   2.0      2.0   70.00
21            1755   3.0      1.0  122.00
24             510   1.0      0.0   25.25
...            ...   ...      ...     ...
13291          812   1.0      0.0   26.00
13292         1440   2.0      2.0   63.93
13295         1810   3.0      2.0  112.00
13299  2830 - 2882   5.0      0.0  154.50
13318         4689   4.0      1.0  488.00

[2739 rows x 8 columns]
```

In [58]:
```
high_value = real_estate[real_estate['price'] > 100]
print('Properties with high price:', high_value)
```

```
Properties with high price:                   area_type   availability           loc
ation       size  \
1               Plot  Area  Ready To Move   Chikka Tirupathi  4 Bedroom
6      Super built-up  Area         18-May   Old Airport Road      4 BHK
7      Super built-up  Area  Ready To Move       Rajaji Nagar      4 BHK
9               Plot  Area  Ready To Move       Gandhi Bazar  6 Bedroom
11              Plot  Area  Ready To Move         Whitefield  4 Bedroom
...                      ...           ...                ...        ...
13311           Plot  Area  Ready To Move   Ramamurthy Nagar  7 Bedroom
13314  Super built-up  Area  Ready To Move  Green Glen Layout      3 BHK
13315        Built-up  Area  Ready To Move         Whitefield  5 Bedroom
13316  Super built-up  Area  Ready To Move      Richards Town      4 BHK
13318  Super built-up  Area         18-Jun    Padmanabhanagar      4 BHK

      total_sqft  bath  balcony  price
1           2600   5.0      3.0  120.0
6           2732   4.0      2.0  204.0
7           3300   4.0      2.0  600.0
9           1020   6.0      2.0  370.0
11          2785   5.0      3.0  295.0
...          ...   ...      ...    ...
13311       1500   9.0      2.0  250.0
13314       1715   3.0      3.0  112.0
13315       3453   4.0      0.0  231.0
13316       3600   5.0      2.0  400.0
13318       4689   4.0      1.0  488.0

[4103 rows x 8 columns]
```

In [59]:
```
#Step 5: Aggregate Data for Insights
avg_price_by_location = real_estate.groupby('location')['price'].mean().sort_values
print("Average price by location:\n",avg_price_by_location)

avg_price_by_area = real_estate.groupby('area_type')['price'].mean()
print("Average price by area type:\n",avg_price_by_area)
```

```
Average price by location:
 location
Cubbon Road             1900.000000
Ashok Nagar             1486.000000
Defence Colony          1167.714286
Yemlur                  1093.388889
Church Street           1068.000000
D Souza Layout          1015.000000
Sadashiva Nagar         1011.100000
Sindhi Colony            988.000000
Srinivas Colony          922.000000
5th Block Jayanagar      905.000000
Name: price, dtype: float64
Average price by area type:
 area_type
Built-up  Area          104.285498
Carpet  Area             89.502356
Plot  Area              208.495486
Super built-up  Area     92.971757
Name: price, dtype: float64
```

In [60]:
```
# Step 8: Identify and Handle Outliers

Q1 = real_estate['price'].quantile(0.25)
Q3 = real_estate['price'].quantile(0.75)
IQR = Q3 - Q1


lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

real_estate = real_estate[(real_estate['price'] >= lower_bound) & (real_estate['pri

print("Outliers handled successfully!")
```

```
Outliers handled successfully!
```

In [61]:
```
# Step 9: Export Cleaned Data for Further Analysis
real_estate.to_csv("Cleaned_Bengaluru_House_Data.csv", index=False)
print("Cleaned dataset saved successfully!")
```

```
Cleaned dataset saved successfully!
```