In [1]:

```python
# Data Manipulation
import pandas as pd
import numpy as np

# Data Visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Machine Learning
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.experimental import enable_hist_gradient_boosting
from sklearn.ensemble import HistGradientBoostingRegressor

from sklearn.metrics import mean_squared_error, r2_score

# Optional Libraries for Advanced Techniques
import xgboost as xgb
import scipy
import statsmodels.api as sm
import warnings
warnings.filterwarnings('ignore')
```

```
C:\Users\chand\anaconda3\Lib\site-packages\sklearn\experimental\enable_hist_gradient_boosting.py:1
5: UserWarning: Since version 1.0, it is not needed to import enable_hist_gradient_boosting anymor
e. HistGradientBoostingClassifier and HistGradientBoostingRegressor are now stable and can be norma
lly imported from sklearn.ensemble.
  warnings.warn(
```

In [2]:

```python
data = pd.read_excel('innercity.xlsx')
```

In [3]:

```python
data.shape
```

Out[3]:

```
(21613, 23)
```

In [4]:

```python
data.describe()
```

Out[4]:

	cid	price	room_bed	room_bath	living_measure	lot_measure	sight	quality	c
count	2.161300e+04	2.161300e+04	21505.000000	21505.000000	21596.000000	2.157100e+04	21556.000000	21612.000000	2.
mean	4.580302e+09	5.401822e+05	3.371355	2.115171	2079.860761	1.510458e+04	0.234366	7.656857	
std	2.876566e+09	3.673622e+05	0.930289	0.770248	918.496121	4.142362e+04	0.766438	1.175484	
min	1.000102e+06	7.500000e+04	0.000000	0.000000	290.000000	5.200000e+02	0.000000	1.000000	
25%	2.123049e+09	3.219500e+05	3.000000	1.750000	1429.250000	5.040000e+03	0.000000	7.000000	
50%	3.904930e+09	4.500000e+05	3.000000	2.250000	1910.000000	7.618000e+03	0.000000	7.000000	
75%	7.308900e+09	6.450000e+05	4.000000	2.500000	2550.000000	1.068450e+04	0.000000	8.000000	
max	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e+06	4.000000	13.000000	

In [5]:

```python
pd.set_option('display.float_format',lambda x:'%.5f' % x) # price is showing some scientific notati
```

In [6]:

```python
data.describe()
```

Out[6]:

	cid	price	room_bed	room_bath	living_measure	lot_measure	sight	quality
count	21613.00000	21613.00000	21505.00000	21505.00000	21596.00000	21571.00000	21556.00000	21612.00000
mean	4580301520.86499	540182.15879	3.37136	2.11517	2079.86076	15104.58328	0.23437	7.65686
std	2876565571.31205	367362.23172	0.93029	0.77025	918.49612	41423.61939	0.76644	1.17548
min	1000102.00000	75000.00000	0.00000	0.00000	290.00000	520.00000	0.00000	1.00000
25%	2123049194.00000	321950.00000	3.00000	1.75000	1429.25000	5040.00000	0.00000	7.00000
50%	3904930410.00000	450000.00000	3.00000	2.25000	1910.00000	7618.00000	0.00000	7.00000
75%	7308900445.00000	645000.00000	4.00000	2.50000	2550.00000	10684.50000	0.00000	8.00000
max	9900000190.00000	7700000.00000	33.00000	8.00000	13540.00000	1651359.00000	4.00000	13.00000

In [7]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 23 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   cid               21613 non-null  int64
 1   dayhours          21613 non-null  object
 2   price             21613 non-null  int64
 3   room_bed          21505 non-null  float64
 4   room_bath         21505 non-null  float64
 5   living_measure    21596 non-null  float64
 6   lot_measure       21571 non-null  float64
 7   ceil              21572 non-null  object
 8   coast             21612 non-null  object
 9   sight             21556 non-null  float64
 10  condition         21556 non-null  object
 11  quality           21612 non-null  float64
 12  ceil_measure      21612 non-null  float64
 13  basement          21612 non-null  float64
 14  yr_built          21612 non-null  object
 15  yr_renovated      21613 non-null  int64
 16  zipcode           21613 non-null  int64
 17  lat               21613 non-null  float64
 18  long              21613 non-null  object
 19  living_measure15  21447 non-null  float64
 20  lot_measure15     21584 non-null  float64
 21  furnished         21584 non-null  float64
 22  total_area        21584 non-null  object
dtypes: float64(12), int64(4), object(7)
memory usage: 3.8+ MB
```

In [8]:

```
data.isnull().sum()
```

Out[8]:
```
cid                    0
dayhours               0
price                  0
room_bed             108
room_bath            108
living_measure        17
lot_measure           42
ceil                  41
coast                  1
sight                 57
condition             57
quality                1
ceil_measure           1
basement               1
yr_built               1
yr_renovated           0
zipcode                0
lat                    0
long                   0
living_measure15     166
lot_measure15         29
furnished             29
total_area            29
dtype: int64
```

In [9]:
```
data.isnull().sum().sum()
```

Out[9]:
```
688
```

In [10]:
```
# value counts of each feature
def value_count(data):
    for var in data.columns:
        print(data[var].value_counts())
        print("--------------------------")
```

In [11]:
```
value_count(data)
```

Loading [MathJax]/extensions/Safe.js

```
cid
795000620     3
5101405604    2
9809000020    2
7853420110    2
6021500970    2
             ..
7871500485    1
2022069200    1
9808630120    1
7302000210    1
8805900430    1
Name: count, Length: 21436, dtype: int64
--------------------------
dayhours
20140623T000000    142
20140625T000000    131
20140626T000000    131
20140708T000000    127
20150427T000000    126
                  ...
20150515T000000      1
20150110T000000      1
20140803T000000      1
20150131T000000      1
20140830T000000      1
Name: count, Length: 372, dtype: int64
--------------------------
price
450000    172
350000    172
550000    159
500000    152
425000    150
         ...
919000      1
364988      1
362764      1
849900      1
685530      1
Name: count, Length: 3625, dtype: int64
--------------------------
room_bed
3.00000     9767
4.00000     6854
2.00000     2747
5.00000     1595
6.00000      270
1.00000      197
7.00000       38
8.00000       13
0.00000       13
9.00000        6
10.00000       3
33.00000       1
11.00000       1
Name: count, dtype: int64
--------------------------
room_bath
2.50000    5358
1.00000    3829
1.75000    3031
2.25000    2039
2.00000    1917
1.50000    1439
2.75000    1178
3.00000     750
3.50000     726
3.25000     588
3.75000     155
4.00000     135
```
Loading [MathJax]/extensions/Safe.js

```
4.50000     100
4.25000      78
0.75000      72
4.75000      23
5.00000      21
5.25000      13
5.50000      10
0.00000      10
1.25000       9
6.00000       6
5.75000       4
0.50000       4
8.00000       2
6.75000       2
6.50000       2
6.25000       2
7.50000       1
7.75000       1
Name: count, dtype: int64
--------------------------
living_measure
1300.00000    138
1400.00000    134
1440.00000    133
1010.00000    129
1800.00000    129
             ...
1728.00000      1
5240.00000      1
2105.00000      1
3845.00000      1
2253.00000      1
Name: count, Length: 1038, dtype: int64
--------------------------
lot_measure
5000.00000      356
6000.00000      290
4000.00000      251
7200.00000      219
4800.00000      120
               ...
641203.00000      1
913.00000         1
12286.00000       1
8749.00000        1
60467.00000       1
Name: count, Length: 9765, dtype: int64
--------------------------
ceil
1      10647
2       8210
1.5     1905
3        610
2.5      161
$         30
3.5        8
`          1
Name: count, dtype: int64
--------------------------
coast
0    21421
1      161
$       30
Name: count, dtype: int64
--------------------------
sight
0.00000    19437
2.00000      959
3.00000      510
1.00000      332
4.00000      318
```
Loading [MathJax]/extensions/Safe.js

```
Name: count, dtype: int64
-------------------------
condition
3    13978
4     5655
5     1694
2      171
1       30
$       28
Name: count, dtype: int64
-------------------------
quality
7.00000     8981
8.00000     6067
9.00000     2615
6.00000     2038
10.00000    1134
11.00000     399
5.00000      242
12.00000      90
4.00000       29
13.00000      13
3.00000        3
1.00000        1
Name: count, dtype: int64
-------------------------
ceil_measure
1300.00000    212
1010.00000    210
1200.00000    206
1220.00000    192
1140.00000    184
             ...
2481.00000      1
1728.00000      1
2105.00000      1
3845.00000      1
2253.00000      1
Name: count, Length: 946, dtype: int64
-------------------------
basement
0.00000       13125
600.00000       221
700.00000       218
500.00000       214
800.00000       206
              ...
4130.00000        1
2050.00000        1
784.00000         1
518.00000         1
2180.00000        1
Name: count, Length: 306, dtype: int64
-------------------------
yr_built
2014    559
2006    454
2005    450
2004    433
2003    421
       ...
1901     29
1902     27
1935     24
1934     21
$        14
Name: count, Length: 117, dtype: int64
-------------------------
yr_renovated
0       20699
2014       91
```

```
2013       37
2003       36
2007       35
        ...
1944        1
1948        1
1959        1
1951        1
1954        1
Name: count, Length: 70, dtype: int64
--------------------------
zipcode
98103    602
98038    590
98115    583
98052    574
98117    553
        ...
98102    105
98010    100
98024     81
98148     57
98039     50
Name: count, Length: 70, dtype: int64
--------------------------
lat
47.54910    17
47.68460    17
47.66240    17
47.53220    17
47.67110    16
            ..
47.61530     1
47.75820     1
47.29430     1
47.29390     1
47.39150     1
Name: count, Length: 5034, dtype: int64
--------------------------
long
-122.29000    116
-122.30000    111
-122.36200    104
-122.29100    100
-122.37200     99
             ...
-122.47400      1
-121.71100      1
-121.84500      1
-121.73700      1
-121.94700      1
Name: count, Length: 753, dtype: int64
--------------------------
living_measure15
1540.00000    197
1440.00000    193
1560.00000    192
1500.00000    178
1580.00000    167
             ...
1813.00000      1
1336.00000      1
2005.00000      1
2049.00000      1
1786.00000      1
Name: count, Length: 774, dtype: int64
--------------------------
lot_measure15
5000.00000    427
4000.00000    357
6000.00000    287
```

```
7200.00000    211
4800.00000    145
             ...
4862.00000      1
5228.00000      1
9354.00000      1
2378.00000      1
7604.00000      1
Name: count, Length: 8682, dtype: int64
--------------------------
furnished
0.00000    17338
1.00000     4246
Name: count, dtype: int64
--------------------------
total_area
$        39
6770     19
5940     19
7330     19
9060     19
         ..
15707     1
5355      1
12215     1
46580     1
38122     1
Name: count, Length: 11145, dtype: int64
--------------------------
```

cat variables = cid,coast,sight,condition,quality,zipcode,furnished,ceil,room_bed,room_bath # Num variables = dayhours,price,living_measure,lot_measure,ceil_measure,basement,yr_built,yr_renovated,lat,long,living_measure15,lot_measure15,total_area

In [12]:

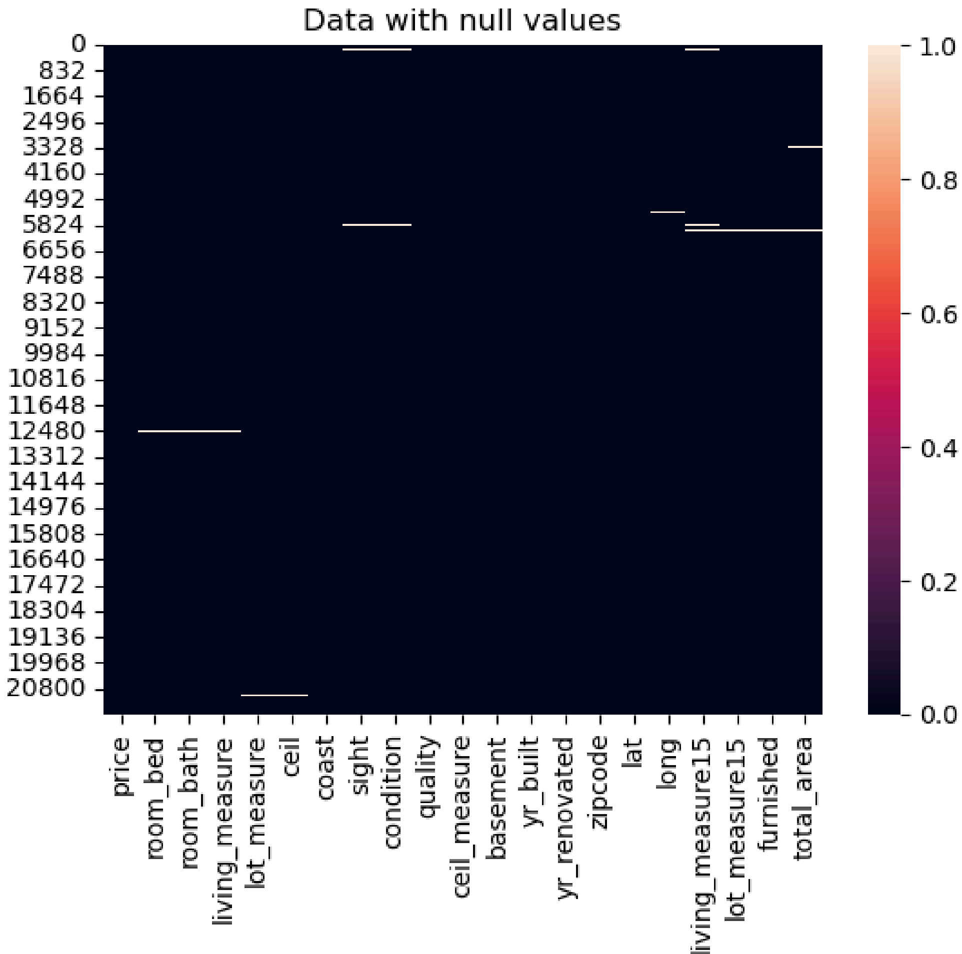
```
data = data.replace('$',np.NaN)
data = data.replace('`',np.NaN)
```

In [13]:

```
data = data.drop(['cid','dayhours'],axis='columns')
```

In [14]:

```
sns.heatmap(data.isnull())

plt.title('Data with null values')
plt.show()
```



In [15]:

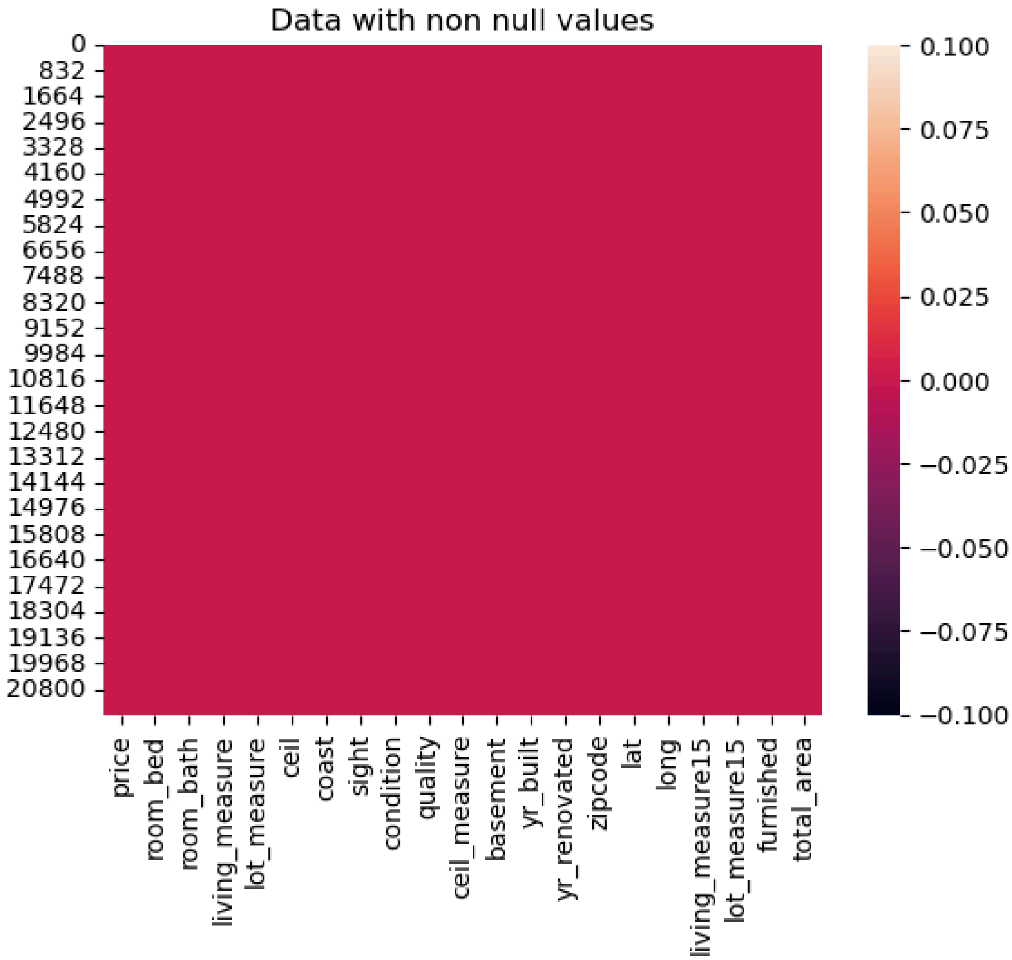
```
z = data["coast"].mode()[0]
data["coast"].fillna(z,inplace=True)

z = data["sight"].mode()[0]
data["sight"].fillna(z,inplace=True)

z = data["condition"].mode()[0]
data["condition"].fillna(z,inplace=True)

z = data["quality"].mode()[0]
data["quality"].fillna(z,inplace=True)

z = data["furnished"].mode()[0]
data["furnished"].fillna(z,inplace=True)

z = data["room_bed"].mode()[0]
data["room_bed"].fillna(z,inplace=True)

z = data["room_bath"].mode()[0]
data["room_bath"].fillna(z,inplace=True)

z = data["ceil"].mode()[0]
data["ceil"].fillna(z,inplace=True)
```

In [16]:

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='mean')
data = pd.DataFrame(imputer.fit_transform(data), columns=data.columns)
```

In [17]:

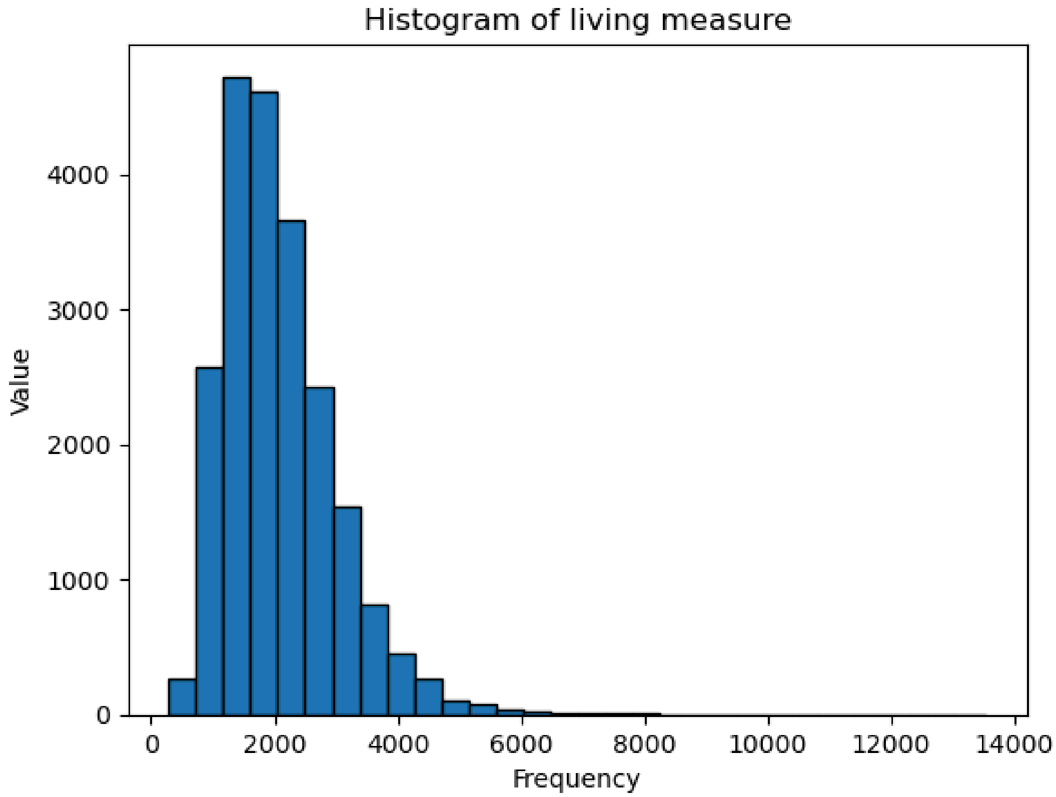
```
sns.heatmap(data.isnull())

plt.title('Data with non null values')
plt.show()
```

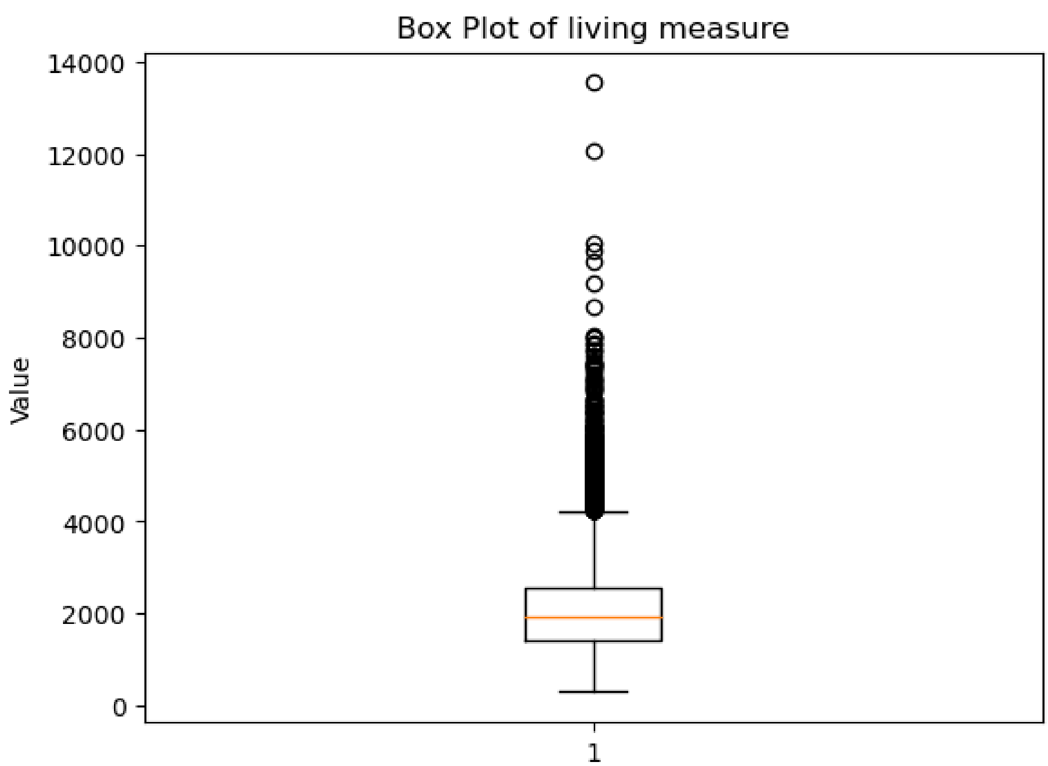


Univariate analysis of features

In [18]:

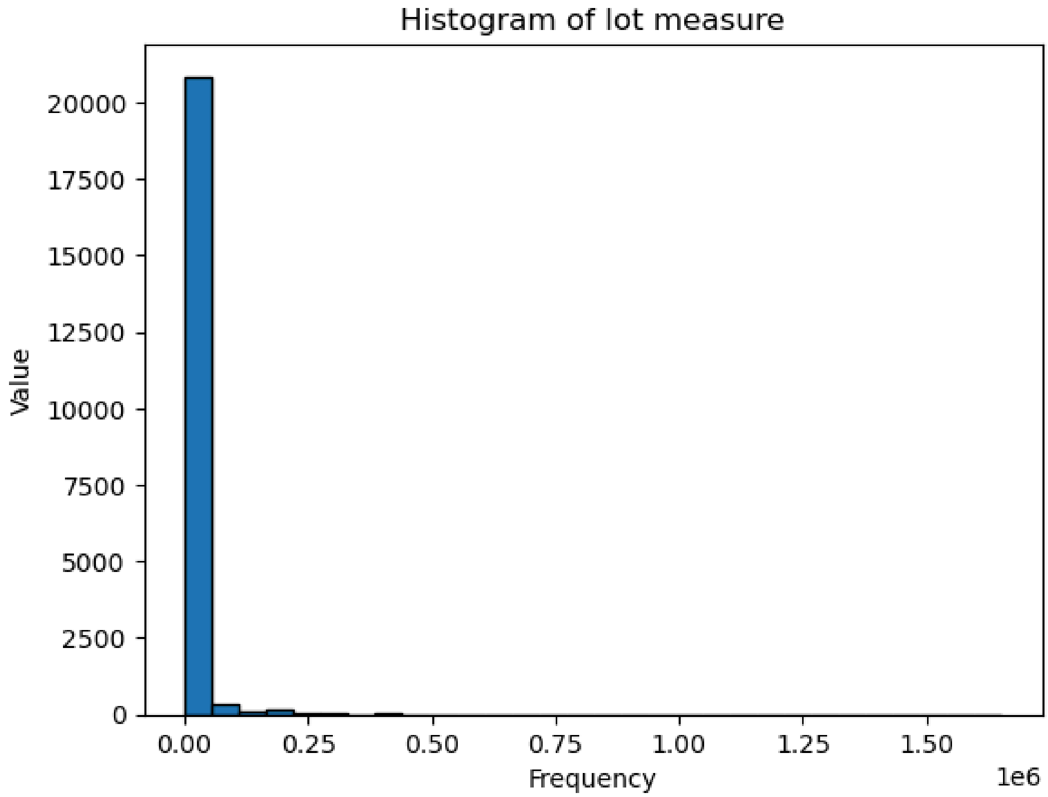
```
plt.hist(data['living_measure'], bins=30, edgecolor='black')
plt.title('Histogram of living measure')
plt.xlabel('Frequency')
plt.ylabel('Value')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/hist of living.png")
plt.show()


plt.boxplot(data['living_measure'])
plt.title('Box Plot of living measure')
plt.ylabel('Value')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/box of living.png")
plt.show()
```

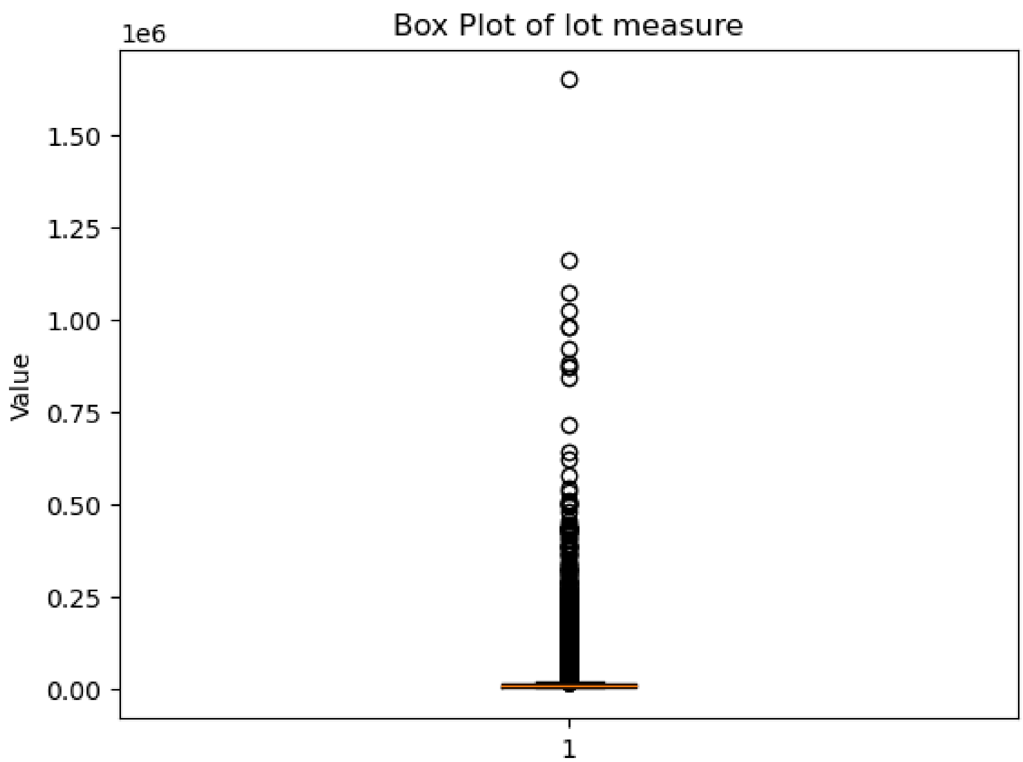




In [19]:

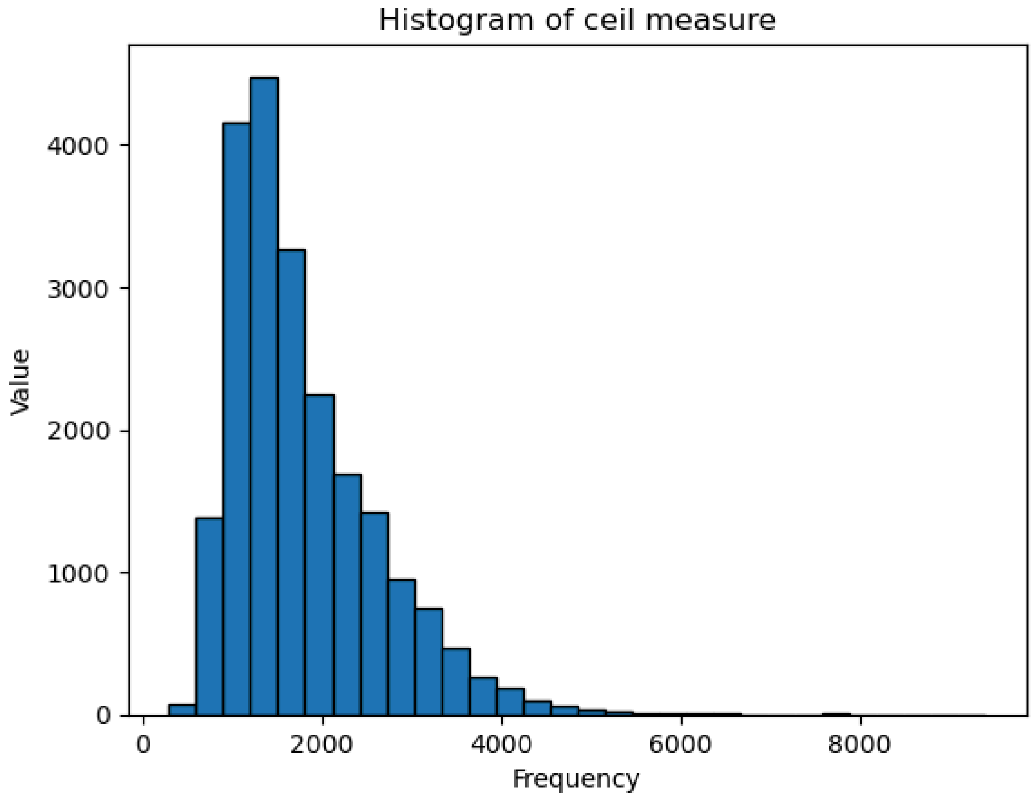
```
plt.hist(data['lot_measure'], bins=30, edgecolor='black')
plt.title('Histogram of lot measure')
plt.xlabel('Frequency')
plt.ylabel('Value')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/hist of lot.png")
plt.show()


plt.boxplot(data['lot_measure'])
plt.title('Box Plot of lot measure')
plt.ylabel('Value')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/box of lot.png")
plt.show()
```

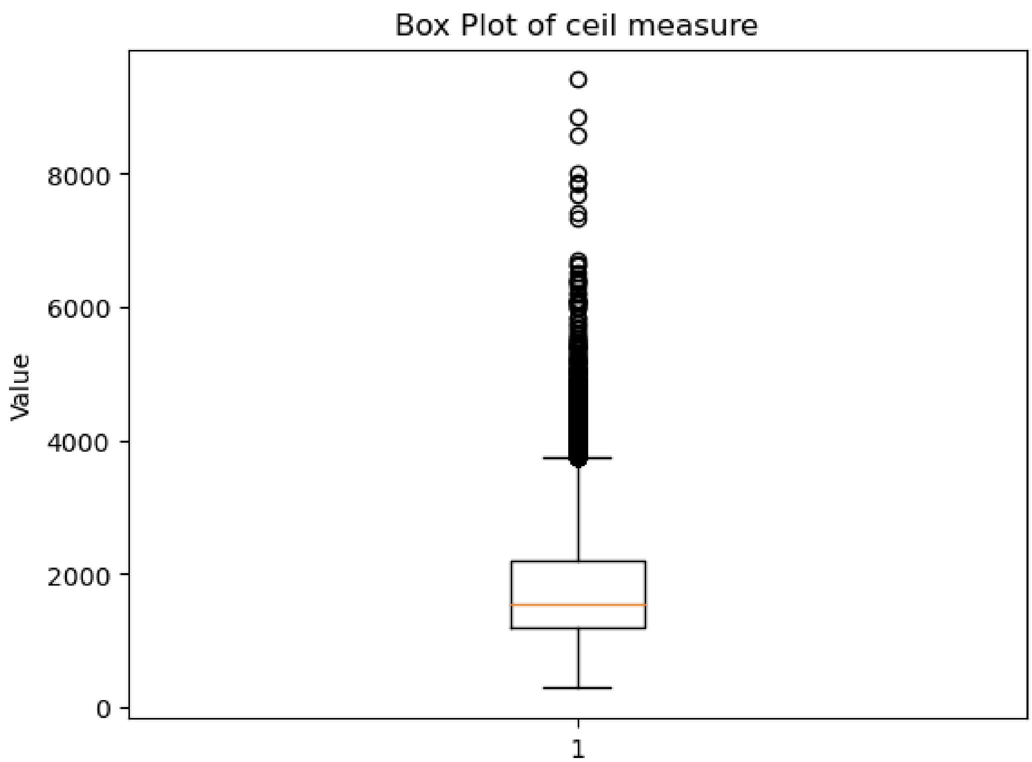




In [20]:

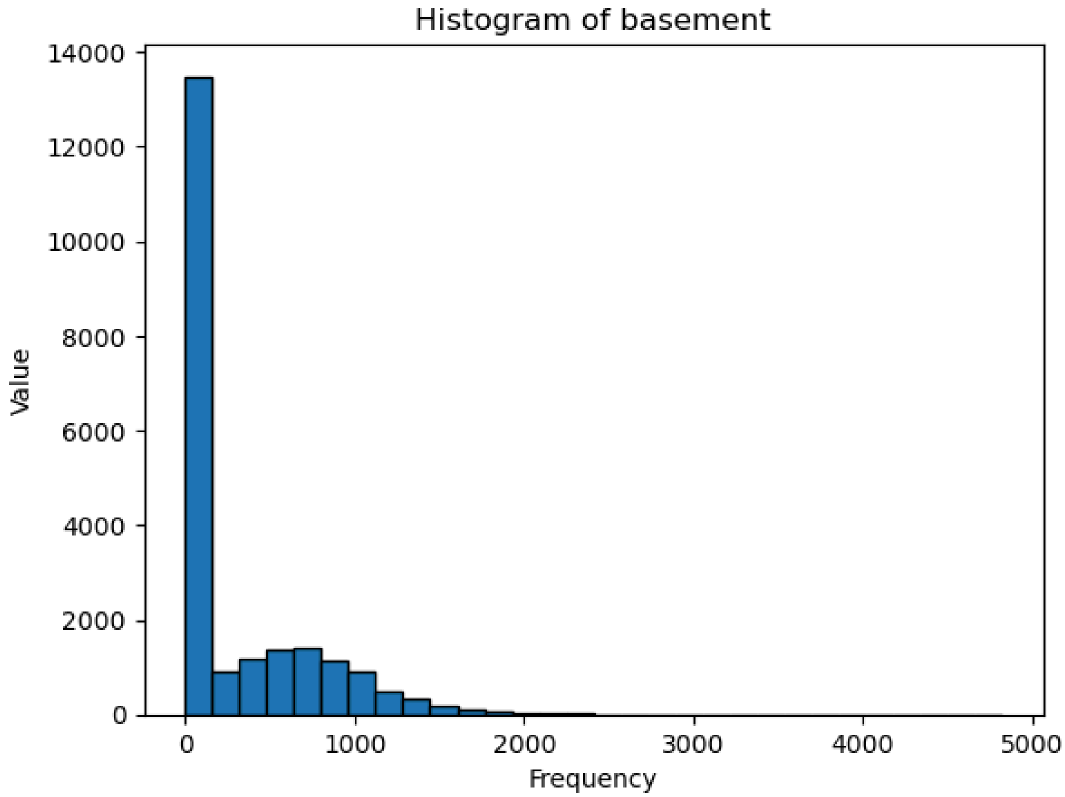
```
plt.hist(data['ceil_measure'], bins=30, edgecolor='black')
plt.title('Histogram of ceil measure')
plt.xlabel('Frequency')
plt.ylabel('Value')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/hist of ceil measure.png")
plt.show()


plt.boxplot(data['ceil_measure'])
plt.title('Box Plot of ceil measure')
plt.ylabel('Value')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/box of ceil measure.png")
plt.show()
```

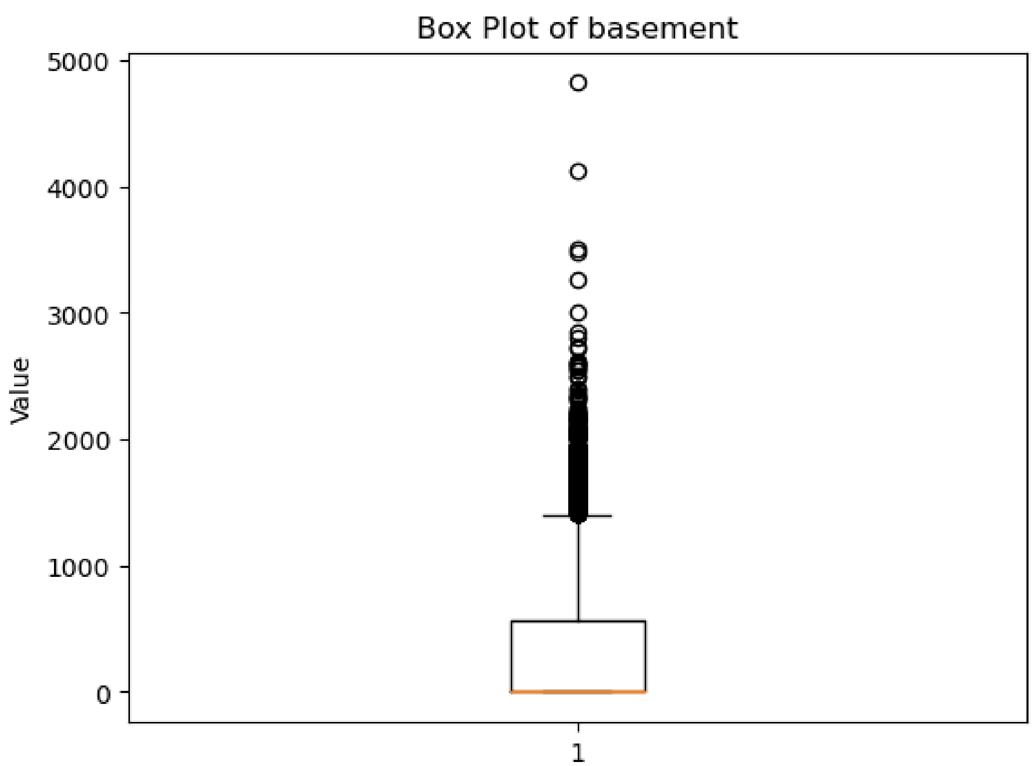




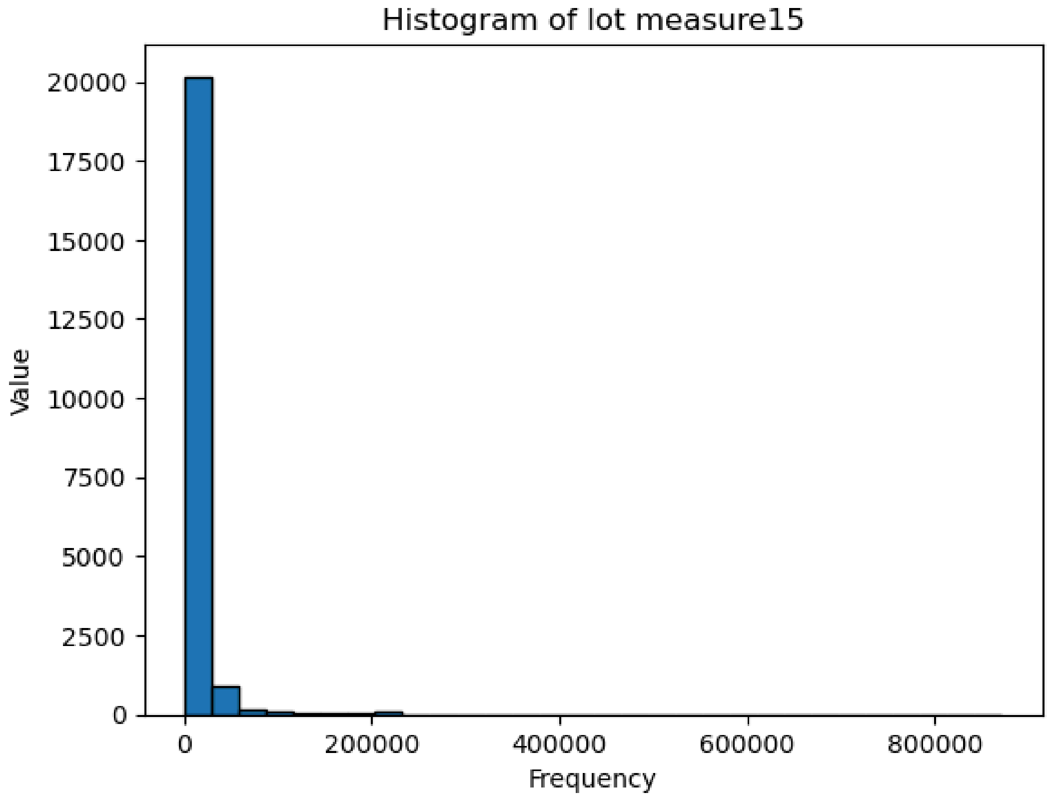
In [21]:

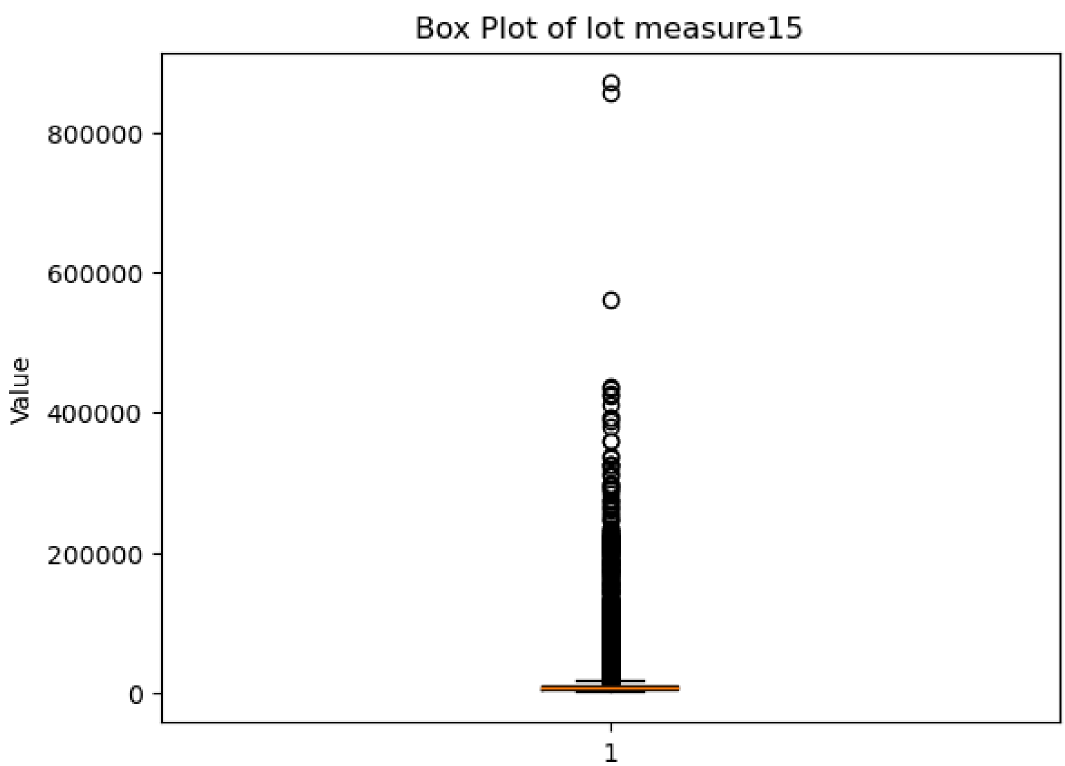
```
plt.hist(data['basement'], bins=30, edgecolor='black')
plt.title('Histogram of basement')
plt.xlabel('Frequency')
plt.ylabel('Value')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/hist of basement.png")
plt.show()


plt.boxplot(data['basement'])
plt.title('Box Plot of basement')
plt.ylabel('Value')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/box of basement.png")
plt.show()
```





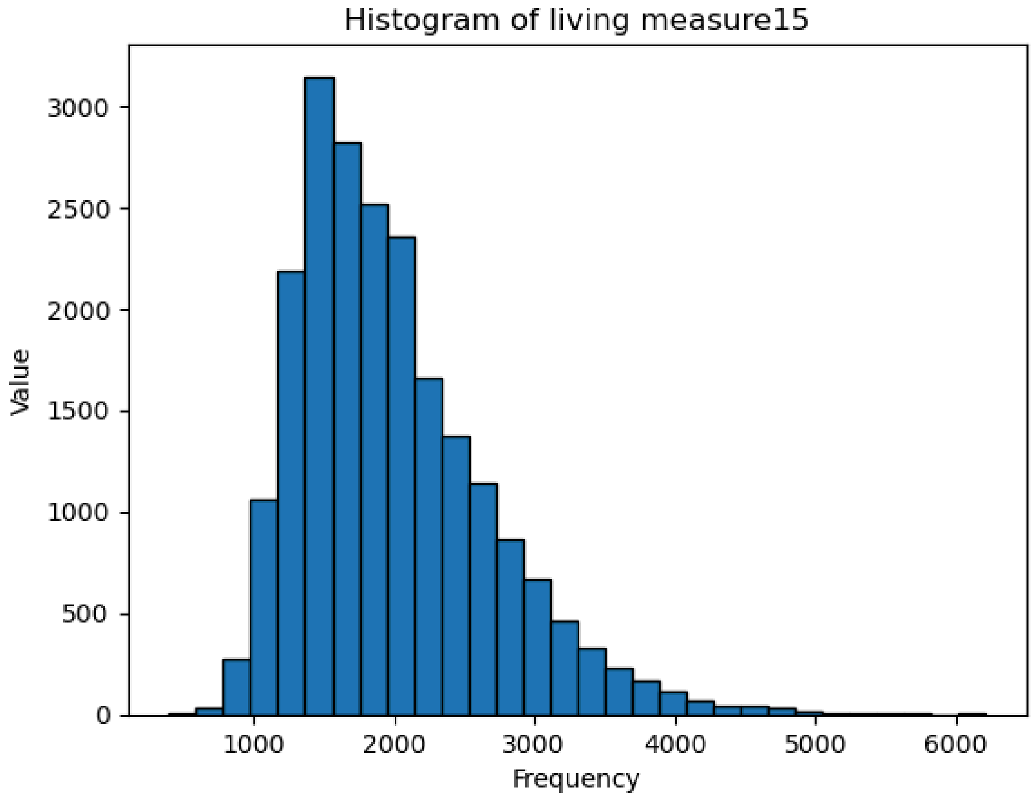
In [22]:
```
plt.hist(data['lot_measure15'], bins=30, edgecolor='black')
plt.title('Histogram of lot measure15')
plt.xlabel('Frequency')
plt.ylabel('Value')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/hist lot measure15.png")
plt.show()


plt.boxplot(data['lot_measure15'])
plt.title('Box Plot of lot measure15')
plt.ylabel('Value')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/box lot measure15.png")
plt.show()
```

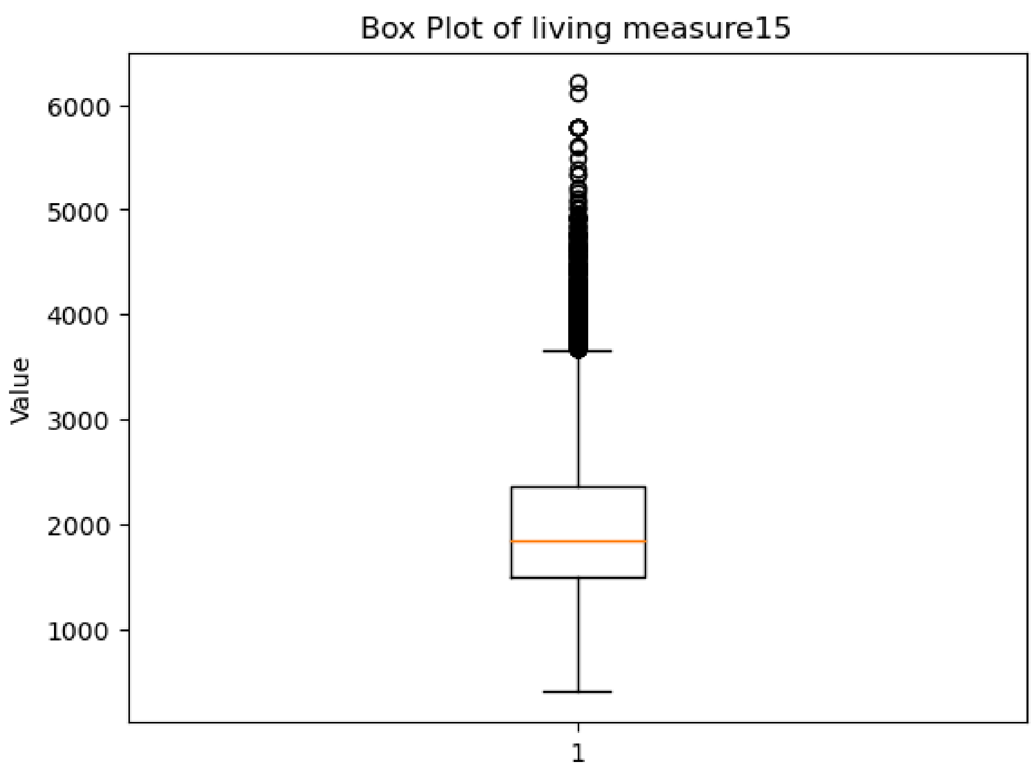




In [23]:

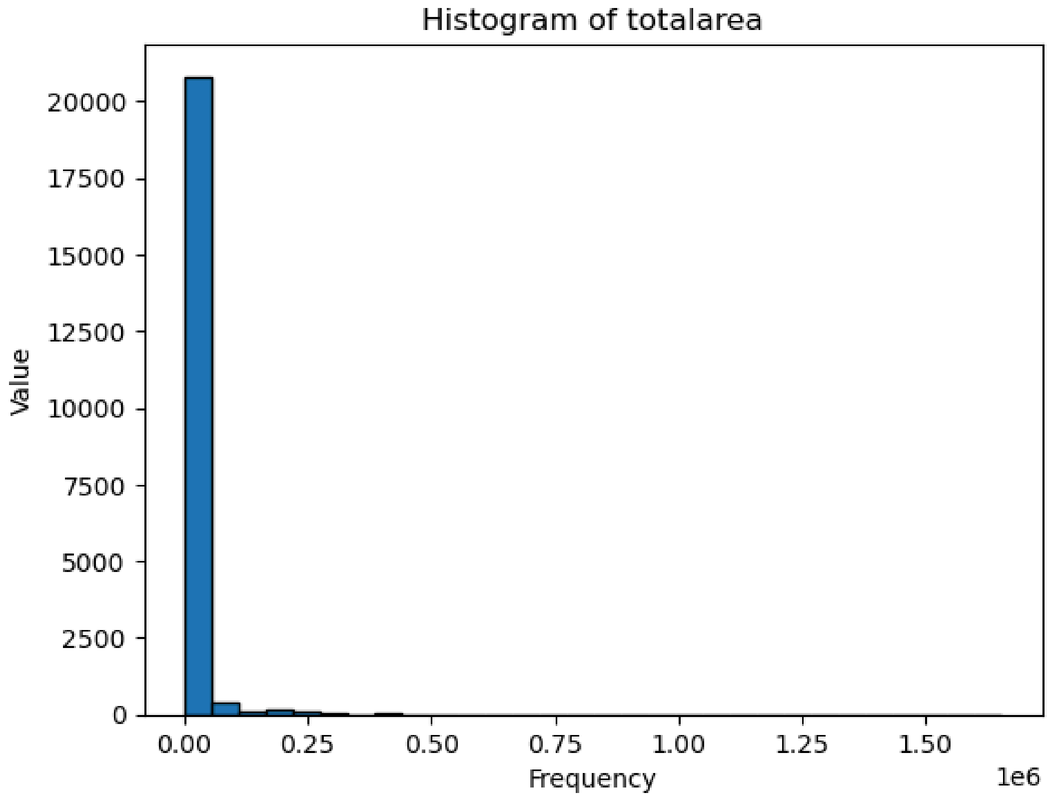
```
plt.hist(data['living_measure15'], bins=30, edgecolor='black')
plt.title('Histogram of living measure15')
plt.xlabel('Frequency')
plt.ylabel('Value')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/hist of living measure15.png")
plt.show()


plt.boxplot(data['living_measure15'])
plt.title('Box Plot of living measure15')
plt.ylabel('Value')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/box of living measure15.png")
plt.show()
```

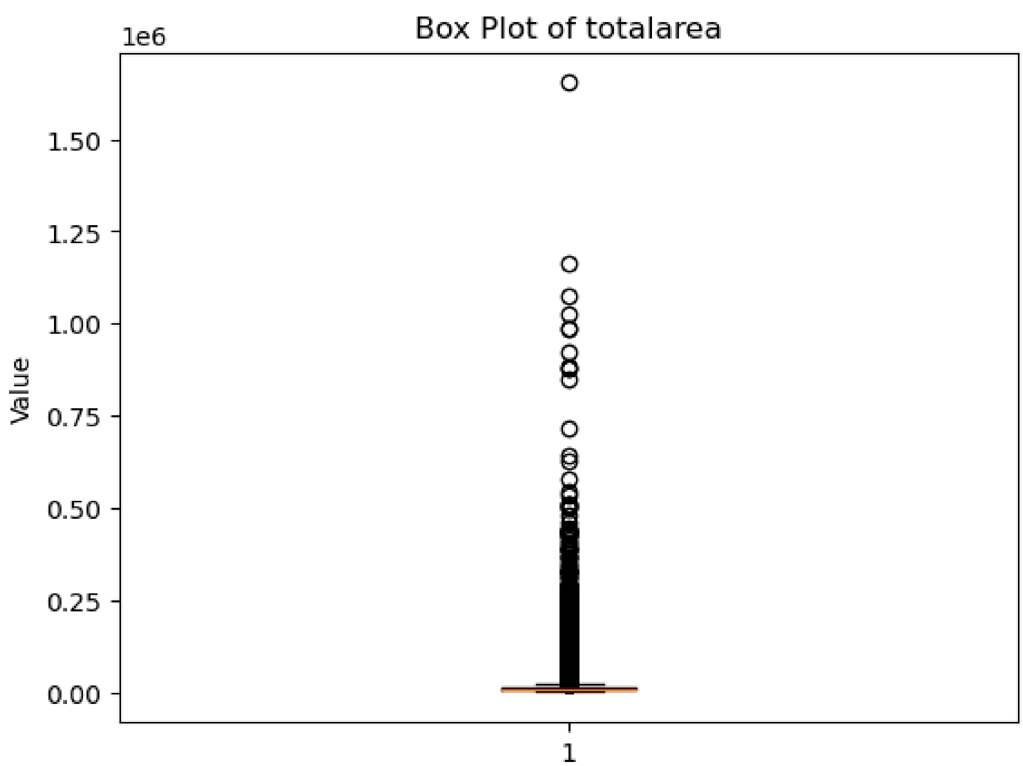




In [24]:

```
plt.hist(data['total_area'], bins=30, edgecolor='black')
plt.title('Histogram of totalarea')
plt.xlabel('Frequency')
plt.ylabel('Value')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/hist of total area.png")
plt.show()


plt.boxplot(data['total_area'])
plt.title('Box Plot of totalarea')
plt.ylabel('Value')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/box of total area.png")
plt.show()
```

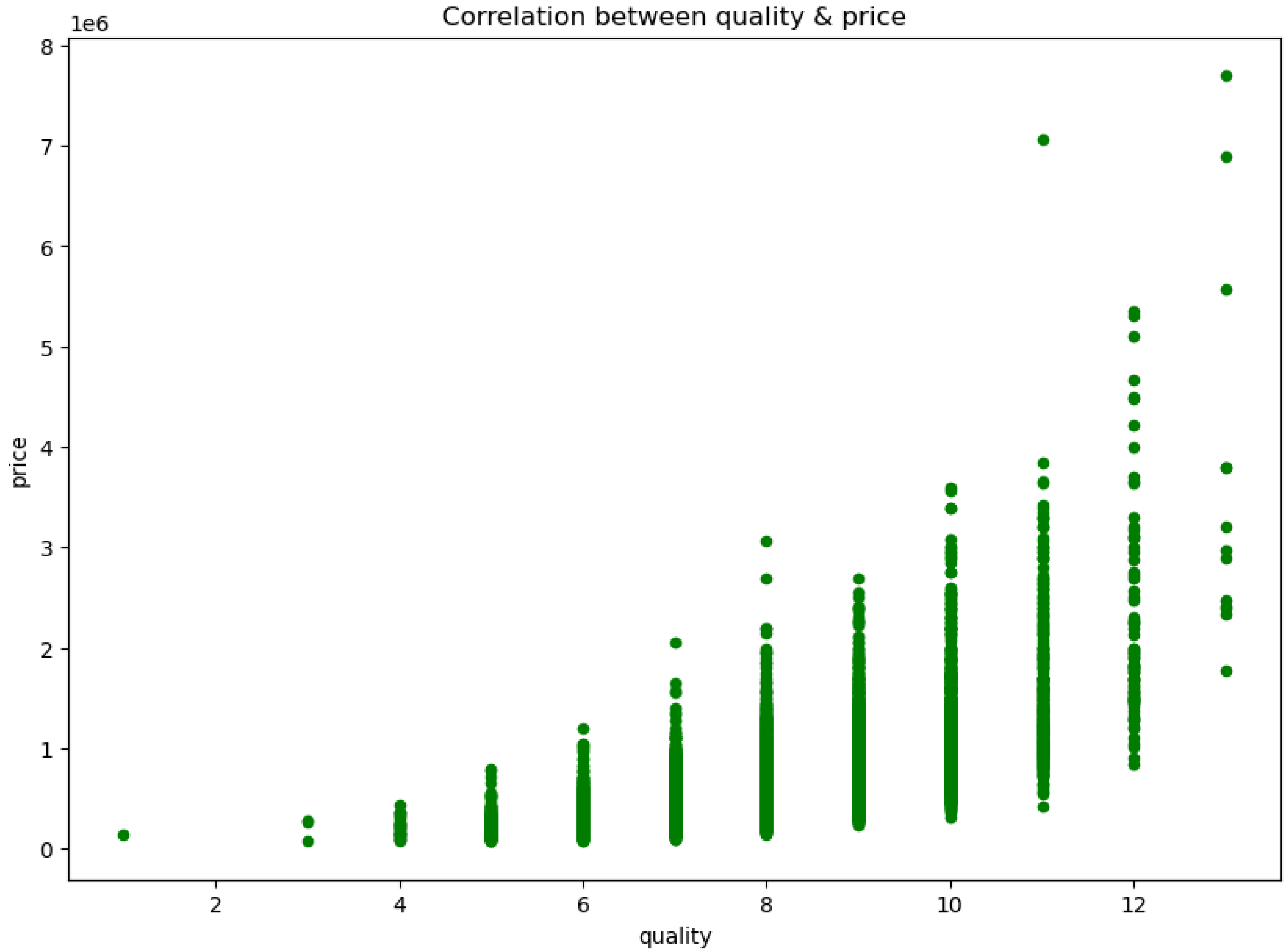




In [25]:
```
data.plot(kind='scatter',x= 'quality',y = 'price',c='g',figsize=(10,7))

plt.title('Correlation between quality & price')
```

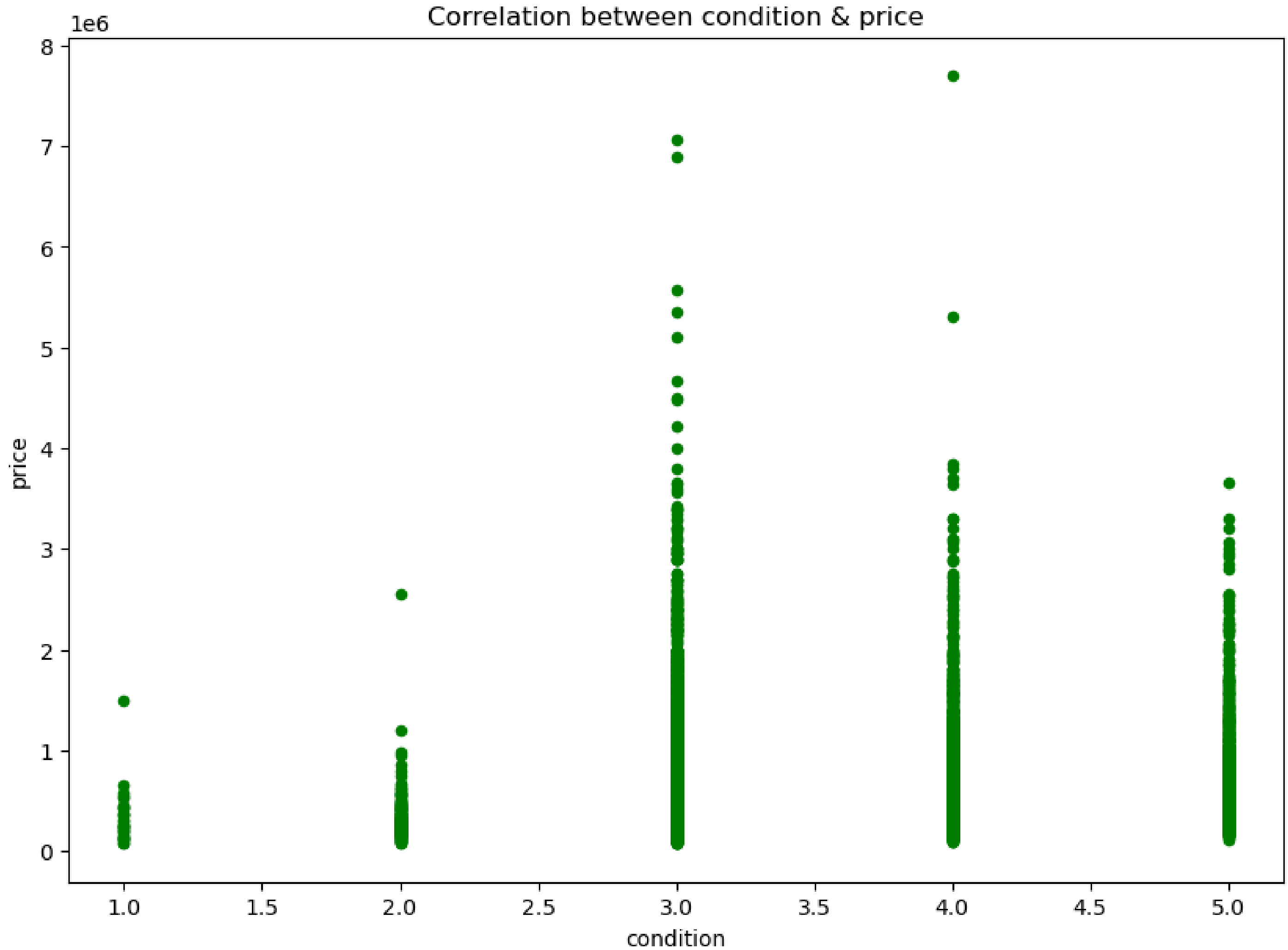
Out[25]: Text(0.5, 1.0, 'Correlation between quality & price')

Loading [MathJax]/extensions/Safe.js



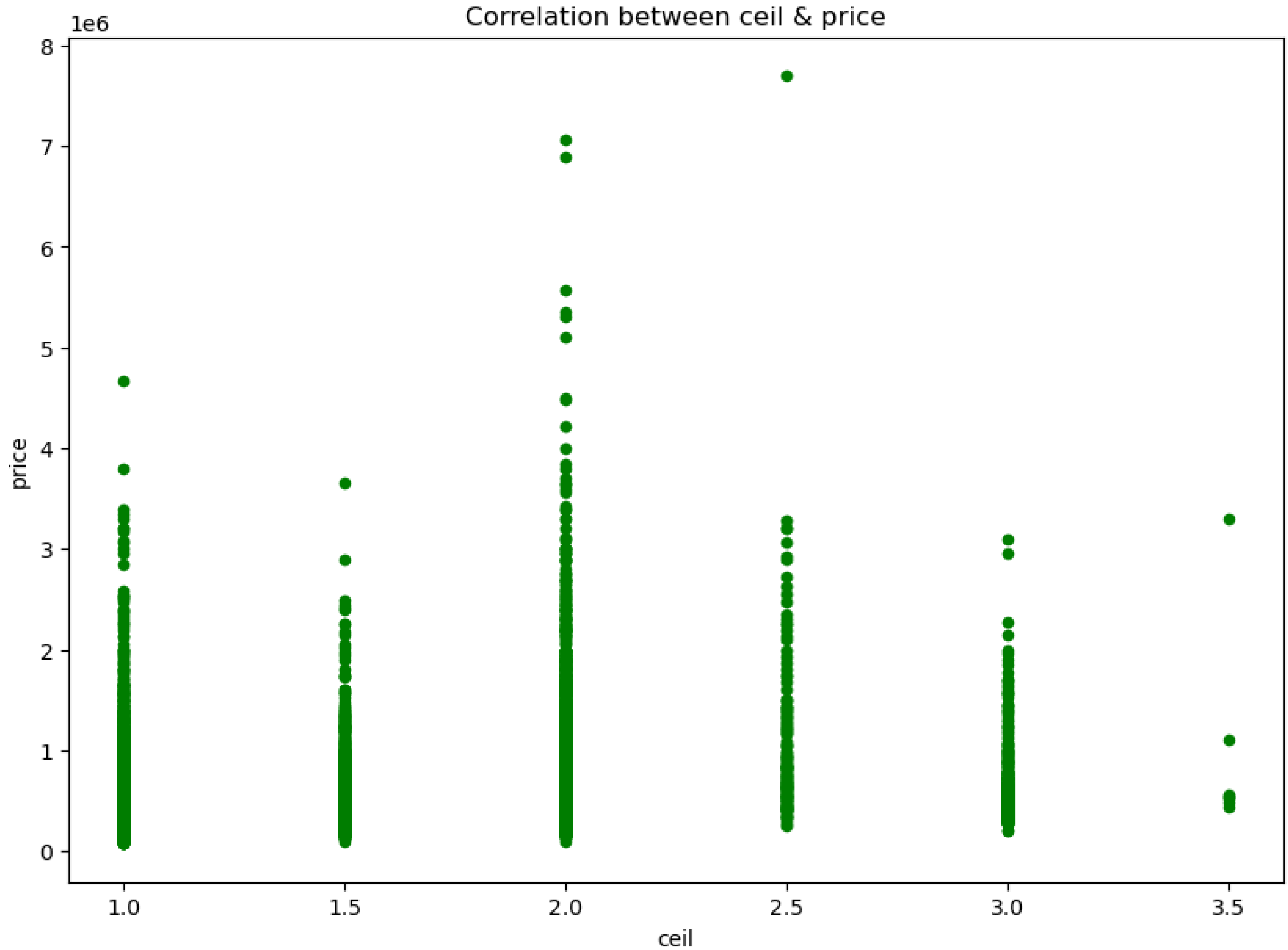
In [26]:
```
data.plot(kind='scatter',x= 'condition',y = 'price',c='g',figsize=(10,7))

plt.title('Correlation between condition & price')
```

Out[26]:
```
Text(0.5, 1.0, 'Correlation between condition & price')
```

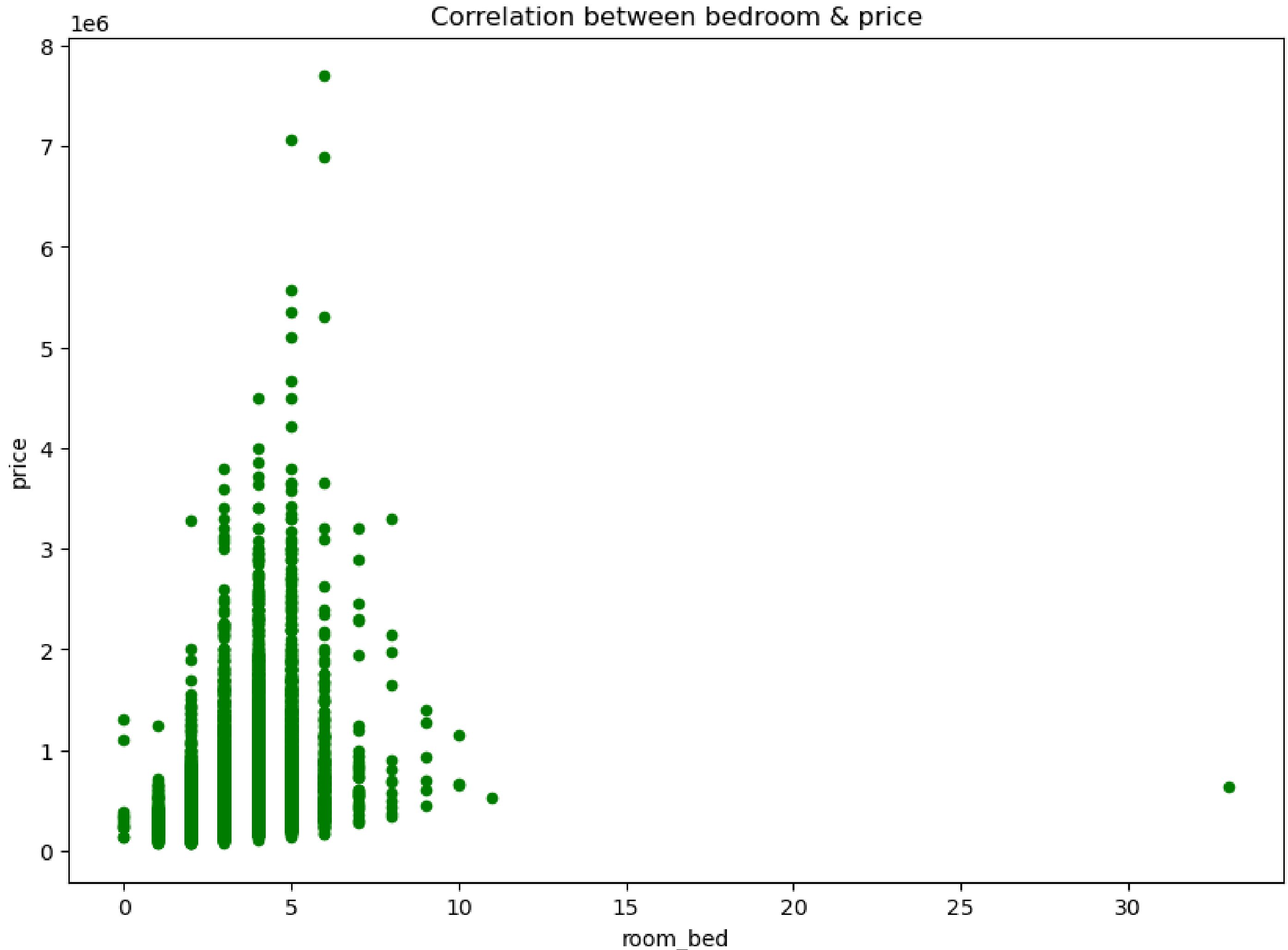


In [27]:
```
data.plot(kind='scatter',x= 'ceil',y = 'price',c='g',figsize=(10,7))

plt.title('Correlation between ceil & price')
```

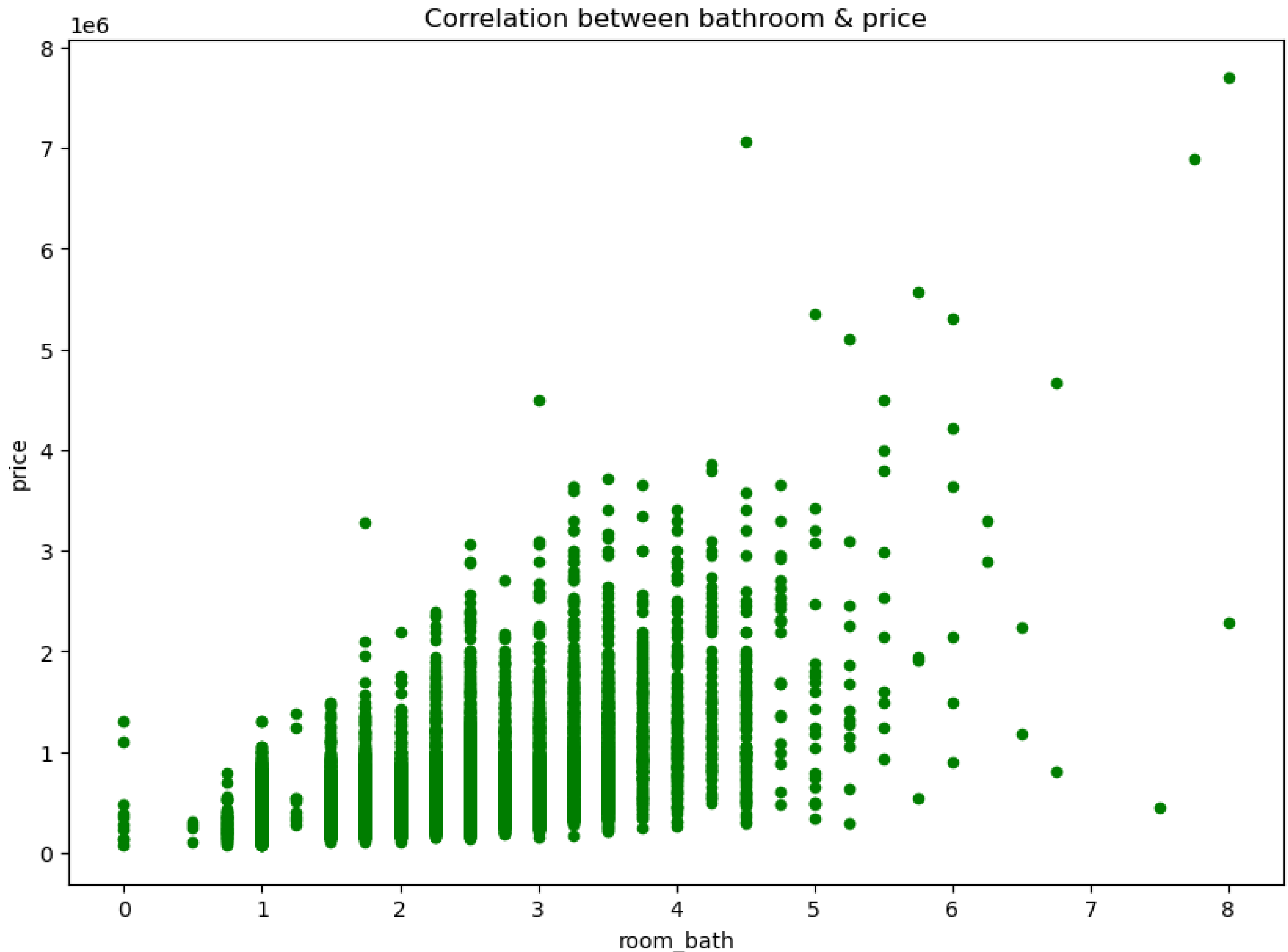
Out[27]:
```
Text(0.5, 1.0, 'Correlation between ceil & price')
```



In [28]:
```
data.plot(kind='scatter',x= 'room_bed',y = 'price',c='g',figsize=(10,7))

plt.title('Correlation between bedroom & price')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/scatter bedroom.png")
plt.show()
```



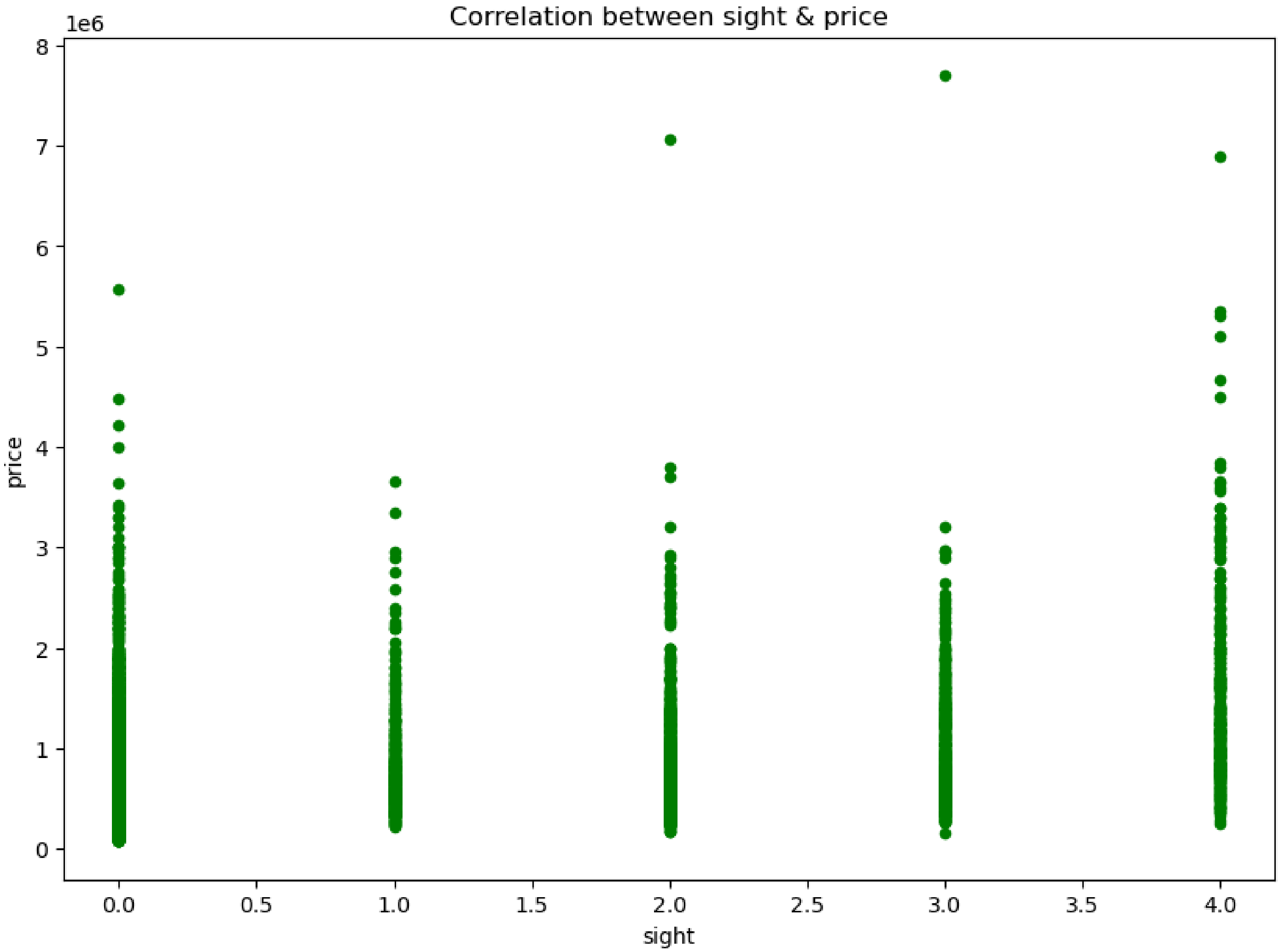
In [29]:
```
data.plot(kind='scatter',x= 'room_bath',y = 'price',c='g',figsize=(10,7))

plt.title('Correlation between bathroom & price')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/scatter bathroom.png")
plt.show()
```



In [30]:

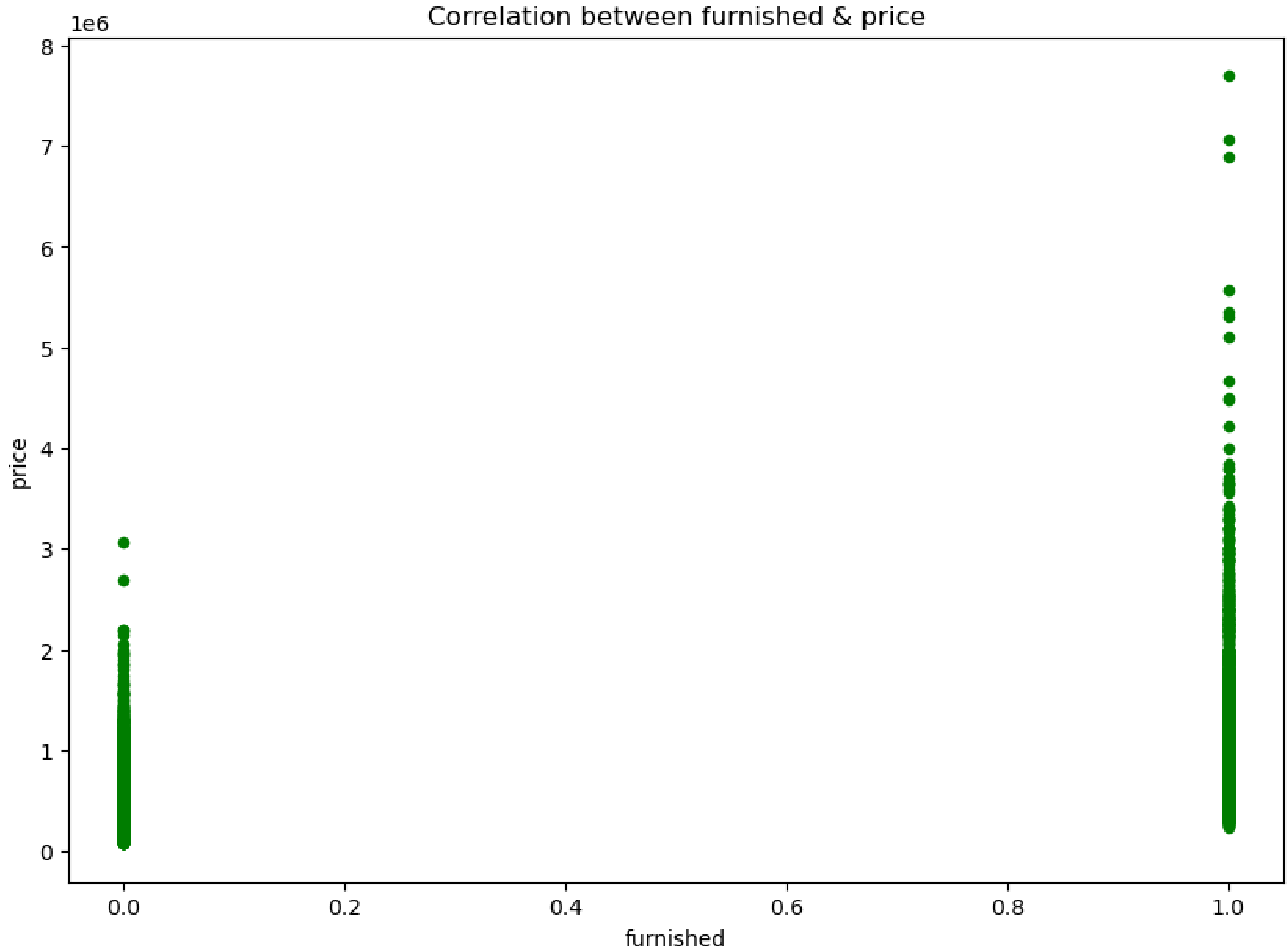
```
data.plot(kind='scatter',x= 'sight',y = 'price',c='g',figsize=(10,7))

plt.title('Correlation between sight & price')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/scatter sight.png")
plt.show()
```

Loading [MathJax]/extensions/Safe.js



In [31]:
```
data.plot(kind='scatter',x= 'furnished',y = 'price',c='g',figsize=(10,7))

plt.title('Correlation between furnished & price')
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/scatter furnished.png")
plt.show()
```

Loading [MathJax]/extensions/Safe.js



Insights :

1. in living measure and living measure15 there is high skeweness towards right.
2. we can see Total area and lot measure are highly correlated with each other.
3. we can see more outliers detected in living measure, lot measure, ceil measure, total area, basement, living measure 15, lotmeasure 15, and room bath when correlating with target variable Price.
4. when it is furnished prices are showing a bit high.
5. we can see there is no much price variations depending on the number of sights.
6. Bedrooms ranges between 3 to 6 are having mostly higher prices.
7. most of houses sold are of 2 nd floor, but the price are ranging from lower to higher of the same no of floors.
8. condition of the house which are given 3 are most sold and prices are also high.
9. quality and price are positively correlated with each other as quality number increasing price is also increasing.

Bivariate Analysis

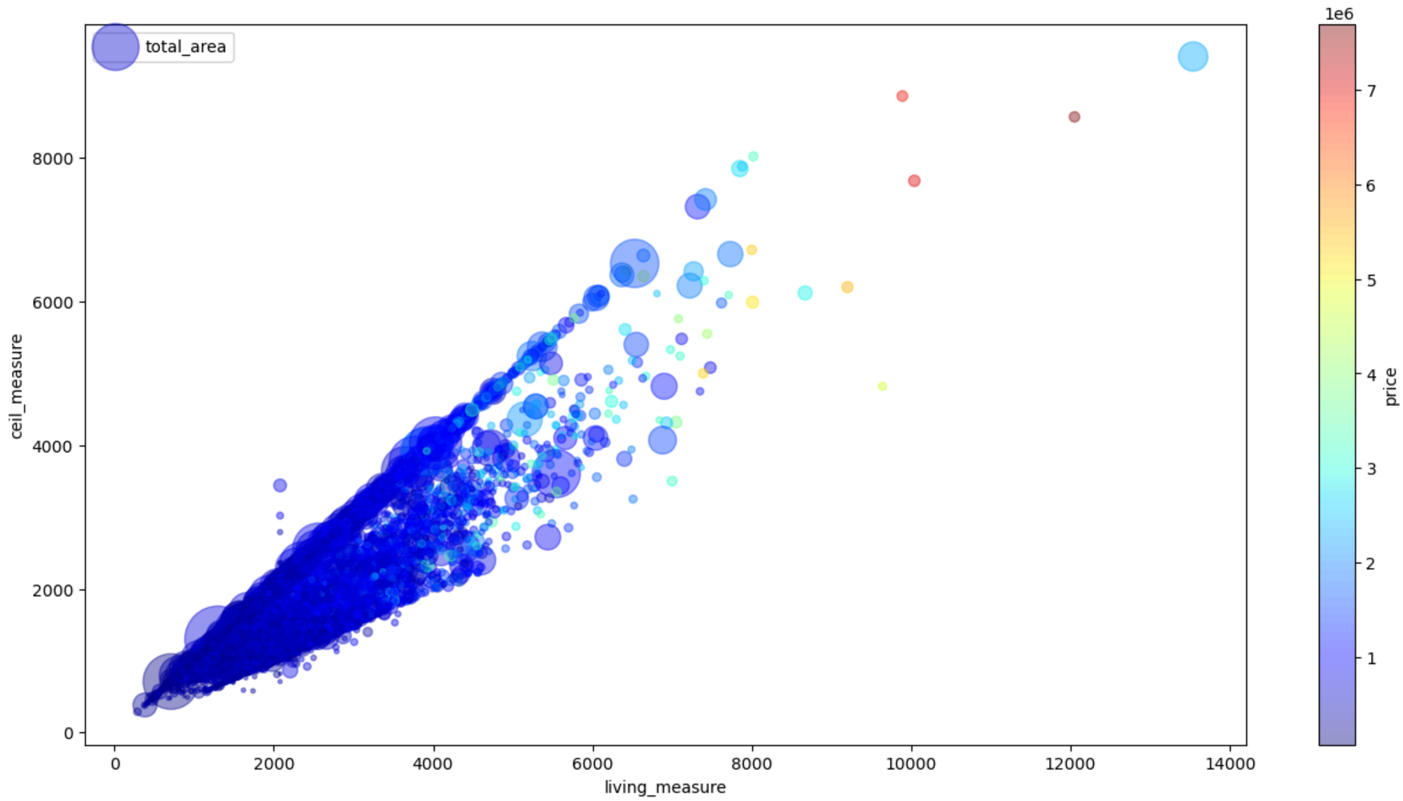
In [32]:

```
data.plot(kind='scatter',x = 'living_measure',y = 'ceil_measure',alpha= 0.4,
         s= data['total_area']/1000,
         label = 'total_area',
         c ='price',
         cmap = plt.get_cmap('jet'),colorbar = True,figsize = (16,8))
plt.legend()
```

Out[32]:

```
<matplotlib.legend.Legend at 0x232d25f2490>
```

Loading [MathJax]/extensions/Safe.js

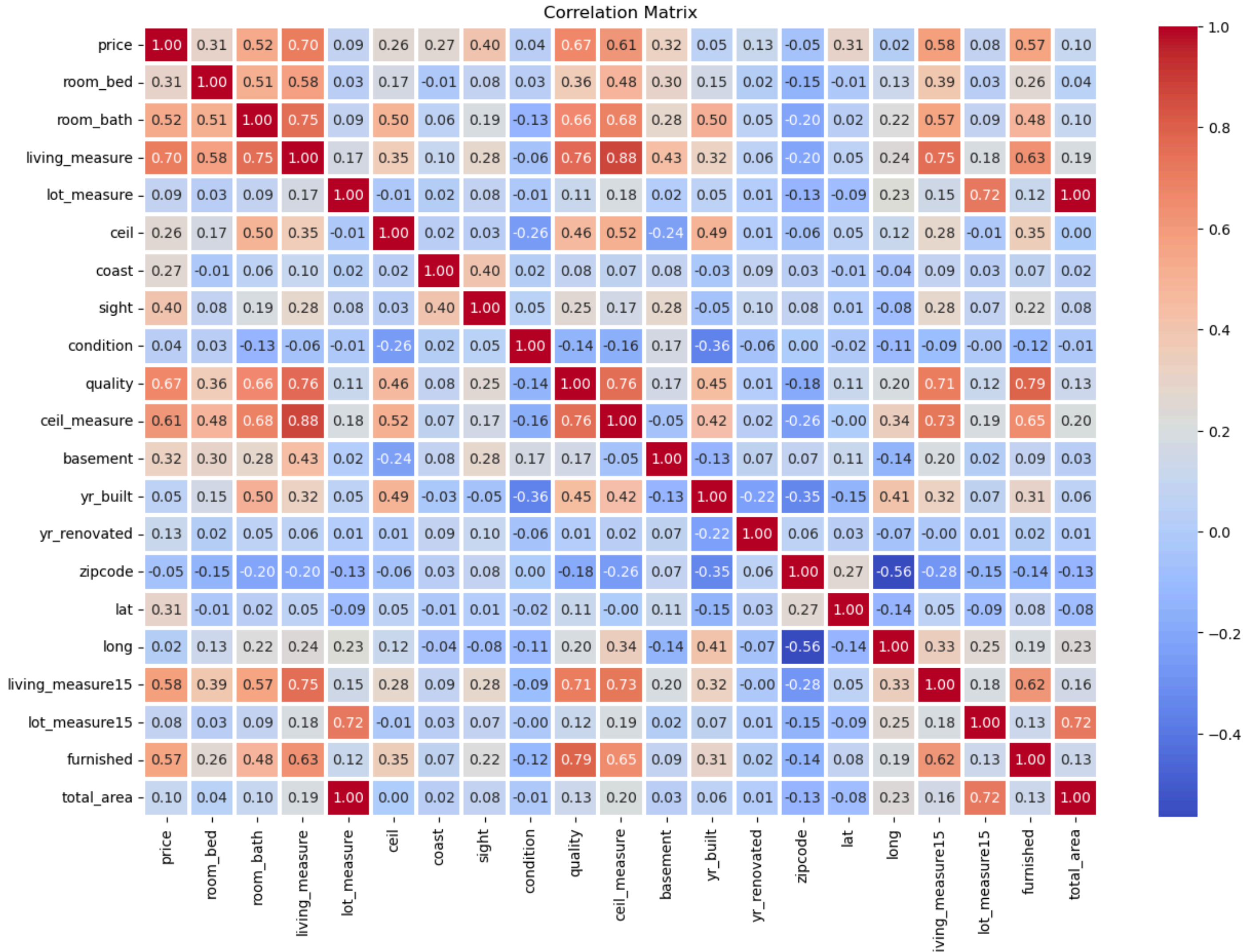


we can see best fit line among these features in correaltion with price

In [33]:
```
correlation_matrix = data.corr()
```

In [34]:
```
plt.figure(figsize=(15, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=1.5)
plt.title('Correlation Matrix')
```

Out[34]:
```
Text(0.5, 1.0, 'Correlation Matrix')
```



Insights : 1.total area and lot measure are perfectly correlated with each other. so, we can eleminate one feature among these. eliminating lotmeasure and considering totalarea.

1. room bed, room bath, living measure,quality, ceil measure, basement,lat,living measure 15, furnished are highly correlated with each other.

Treatment of outliers

In [35]:

```
df = pd.DataFrame(data)
```

In [36]:

```
# Function to cap outliers
def cap_outliers(df, lower_percentile=0.01, upper_percentile=0.99):
    df_capped = df.copy()
    for column in df_capped.columns:
        lower_cap = df_capped[column].quantile(lower_percentile)
        upper_cap = df_capped[column].quantile(upper_percentile)
        df_capped[column] = np.where(df_capped[column] < lower_cap, lower_cap, df_capped[column])
        df_capped[column] = np.where(df_capped[column] > upper_cap, upper_cap, df_capped[column])
    return df_capped

# Applying the function to cap outliers
df_capped = cap_outliers(df, lower_percentile=0.01, upper_percentile=0.99)

print("Original DataFrame:")
print(df)

print("\nDataFrame after capping outliers:")
print(df_capped)
```

```
Original DataFrame:
              price  room_bed  room_bath  living_measure  lot_measure    ceil  \
0      600000.00000   4.00000    1.75000      3050.00000   9440.00000 1.00000
1      190000.00000   2.00000    1.00000       670.00000   3101.00000 1.00000
2      735000.00000   4.00000    2.75000      3040.00000   2415.00000 2.00000
3      257000.00000   3.00000    2.50000      1740.00000   3721.00000 2.00000
4      450000.00000   2.00000    1.00000      1120.00000   4590.00000 1.00000
...             ...       ...        ...             ...          ...     ...
21608  685530.00000   4.00000    2.50000      3130.00000  60467.00000 2.00000
21609  535000.00000   2.00000    1.00000      1030.00000   4841.00000 1.00000
21610  998000.00000   3.00000    3.75000      3710.00000  34412.00000 2.00000
21611  262000.00000   4.00000    2.50000      1560.00000   7800.00000 2.00000
21612 1150000.00000   4.00000    2.50000      1940.00000   4875.00000 2.00000

        coast   sight  condition  quality  ...   basement   yr_built  \
0     0.00000 0.00000    3.00000  8.00000  ... 1250.00000 1966.00000
1     0.00000 0.00000    4.00000  6.00000  ...    0.00000 1948.00000
2     1.00000 4.00000    3.00000  8.00000  ...    0.00000 1966.00000
3     0.00000 0.00000    3.00000  8.00000  ...    0.00000 2009.00000
4     0.00000 0.00000    3.00000  7.00000  ...    0.00000 1924.00000
...       ...     ...        ...      ...  ...        ...        ...
21608 0.00000 0.00000    3.00000  9.00000  ...    0.00000 1996.00000
21609 0.00000 0.00000    3.00000  7.00000  ...  110.00000 1939.00000
21610 0.00000 0.00000    3.00000 10.00000  ...  800.00000 1978.00000
21611 0.00000 0.00000    3.00000  7.00000  ...    0.00000 1997.00000
21612 0.00000 0.00000    4.00000  9.00000  ...    0.00000 1925.00000

       yr_renovated     zipcode      lat       long  living_measure15  \
0           0.00000 98034.00000 47.72280 -122.18300        2020.00000
1           0.00000 98118.00000 47.55460 -122.27400        1660.00000
2           0.00000 98118.00000 47.51880 -122.25600        2620.00000
3           0.00000 98002.00000 47.33630 -122.21300        2030.00000
4           0.00000 98118.00000 47.56630 -122.28500        1120.00000
...             ...         ...      ...        ...               ...
21608       0.00000 98014.00000 47.66180 -121.96200        2780.00000
21609       0.00000 98103.00000 47.68600 -122.34100        1530.00000
21610       0.00000 98075.00000 47.58880 -122.04000        2390.00000
21611       0.00000 98168.00000 47.51400 -122.31600        1160.00000
21612       0.00000 98112.00000 47.64270 -122.30400        1790.00000

       lot_measure15  furnished  total_area
0         8660.00000    0.00000 12490.00000
1         4100.00000    0.00000  3771.00000
2         2433.00000    0.00000  5455.00000
3         3794.00000    0.00000  5461.00000
4         5100.00000    0.00000  5710.00000
...              ...        ...         ...
21608    44224.00000    1.00000 63597.00000
21609     4944.00000    0.00000  5871.00000
21610    34412.00000    1.00000 38122.00000
21611     7800.00000    0.00000  9360.00000
21612     4875.00000    1.00000  6815.00000

[21613 rows x 21 columns]

DataFrame after capping outliers:
              price  room_bed  room_bath  living_measure  lot_measure    ceil  \
0      600000.00000   4.00000    1.75000      3050.00000   9440.00000 1.00000
1      190000.00000   2.00000    1.00000       720.00000   3101.00000 1.00000
2      735000.00000   4.00000    2.75000      3040.00000   2415.00000 2.00000
3      257000.00000   3.00000    2.50000      1740.00000   3721.00000 2.00000
4      450000.00000   2.00000    1.00000      1120.00000   4590.00000 1.00000
...             ...       ...        ...             ...          ...     ...
21608  685530.00000   4.00000    2.50000      3130.00000  60467.00000 2.00000
21609  535000.00000   2.00000    1.00000      1030.00000   4841.00000 1.00000
21610  998000.00000   3.00000    3.75000      3710.00000  34412.00000 2.00000
21611  262000.00000   4.00000    2.50000      1560.00000   7800.00000 2.00000
21612 1150000.00000   4.00000    2.50000      1940.00000   4875.00000 2.00000

        coast   sight  condition  quality  ...   basement   yr_built  \
0     0.00000 0.00000    3.00000  8.00000  ... 1250.00000 1966.00000
```

```
1     0.00000 0.00000    4.00000  6.00000  ...    0.00000 1948.00000
2     0.00000 4.00000    3.00000  8.00000  ...    0.00000 1966.00000
3     0.00000 0.00000    3.00000  8.00000  ...    0.00000 2009.00000
4     0.00000 0.00000    3.00000  7.00000  ...    0.00000 1924.00000
...       ...     ...        ...      ...  ...        ...        ...
21608 0.00000 0.00000    3.00000  9.00000  ...    0.00000 1996.00000
21609 0.00000 0.00000    3.00000  7.00000  ...  110.00000 1939.00000
21610 0.00000 0.00000    3.00000 10.00000  ...  800.00000 1978.00000
21611 0.00000 0.00000    3.00000  7.00000  ...    0.00000 1997.00000
21612 0.00000 0.00000    4.00000  9.00000  ...    0.00000 1925.00000

       yr_renovated     zipcode      lat       long  living_measure15  \
0           0.00000 98034.00000 47.72280 -122.18300        2020.00000
1           0.00000 98118.00000 47.55460 -122.27400        1660.00000
2           0.00000 98118.00000 47.51880 -122.25600        2620.00000
3           0.00000 98002.00000 47.33630 -122.21300        2030.00000
4           0.00000 98118.00000 47.56630 -122.28500        1120.00000
...             ...         ...      ...        ...               ...
21608       0.00000 98014.00000 47.66180 -121.96200        2780.00000
21609       0.00000 98103.00000 47.68600 -122.34100        1530.00000
21610       0.00000 98075.00000 47.58880 -122.04000        2390.00000
21611       0.00000 98168.00000 47.51400 -122.31600        1160.00000
21612       0.00000 98112.00000 47.64270 -122.30400        1790.00000

       lot_measure15  furnished  total_area
0         8660.00000    0.00000 12490.00000
1         4100.00000    0.00000  3771.00000
2         2433.00000    0.00000  5455.00000
3         3794.00000    0.00000  5461.00000
4         5100.00000    0.00000  5710.00000
...              ...        ...         ...
21608    44224.00000    1.00000 63597.00000
21609     4944.00000    0.00000  5871.00000
21610    34412.00000    1.00000 38122.00000
21611     7800.00000    0.00000  9360.00000
21612     4875.00000    1.00000  6815.00000

[21613 rows x 21 columns]
```

In [37]:
```
# Removal of less correlated features with price
```

In [38]:
```
data = df_capped.drop(['lot_measure','zipcode'],axis = 'columns')
```

In [39]:
```
data.shape
```

Out[39]:
```
(21613, 19)
```

Splitting Data into Train and Test set

In [40]:
```
x = data.drop('price',axis=1)
y = data['price']
print('shape of x =',x.shape)
print('shape of y =',y.shape)
```

```
shape of x = (21613, 18)
shape of y = (21613,)
```

In [41]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.20,random_state=50)
print('shape of x_train=',x_train.shape)
print('shape of y_train=',y_train.shape)
print('shape of x_test=',x_test.shape)
print('shape of y_test=',y_test.shape)
```

```
shape of x_train= (17290, 18)
shape of y_train= (17290,)
shape of x_test= (4323, 18)
shape of y_test= (4323,)
```

Loading [MathJax]/extensions/Safe.js

In [42]:
```python
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

Feature Scaling

In [43]:
```python
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
sc.fit(x_train)
x_train = sc.transform(x_train)
x_test = sc.transform(x_test)
```

Forward Feature Selection

In [99]:
```python
np.random.seed(0)
df_train, df_test = train_test_split(data, train_size = 0.8, test_size = 0.2, random_state = 100)
```

In [100…
```python
df_train.head()
```

Out[100]:

	price	room_bed	room_bath	living_measure	ceil	coast	sight	condition	quality	ceil_measure	ba
16000	575000.00000	5.00000	1.75000	2980.00000	1.00000	0.00000	0.00000	4.00000	9.00000	2230.00000	75(
11286	325500.00000	3.00000	1.50000	1540.00000	1.00000	0.00000	0.00000	4.00000	7.00000	1190.00000	35(
3201	250000.00000	2.00000	1.00000	720.00000	1.00000	0.00000	0.00000	3.00000	6.00000	700.00000	(
11049	264000.00000	3.00000	1.50000	1470.00000	1.00000	0.00000	0.00000	4.00000	7.00000	1470.00000	(
9716	415000.00000	2.00000	1.00000	1070.00000	1.00000	0.00000	0.00000	3.00000	7.00000	1070.00000	(

In [101…
```python
y_train = df_train.pop('price')
X_train = df_train
```

In [102…
```python
X_train_1 = X_train['room_bed']
```

In [103…
```python
#Add a constant
X_train_1c = sm.add_constant(X_train_1)

 #Create a first fitted model
lr_1 = sm.OLS(y_train, X_train_1c).fit()
```

In [104…
```python
print(lr_1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.111
Model:                            OLS   Adj. R-squared:                  0.111
Method:                 Least Squares   F-statistic:                     2161.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:27:54   Log-Likelihood:            -2.4237e+05
No. Observations:               17290   AIC:                         4.847e+05
Df Residuals:                   17288   BIC:                         4.848e+05
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        1.22e+05   9107.369     13.395      0.000    1.04e+05     1.4e+05
room_bed    1.215e+05   2614.195     46.488      0.000    1.16e+05    1.27e+05
==============================================================================
Omnibus:                     6750.336   Durbin-Watson:                   1.987
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            30345.747
Skew:                           1.887   Prob(JB):                         0.00
Kurtosis:                       8.280   Cond. No.                         15.2
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [105…

```python
X_train_2 = X_train[['room_bed', 'room_bath']]
```

In [106…

```python
X_train_2c = sm.add_constant(X_train_2)
lr_2 = sm.OLS(y_train, X_train_2c).fit()
```

In [107…

```python
print(lr_2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.279
Model:                            OLS   Adj. R-squared:                  0.279
Method:                 Least Squares   F-statistic:                     3346.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:28:15   Log-Likelihood:            -2.4056e+05
No. Observations:               17290   AIC:                         4.811e+05
Df Residuals:                   17287   BIC:                         4.812e+05
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        893.3842   8421.231      0.106      0.916   -1.56e+04    1.74e+04
room_bed    3.023e+04   2759.126     10.957      0.000    2.48e+04    3.56e+04
room_bath   2.033e+05   3203.511     63.461      0.000    1.97e+05     2.1e+05
==============================================================================
Omnibus:                     5575.886   Durbin-Watson:                   1.985
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            20660.865
Skew:                           1.594   Prob(JB):                         0.00
Kurtosis:                       7.303   Cond. No.                         18.0
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [108…

```python
X_train_3 = X_train[['room_bed', 'room_bath','living_measure']]
X_train_3c = sm.add_constant(X_train_3)
lr_3 = sm.OLS(y_train, X_train_3c).fit()
print(lr_3.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.503
Model:                            OLS   Adj. R-squared:                  0.503
Method:                 Least Squares   F-statistic:                     5837.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:28:25   Log-Likelihood:            -2.3734e+05
No. Observations:               17290   AIC:                         4.747e+05
Df Residuals:                   17286   BIC:                         4.747e+05
Df Model:                           3
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
const           1.102e+05   7099.295     15.517      0.000    9.62e+04    1.24e+05
room_bed        -5.14e+04   2469.894    -20.812      0.000   -5.62e+04   -4.66e+04
room_bath       9727.9007   3446.104      2.823      0.005    2973.189    1.65e+04
living_measure   277.6246      3.143     88.319      0.000     271.463     283.786
==============================================================================
Omnibus:                     4183.636   Durbin-Watson:                   1.989
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            13475.950
Skew:                           1.226   Prob(JB):                         0.00
Kurtosis:                       6.562   Cond. No.                     9.84e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.84e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [109…

```python
X_train_4 = X_train[['room_bed', 'room_bath','living_measure','ceil']]
X_train_4c = sm.add_constant(X_train_4)
lr_4 = sm.OLS(y_train, X_train_4c).fit()
print(lr_4.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.503
Model:                            OLS   Adj. R-squared:                  0.503
Method:                 Least Squares   F-statistic:                     4382.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:28:35   Log-Likelihood:            -2.3734e+05
No. Observations:               17290   AIC:                         4.747e+05
Df Residuals:                   17285   BIC:                         4.747e+05
Df Model:                           4
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
const            1.01e+05   7786.385     12.977      0.000    8.58e+04    1.16e+05
room_bed        -5.06e+04   2485.452    -20.359      0.000   -5.55e+04   -4.57e+04
room_bath       5526.2793   3748.171      1.474      0.140   -1820.515    1.29e+04
living_measure   277.5105      3.143     88.293      0.000     271.350     283.671
ceil            1.039e+04   3650.359      2.847      0.004    3237.431    1.75e+04
==============================================================================
Omnibus:                     4200.630   Durbin-Watson:                   1.988
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            13592.397
Skew:                           1.230   Prob(JB):                         0.00
Kurtosis:                       6.580   Cond. No.                     1.08e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.08e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [110…

```python
X_train_5 = X_train[['room_bed', 'room_bath','living_measure','ceil','sight']]
X_train_5c = sm.add_constant(X_train_5)
lr_5 = sm.OLS(y_train, X_train_5c).fit()
print(lr_5.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.546
Model:                            OLS   Adj. R-squared:                  0.546
Method:                 Least Squares   F-statistic:                     4159.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:28:44   Log-Likelihood:            -2.3656e+05
No. Observations:               17290   AIC:                         4.731e+05
Df Residuals:                   17284   BIC:                         4.732e+05
Df Model:                           5
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
const           9.202e+04   7448.124     12.355      0.000    7.74e+04    1.07e+05
room_bed       -3.974e+04   2391.627    -16.618      0.000   -4.44e+04   -3.51e+04
room_bath       4525.2483   3583.805      1.263      0.207   -2499.372    1.15e+04
living_measure   247.2377      3.098     79.815      0.000     241.166     253.309
ceil            2.143e+04   3500.924      6.120      0.000    1.46e+04    2.83e+04
sight            8.89e+04   2206.160     40.296      0.000    8.46e+04    9.32e+04
==============================================================================
Omnibus:                     3515.741   Durbin-Watson:                   1.994
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            10745.182
Skew:                           1.050   Prob(JB):                         0.00
Kurtosis:                       6.241   Cond. No.                     1.09e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.09e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [111…

```
X_train_6 = X_train[['room_bed', 'room_bath','living_measure','ceil','sight','condition']]
X_train_6c = sm.add_constant(X_train_6)
lr_6 = sm.OLS(y_train, X_train_6c).fit()
print(lr_6.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.554
Model:                            OLS   Adj. R-squared:                  0.554
Method:                 Least Squares   F-statistic:                     3581.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:28:53   Log-Likelihood:            -2.3641e+05
No. Observations:               17290   AIC:                         4.728e+05
Df Residuals:                   17283   BIC:                         4.729e+05
Df Model:                           6
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
const           -8.43e+04   1.24e+04     -6.796      0.000   -1.09e+05       -6e+04
room_bed       -4.279e+04   2376.577    -18.004      0.000   -4.74e+04   -3.81e+04
room_bath       7552.8217   3556.033      2.124      0.034     582.638    1.45e+04
living_measure   246.9299      3.070     80.430      0.000     240.912     252.948
ceil            3.584e+04   3564.244     10.057      0.000    2.89e+04    4.28e+04
sight           8.663e+04   2190.288     39.552      0.000    8.23e+04    9.09e+04
condition       4.676e+04   2643.827     17.686      0.000    4.16e+04    5.19e+04
==============================================================================
Omnibus:                     3507.961   Durbin-Watson:                   1.992
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            10874.447
Skew:                           1.043   Prob(JB):                         0.00
Kurtosis:                       6.278   Cond. No.                     1.79e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.79e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [112…

```
X_train_7 = X_train[['room_bed', 'room_bath','living_measure','ceil','sight','condition','quality']]
X_train_7c = sm.add_constant(X_train_7)
```

Loading [MathJax]/extensions/Safe.js

```
lr_7 = sm.OLS(y_train, X_train_7c).fit()
print(lr_7.summary())
```

```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                  price   R-squared:                       0.606
Model:                            OLS   Adj. R-squared:                  0.606
Method:                 Least Squares   F-statistic:                     3795.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:29:04   Log-Likelihood:            -2.3534e+05
No. Observations:               17290   AIC:                         4.707e+05
Df Residuals:                   17282   BIC:                         4.708e+05
Df Model:                           7                                         
Covariance Type:            nonrobust                                         
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
const          -6.935e+05   1.73e+04    -40.054      0.000   -7.27e+05    -6.6e+05
room_bed       -2.247e+04   2274.980     -9.878      0.000   -2.69e+04    -1.8e+04
room_bath       -1.43e+04   3374.964     -4.236      0.000   -2.09e+04   -7682.333
living_measure   154.0378      3.484     44.210      0.000     147.208     160.867
ceil             324.4551   3433.372      0.095      0.925   -6405.301    7054.211
sight           7.928e+04   2065.222     38.390      0.000    7.52e+04    8.33e+04
condition       5.734e+04   2495.816     22.975      0.000    5.24e+04    6.22e+04
quality         1.042e+05   2189.002     47.610      0.000    9.99e+04    1.09e+05
==============================================================================
Omnibus:                     3780.093   Durbin-Watson:                   2.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            12543.081
Skew:                           1.099   Prob(JB):                         0.00
Kurtosis:                       6.547   Cond. No.                     2.62e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.62e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [113…

```
X_train_8 = X_train[['room_bed', 'room_bath','living_measure','ceil','sight','condition','quality',
X_train_8c = sm.add_constant(X_train_8)
lr_8 = sm.OLS(y_train, X_train_8c).fit()
print(lr_8.summary())
```

Loading [MathJax]/extensions/Safe.js

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.607
Model:                            OLS   Adj. R-squared:                  0.607
Method:                 Least Squares   F-statistic:                     3342.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:29:14   Log-Likelihood:            -2.3531e+05
No. Observations:               17290   AIC:                         4.706e+05
Df Residuals:                   17281   BIC:                         4.707e+05
Df Model:                           8
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
const          -7.09e+05   1.74e+04    -40.778      0.000   -7.43e+05   -6.75e+05
room_bed      -2.262e+04   2270.805     -9.962      0.000   -2.71e+04   -1.82e+04
room_bath     -1.867e+04   3411.491     -5.472      0.000   -2.54e+04    -1.2e+04
living_measure  180.8490      4.799     37.687      0.000     171.443     190.255
ceil           1.328e+04   3781.052      3.512      0.000    5867.369    2.07e+04
sight          7.654e+04   2088.951     36.641      0.000    7.24e+04    8.06e+04
condition      5.519e+04   2505.304     22.028      0.000    5.03e+04    6.01e+04
quality        1.074e+05   2220.837     48.380      0.000    1.03e+05    1.12e+05
ceil_measure    -37.1867      4.586     -8.109      0.000     -46.176     -28.197
==============================================================================
Omnibus:                     3872.026   Durbin-Watson:                   1.999
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            13334.087
Skew:                           1.113   Prob(JB):                         0.00
Kurtosis:                       6.682   Cond. No.                     3.47e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.47e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [114…

```
X_train_9 = X_train[['room_bed', 'room_bath','living_measure','ceil','sight','condition','quality',
X_train_9c = sm.add_constant(X_train_9)
lr_9= sm.OLS(y_train, X_train_9c).fit()
print(lr_9.summary())
```

Loading [MathJax]/extensions/Safe.js

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.608
Model:                            OLS   Adj. R-squared:                  0.608
Method:                 Least Squares   F-statistic:                     2978.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:29:25   Log-Likelihood:            -2.3529e+05
No. Observations:               17290   AIC:                         4.706e+05
Df Residuals:                   17280   BIC:                         4.707e+05
Df Model:                           9
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
const           -7.08e+05   1.74e+04    -40.748      0.000   -7.42e+05   -6.74e+05
room_bed       -2.316e+04   2271.239    -10.199      0.000   -2.76e+04   -1.87e+04
room_bath       -1.92e+04   3410.183     -5.629      0.000   -2.59e+04   -1.25e+04
living_measure    69.4764     21.332      3.257      0.001      27.663     111.290
ceil            1.349e+04   3778.230      3.570      0.000    6083.746    2.09e+04
sight           7.584e+04   2091.354     36.264      0.000    7.17e+04    7.99e+04
condition       5.517e+04   2503.299     22.040      0.000    5.03e+04    6.01e+04
quality         1.075e+05   2219.066     48.432      0.000    1.03e+05    1.12e+05
ceil_measure      74.7100     21.381      3.494      0.000      32.801     116.619
basement         113.8580     21.250      5.358      0.000      72.205     155.511
==============================================================================
Omnibus:                     3871.902   Durbin-Watson:                   1.999
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            13378.780
Skew:                           1.112   Prob(JB):                         0.00
Kurtosis:                       6.692   Cond. No.                     3.49e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.49e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [115…

```python
X_train_10 = X_train[['room_bed', 'room_bath','living_measure','ceil','sight','condition','quality'
X_train_10c = sm.add_constant(X_train_10)
lr_10 = sm.OLS(y_train, X_train_10c).fit()
print(lr_10.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.663
Model:                            OLS   Adj. R-squared:                  0.663
Method:                 Least Squares   F-statistic:                     3402.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:29:35   Log-Likelihood:            -2.3398e+05
No. Observations:               17290   AIC:                         4.680e+05
Df Residuals:                   17279   BIC:                         4.681e+05
Df Model:                          10
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
const           5.856e+06   1.24e+05     47.048      0.000    5.61e+06     6.1e+06
room_bed       -2.903e+04   2108.392    -13.769      0.000   -3.32e+04   -2.49e+04
room_bath       4.029e+04   3353.371     12.015      0.000    3.37e+04    4.69e+04
living_measure    54.4814     19.778      2.755      0.006      15.715      93.247
ceil            3.517e+04   3526.171      9.974      0.000    2.83e+04    4.21e+04
sight           6.162e+04   1957.106     31.484      0.000    5.78e+04    6.55e+04
condition       1.866e+04   2420.041      7.712      0.000    1.39e+04    2.34e+04
quality         1.262e+05   2086.904     60.450      0.000    1.22e+05     1.3e+05
ceil_measure      76.9001     19.821      3.880      0.000      38.049     115.751
basement          85.5015     19.707      4.339      0.000      46.874     124.129
yr_built       -3389.9601     63.741    -53.183      0.000   -3514.899   -3265.021
==============================================================================
Omnibus:                     3833.182   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            15747.143
Skew:                           1.045   Prob(JB):                         0.00
Kurtosis:                       7.182   Cond. No.                     3.15e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.15e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [116…

```
X_train_11 = X_train[['room_bed', 'room_bath','living_measure','ceil','sight','condition','quality'
X_train_11c = sm.add_constant(X_train_11)
lr_11 = sm.OLS(y_train, X_train_11c).fit()
print(lr_11.summary())
```

Loading [MathJax]/extensions/Safe.js

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.663
Model:                            OLS   Adj. R-squared:                  0.663
Method:                 Least Squares   F-statistic:                     3097.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:29:45   Log-Likelihood:            -2.3398e+05
No. Observations:               17290   AIC:                         4.680e+05
Df Residuals:                   17278   BIC:                         4.681e+05
Df Model:                          11
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
const           5.683e+06   1.32e+05     43.114      0.000    5.42e+06    5.94e+06
room_bed       -2.875e+04   2108.704    -13.633      0.000   -3.29e+04   -2.46e+04
room_bath       3.848e+04   3382.904     11.376      0.000    3.19e+04    4.51e+04
living_measure    55.1685     19.770      2.791      0.005      16.417      93.920
ceil            3.466e+04   3527.027      9.827      0.000    2.77e+04    4.16e+04
sight           6.121e+04   1959.041     31.243      0.000    5.74e+04     6.5e+04
condition       2.036e+04   2456.808      8.288      0.000    1.55e+04    2.52e+04
quality         1.261e+05   2086.029     60.463      0.000    1.22e+05     1.3e+05
ceil_measure      76.1737     19.813      3.845      0.000      37.338     115.010
basement          84.9765     19.699      4.314      0.000      46.365     123.588
yr_built       -3304.0119     67.311    -49.086      0.000   -3435.949   -3172.075
yr_renovated      14.6419      3.698      3.959      0.000       7.393      21.891
==============================================================================
Omnibus:                     3808.060   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            15590.469
Skew:                           1.039   Prob(JB):                         0.00
Kurtosis:                       7.162   Cond. No.                     3.34e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.34e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [117…

```
X_train_12 = X_train[['room_bed', 'room_bath','living_measure','ceil','sight','condition','quality'
X_train_12c = sm.add_constant(X_train_12)
lr_12 = sm.OLS(y_train, X_train_12c).fit()
print(lr_12.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.719
Model:                            OLS   Adj. R-squared:                  0.718
Method:                 Least Squares   F-statistic:                     3677.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:29:56   Log-Likelihood:            -2.3243e+05
No. Observations:               17290   AIC:                         4.649e+05
Df Residuals:                   17277   BIC:                         4.650e+05
Df Model:                          12
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
const          -2.306e+07   5.08e+05    -45.354      0.000   -2.41e+07   -2.21e+07
room_bed       -2.368e+04   1930.217    -12.269      0.000   -2.75e+04   -1.99e+04
room_bath       3.568e+04   3093.789     11.532      0.000    2.96e+04    4.17e+04
living_measure    48.7150     18.079      2.695      0.007      13.279      84.151
ceil            1.239e+04   3247.838      3.814      0.000    6020.025    1.88e+04
sight           6.897e+04   1796.358     38.394      0.000    6.54e+04    7.25e+04
condition        3.14e+04   2254.553     13.925      0.000     2.7e+04    3.58e+04
quality         1.055e+05   1940.103     54.391      0.000    1.02e+05    1.09e+05
ceil_measure      97.9275     18.122      5.404      0.000      62.407     133.448
basement          74.7259     18.014      4.148      0.000      39.417     110.035
yr_built       -2409.9404     63.440    -37.988      0.000   -2534.289   -2285.592
yr_renovated      24.3296      3.386      7.185      0.000      17.693      30.967
lat             5.697e+05   9790.704     58.191      0.000    5.51e+05    5.89e+05
==============================================================================
Omnibus:                     5132.009   Durbin-Watson:                   1.994
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            26270.408
Skew:                           1.343   Prob(JB):                         0.00
Kurtosis:                       8.408   Cond. No.                     1.41e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.41e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [118…

```python
X_train_13 = X_train[['room_bed', 'room_bath','living_measure','ceil','sight','condition','quality'
X_train_13c = sm.add_constant(X_train_13)
lr_13 = sm.OLS(y_train, X_train_13c).fit()
print(lr_13.summary())
```

Loading [MathJax]/extensions/Safe.js

```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                  price   R-squared:                       0.719
Model:                            OLS   Adj. R-squared:                  0.719
Method:                 Least Squares   F-statistic:                     3405.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:30:06   Log-Likelihood:            -2.3241e+05
No. Observations:               17290   AIC:                         4.648e+05
Df Residuals:                   17276   BIC:                         4.650e+05
Df Model:                          13                                         
Covariance Type:            nonrobust                                         
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
const          -3.188e+07   1.45e+06    -21.966      0.000   -3.47e+07    -2.9e+07
room_bed       -2.382e+04   1928.041    -12.354      0.000   -2.76e+04      -2e+04
room_bath       3.523e+04   3090.884     11.398      0.000    2.92e+04    4.13e+04
living_measure    49.6604     18.058      2.750      0.006      14.266      85.055
ceil            8199.8044   3307.525      2.479      0.013    1716.720    1.47e+04
sight           6.814e+04   1798.746     37.883      0.000    6.46e+04    7.17e+04
condition       3.227e+04   2255.871     14.303      0.000    2.78e+04    3.67e+04
quality         1.043e+05   1947.671     53.527      0.000       1e+05    1.08e+05
ceil_measure     102.5139     18.114      5.659      0.000      67.009     138.019
basement          72.2961     17.997      4.017      0.000      37.021     107.571
yr_built       -2265.3342     67.170    -33.726      0.000   -2396.994   -2133.675
yr_renovated      25.2292      3.385      7.454      0.000      18.595      31.864
lat             5.678e+05   9783.554     58.037      0.000    5.49e+05    5.87e+05
long           -7.064e+04   1.09e+04     -6.488      0.000    -9.2e+04   -4.93e+04
==============================================================================
Omnibus:                     5026.849   Durbin-Watson:                   1.997
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            25297.869
Skew:                           1.319   Prob(JB):                         0.00
Kurtosis:                       8.307   Cond. No.                     4.03e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.03e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [119…

```
X_train_14 = X_train[['room_bed', 'room_bath','living_measure','ceil','sight','condition','quality'
X_train_14c = sm.add_constant(X_train_14)
lr_14 = sm.OLS(y_train, X_train_14c).fit()
print(lr_14.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.723
Model:                            OLS   Adj. R-squared:                  0.722
Method:                 Least Squares   F-statistic:                     3214.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:30:15   Log-Likelihood:            -2.3231e+05
No. Observations:               17290   AIC:                         4.646e+05
Df Residuals:                   17275   BIC:                         4.648e+05
Df Model:                          14
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const             -3.595e+07   1.47e+06    -24.443      0.000   -3.88e+07   -3.31e+07
room_bed          -2.338e+04   1917.078    -12.195      0.000   -2.71e+04   -1.96e+04
room_bath          3.562e+04   3073.030     11.590      0.000    2.96e+04    4.16e+04
living_measure       33.6767     17.988      1.872      0.061      -1.581      68.934
ceil               1.444e+04   3317.239      4.353      0.000    7938.195    2.09e+04
sight               6.41e+04   1810.553     35.405      0.000    6.06e+04    6.77e+04
condition          3.353e+04   2244.505     14.939      0.000    2.91e+04    3.79e+04
quality            9.624e+04   2016.021     47.740      0.000    9.23e+04       1e+05
ceil_measure         97.7631     18.012      5.428      0.000      62.458     133.068
basement             76.4728     17.894      4.274      0.000      41.398     111.547
yr_built          -2237.9927     66.807    -33.500      0.000   -2368.940   -2107.045
yr_renovated         27.8113      3.370      8.253      0.000      21.206      34.417
lat                5.632e+05   9732.054     57.869      0.000    5.44e+05    5.82e+05
long              -1.053e+05   1.11e+04     -9.490      0.000   -1.27e+05   -8.35e+04
living_measure15     46.7327      3.275     14.272      0.000      40.314      53.151
==============================================================================
Omnibus:                     5059.307   Durbin-Watson:                   1.997
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            25422.964
Skew:                           1.328   Prob(JB):                         0.00
Kurtosis:                       8.313   Cond. No.                     4.76e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.76e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [120…

```
X_train_15 = X_train[['room_bed', 'room_bath','living_measure','ceil','sight','condition','quality'
X_train_15c = sm.add_constant(X_train_15)
lr_15 = sm.OLS(y_train, X_train_15c).fit()
print(lr_15.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.723
Model:                            OLS   Adj. R-squared:                  0.722
Method:                 Least Squares   F-statistic:                     2999.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:30:25   Log-Likelihood:            -2.3231e+05
No. Observations:               17290   AIC:                         4.646e+05
Df Residuals:                   17274   BIC:                         4.648e+05
Df Model:                          15
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const            -3.568e+07   1.51e+06    -23.621      0.000   -3.86e+07   -3.27e+07
room_bed         -2.352e+04   1925.178    -12.216      0.000   -2.73e+04   -1.97e+04
room_bath         3.551e+04   3075.740     11.546      0.000    2.95e+04    4.15e+04
living_measure      34.3605     18.008      1.908      0.056      -0.937      69.658
ceil              1.422e+04   3328.755      4.272      0.000    7695.530    2.07e+04
sight             6.417e+04   1812.512     35.403      0.000    6.06e+04    6.77e+04
condition         3.356e+04   2244.838     14.950      0.000    2.92e+04     3.8e+04
quality           9.622e+04   2016.242     47.723      0.000    9.23e+04       1e+05
ceil_measure        97.4680     18.016      5.410      0.000      62.155     132.781
basement            75.9990     17.904      4.245      0.000      40.905     111.093
yr_built         -2239.3077     66.828    -33.509      0.000   -2370.297   -2108.319
yr_renovated        27.8385      3.370      8.260      0.000      21.232      34.445
lat               5.626e+05   9757.735     57.659      0.000    5.43e+05    5.82e+05
long             -1.033e+05   1.14e+04     -9.082      0.000   -1.26e+05    -8.1e+04
living_measure15    46.8205      3.276     14.290      0.000      40.398      53.243
lot_measure15       -0.0558      0.070     -0.797      0.426      -0.193       0.082
==============================================================================
Omnibus:                     5058.778   Durbin-Watson:                   1.997
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            25402.440
Skew:                           1.329   Prob(JB):                         0.00
Kurtosis:                       8.310   Cond. No.                     2.75e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.75e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [121…

```python
X_train_16 = X_train[['room_bed', 'room_bath','living_measure','ceil','sight','condition','quality'
X_train_16c = sm.add_constant(X_train_16)
lr_16 = sm.OLS(y_train, X_train_16c).fit()
print(lr_16.summary())
```

Loading [MathJax]/extensions/Safe.js

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.725
Model:                            OLS   Adj. R-squared:                  0.724
Method:                 Least Squares   F-statistic:                     2842.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:30:33   Log-Likelihood:            -2.3224e+05
No. Observations:               17290   AIC:                         4.645e+05
Df Residuals:                   17273   BIC:                         4.646e+05
Df Model:                          16
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const             -3.59e+07    1.5e+06    -23.858      0.000   -3.88e+07    -3.3e+07
room_bed         -2.184e+04   1923.077    -11.358      0.000   -2.56e+04   -1.81e+04
room_bath         3.864e+04   3075.514     12.564      0.000    3.26e+04    4.47e+04
living_measure      30.6078     17.941      1.706      0.088      -4.559      65.774
ceil              1.489e+04   3316.315      4.489      0.000    8385.145    2.14e+04
sight             6.377e+04   1805.796     35.314      0.000    6.02e+04    6.73e+04
condition         3.434e+04   2237.119     15.350      0.000       3e+04    3.87e+04
quality           8.083e+04   2403.044     33.638      0.000    7.61e+04    8.55e+04
ceil_measure        94.9307     17.947      5.290      0.000      59.753     130.109
basement            79.0144     17.837      4.430      0.000      44.053     113.976
yr_built         -2176.1410     66.788    -32.583      0.000   -2307.053   -2045.229
yr_renovated        28.5879      3.358      8.514      0.000      22.006      35.170
lat               5.644e+05   9721.056     58.062      0.000    5.45e+05    5.83e+05
long             -1.042e+05    1.13e+04     -9.201      0.000   -1.26e+05    -8.2e+04
living_measure15    44.3939      3.270     13.575      0.000      37.984      50.804
lot_measure15       -0.0647      0.070     -0.927      0.354      -0.202       0.072
furnished         6.269e+04   5376.038     11.661      0.000    5.22e+04    7.32e+04
==============================================================================
Omnibus:                     4979.300   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            25232.524
Skew:                           1.303   Prob(JB):                         0.00
Kurtosis:                       8.314   Cond. No.                     2.75e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.75e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [122…

```
X_train_17 = X_train[['room_bed', 'room_bath','living_measure','ceil','sight','condition','quality'
X_train_17c = sm.add_constant(X_train_17)
lr_17 = sm.OLS(y_train, X_train_17c).fit()
print(lr_17.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.725
Model:                            OLS   Adj. R-squared:                  0.725
Method:                 Least Squares   F-statistic:                     2677.
Date:                Sat, 15 Jun 2024   Prob (F-statistic):               0.00
Time:                        21:30:46   Log-Likelihood:            -2.3223e+05
No. Observations:               17290   AIC:                         4.645e+05
Df Residuals:                   17272   BIC:                         4.646e+05
Df Model:                          17
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const            -3.619e+07   1.51e+06    -23.994      0.000   -3.91e+07   -3.32e+07
room_bed         -2.169e+04   1923.605    -11.273      0.000   -2.55e+04   -1.79e+04
room_bath         3.865e+04   3074.953     12.569      0.000    3.26e+04    4.47e+04
living_measure      30.0065     17.939      1.673      0.094      -5.156      65.169
ceil              1.509e+04   3316.552      4.549      0.000    8586.867    2.16e+04
sight             6.374e+04   1805.501     35.303      0.000    6.02e+04    6.73e+04
condition         3.452e+04   2237.695     15.427      0.000    3.01e+04    3.89e+04
quality           8.071e+04   2403.043     33.587      0.000     7.6e+04    8.54e+04
ceil_measure        94.5596     17.944      5.270      0.000      59.387     129.732
basement            78.9294     17.833      4.426      0.000      43.974     113.885
yr_built         -2168.1945     66.841    -32.438      0.000   -2299.209   -2037.180
yr_renovated        28.6399      3.357      8.531      0.000      22.059      35.220
lat               5.651e+05   9722.953     58.125      0.000    5.46e+05    5.84e+05
long             -1.062e+05   1.13e+04     -9.355      0.000   -1.28e+05   -8.39e+04
living_measure15    44.9370      3.276     13.717      0.000      38.516      51.358
lot_measure15       -0.3293      0.120     -2.740      0.006      -0.565      -0.094
furnished         6.283e+04   5375.279     11.688      0.000    5.23e+04    7.34e+04
total_area           0.2320      0.086      2.705      0.007       0.064       0.400
==============================================================================
Omnibus:                     4988.226   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            25385.528
Skew:                           1.304   Prob(JB):                         0.00
Kurtosis:                       8.333   Cond. No.                     4.57e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.57e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [ ]:

In [68]:

```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error
lr = LinearRegression()
lr_lasso = Lasso()
lr_ridge = Ridge()
```

Linear Regression

In [69]:

```
def rmse(y_test,y_pred):
    return np.sqrt(mean_squared_error(y_test,y_pred))
```

In [89]:

```
lr.fit(x_train,y_train)
lr_score = lr.score(x_test,y_test)
lr_rmse = rmse(y_test,lr.predict(x_test))
lr_score,lr_rmse
```

Out[89]:

```
(0.7254363851093831, 166817.19552602695)
```

In [90]:

```
lr_lasso.fit(x_train,y_train)
[illegible]e = lr_lasso.score(x_test,y_test)
```

```
lr_lasso_rmse = rmse(y_test,lr_lasso.predict(x_test))
lr_lasso_score,lr_lasso_rmse
```

Out[90]: (0.7254413870451293, 166815.67600077228)

In [91]:
```
lr_ridge.fit(x_train,y_train)
lr_ridge_score = lr_ridge.score(x_test,y_test)
lr_ridge_rmse = rmse(y_test,lr_ridge.predict(x_test))
lr_ridge_score,lr_ridge_rmse
```

Out[91]: (0.7254430166091722, 166815.18095669438)

Random Forest

In [92]:
```
from sklearn.ensemble import RandomForestRegressor
rfr = RandomForestRegressor()
rfr.fit(x_train,y_train)
rfr_score = rfr.score(x_test,y_test)
rfr_rmse = rmse(y_test,rfr.predict(x_test))
rfr_score,rfr_rmse
```

Out[92]: (0.8826477691506704, 109059.87575862813)

XG Boost

In [93]:
```
!pip install xgboost
import xgboost
xgb_reg = xgboost.XGBRegressor()
xgb_reg.fit(x_train,y_train)
xgb_reg_score = xgb_reg.score(x_test,y_test)
xgb_reg_rmse = rmse(y_test,xgb_reg.predict(x_test))
xgb_reg_score,xgb_reg_rmse
```

```
Requirement already satisfied: xgboost in c:\users\chand\anaconda3\lib\site-packages (2.0.3)
Requirement already satisfied: numpy in c:\users\chand\anaconda3\lib\site-packages (from xgboost)
(1.24.3)
Requirement already satisfied: scipy in c:\users\chand\anaconda3\lib\site-packages (from xgboost)
(1.11.1)
```

Out[93]: (0.8922375052641274, 104508.8676260676)

Gradient Boost

In [97]:
```
from sklearn.experimental import enable_hist_gradient_boosting  # noqa
from sklearn.ensemble import HistGradientBoostingRegressor

hgb = HistGradientBoostingRegressor()
hgb.fit(x_train, y_train)
hgb_score = hgb.score(x_test, y_test)
hgb_rmse = rmse(y_test, hgb.predict(x_test))
hgb_score, hgb_rmse
```

Out[97]: (0.9017984192400529, 99765.08180871671)

In [98]:
```
print(pd.DataFrame([{'Model':'Linear Regression','Score':lr_score,'RMSE':lr_rmse},

              {'Model':'Lasso','Score':lr_lasso_score,'RMSE':lr_lasso_rmse},

              {'Model':'Ridge','Score':lr_ridge_score,'RMSE':lr_ridge_rmse},

              {'Model':'Random Forest','Score':rfr_score,'RMSE':rfr_rmse},

              {'Model':'XG Boost','Score':xgb_reg_score,'RMSE':xgb_reg_rmse},
```

```
                {'Model':'Gradient Boost','Score':hgb_score,'RMSE':hgb_rmse}],

               columns = ['Model','Score','RMSE']))
```

```
                Model    Score          RMSE
0  Linear Regression 0.72544 166817.19553
1              Lasso 0.72544 166815.67600
2              Ridge 0.72544 166815.18096
3      Random Forest 0.88265 109059.87576
4           XG Boost 0.89224 104508.86763
5     Gradient Boost 0.90180  99765.08181
```

Cross validation

In [77]:
```
from sklearn.model_selection import cross_val_score
```

In [78]:
```
scores = cross_val_score(hgb,x_train,y_train,scoring="neg_mean_squared_error",cv=5)
rmse_score = np.sqrt(-scores)
```

In [79]:
```
rmse_score
```

Out[79]:
```
array([100351.95232281, 101954.61879501, 109014.75801357, 106848.62855815,
       101967.82212084])
```

In [80]:
```
rmse_score.mean()
```

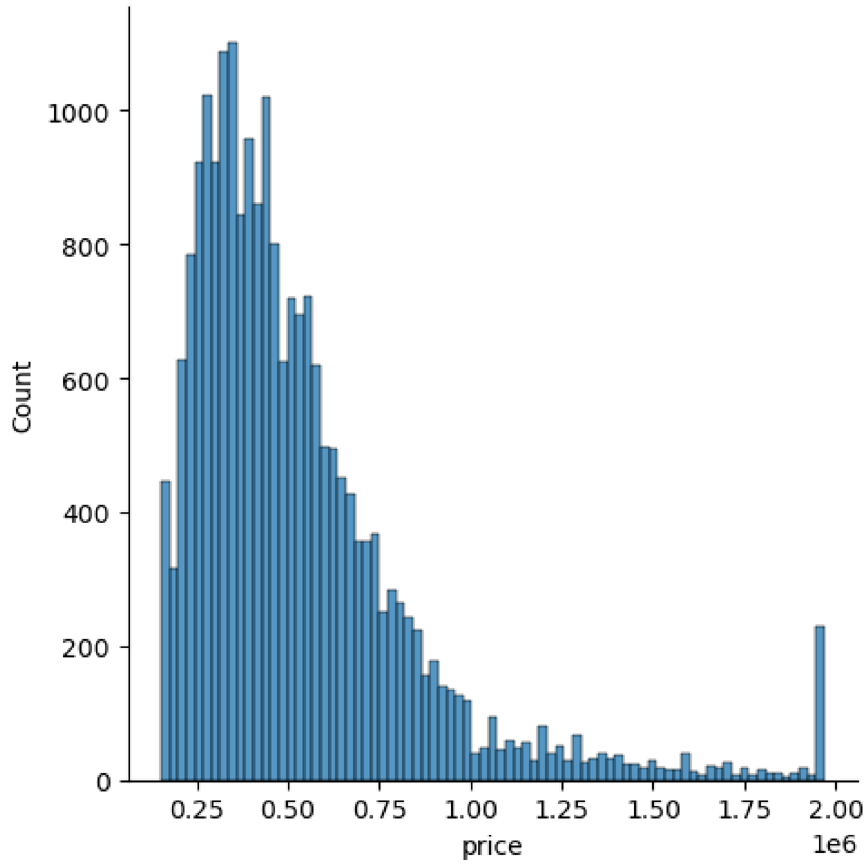
Out[80]:
```
104027.55596207861
```

In [81]:
```
from sklearn.model_selection import KFold,cross_val_score
cvs = cross_val_score(hgb,x_train,y_train,cv=10)
cvs,cvs.mean()
```

Out[81]:
```
(array([0.89825161, 0.90875261, 0.89661882, 0.90056747, 0.88493343,
        0.88812946, 0.88821818, 0.89366098, 0.88645257, 0.90860966]),
 0.8954194797028545)
```

adjusting Skewness of price with log

In [123…
```
sns.displot(data['price'])
plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/price.png")
plt.show()
```



In [124…

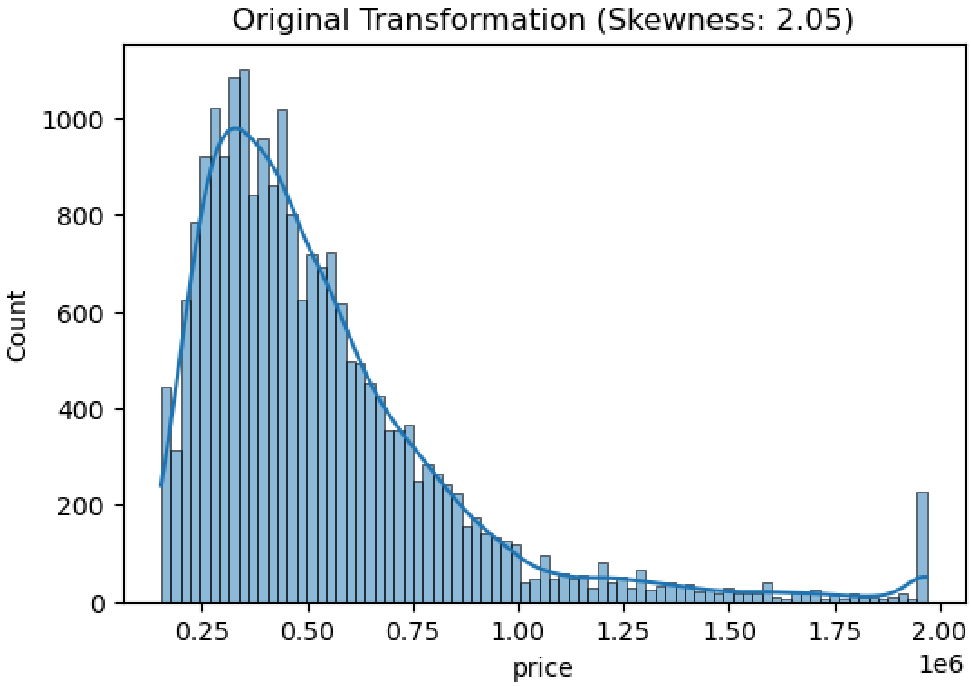
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import skew


# Assuming `y` is the target variable
y_log_transformed = np.log(y + 1)  # Adding 1 to avoid log(0)
```

In [125…

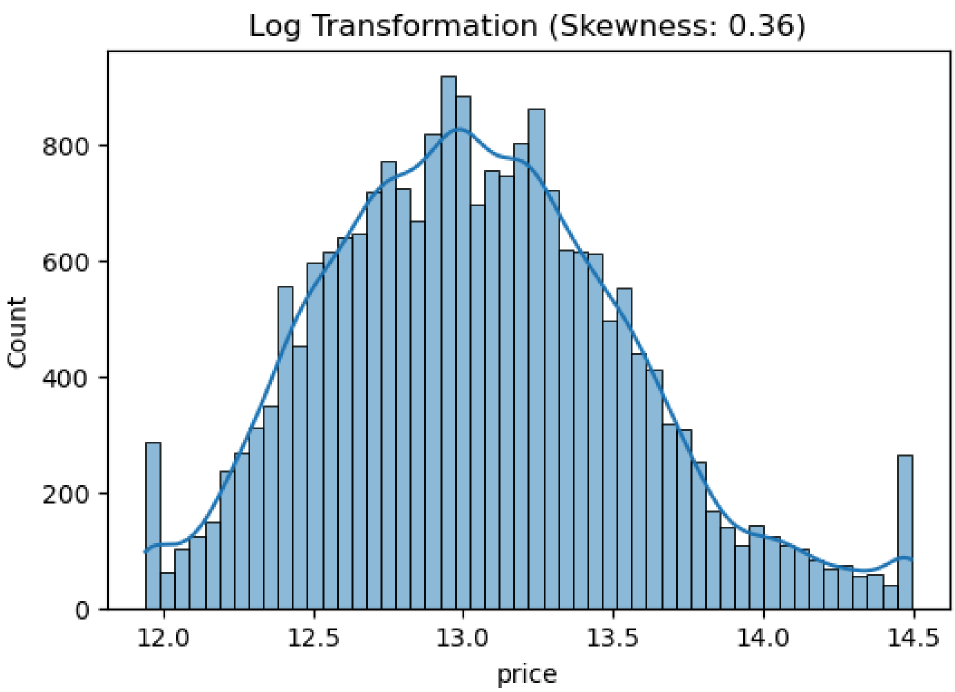
```
from scipy.stats import boxcox

y_boxcox_transformed, _ = boxcox(y + 1)  # Adding 1 to ensure all values are positive
```

In [126…

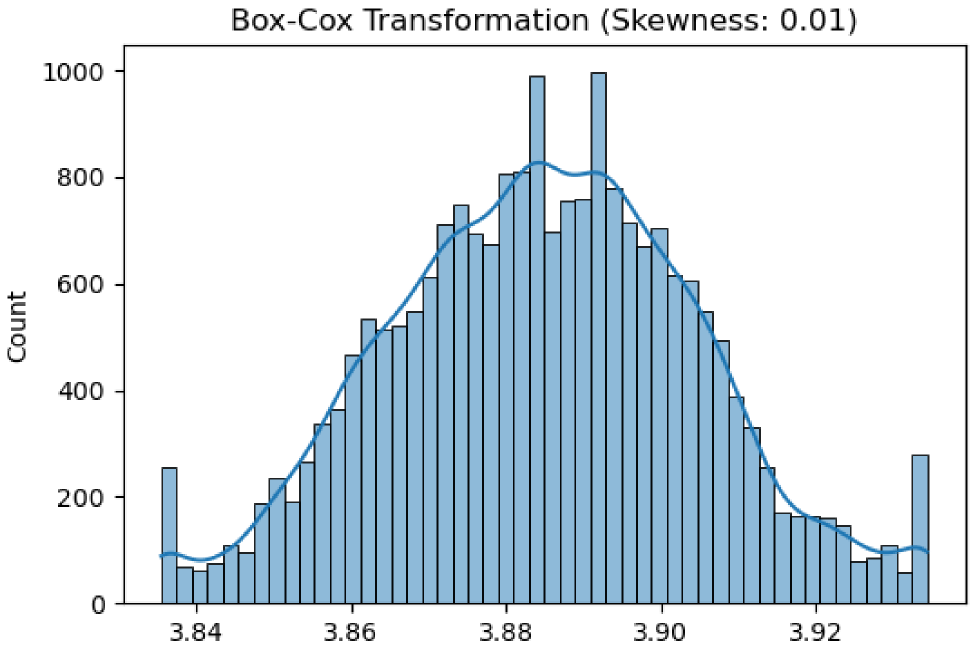
```
   transformations={'Original': y,
    'Log': y_log_transformed,
    'Box-Cox': y_boxcox_transformed,}




for name, data in transformations.items():
      plt.figure(figsize=(6, 4))
      sns.histplot(data, kde=True)
      plt.title(f'{name} Transformation (Skewness: {skew(data):.2f})')
      plt.savefig("C:/Users/chand/OneDrive/Desktop/graphs/price transformation.png")
      plt.show()
```



Original Transformation (Skewness: 2.05)
Count
price
1e6



Log Transformation (Skewness: 0.36)
Count
price


Box-Cox Transformation (Skewness: 0.01)
Count
1000
800
600
400
200
0
3.84
3.86
3.88
3.90
3.92

In [ ]: