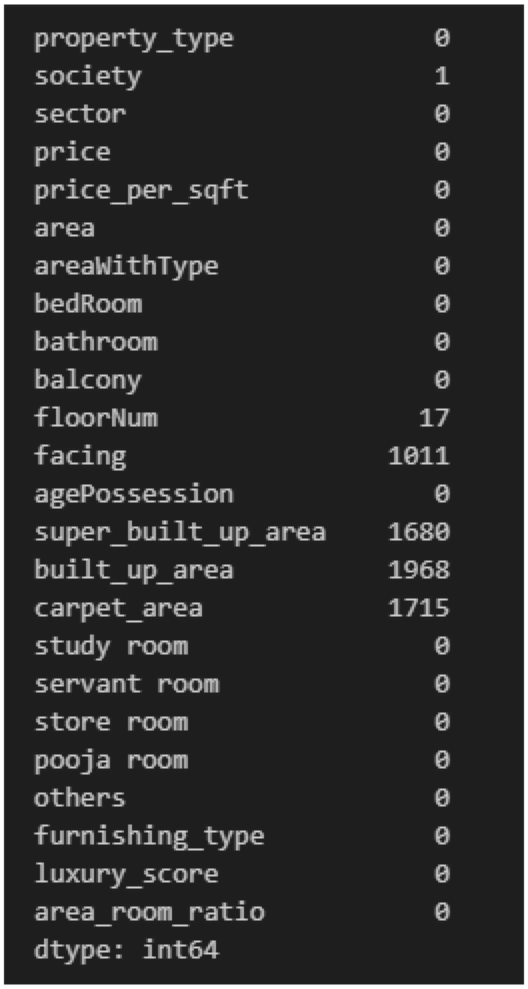
Capstone Project (Missing Value Imputation)

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
pd.set_option('display.max_columns', None)
df = pd.read_csv('gurgaon_properties_outlier_treated.csv')
```

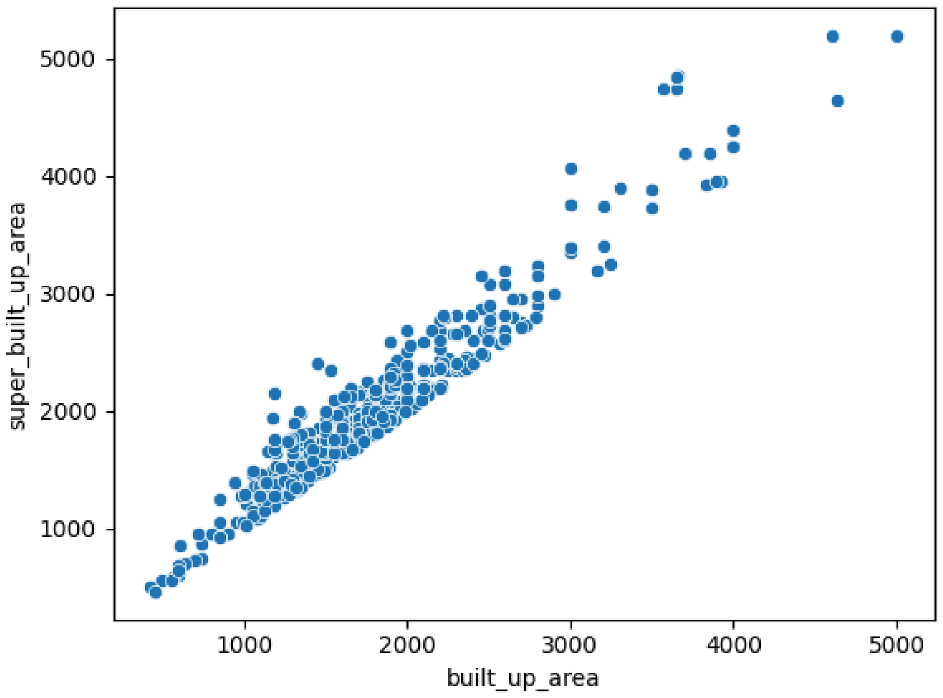
```
df.isnull().sum()
```



```
property_type              0
society                    1
sector                     0
price                      0
price_per_sqft             0
area                       0
areaWithType               0
bedRoom                    0
bathroom                   0
balcony                    0
floorNum                  17
facing                  1011
agePossession              0
super_built_up_area     1680
built_up_area           1968
carpet_area             1715
study room                 0
servant room               0
store room                 0
pooja room                 0
others                     0
furnishing_type            0
luxury_score               0
area_room_ratio            0
dtype: int64
```

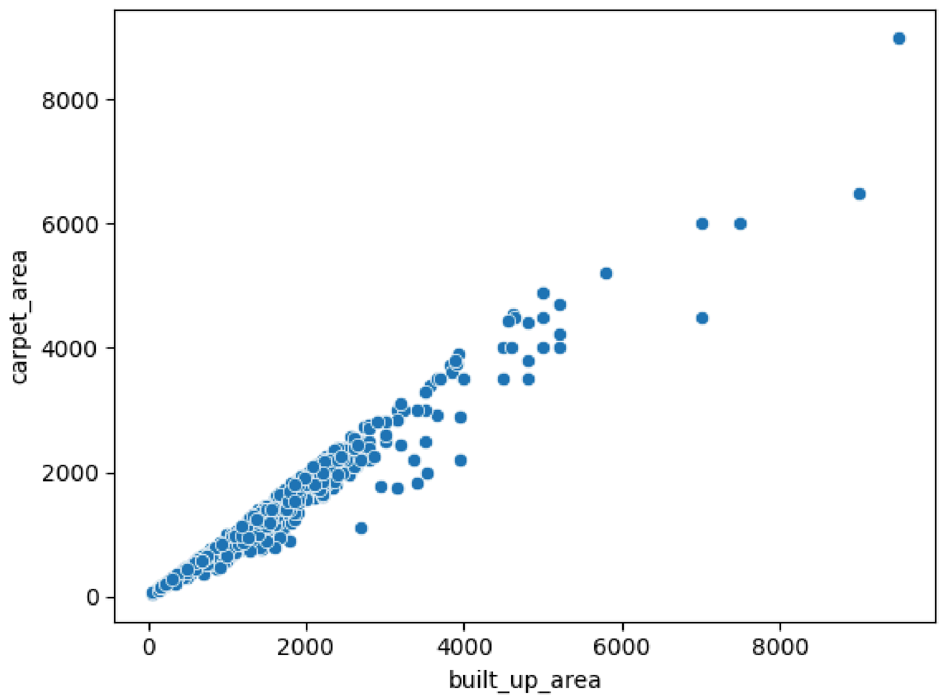
built_up_area vs super_built_up_area:

```
sns.scatterplot(x=df['built_up_area'],y= df['super_built_up_area'])
```



built_up_area vs carpet_area:

```
sns.scatterplot(x=df['built_up_area'],y=df['carpet_area'])
```



```
((df['super_built_up_area'].isnull()) & (df['built_up_area'].isnull()) & (df['carpet_
area'].isnull())).sum()

Output: 0
```

👆 There is not a single row where all 3 values are absent.

Rows where all 3 are present:

```
all_present_df = df[~((df['super_built_up_area'].isnull()) | (df['built_up_area'].is
null()) | (df['carpet_area'].isnull()))]
```

`df['column'].isnull()`

- This checks if the values in the column are missing (`NaN`).
- For example:

  - `df['super_built_up_area'].isnull()` returns `True` for rows where `super_built_up_area` is `NaN`, and `False` otherwise.

Combining Conditions with `|` (OR)

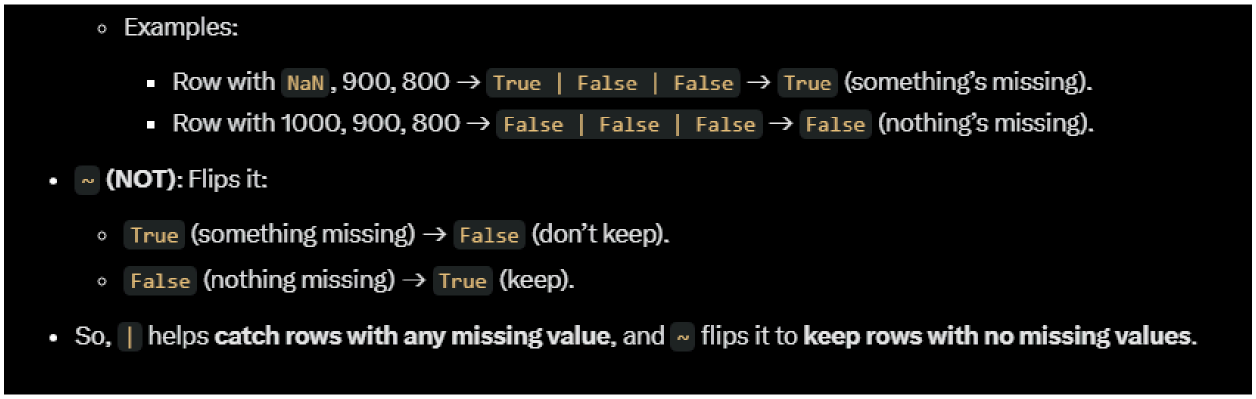
- The `|` operator is used to combine the conditions for the three columns.
- `(df['super_built_up_area'].isnull()) | (df['built_up_area'].isnull()) | (df['carpet_area'].isnull())`:
  - This returns `True` for rows where **any of the three columns** have missing values (`NaN`).

`~` (NOT)

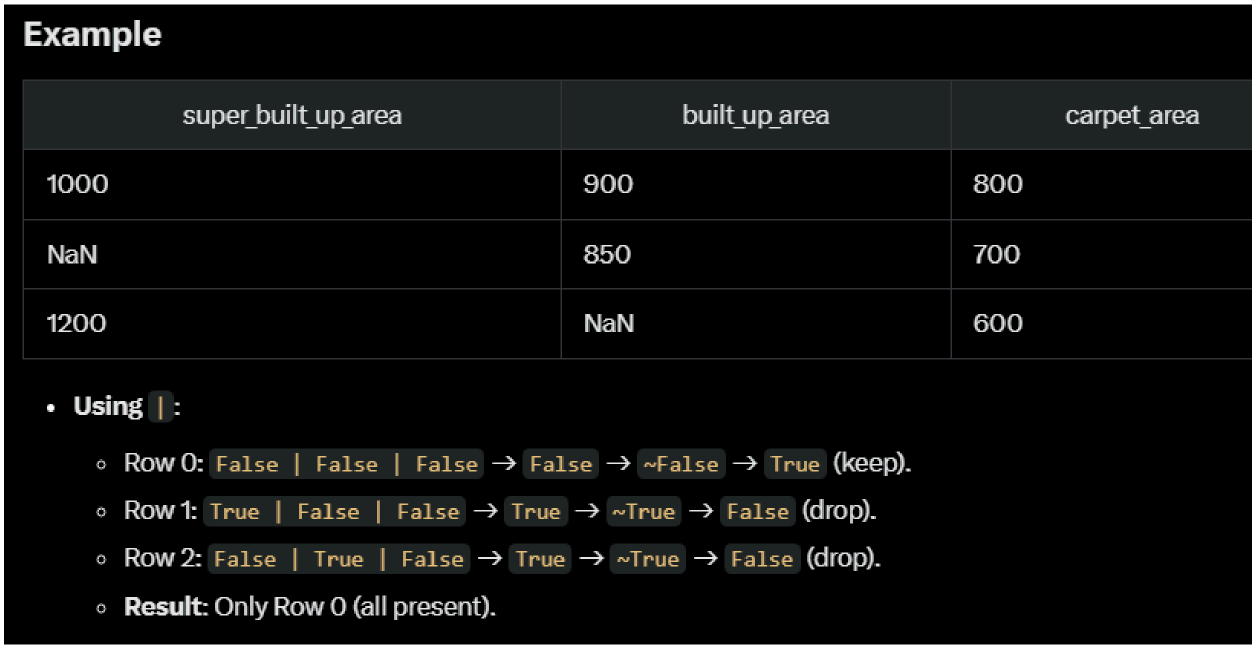
- The `~` operator negates the condition.
- `~(...)` means "not" the condition inside the parentheses.
- So, `~((df['super_built_up_area'].isnull()) | (df['built_up_area'].isnull()) | (df['carpet_area'].isnull()))`:
  - Returns `True` for rows where **none of the three columns** have missing values (`NaN`).

Filtering the DataFrame

- `df[~(...)]` filters the DataFrame to keep only the rows where the condition inside `~(...)` is `True`.
- In this case, it keeps rows where **all three columns** (`super_built_up_area`, `built_up_area`, and `carpet_area`) have **non-missing values**.



  - Examples:
    - Row with `NaN`, 900, 800 → `True | False | False` → `True` (something's missing).
    - Row with 1000, 900, 800 → `False | False | False` → `False` (nothing's missing).
- `~` **(NOT):** Flips it:
  - `True` (something missing) → `False` (don't keep).
  - `False` (nothing missing) → `True` (keep).
- So, `|` helps **catch rows with any missing value**, and `~` flips it to **keep rows with no missing values.**



Example

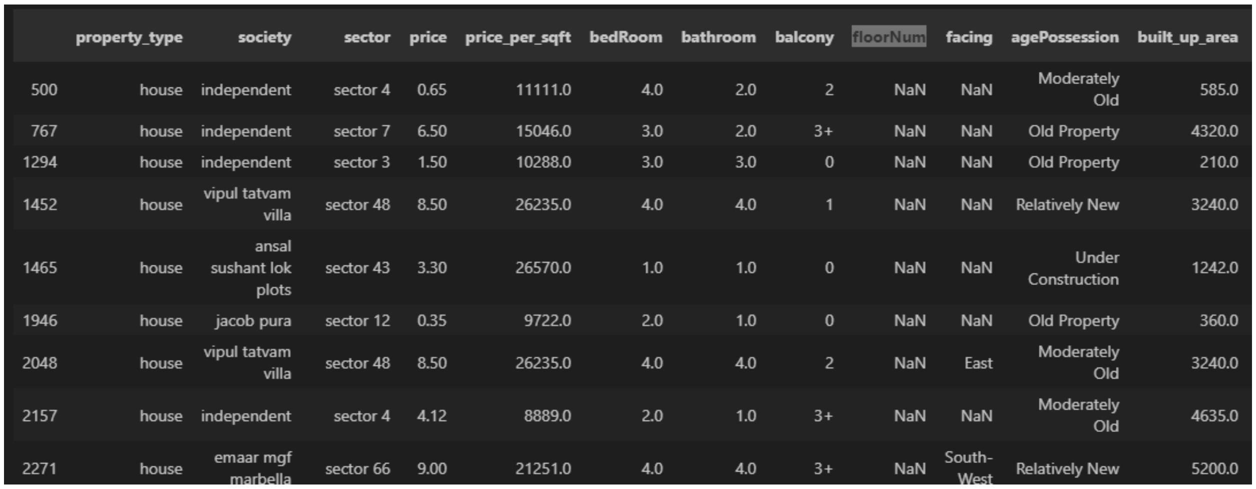
super_built_up_area	built_up_area	carpet_area
1000	900	800
NaN	850	700
1200	NaN	600

- **Using `|`:**
  - Row 0: `False | False | False` → `False` → `~False` → `True` (keep).
  - Row 1: `True | False | False` → `True` → `~True` → `False` (drop).
  - Row 2: `False | True | False` → `True` → `~True` → `False` (drop).
  - **Result:** Only Row 0 (all present).

```
all_present_df.shape

Output: (531, 24)
```

floorNum

```
df[df['floorNum'].isnull()]
```



	property_type	society	sector	price	price_per_sqft	bedRoom	bathroom	balcony	floorNum	facing	agePossession	built_up_area
500	house	independent	sector 4	0.65	11111.0	4.0	2.0	2	NaN	NaN	Moderately Old	585.0
767	house	independent	sector 7	6.50	15046.0	3.0	2.0	3+	NaN	NaN	Old Property	4320.0
1294	house	independent	sector 3	1.50	10288.0	3.0	3.0	0	NaN	NaN	Old Property	210.0
1452	house	vipul tatvam villa	sector 48	8.50	26235.0	4.0	4.0	1	NaN	NaN	Relatively New	3240.0
1465	house	ansal sushant lok plots	sector 43	3.30	26570.0	1.0	1.0	0	NaN	NaN	Under Construction	1242.0
1946	house	jacob pura	sector 12	0.35	9722.0	2.0	1.0	0	NaN	NaN	Old Property	360.0
2048	house	vipul tatvam villa	sector 48	8.50	26235.0	4.0	4.0	2	NaN	East	Moderately Old	3240.0
2157	house	independent	sector 4	4.12	8889.0	2.0	1.0	3+	NaN	NaN	Moderately Old	4635.0
2271	house	emaar mgf marbella	sector 66	9.00	21251.0	4.0	4.0	3+	NaN	South-West	Relatively New	5200.0

- Calculate mean of the floor number of houses:

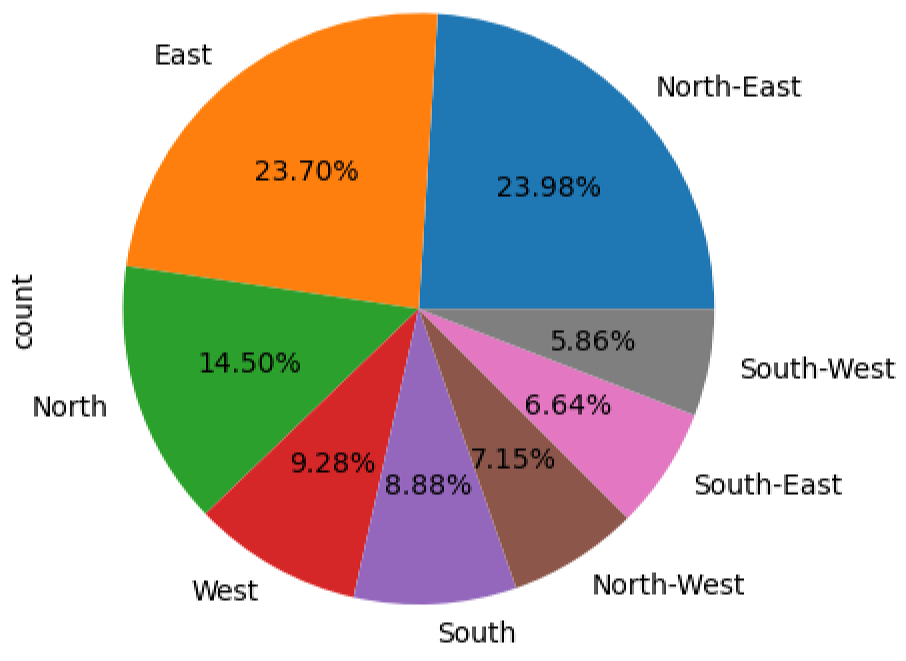
```
df[df['property_type'] == 'house']['floorNum'].median()

Output:2
```

- Fill missing floor no. with 2

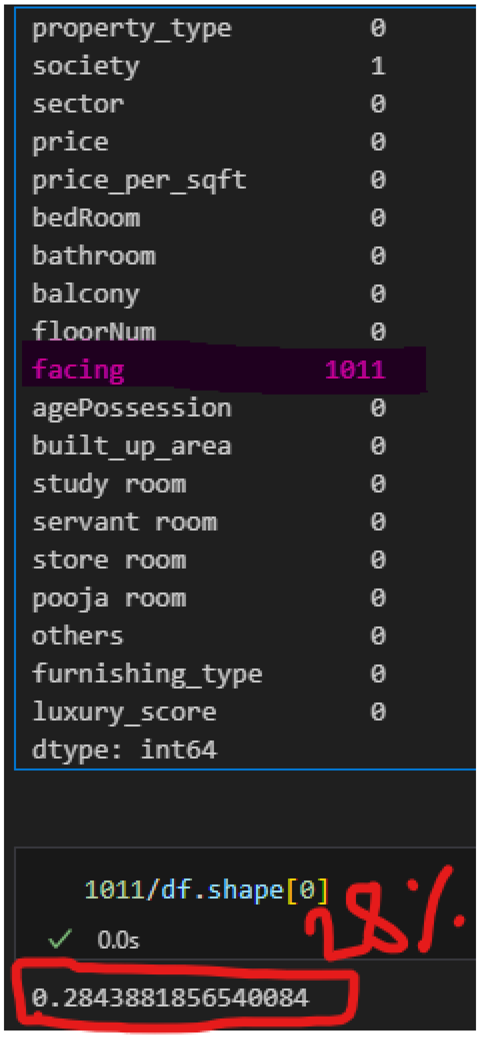
```
df['floorNum'].fillna(2.0,inplace=True)
```

Facing

```
df['facing'].value_counts().plot(kind='pie',autopct='%0.2f%%')
```



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



```
property_type        0
society              1
sector               0
price                0
price_per_sqft       0
bedRoom              0
bathroom             0
balcony              0
floorNum             0
facing            1011
agePossession        0
built_up_area        0
study room           0
servant room         0
store room           0
pooja room           0
others               0
furnishing_type      0
luxury_score         0
dtype: int64
```

```
1011/df.shape[0]
```

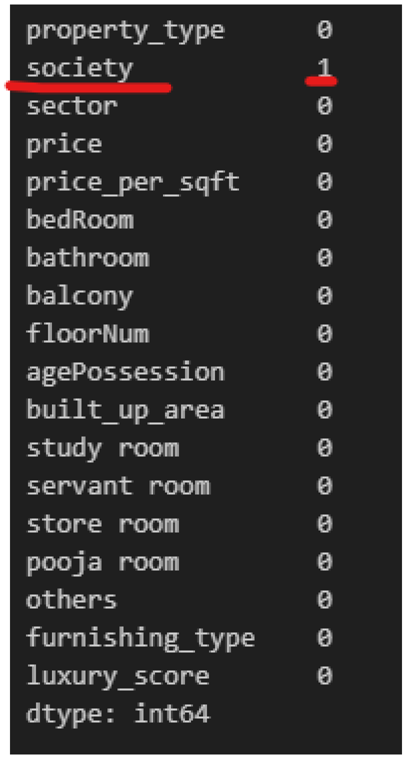
28%

```
0.0s
0.2843881856540084
```

- There are 28% missing values
- Decided to drop the column

```
df.drop(columns=['facing'],inplace=True)
```

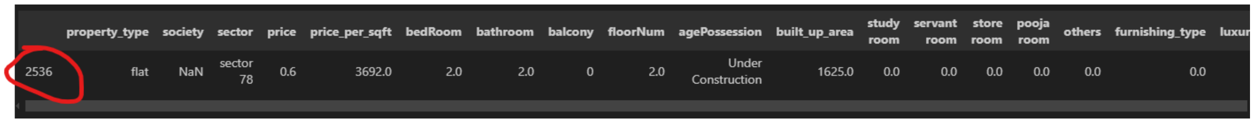
```
df.isnull().sum()
```



```
property_type      0
society            1
sector             0
price              0
price_per_sqft     0
bedRoom            0
bathroom           0
balcony            0
floorNum           0
agePossession      0
built_up_area      0
study room         0
servant room       0
store room         0
pooja room         0
others             0
furnishing_type    0
luxury_score       0
dtype: int64
```

- We can check the index & drop this row

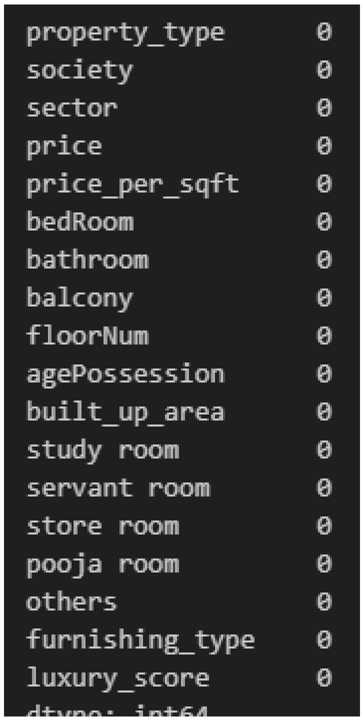
```
df[df.society.isnull()]
```



	property_type	society	sector	price	price_per_sqft	bedRoom	bathroom	balcony	floorNum	agePossession	built_up_area	study room	servant room	store room	pooja room	others	furnishing_type	luxur
2536	flat	NaN	sector 78	0.6	3692.0	2.0	2.0	0	2.0	Under Construction	1625.0	0.0	0.0	0.0	0.0	0.0	0.0	

```
df.drop(index=[2536],inplace=True)
```

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



```
property_type      0
society            0
sector             0
price              0
price_per_sqft     0
bedRoom            0
bathroom           0
balcony            0
floorNum           0
agePossession      0
built_up_area      0
study room         0
servant room       0
store room         0
pooja room         0
others             0
furnishing_type    0
luxury_score       0
dtype: int64
```