Data_Preprocessing_Đề_bài_thực_hành

August 31, 2021

1 Các nội dung chính

1. Mục tiêu:

- Nắm được các bước cơ bản trong khâu tiền xử lí dữ liệu.

2. Dữ liệu:

- Dữ liệu bất động sản - *Bengaluru_House_Data* > Gồm các trường dữ liệu: location, size, total_sqft, price, ...

  Link Kaggle: https://www.kaggle.com/amitabhajoy/bengaluru-house-price-data

3. Yêu cầu:

- Sử dụng các công cụ (Pandas, Seaborn, ...) để thực hiện xem xét, đánh giá đặc điểm của dữ liệu, từ đó đưa ra phương án tiền xử lí dữ liệu (làm sạch, trích xuất thông tin ban đầu, ...)

2 Nội dung thực hành

[6]:
```
#Nếu chạy trên Google Colab thì cần kết nối với máy chủ trước
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

2.0.1 Import Libraries

[7]:
```
import numpy as np
import pandas as pd #Giúp làm việc với các dữ liệu dạng bảng
import matplotlib.pyplot as plt #Thư viện hỗ trợ trực quan hóa dữ liệu
import seaborn as sns #Thư viện giúp trực quan hóa dữ liệu, được xây trên␣
↪matplotlib
```

2.0.2 Load dataset

1. Đọc dữ liệu bằng pandas, dạng dataframe

[8]:
```
%cd /content/drive/MyDrive/Preprocessing_practice/1.Practice/
 ↪Bangalore_House_Price_data
# Nếu chạy trên colab thì cũng cần trỏ tới thư mục phù hợp để lấy data
# cd DIR_PATH
```

```
/content/drive/MyDrive/Preprocessing_practice/1.Practice/Bangalore_House_Price_d
ata
```

[9]:
```
path = "Bengaluru_House_Data.csv"
df_raw = pd.read_csv(path)
df_raw.shape
```

[9]:
```
(13320, 9)
```

2. Review 5 sample đầu tiên

[10]:
```
df_raw.head() # return DataFrame
```

[10]:
```
              area_type   availability  … balcony   price
0  Super built-up  Area         19-Dec  …     1.0   39.07
1            Plot  Area  Ready To Move  …     3.0  120.00
2        Built-up  Area  Ready To Move  …     3.0   62.00
3  Super built-up  Area  Ready To Move  …     1.0   95.00
4  Super built-up  Area  Ready To Move  …     1.0   51.00

[5 rows x 9 columns]
```

3. Review 5 sample cuối cùng

[11]:
```
df_raw.tail()
```

[11]:
```
                  area_type   availability  … balcony  price
13315        Built-up  Area  Ready To Move  …     0.0  231.0
13316  Super built-up  Area  Ready To Move  …     NaN  400.0
13317        Built-up  Area  Ready To Move  …     1.0   60.0
13318  Super built-up  Area         18-Jun  …     1.0  488.0
13319  Super built-up  Area  Ready To Move  …     1.0   17.0

[5 rows x 9 columns]
```

2.0.3 Exploratory Data Analysis (EDA)

[12]:
```
df = df_raw.copy() #Tạo bản sao để thực hiện EDA
```

1. Thông tin cơ bản về dữ liệu, tên trường, số giá trị non-null của từng trường, kiểu dữ liệu của từng trường

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

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

2. Thống kê 1 số thuộc tính cơ bản của dữ liệu, bao gồm count, mean, std, min, max, quartile

[14]:
```
df.describe()
```

[14]:

	bath	balcony	price
count	13247.000000	12711.000000	13320.000000
mean	2.692610	1.584376	112.565627
std	1.341458	0.817263	148.971674
min	1.000000	0.000000	8.000000
25%	2.000000	1.000000	50.000000
50%	2.000000	2.000000	72.000000
75%	3.000000	2.000000	120.000000
max	40.000000	3.000000	3600.000000

3. Thống kê các giá trị duy nhất của từng trường và số lần xuất hiện của chúng

[15]:
```
def value_count(df):
  for var in df.columns:
    print(df[var].value_counts())
    print("--------------------------------")

value_count(df)
```

```
Super built-up  Area    8790
Built-up  Area          2418
Plot  Area              2025
Carpet  Area              87
Name: area_type, dtype: int64
--------------------------------
Ready To Move    10581
18-Dec             307
18-May             295
```

```
18-Apr             271
18-Aug             200
                   …
17-Jan               1
14-Jul               1
16-Jan               1
16-Nov               1
16-Oct               1
Name: availability, Length: 81, dtype: int64
--------------------------------
Whitefield              540
Sarjapur  Road          399
Electronic City         302
Kanakpura Road          273
Thanisandra             234
                        …
Kanakapura Road           1
KHB Colony Extension      1
Gollarahatti              1
MM Layout                 1
Hal old airport road      1
Name: location, Length: 1305, dtype: int64
--------------------------------
2 BHK         5199
3 BHK         4310
4 Bedroom      826
4 BHK          591
3 Bedroom      547
1 BHK          538
2 Bedroom      329
5 Bedroom      297
6 Bedroom      191
1 Bedroom      105
8 Bedroom       84
7 Bedroom       83
5 BHK           59
9 Bedroom       46
6 BHK           30
7 BHK           17
1 RK            13
10 Bedroom      12
9 BHK            8
8 BHK            5
10 BHK           2
11 BHK           2
11 Bedroom       2
12 Bedroom       1
14 BHK           1
```

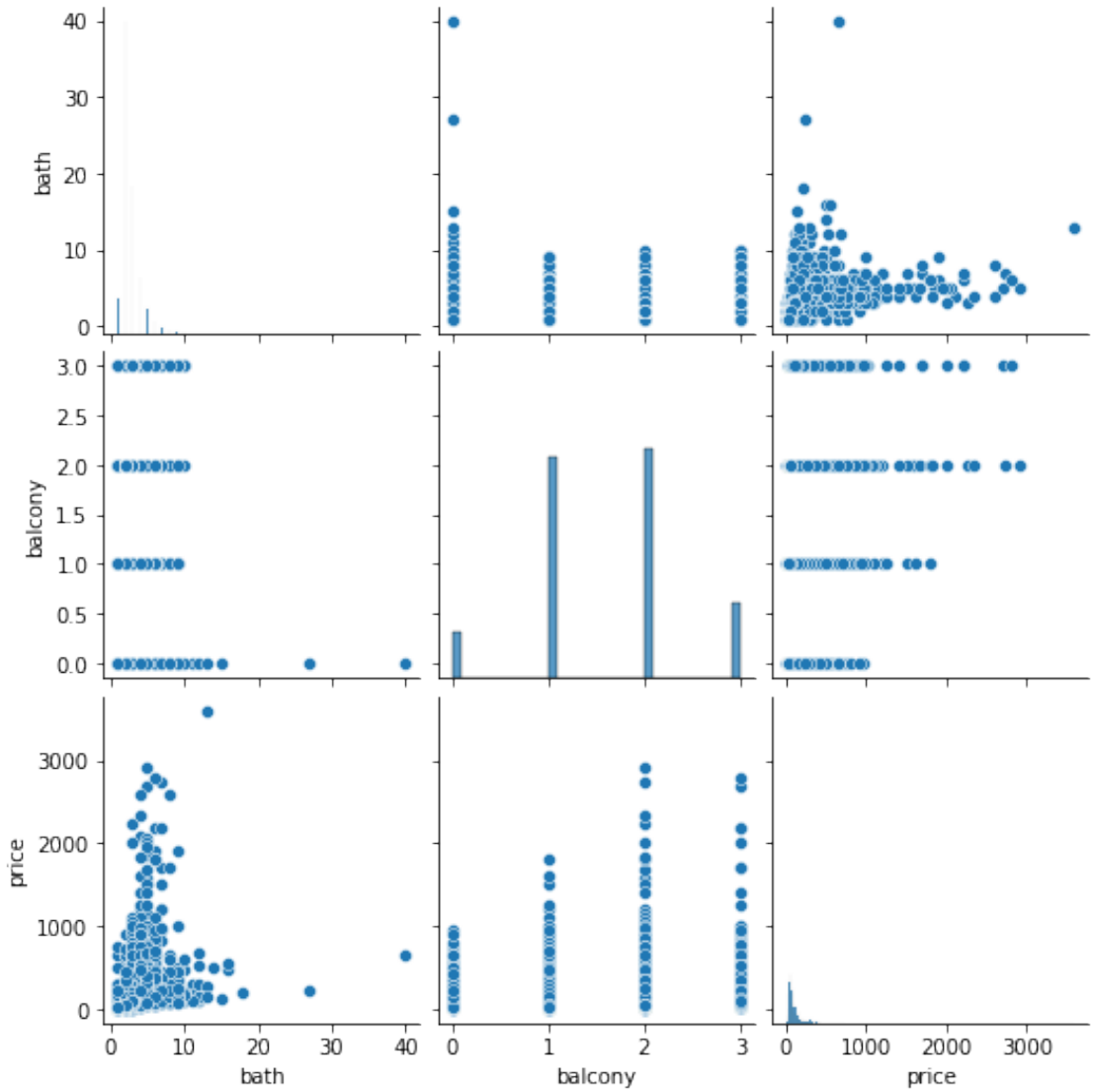
```
19 BHK           1
43 Bedroom       1
18 Bedroom       1
13 BHK           1
16 BHK           1
27 BHK           1
Name: size, dtype: int64
--------------------------------
GrrvaGr    80
PrarePa    76
Prtates    59
Sryalan    59
GMown E    56
           ..
Fosic C     1
Esncyth     1
Suashen     1
Pridel      1
Siite E     1
Name: society, Length: 2688, dtype: int64
--------------------------------
1200             843
1100             221
1500             205
2400             196
600              180
                 …
2181               1
1642               1
651                1
2395               1
1500Sq. Meter      1
Name: total_sqft, Length: 2117, dtype: int64
--------------------------------
2.0     6908
3.0     3286
4.0     1226
1.0      788
5.0      524
6.0      273
7.0      102
8.0       64
9.0       43
10.0      13
12.0       7
13.0       3
11.0       3
16.0       2
```

```
27.0       1
18.0       1
40.0       1
15.0       1
14.0       1
Name: bath, dtype: int64
--------------------------------
2.0    5113
1.0    4897
3.0    1672
0.0    1029
Name: balcony, dtype: int64
--------------------------------
75.00    310
65.00    302
55.00    275
60.00    270
45.00    240
         …
81.55      1
69.49      1
42.18      1
70.25      1
74.82      1
Name: price, Length: 1994, dtype: int64
--------------------------------
```

4. Xem xét tương quan về giá trị của các cặp trường số

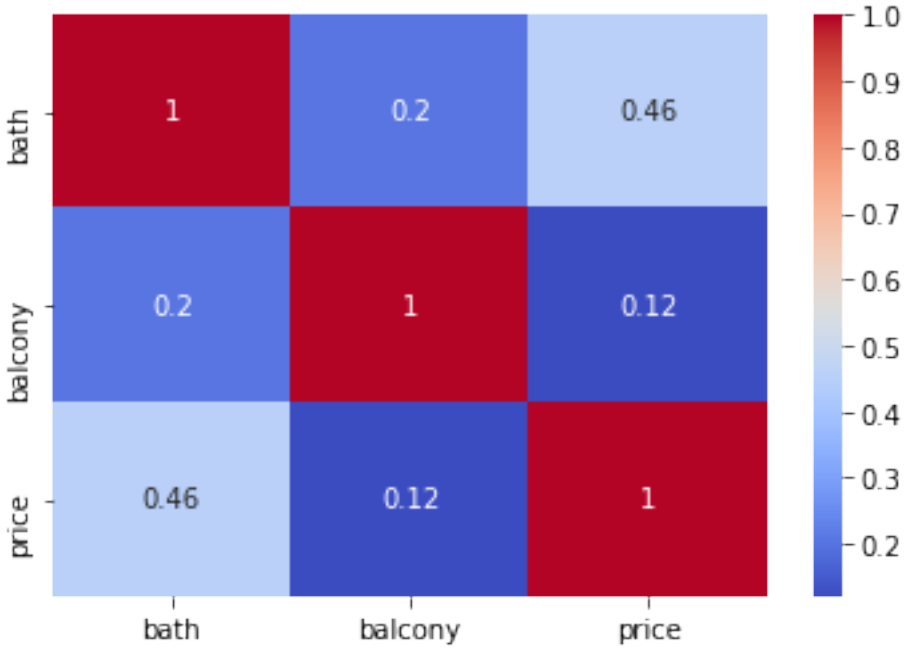
[16]:
```
sns.pairplot(df)
```

[16]:
```
<seaborn.axisgrid.PairGrid at 0x7f59befb89d0>
```



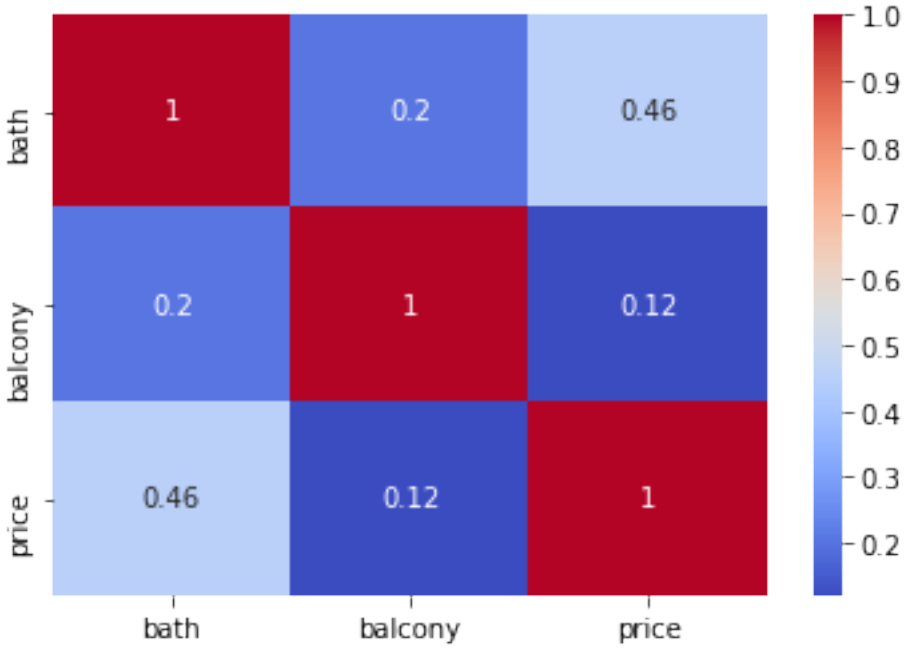
[17]:
```
num_vars = ["bath", "balcony", "price"]
sns.heatmap(df[num_vars].corr(),cmap="coolwarm", annot=True)
```

[17]: <matplotlib.axes._subplots.AxesSubplot at 0x7f59b4ed2250>



[18]:
```
# num_vars = ["bath", "balcony", "price"]
sns.heatmap(df.corr(),cmap="coolwarm", annot=True)
```

[18]: <matplotlib.axes._subplots.AxesSubplot at 0x7f59d353d990>



2.0.4 Preare Data for Machine Learning Model

Data cleaning

1. Thống kê số lượng và tỉ lệ giá trị null của từng thuộc tính

[19]:
```
df.isnull().sum() #So luong gia tri null cua tung thuoc tinh
```

[19]:
```
area_type          0
availability       0
location           1
size              16
society         5502
total_sqft         0
bath              73
balcony          609
price              0
dtype: int64
```

[20]:
```
df.isnull().mean()*100  # Tỷ lệ giá trị null của từng thuộc tính
```

[20]:
```
area_type         0.000000
availability      0.000000
location          0.007508
size              0.120120
society          41.306306
total_sqft        0.000000
bath              0.548048
balcony           4.572072
price             0.000000
dtype: float64
```

2. Loại đi trường society vì tỉ lệ null cao (41%)

[21]:
```
df2 = df.drop('society', axis='columns')
df2.shape
```

[21]:
```
(13320, 8)
```

3. Thay thế giá trị null trong trường balcony bằng giá trị trung bình của các giá trị not null

[22]:
```
df2['balcony'] = df2['balcony'].fillna(df2['balcony'].mean())
df2.isnull().sum()
```

[22]:
```
area_type        0
availability     0
location         1
size            16
total_sqft       0
```

```
bath            73
balcony          0
price            0
dtype: int64
```

4. Xóa đi các điểm dữ liệu (hàng) có giá trị nan (không có giá trị)

[23]:
```
df3 = df2.dropna()
df3.shape
```

[23]:
```
(13246, 8)
```

[24]:
```
df3.isnull().sum() #Thống kê lại xem đã xử lí hết các dữ liệu null hay chưa?
```

[24]:
```
area_type       0
availability    0
location        0
size            0
total_sqft      0
bath            0
balcony         0
price           0
dtype: int64
```

Feature Engineering

[25]:
```
#Cho phép in ra toàn bộ các giá trị output có thể của câu lệnh
pd.set_option("display.max_columns", None)
pd.set_option("display.max_rows", None)
```

1. Converting 'total_sqft' cat feature in numeric

[26]:
```
df3['total_sqft'].value_counts()
```

[26]:
```
1200                843
1100                221
1500                204
2400                195
600                 180
1000                172
1350                132
1050                123
1300                117
1250                114
900                 112
1400                108
1800                104
1150                101
```

1600	100
1140	91
2000	82
1450	70
1650	69
800	67
3000	66
1075	66
1020	63
2500	62
1160	60
1125	60
1550	60
950	59
1180	58
1700	58
1260	57
1255	56
1220	55
1080	55
1070	53
750	52
700	52
4000	48
1175	48
1225	48
1320	46
1240	46
2100	46
1230	45
1060	45
1210	44
850	43
1280	42
1185	41
1270	41
1410	40
1170	40
1190	40
1750	39
1025	38
1330	38
1290	37
1850	37
1310	37
1065	36
1194	36

1090	35
1215	35
500	34
1360	33
2700	33
1115	33
1464	32
1900	32
1120	32
3500	32
1530	31
2200	31
1430	31
1205	31
1340	31
1035	30
1165	30
1560	30
1145	29
1130	29
3600	29
1128	29
1275	28
2800	28
1040	28
1355	28
1680	27
1420	27
1155	27
1105	27
1590	25
1760	25
1245	25
1216	25
650	25
1460	25
1305	24
2600	24
1475	24
1010	24
883	23
1110	23
1030	23
1440	23
1495	22
1246	22
1575	22

985	22
1027	21
1315	21
1610	21
1325	21
1385	21
1370	21
3200	21
1470	21
1243	21
1015	21
925	20
660	20
5000	20
1390	20
550	20
1480	20
1540	20
1645	19
1640	19
1570	19
1265	19
920	19
1365	19
1520	19
1820	18
975	18
1196	18
1665	18
1525	18
980	18
2072	18
940	18
1095	18
1485	18
1195	18
1345	18
1012	18
4800	18
400	17
960	17
1375	17
525	17
1295	17
1157	17
1232	17
1045	16

1830	16
1655	16
1425	16
1197	16
645	16
1235	16
935	16
1490	16
3300	16
1720	16
1418	16
1135	16
2300	16
4500	15
1380	15
1116	15
1153	15
1660	15
1152	15
1950	15
1740	15
1580	15
1285	15
630	15
1085	15
1920	15
450	14
675	14
1141	14
905	14
1630	14
1595	14
720	14
1445	14
1510	13
1535	13
1615	13
1730	13
3800	13
1082	13
1690	13
1710	13
3900	13
1314	13
2250	13
1282	13
1005	13

1625	13
840	13
1875	13
2215	13
970	13
1455	12
1056	12
1404	12
1339	12
1843	12
1435	12
1296	12
1565	12
1890	12
3100	12
1252	12
1691	12
2900	12
1007	12
1151	12
1033	12
1685	11
1452	11
540	11
1405	11
1515	11
1174	11
984	11
1346	11
1162	11
1206	11
1804	11
2150	11
845	11
1639	11
965	11
1063	10
3596	10
1620	10
957	10
918	10
995	10
760	10
930	10
1465	10
770	10
1308	10

1198	10
1161	10
880	10
4200	10
1415	10
620	10
1555	10
1256	10
1810	10
3400	10
2350	10
1790	9
1047	9
1419	9
1156	9
1870	9
1482	9
2750	9
1693	9
1092	9
910	9
674	9
1715	9
1223	9
3150	9
1108	9
1724	9
1745	9
825	9
1725	9
3750	9
2475	9
1756	9
1224	9
890	9
2180	9
945	9
6000	9
1186	9
1605	9
705	9
1222	9
1113	9
2790	9
3250	9
780	9
875	8

1476	8
1187	8
1395	8
1835	8
1088	8
2650	8
990	8
1026	8
2990	8
656	8
1133	8
1335	8
1705	8
1703	8
1322	8
1670	8
1447	8
3122	8
1299	8
1221	8
1146	8
1352	8
2145	8
1636	8
4400	8
1532	8
1178	8
680	8
870	8
1163	8
5400	8
812	8
1192	8
1089	8
1508	7
1306	7
1272	7
615	7
1251	7
1183	7
1847	7
1096	7
982	7
1910	7
1139	7
1427	7
1069	7

1093	7
1132	7
1374	7
1357	7
1036	7
1349	7
1101	7
1342	7
1262	7
1166	7
1718	7
2050	7
1840	7
1041	7
1397	7
1179	7
1880	7
708	7
1762	7
1254	7
1545	7
1564	7
1044	7
1392	7
1917	7
1199	7
1277	7
7500	7
1112	7
1276	7
1143	7
820	7
1127	6
520	6
1083	6
710	6
1173	6
1126	6
654	6
1735	6
1602	6
1785	6
1167	6
1541	6
933	6
440	6
860	6

3895	6
810	6
929	6
1107	6
1787	6
1062	6
2275	6
1583	6
1512	6
1453	6
1333	6
1176	6
605	6
1053	6
1257	6
560	6
1274	6
1675	6
1102	6
1213	6
1571	6
1019	6
1451	6
830	6
1372	6
1028	6
2040	6
1268	6
2010	6
2280	6
1770	6
1303	6
1936	6
924	6
610	6
7000	6
1717	6
625	6
530	6
1719	6
1614	6
1855	6
1362	6
1856	6
1084	6
2690	6
1242	6

936	6
1780	6
1313	6
1181	6
1893	6
1635	6
1755	6
2225	6
1286	6
2425	6
1016	6
1846	6
1991	6
1697	6
1247	6
1411	6
1278	6
2774	6
1484	6
1304	6
1656	6
2480	6
1328	5
1297	5
1754	5
1253	5
1055	5
1098	5
1267	5
635	5
915	5
1364	5
1801	5
1505	5
1009	5
1293	5
2360	5
2340	5
4050	5
1925	5
1868	5
1109	5
545	5
1057	5
420	5
1573	5
4395	5

1458	5
2830 - 2882	5
715	5
1241	5
1930	5
2254	5
1354	5
1329	5
1884	5
1263	5
1351	5
907	5
1825	5
2850	5
2710	5
1123	5
919	5
640	5
2119	5
1586	5
1862	5
1269	5
1865	5
1444	5
1852	5
1935	5
1567	5
1585	5
1168	5
1403	5
1457	5
865	5
1533	5
1309	5
1559	5
410	5
1204	5
1664	5
1358	5
1148	5
1021	5
1052	5
1142	5
1188	5
1208	5
510	5
1326	5

1408	5
1749	5
1022	5
1226	5
1031	5
1798	5
1231	5
1683	5
1077	5
1436	5
1662	5
3450	5
1424	5
1344	5
1184	5
740	5
1654	5
1076	5
1826	5
1244	5
1738	5
1765	5
1073	4
1158	4
1521	4
416	4
3700	4
1569	4
12000	4
460	4
1634	4
1975	4
1517	4
1058	4
2439	4
1611	4
1386	4
1407	4
2160	4
1219	4
921	4
602	4
1074	4
996	4
4100	4
1134	4
1169	4

1008	4
1767	4
1945	4
2760	4
1207	4
1965	4
2240	4
1334	4
877	4
4750	4
1059	4
1061	4
861	4
1201	4
814	4
1091	4
1616	4
1711	4
1343	4
1081	4
2135	4
1367	4
595	4
1234	4
580	4
1171	4
1466	4
1752.12	4
10000	4
1980	4
1938	4
1072	4
1732	4
1842	4
755	4
2273	4
1394	4
1037	4
1118	4
1149	4
1933	4
1489	4
993	4
1776	4
1236	4
1864	4
1104	4

1332	4
1702	4
902	4
1203	4
775	4
1459	4
2640	4
497	4
1291	4
967	4
2560	4
1258	4
1279	4
1837	4
1138	4
884	4
1608	4
3850	4
1976	4
1592	4
1504	4
1537	4
418	4
1904	4
2450	4
1046	4
662	4
2070	4
795	4
1698	4
1318	4
1806	4
1079	4
1596	4
1885	4
1259	4
1353	4
8000	4
1238	4
914	4
360	4
1708	4
1891	4
955	4
2805	4
1111	4
1494	4

972	4
1034	4
1432	4
992	4
1563	4
966	4
1881	4
927	4
2830	4
1067	4
1929	4
1695	4
665	4
1051	4
3520	4
3436	4
946	4
1795	4
1202	4
1449	4
425	3
1539	3
829	3
1626	3
1644	3
1588	3
2099	3
735	3
1448	3
1805	3
1819	3
670	3
1117	3
2330	3
4111	3
2357	3
2559	3
3335	3
2062	3
1713	3
1307	3
896	3
2065	3
1519	3
2390	3
3630 - 3800	3
1531	3

3630	3
1227	3
1233	3
2503	3
2159	3
1704	3
1356	3
1942	3
1594	3
648	3
2610	3
2017	3
1783	3
2095	3
2550	3
1984	3
2020	3
1536	3
1066	3
3730	3
3252	3
1919	3
1488	3
1024	3
923	3
1147	3
1694	3
3155	3
2422	3
1553	3
937	3
1491	3
1312	3
1301	3
663	3
891	3
1209	3
1985	3
5100	3
1779	3
1416	3
1652	3
1603	3
3526	3
1011	3
375	3
1513	3

1097	3
1632	3
1384	3
1706	3
1154	3
1678	3
1039	3
999	3
1032	3
1897	3
1086	3
1212	3
1896	3
7200	3
782	3
2483	3
4600	3
2770	3
1682	3
1591	3
1692	3
2940	3
1172	3
1653	3
1496	3
2144	3
1018	3
1068	3
1426	3
2257	3
703	3
2230	3
1751	3
1566	3
745	3
1817	3
2880	3
690	3
1777.26	3
3040	3
823	3
994	3
2732	3
1164	3
2061	3
1469	3
912	3

658	3
3205	3
1758	3
702	3
565	3
1832	3
1576	3
2502	3
1182	3
1858	3
2367	3
480	3
3385	3
2289	3
1726	3
1527	3
1860	3
1336	3
1768	3
2087.01	3
1903	3
1477	3
1347	3
1071	3
1518	3
1341	3
1121	3
6500	3
1294	3
909	3
1302	3
664	3
1599	3
1621	3
3050	3
1017	3
958	3
575	3
1006	3
1845	3
1689	3
1853	3
3067	3
1788	3
1191	3
1523	3
1960	3

2440	3
2060	3
1478	3
1584	3
6200	3
661	3
1577	3
2470	3
1129	3
2325	3
1043	3
1144	3
672	3
9600	3
1124	3
1398	3
1382	3
1428	3
435	3
1737	3
2080	3
2882	3
1672	3
2292	3
1311	3
2106	3
1839	3
1498	3
1094	3
3626	2
2075	2
2689	2
1808	2
3675	2
4104	2
1618	2
2030	2
1237	2
2002	2
1677	2
2041	2
1552	2
1348	2
834	2
613 - 648	2
1676	2
2172	2

485	2
1763	2
1546	2
1799	2
3508	2
2460	2
1607	2
1363	2
1365 - 1700	2
2493	2
4003	2
2950	2
1468	2
1666	2
2403	2
726	2
1115 - 1130	2
1499	2
1371	2
725	2
952	2
1248	2
1359	2
10961	2
4260	2
2376	2
971	2
1752	2
1136	2
350	2
1999	2
4346	2
2320	2
2540	2
524 - 894	2
589	2
2197	2
1214	2
2524	2
2780	2
3875	2
1823	2
1819.18	2
730	2
1419.59	2
3761	2
2569	2

1438	2
1211	2
1368	2
922	2
1949	2
3012	2
3633	2
2025	2
1584.01	2
1431	2
1002	2
2392	2
2047	2
1789	2
2214	2
1814	2
1106	2
1688	2
2409	2
3198	2
1023	2
1284	2
1996	2
644	2
1667	2
1524	2
4700	2
1757	2
686	2
2006	2
991	2
1099	2
1261	2
1319	2
1581	2
1863	2
1078	2
1249	2
1417	2
1323	2
3657	2
1391	2
655	2
697	2
1894	2
2849	2
2675	2

682	2
2111	2
3356	2
645 - 936	2
1421	2
3420	2
1995	2
1239	2
2520	2
2375	2
997	2
2665	2
1402	2
1283	2
1396	2
711	2
1381	2
3960	2
432	2
1548	2
3425	2
1950.2	2
2420	2
1836	2
2658	2
1914	2
2089	2
1107.83	2
3090	2
1970	2
1651	2
2462	2
4144	2
1229	2
2264	2
7800	2
5800	2
1327	2
1003	2
1739	2
3951	2
3785	2
1948	2
973	2
1601	2
1423	2
2321	2

986	2
1388	2
3035	2
3206	2
4250	2
2093	2
2666	2
1542	2
2045	2
1674	2
646	2
4190	2
3522	2
1827	2
1684	2
585	2
3126	2
596	2
3754	2
6136	2
1366	2
1786	2
835	2
3260	2
799	2
1622	2
805	2
1578	2
1376	2
1454	2
570	2
876	2
2210	2
1623	2
1679	2
620 - 934	2
793	2
606	2
381 - 535	2
527	2
712	2
2268	2
693	2
509	2
6652	2
1001	2
1442	2

1331	2
1744	2
3216	2
2720	2
1538	2
1721	2
1409	2
1643	2
1048	2
3930	2
2140	2
794	2
1159	2
3024	2
470	2
1699	2
3262	2
1743	2
1937	2
4025	2
3940	2
1493	2
1255 - 1350	2
3295	2
2556	2
2238	2
381	2
3360	2
1612	2
1876	2
4235	2
551	2
2260	2
142.61Sq. Meter	2
901	2
1434	2
1859	2
1507	2
1013	2
1492	2
2265	2
1556	2
882	2
1804 - 2273	2
2259	2
815	2
11000	2

734	2
3758	2
1613	2
2024	2
8321	2
5500	2
3009	2
2170	2
1441	2
913	2
3170	2
943	2
1746	2
827	2
1633	2
2430	2
1728	2
1414	2
633	2
1709	2
2378	2
535	2
1369.1	2
3565	2
1516	2
2456	2
3408	2
1137	2
1554	2
765	2
1866	2
938	2
2051	2
1463	2
1281	2
4300	2
1486	2
1707	2
2388	2
1844	2
3095	2
1934	2
1298	2
849	2
2580 - 2591	2
1549	2
856	2

3004	2
3453	2
1103	2
2266	2
2777.29	2
1210 - 1477	1
2429	1
188.89Sq. Yards	1
1558	1
1331.95	1
3504	1
2651	1
864	1
871	1
1831	1
660 - 780	1
284	1
4273	1
3Cents	1
1686	1
2155	1
1932.47	1
4830	1
3033	1
385 - 440	1
3480	1
2423	1
3530	1
2337	1
1565 - 1595	1
785	1
2328	1
3532	1
1681	1
2431	1
2121	1
132Sq. Yards	1
2120	1
2504	1
3204	1
2015	1
1.26Acres	1
4408	1
1723	1
939	1
2733	1
660 - 700	1

1660 - 1805	1
697Sq. Meter	1
3175	1
799 - 803	1
1818	1
4694	1
2615	1
4450	1
2932	1
623	1
167Sq. Meter	1
1902	1
627	1
1383	1
3117	1
4075	1
2462 - 2467	1
2453	1
1218	1
540 - 740	1
3968	1
1589	1
2110	1
774	1
3692	1
2894	1
547.34 - 827.31	1
2043	1
5480	1
2144.6	1
3860	1
4900 - 4940	1
2171	1
2650 - 2990	1
818	1
3329	1
2167	1
753	1
3595	1
1349 - 3324	1
684 - 810	1
45	1
2704	1
2162	1
2098	1
1217	1
1987	1

3179	1
204Sq. Meter	1
3016	1
2048	1
3418	1
3664	1
1918	1
953	1
3075	1
11338	1
1	1
1200 - 1470	1
4482	1
3584	1
2505	1
117Sq. Yards	1
3230	1
581.91	1
3369 - 3464	1
2479.13	1
1563.05	1
2006.8	1
3381	1
4900	1
9200	1
1561	1
650 - 760	1
5.31Acres	1
1763.25	1
1646	1
808	1
2162.03	1
2380	1
2528 - 3188	1
2064	1
4500 - 5540	1
84.53Sq. Meter	1
1511	1
2028	1
4166	1
884 - 1116	1
502	1
1816	1
505	1
916	1
1733.5	1
4818	1

2088	1
983	1
1230 - 1490	1
2912	1
2023.71	1
3537	1
2611	1
1998	1
2384	1
2444	1
1604	1
2134	1
2401	1
598 - 958	1
1981	1
2172.65	1
2980	1
1558.67	1
1551	1
38Guntha	1
300	1
695	1
5515	1
2370	1
1139.7	1
1469 - 1766	1
3445	1
755 - 770	1
1879	1
1189	1
1321	1
3025	1
3362	1
2416	1
1181.7	1
1567.2	1
1915	1
3463	1
2470 - 2790	1
5422	1
2270	1
6613	1
532	1
2108	1
2925	1
3125	1
4097	1

934	1
1264	1
3489	1
2246	1
1180 - 1630	1
1330.74	1
3290	1
717	1
1777	1
2679	1
817	1
3425 - 3435	1
1628	1
2245	1
7400	1
6729	1
3680	1
1361	1
2996	1
3161	1
2519	1
1783 - 1878	1
4850	1
3339	1
2776	1
840 - 1010	1
1623.29	1
3628	1
850 - 1093	1
2465	1
3073	1
650 - 665	1
1401	1
1652.5	1
2476	1
10624	1
3670	1
120Sq. Yards	1
3516	1
1673	1
1004	1
613	1
706	1
1510 - 1670	1
1125 - 1500	1
2282	1
4960	1

122Sq. Yards	1
1716	1
1905	1
2319	1
1669	1
2404	1
1761	1
451	1
1641	1
629 - 1026	1
1668	1
961	1
1593	1
1815	1
2153	1
2601	1
2885	1
1778	1
4827	1
1736	1
2842	1
615 - 985	1
2415	1
1775	1
866	1
2251	1
3535	1
2363	1
2372	1
315Sq. Yards	1
1631	1
2597	1
4303	1
832	1
2526	1
911	1
4510	1
1606	1
1722	1
5700	1
3366	1
1234.6	1
5965	1
5530	1
1542.14	1
1160 - 1195	1
3602	1

4209	1
10030	1
1413	1
35000	1
1792	1
2150 - 2225	1
1473	1
2875	1
2195	1
4125Perch	1
896.9	1
1049	1
3056	1
488	1
1230 - 1290	1
1888	1
1940	1
1617	1
142.84Sq. Meter	1
3734	1
11890	1
980 - 1030	1
3950	1
947	1
1266	1
2557	1
2999.97	1
516	1
1471	1
2138	1
2408	1
2791	1
4110	1
3870	1
963	1
1609	1
3297	1
2646	1
1145 - 1340	1
1691.2	1
1289	1
2825	1
2118	1
943 - 1220	1
628	1
4550	1
4170	1

2176	1
2663	1
1872	1
8840	1
1747	1
527 - 639	1
1324	1
2247	1
673	1
1627.86	1
8400	1
2079	1
4470	1
2695	1
1629	1
858	1
6688	1
959	1
1195 - 1440	1
1379	1
2008	1
4000 - 5249	1
2086	1
2721	1
908	1
947.55	1
567	1
86.72Sq. Meter	1
5080	1
2302	1
7514	1
1300 - 1405	1
1452.19	1
1467	1
1400 - 1421	1
1208.51	1
1160 - 1315	1
2236	1
2563 - 2733	1
4278	1
3978	1
664 - 722	1
1255 - 1375	1
772	1
2735	1
1522	1
456	1

16335	1
3760	1
1377	1
2435	1
2801.25	1
2317	1
857	1
500Sq. Yards	1
2383	1
1828	1
987	1
1734	1
1886	1
1574	1
361.33Sq. Yards	1
5425	1
787	1
1020.07	1
854 - 960	1
1076 - 1199	1
977	1
2928	1
1689.28	1
34.46Sq. Meter	1
6600	1
1626.6	1
2365	1
2901	1
2169	1
351	1
596 - 861	1
754	1
1004 - 1204	1
1451.5	1
3913	1
806	1
3131	1
1776.42	1
36000	1
638	1
1412	1
2107	1
3563	1
2182	1
948	1
5150	1
1054	1

5656	1
1273	1
5270	1
1235 - 1410	1
3309	1
2031	1
978	1
1119	1
888	1
1500Cents	1
30Acres	1
552	1
1462	1
1568	1
870 - 1080	1
1439	1
1547	1
1916	1
1548.3	1
4460	1
2795	1
706 - 716	1
769	1
7150	1
3259	1
1990	1
1429	1
2592	1
897	1
3040Sq. Meter	1
2396	1
11	1
1797	1
2206	1
6830	1
1010 - 1300	1
42000	1
1829	1
612	1
1587	1
2968	1
539	1
3307 - 3464	1
4041	1
580 - 650	1
1452.55	1
1287	1

2297	1
2342	1
1005.03 - 1252.49	1
1637	1
3071	1
5384	1
1100Sq. Meter	1
1502	1
340	1
24Sq. Meter	1
1557	1
250	1
5985	1
581	1
4239	1
2171.66	1
1215 - 1495	1
621	1
956	1
4634	1
873	1
1288	1
3060	1
3815	1
2800 - 2870	1
1550 - 1590	1
3990	1
5665.84	1
904	1
3746	1
1989	1
1753	1
2970	1
1922	1
5600	1
714	1
396	1
1270 - 1275	1
45Sq. Yards	1
2039	1
784	1
3554	1
3010	1
2185	1
747	1
1660.4	1
2582	1

445	1
1529	1
2792	1
2168	1
1901	1
3606	1
3855	1
3300 - 3335	1
1520 - 1759	1
3067 - 8156	1
3820	1
276	1
524	1
1408 - 1455	1
14000	1
3144	1
2113	1
4428	1
783 - 943	1
2845	1
885	1
3245	1
1000Sq. Meter	1
2005	1
2725 - 3250	1
2130	1
2204	1
1052 - 1322	1
1338	1
2082	1
2173	1
688	1
2775	1
534	1
712 - 938	1
2496	1
3124	1
1389	1
2090	1
540 - 670	1
3210	1
1193	1
1824	1
3555	1
2546	1
4040	1
666	1

1369	1
24Guntha	1
30400	1
6Acres	1
842	1
555	1
2631	1
4051	1
2112.95	1
3729	1
694	1
5	1
1337	1
655 - 742	1
3811	1
2059	1
3027	1
2220	1
151.11Sq. Yards	1
824	1
3405.1	1
941	1
2100 - 2850	1
1250 - 1305	1
668	1
2405	1
2424	1
4350	1
1378	1
1266.67	1
1437 - 1629	1
2485	1
2940Sq. Yards	1
1000 - 1285	1
764	1
3045	1
2489	1
1732.46	1
20000	1
4000 - 4450	1
2956	1
1733	1
2132	1
2041 - 2090	1
2386	1
3905	1
1962	1

3227	1
3884	1
964	1
1.25Acres	1
2736	1
1907	1
2019	1
833	1
4689	1
2125	1
813	1
3401	1
722	1
1688.12	1
704	1
989	1
1939	1
5666 - 5669	1
951	1
3235	1
4772	1
2.09Acres	1
1087	1
1659	1
492	1
610 - 615	1
777.4	1
1648	1
2806 - 3019	1
296	1
1150 - 1194	1
2137	1
4355	1
1719.3	1
2501	1
616	1
1015 - 1540	1
1525.84	1
2570	1
2Acres	1
685	1
934 - 1437	1
3092	1
892	1
2293	1
790	1
3005	1

3515	1
2710 - 3360	1
4640	1
302	1
3496	1
4856	1
24	1
499	1
588	1
826	1
5108	1
60	1
3190	1
133.3Sq. Yards	1
2758	1
1909	1
78.03Sq. Meter	1
1793	1
3042	1
866.28	1
1443	1
462	1
2283	1
4382	1
2406	1
1741	1
3280	1
2863	1
2785	1
1766	1
469	1
3569	1
881	1
1100Sq. Yards	1
2105	1
4356	1
1731	1
855	1
2495	1
2515	1
1712	1
583	1
3350	1
2035	1
850 - 1060	1
2507	1
3770	1

52272	1
2077	1
2023	1
1255 - 1863	1
2533	1
8500	1
1861	1
1269.72	1
520 - 645	1
395	1
893	1
596 - 804	1
4320	1
2232	1
2003	1
1373	1
3508 - 4201	1
797	1
2572	1
886	1
1874	1
540 - 565	1
2285	1
5200	1
2448	1
910.2	1
2872	1
670 - 980	1
2274.24	1
766	1
2400 - 2600	1
716Sq. Meter	1
2295	1
45.06Sq. Meter	1
5924	1
1554.3	1
1649	1
5720	1
981 - 1249	1
3467.86	1
3301.8	1
2820	1
1483	1
2955	1
4201	1
1133 - 1384	1
888 - 1290	1

1562	1
15Acres	1
1544	1
981	1
1079 - 1183	1
1450 - 1950	1
4446	1
704 - 730	1
1445 - 1455	1
30000	1
1701	1
3560	1
6040	1
2461	1
3589	1
2087	1
461.82	1
1687	1
3876	1
1316	1
667	1
2249.81	1
3410	1
6150	1
2779	1
1857	1
2497	1
3090 - 5002	1
1205.47	1
1877	1
1506	1
3044	1
929 - 1078	1
1974	1
944	1
3435	1
1113.27	1
3580	1
1113.12	1
2312	1
4723	1
727	1
1Grounds	1
1053.4	1
1748	1
3197	1
1114	1

2026	1
605 - 624	1
9000	1
300Sq. Yards	1
1140 - 1250	1
1014	1
3270	1
475	1
942 - 1117	1
691	1
2920	1
1317	1
4304	1
633 - 666	1
1892	1
2249.81 - 4112.19	1
750 - 800	1
1769	1
3621	1
976	1
2223	1
1437	1
1503	1
3293	1
4007	1
515	1
2787	1
669	1
2437	1
590	1
26136	1
4920	1
1042 - 1105	1
804.1	1
2648	1
1120 - 1145	1
1867	1
763 - 805	1
2625	1
1574Sq. Yards	1
607	1
998	1
2826	1
2395	1
3080	1
2127	1
671	1

```
2511                  1
1932                  1
10200                 1
4560                  1
2916                  1
2122                  1
4290                  1
624                   1
3019                  1
5230                  1
3671                  1
770 - 841             1
2316                  1
651                   1
1070 - 1315           1
1642                  1
3160                  1
3640                  1
1393                  1
906                   1
315                   1
1921                  1
15                    1
1370.07               1
869                   1
1248.52               1
792                   1
614                   1
4360                  1
1597                  1
2181                  1
1500Sq. Meter         1
Name: total_sqft, dtype: int64
```

2. Chuyển trường total_sqft thành kiểu float

[27]:
```
total_sqft_float = []
for str_val in df3['total_sqft']:
  try:
    total_sqft_float.append(float(str_val))
  except:
    try:
      temp = []
      temp = str_val.split('-')
      total_sqft_float.append((float(temp[0])+float(temp[-1]))/2)
    except:
      total_sqft_float.append(np.nan)
```

[28]:
```
df4 = df3.reset_index(drop=True)
```

3. Thêm trường total_sqft kiểu float

[29]:
```
df5 = df4.join(pd.DataFrame({'total_sqft_float':total_sqft_float}))
df5.head() #Quan sát kết quả sau khi xử lí
```

[29]:

	area_type	availability	location	size	total_sqft	bath	balcony	price	total_sqft_float
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	1056	2.0	1.0	39.07	1056.0
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	2600	5.0	3.0	120.00	2600.0
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	1440	2.0	3.0	62.00	1440.0
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	1521	3.0	1.0	95.00	1521.0
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	1200	2.0	1.0	51.00	1200.0

6. Thông tin về số điểm dữ liệu null của từng trường

[30]:
```
df5.isnull().sum()
```

[30]:
```
area_type            0
availability         0
location             0
size                 0
total_sqft           0
bath                 0
balcony              0
price                0
total_sqft_float    46
dtype: int64
```

7. Bỏ đi các điểm dữ liệu (hàng) có giá trị null

[31]:
```
df6 = df5.dropna()
df6.shape
```

[31]:
```
(13200, 9)
```

8. Xem lại thông tin của dataframe

[32]:
```
df6.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13200 entries, 0 to 13245
```

```
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   area_type         13200 non-null  object
 1   availability      13200 non-null  object
 2   location          13200 non-null  object
 3   size              13200 non-null  object
 4   total_sqft        13200 non-null  object
 5   bath              13200 non-null  float64
 6   balcony           13200 non-null  float64
 7   price             13200 non-null  float64
 8   total_sqft_float  13200 non-null  float64
dtypes: float64(4), object(5)
memory usage: 1.0+ MB
```

9. Làm việc với feature: *size*

[33]:
```
# Quan sát sự phân bố giá trị của trường 'size' với value_counts
df6['size'].value_counts()
```

[33]:
```
2 BHK         5192
3 BHK         4277
4 Bedroom      816
4 BHK          574
3 Bedroom      541
1 BHK          527
2 Bedroom      325
5 Bedroom      293
6 Bedroom      190
1 Bedroom      100
7 Bedroom       83
8 Bedroom       83
5 BHK           56
9 Bedroom       45
6 BHK           30
7 BHK           17
1 RK            13
10 Bedroom      12
9 BHK            7
8 BHK            5
10 BHK           2
11 BHK           2
11 Bedroom       2
12 Bedroom       1
14 BHK           1
19 BHK           1
43 Bedroom       1
18 Bedroom       1
```

```
13 BHK           1
16 BHK           1
27 BHK           1
Name: size, dtype: int64
```

[34]:
```
#Chuyển thuộc tính số phòng từ dạng category về dạng numeric
size_int = []
for str_val in df6['size']:
  temp=[]
  temp = str_val.split(" ")
  try:
    size_int.append(int(temp[0]))
  except:
    size_int.append(np.nan)
    print("Noice = ",str_val)
```

[35]:
```
#Đánh lại index cho các hàng theo dãy số tự nhiên liên tiếp
df6 = df6.reset_index(drop=True)
```

[36]:
```
# Thêm trường dữ liệu số phòng (bhk)
df7 = df6.join(pd.DataFrame({'bhk':size_int}))
df7.shape
```

[36]:
```
(13200, 10)
```

[37]:
```
#In ra kết quả thực hiện các thao tác kể trên?
df7.tail()
```

[37]:

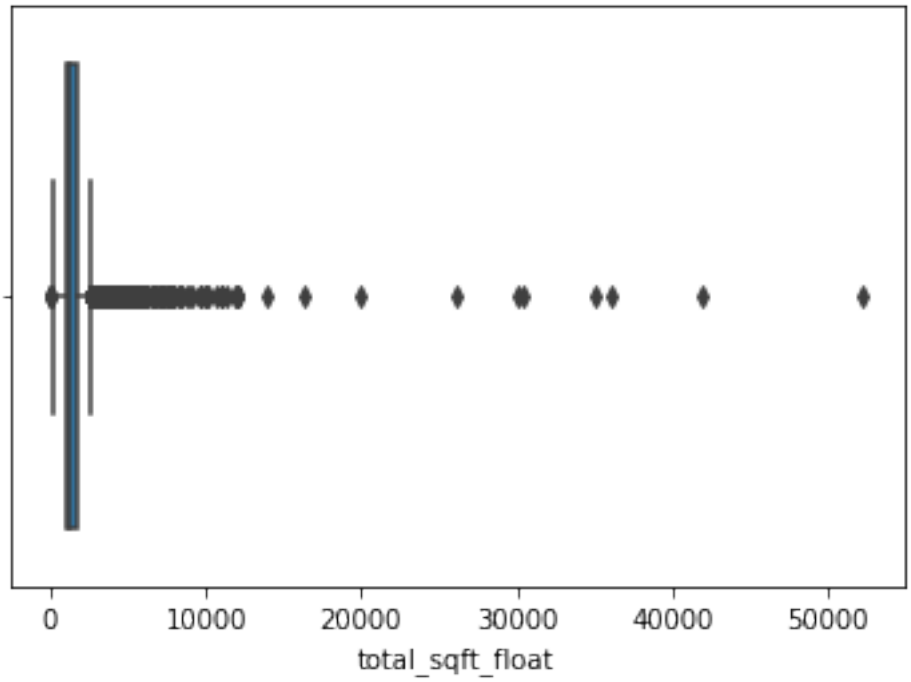
	area_type	availability	location	size	total_sqft	bath	balcony	price	total_sqft_float	bhk
13195	Built-up Area	Ready To Move	Whitefield	5 Bedroom	3453	4.0	0.000000	231.0	3453.0	5
13196	Super built-up Area	Ready To Move	Richards Town	4 BHK	3600	5.0	1.584376	400.0	3600.0	4
13197	Built-up Area	Ready To Move	Raja Rajeshwari Nagar	2 BHK	1141	2.0	1.000000	60.0	1141.0	2
13198	Super built-up Area	18-Jun	Padmanabhanagar	4 BHK	4689	4.0	1.000000	488.0	4689.0	4
13199	Super built-up Area	Ready To Move	Doddathoguru	1 BHK	550	1.0	1.000000	17.0	550.0	1

10. Finding Outlier and Removing

- Dựa trên biểu đồ boxplot vừa thực hiện ở trên/ hoặc công cụ khác để phát hiện và loại bỏ các điểm ngoại lai:

[38]:
```
# Xem xét trường diện tích:
sns.boxplot(x = df7['total_sqft_float'])
```

[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f59b3dfd5d0>
```



[39]:
```
# Chọn ngưỡng diện tích là 350 để xem xét
df7[df7['total_sqft_float']/df7['bhk'] < 350].head()
```

[39]:

	area_type	availability	location	size	total_sqft	bath	balcony	price	total_sqft_float	bhk
9	Plot Area	Ready To Move	Gandhi Bazar	6 Bedroom	1020	6.0	1.584376	370.0	1020.0	6
26	Super built-up Area	Ready To Move	Electronic City	2 BHK	660	1.0	1.000000	23.1	660.0	2
29	Super built-up Area	Ready To Move	Electronic City	3 BHK	1025	2.0	1.000000	47.0	1025.0	3
45	Plot Area	Ready To Move	HSR Layout	8 Bedroom	600	9.0	1.584376	200.0	600.0	8
57	Plot Area	Ready To Move	Murugeshpalya	6 Bedroom	1407	4.0	1.000000	150.0	1407.0	6

[40]:
```
# Loại bỏ đi các điểm dữ liệu có diện tích phòng trung bình >= 350
df8 = df7[~(df7['total_sqft_float']/df7['bhk'] < 350)]
df8.shape
```

[40]: (12106, 10)

[41]:
```
# Tạo thêm trường dữ liệu price_per_sqft (giá/ diện tích feet vuông)
df8['price_per_sqft'] = df8['price']*100000 / df8['total_sqft_float']
df8.head()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

[41]:

	area_type	availability	location	size	\
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	

	total_sqft	bath	balcony	price	total_sqft_float	bhk	price_per_sqft
0	1056	2.0	1.0	39.07	1056.0	2	3699.810606
1	2600	5.0	3.0	120.00	2600.0	4	4615.384615
2	1440	2.0	3.0	62.00	1440.0	3	4305.555556
3	1521	3.0	1.0	95.00	1521.0	3	6245.890861
4	1200	2.0	1.0	51.00	1200.0	2	4250.000000

[42]:
```
df8.price_per_sqft.describe()
```

[42]:
```
count     12106.000000
mean       6184.466889
std        4019.983503
min         267.829813
25%        4200.030048
50%        5261.108523
75%        6800.000000
max      176470.588235
Name: price_per_sqft, dtype: float64
```

3 Bài tập bổ sung (homework)

Phần bài tập này là các câu hỏi mở rộng, làm tiếp theo bài toán ở trên. Học viên cần viết mã để thực hiện các yêu cầu dưới đây:

Bài tập 0: Sử dụng sns.boxplot() để quan sát đặc điểm phân bố dữ liệu của các trường số, mỗi

trường này có outlier ko?

[ ]:
```
# Sử dụng boxplot để quan sát phân bố của dữ liệu và phát hiện ngoại lai của
 ↪từng trường dữ liệu trong vars
# Gợi ý: sns.boxplot(data_field)

vars = ['price', 'total_sqft_float', 'price_per_sqft', 'balcony', 'bath', 'bhk']
plt.figure(figsize=(16,12))

#Code ở đây
```

Bài tập 1: Viết hàm bỏ đi các điểm dữ liệu có price per sqft dựa trên mean, std của các ngôi nhà dựa trên từng vị trí

> Gợi ý: Xét trên từng vị trí (location), ngôi nhà thỏa mãn phải có $price\_per\_sqft \in [mean - std, mean + std]$

[43]:
```
def remove_pps_outliers(df):
  #Code ở đây
#--------
df9 = remove_pps_outliers(df8)
df9.shape
```

[43]:
```
(8888, 11)
```

Bài tập 2: Loại bỏ outlier xét theo trường bkh (số phòng)

Xét theo từng khu vực địa lí và theo từng loại nhà với số lượng phòng khác nhau, có một số ngôi nhà có giá không hợp lí (outliers), hãy tìm cách loại bỏ các outlier này. Cần ghi rõ quy tắc ghi nhận outlier

[61]:
```
def remove_bhk_outliers(df):
  # Code ở đây

df10 = remove_bhk_outliers(df9)
df10.shape
```

[61]:
```
(7194, 11)
```

Bài tập 3: Loại bỏ outlier khi xét trường 'bathroom'

[62]:
```
df10.bath.unique() #Có thể quan sát thấy một số căn nhà có số phòng tắm quá lớn
 ↪(VD: 12!!!)
```

[62]:
```
array([ 3.,  2.,  1.,  4.,  5.,  8.,  9.,  6.,  7., 12.])
```

[ ]:
```
df10[df10.bath > df10.bhk+2]
```

[64]:
```
df11 = #Code ở đây, sao cho: df10[df10.bath < df10.bhk+2]
df11.shape
```

[64]:
```
(7120, 11)
```

[ ]:
```
df11.head()
```

[ ]:
```
# Quan sát lại kết quả sau khi xử lí với boxplot

# (Dùng lại hàm đã code bên trên)
```

Bài tập 4: Xem xét bỏ đi các trường không cần thiết

Gợi ý: bỏ đi ['area_type', 'availability',"location","size","total_sqft"]

[ ]:
```
df12 = #Code ở đây
df12.head()
```

[53]:
```
#Lưu kết quả xử lí cuối cùng:

df12.to_csv("clean_data.csv", index=False)
```

Bài tập 5*: Viết hàm trực quan hóa thể hiện mối tương quan giữa tổng diện tích (total_sqft) và giá nhà (price) theo từng vị trí địa lí (location) (tùy chọn minh họa theo 2 vị trí nào đó), của những căn nhà có 2 hoặc 3 phòng. Và cần phân biệt rõ điểm dữ liệu nào tương ứng với nhà có 2 phòng, điểm nào tương ứng với nhà có 3 phòng?

Gợi ý: Kết quả tương tự như hình dưới/ hoặc biểu đồ khác có ý nghĩa tương đương

[ ]:
```
#Gợi ý: Sử dụng plt.scatter() .... hoạc câu lệnh khác tương đương. Làm với df9

def plot_scatter_chart(df,location):
  #Viết code ở đây

plot_scatter_chart(df9, "Rajaji Nagar")
```

[ ]:
```
plot_scatter_chart(df9, "Hebbal")
```

Bài tập 6*: Thực hiện các câu lệnh để trả lời các câu hỏi dưới đây:

- Thống kê giá nhà theo từng loại khu vực (area_type). Làm với df9:
- xem xét theo từng khu vực, thì giá nhà trung bình (price_per_sqft) là bao nhiêu, tương quan về giá nhà trung bình giữa các khu vực
- Gợi ý: Phần này có thể đưa ra kết quả dạng bảng hoặc biểu đồ (cột, histogram, ...).
- Sử dụng các lệnh: df.groupby(), df.sortvalues(), ... để trích xuất giá trị

- Sử dụng matplotlib: plt.bar(), …

[ ]:
```
# Code ở đây
```