In [2]:
```
import numpy as np
import pandas as pd
import re

import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:
```
data=pd.read_csv('Bengaluru_House_Data.csv')
```

In [4]:
```
data.head()
```

Out[4]:

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

In [5]:
```
data.describe()
```

Out[5]:

	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

In [6]:
```
data.info()
```

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

In [7]:
```
data.isnull().sum()
```

Out[7]:
```
area_type          0
availability       0
location           1
size              16
society         5502
total_sqft         0
bath              73
balcony          609
price              0
dtype: int64
```

In [8]:
```
data['society'].shape
```

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

In [9]:
```
data['size'].unique()
```

Out[9]:
```
array(['2 BHK', '4 Bedroom', '3 BHK', '4 BHK', '6 Bedroom', '3 Bedroom',
       '1 BHK', '1 RK', '1 Bedroom', '8 Bedroom', '2 Bedroom',
       '7 Bedroom', '5 BHK', '7 BHK', '6 BHK', '5 Bedroom', '11 BHK',
       '9 BHK', nan, '9 Bedroom', '27 BHK', '10 Bedroom', '11 Bedroom',
       '10 BHK', '19 BHK', '16 BHK', '43 Bedroom', '14 BHK', '8 BHK',
       '12 Bedroom', '13 BHK', '18 Bedroom'], dtype=object)
```

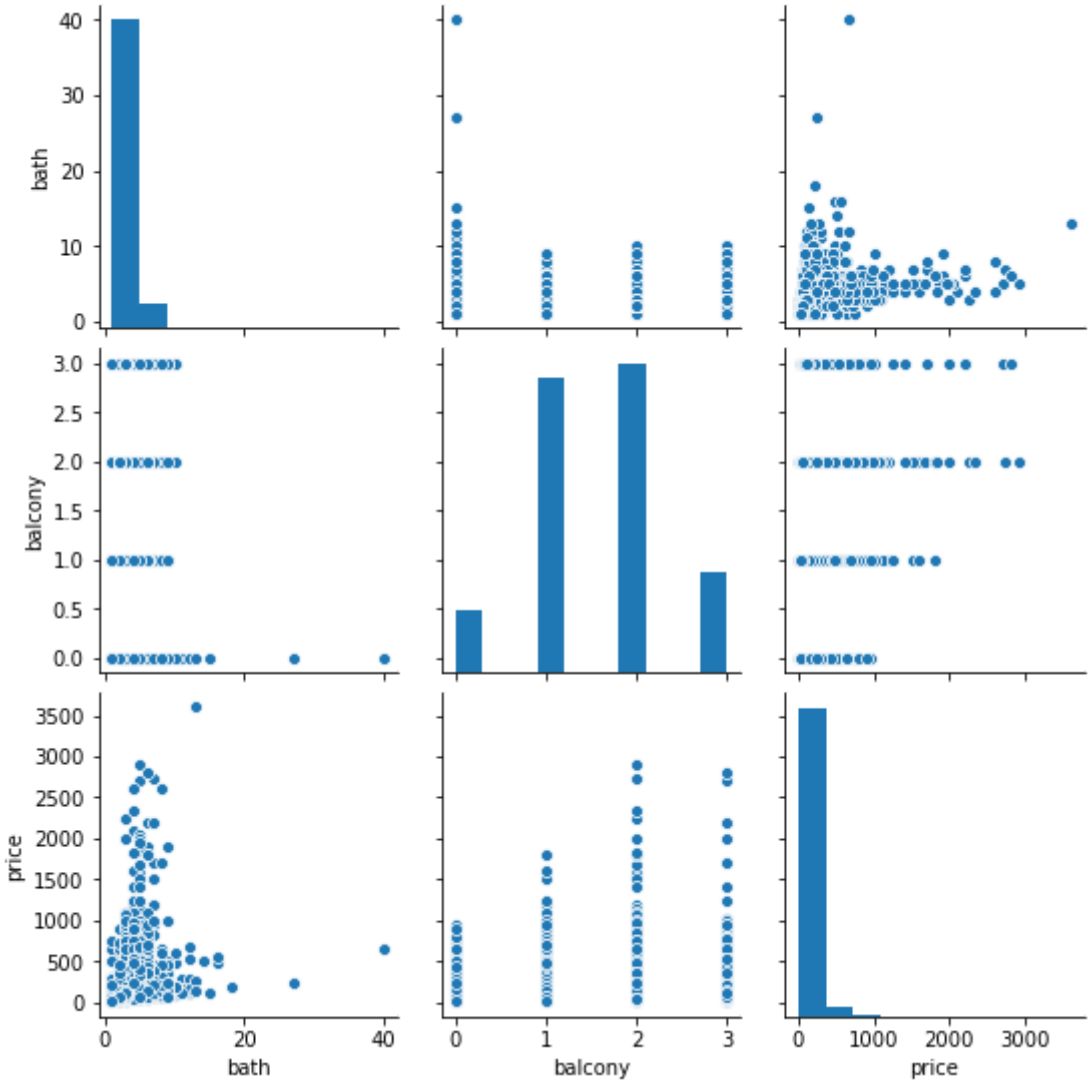
In [10]:
```
data.corr()
```

Out[10]:

	bath	balcony	price
bath	1.000000	0.204201	0.456345
balcony	0.204201	1.000000	0.120355
price	0.456345	0.120355	1.000000

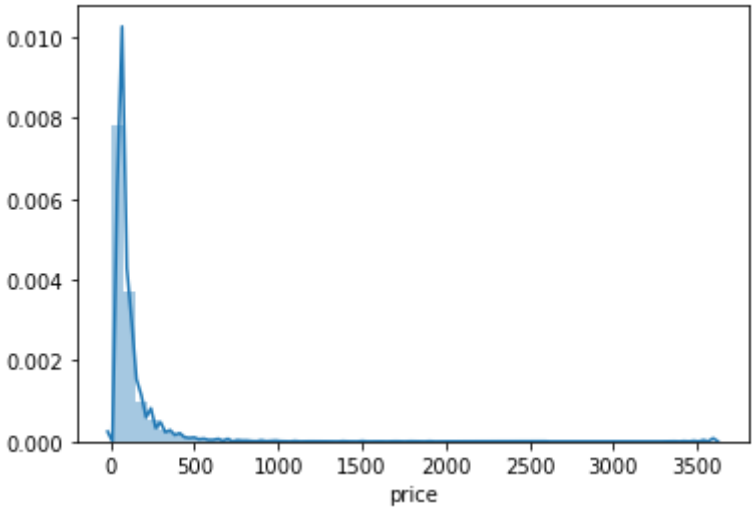
In [11]:
```
sns.pairplot(data)
```

Out[11]: <seaborn.axisgrid.PairGrid at 0x7fdb1caf6a10>



In [12]:
```
sns.distplot(data['price'])
```

Out[12]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdaf24a3490>
```



In [13]:
```
data.select_dtypes(exclude=['object']).describe()
```

Out[13]:

	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

In [14]:
```
corr=data.corr()
```

In [15]:
```
sns.heatmap(corr)
```

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x7fdaf1668a90>

In [16]:
```
from collections import Counter
Counter(data['total_sqft'])
```

Out[16]:
```
Counter({'1056': 12,
         '2600': 24,
         '1440': 23,
         '1521': 4,
         '1200': 843,
         '1170': 40,
         '2732': 3,
         '3300': 16,
         '1310': 37,
         '1020': 63,
         '1800': 104,
         '2785': 1,
         '1000': 172,
         '1100': 221,
         '2250': 13,
         '1175': 48,
         '1180': 58,
         '1540': 20,
         '2770': 3,
         '600': 180,
         '1755': 6,
         '2800': 28,
         '1767': 4,
         '510': 5,
         '1250': 114,
         '660': 20,
         '1610': 21,
         '1151': 12,
         '1025': 38,
         '2100 - 2850': 1,
         '1075': 66,
         '1760': 25,
         '1693': 9,
         '1925': 5,
         '700': 52,
         '1070': 53,
         '1724': 9,
         '1290': 37,
         '1143': 7,
         '1296': 12,
         '1254': 7,
```

```
'1330.74': 1,
'970': 13,
'1459': 4,
'800': 67,
'869': 1,
'1270': 41,
'1670': 8,
'2010': 6,
'1185': 41,
'1600': 101,
'3010 - 3410': 1,
'1500': 205,
'1407': 4,
'840': 13,
'4395': 5,
'845': 11,
'5700': 1,
'1160': 60,
'3000': 66,
'1140': 91,
'1220': 55,
'1350': 133,
'1005': 13,
'500': 34,
'1358': 5,
'1569': 4,
'1240': 46,
'2089': 2,
'1206': 11,
'1150': 101,
'2511': 1,
'460': 4,
'4400': 8,
'1660': 15,
'2957 - 3450': 1,
'1326': 5,
'1325': 21,
'1499': 2,
'1665': 18,
'708': 7,
'1060': 45,
'710': 6,
```

```
'1450': 70,
'2894': 1,
'1330': 38,
'2502': 3,
'650': 25,
'2400': 196,
'1007': 12,
'966': 4,
'1630': 14,
'1640': 19,
'782': 3,
'1260': 57,
'1413': 1,
'1116': 15,
'1530': 31,
'3700': 4,
'2497': 1,
'1436': 5,
'276': 1,
'1427': 7,
'2061': 3,
'3067 - 8156': 1,
'2650': 8,
'1282': 13,
'1050': 123,
'945': 9,
'950': 59,
'1870': 9,
'880': 10,
'1535': 13,
'1360': 33,
'1042 - 1105': 1,
'1280': 42,
'5000': 21,
'3050': 3,
'1563.05': 1,
'1167': 6,
'4000': 48,
'1828': 1,
'890': 9,
'1612': 2,
'1034': 4,
```

```
'1710': 13,
'957': 10,
'2795': 1,
'1125': 60,
'1735': 6,
'2050': 7,
'3750': 9,
'1063': 10,
'1904': 4,
'4200': 10,
'2000': 83,
'1145 - 1340': 1,
'1425': 16,
'1470': 21,
'1300': 117,
'450': 14,
'1152': 15,
'1550': 60,
'400': 17,
'705': 9,
'770': 10,
'1242': 6,
'1700': 58,
'2144': 3,
'1704': 3,
'1846': 6,
'1340': 31,
'1015 - 1540': 1,
'1327': 2,
'1186': 9,
'1783': 3,
'1400': 108,
'980': 18,
'1285': 15,
'912': 3,
'1225': 48,
'1909': 1,
'1359': 2,
'1207': 4,
'1736': 1,
'2850': 5,
'1595': 14,
```

```
'1798': 5,
'1475': 24,
'1580': 15,
'1295': 17,
'3600': 29,
'589': 2,
'1415': 10,
'1787': 6,
'984': 11,
'1520 - 1740': 1,
'2405': 1,
'1080': 55,
'1900': 32,
'805': 2,
'1153': 15,
'1148': 5,
'1110': 23,
'1933': 4,
'3500': 32,
'645': 16,
'1644': 3,
'910': 9,
'1577': 3,
'4050': 5,
'2420': 2,
'900': 112,
'1108': 9,
'3045': 1,
'2900': 12,
'1162': 11,
'1035': 30,
'1464': 32,
'1866': 2,
'1804': 11,
'913': 2,
'1868': 5,
'883': 23,
'1664': 5,
'2026': 1,
'1210': 44,
'4111': 3,
'1762': 7,
```

```
'1252': 12,
'861': 4,
'1420': 27,
'1490': 16,
'1084': 6,
'1015': 21,
'1017': 3,
'1027': 21,
'1069': 7,
'1349': 7,
'1417': 2,
'1863': 2,
'1010': 24,
'1847': 7,
'525': 21,
'1850': 37,
'1438': 2,
'1560': 30,
'850': 43,
'1113': 9,
'1385': 21,
'1128': 29,
'2390': 3,
'1645': 19,
'1192': 8,
'2135': 4,
'1173': 6,
'3122': 8,
'1230': 45,
'11': 1,
'1508': 7,
'1592': 4,
'1388': 2,
'630': 15,
'3252': 3,
'1308': 10,
'530': 6,
'1205': 31,
'930': 10,
'1380': 15,
'2483': 3,
'1166': 7,
```

```
'2023.71': 1,
'1935': 5,
'451': 1,
'1801': 5,
'1451': 6,
'1629': 1,
'1826': 5,
'1245': 25,
'1145': 29,
'825': 9,
'1113.27': 1,
'1460': 25,
'1656': 6,
'1208': 5,
'1910': 7,
'12000': 4,
'550': 20,
'34.46Sq. Meter': 1,
'750': 52,
'1090': 35,
'1991': 6,
'1105': 27,
'985': 22,
'1533': 5,
'1590': 25,
'1120': 32,
'1194': 36,
'1419': 9,
'2150': 11,
'11890': 1,
'1750': 39,
'1404': 12,
'1715': 9,
'1752.12': 4,
'1650': 69,
'1346': 11,
'3309': 1,
'1190': 40,
'1620': 10,
'2450': 4,
'1130': 29,
'1320': 46,
```

```
'4800': 18,
'929': 6,
'1753': 1,
'4500': 15,
'1196': 18,
'1040': 28,
'720': 14,
'1511': 1,
'1545': 7,
'375': 3,
'1062': 6,
'1115': 33,
'1195': 18,
'1246': 22,
'8500': 1,
'2805': 4,
'1584': 3,
'1353': 4,
'1599': 3,
'5230': 1,
'1155': 27,
'1867': 1,
'1251': 7,
'1028': 6,
'1222': 9,
'1372': 6,
'1135': 16,
'1768': 3,
'2610': 3,
'1286': 6,
'2845': 1,
'1195 - 1440': 1,
'3450': 5,
'1102': 6,
'656': 8,
'1780': 6,
'595': 4,
'2225': 6,
'1126': 6,
'4144': 2,
'2100': 46,
'2230': 3,
```

```
'1544': 1,
'1305': 24,
'1200 - 2400': 3,
'967': 4,
'540': 11,
'715': 5,
'2500': 62,
'1578': 2,
'1253': 5,
'961': 1,
'1709': 2,
'416': 4,
'1430': 31,
'1249': 2,
'2791': 1,
'834': 2,
'2060': 3,
'891': 3,
'1133': 8,
'2440': 3,
'940': 18,
'2160': 4,
'4104': 2,
'1790': 9,
'1920': 15,
'1374': 7,
'1445': 14,
'711': 2,
'1720': 16,
'1030': 23,
'1375': 17,
'469': 1,
'3800': 13,
'1820': 18,
'875': 8,
'4125Perch': 1,
'2378': 2,
'3385': 3,
'1641': 1,
'1120 - 1145': 1,
'2200': 31,
'1702': 4,
```

```
'1141': 14,
'2072': 18,
'4400 - 6640': 1,
'3090 - 5002': 1,
'35000': 1,
'1355': 28,
'1019': 6,
'1875': 13,
'1683': 5,
'1515': 11,
'2118': 1,
'1083': 6,
'2300': 16,
'4400 - 6800': 1,
'1092': 9,
'2264': 3,
'1033': 12,
'810': 6,
'1045': 16,
'1337': 1,
'1570': 19,
'1855': 6,
'1823': 2,
'1094': 3,
'1202': 4,
'1688': 2,
'1235': 16,
'3205': 3,
'1077': 5,
'2330': 3,
'425': 3,
'5270': 1,
'1468': 2,
'4300': 2,
'2280': 6,
'1341': 3,
'1279': 4,
'2760': 4,
'1101': 7,
'775': 4,
'667': 1,
'735': 3,
```

```
'4360': 1,
'1215': 35,
'820': 7,
'1160 - 1195': 1,
'1779': 3,
'1000Sq. Meter': 1,
'1694': 3,
'2376': 2,
'1975': 4,
'674': 9,
'445': 1,
'1618': 2,
'2181': 1,
'1556': 2,
'1179': 7,
'1275': 28,
'1615': 13,
'4000 - 5249': 2,
'920': 19,
'1602': 6,
'1176': 6,
'675': 14,
'1352': 8,
'1717': 6,
'10961': 2,
'2119': 5,
'1157': 17,
'1566': 3,
'2830': 4,
'1091': 4,
'3670': 1,
'918': 10,
'1950': 15,
'1695': 4,
'1705': 8,
'1447': 8,
'1114': 1,
'1022': 5,
'3761': 2,
'1339': 12,
'1198': 10,
'1691': 12,
```

```
'1115 - 1130': 2,
'2489': 1,
'1142': 5,
'1976': 4,
'5500': 2,
'1853': 3,
'1567': 5,
'995': 10,
'884': 4,
'1342': 7,
'1345': 18,
'1100Sq. Yards': 1,
'1652': 3,
'1740': 15,
'1278': 6,
'520 - 645': 1,
'1356': 3,
'823': 3,
'1897': 3,
'1575': 23,
'975': 18,
'686': 2,
'1410': 40,
'2238': 2,
'1174': 11,
'793': 2,
'1082': 13,
'1001': 2,
'1554': 2,
'1000 - 1285': 1,
'4239': 1,
'1680': 27,
'2470': 3,
'2825': 1,
'2480': 6,
'1799': 2,
'3606 - 5091': 1,
'1047': 9,
'1495': 22,
'3260': 2,
'1611': 4,
'3206': 2,
```

```
'1639': 11,
'1303': 6,
'650 - 665': 1,
'901': 2,
'1725': 9,
'1396': 2,
'1825': 5,
'1565': 12,
'1891': 4,
'1161': 10,
'633 - 666': 1,
'315': 1,
'665': 4,
'1255': 56,
'2112.95': 1,
'1810': 10,
'1548': 2,
'1485': 18,
'1256': 10,
'2268': 2,
'4100': 4,
'5.31Acres': 1,
'15': 1,
'1843': 12,
'1467': 1,
'1209': 3,
'1315': 21,
'3968': 1,
'1563': 4,
'2169': 1,
'3235': 1,
'1036': 7,
'1662': 5,
'1234': 4,
'1403': 5,
'915': 5,
'3900': 13,
'1405': 11,
'2557': 2,
'1480': 20,
'30Acres': 1,
'6136': 2,
```

```
'1390': 20,
'3100': 12,
'1824': 1,
'1561': 1,
'812': 8,
'1243': 21,
'1637': 1,
'1265': 19,
'24': 1,
'1666': 2,
'1357': 7,
'1093': 7,
'1865': 5,
'1039': 3,
'1985': 3,
'1520': 19,
'6000': 9,
'1117': 3,
'1937': 2,
'1053': 6,
'1625': 13,
'1397': 7,
'1523': 3,
'996': 4,
'3095': 2,
'1445 - 1455': 1,
'697': 2,
'884 - 1116': 1,
'1453': 7,
'3335': 3,
'850 - 1093': 1,
'1281': 2,
'565': 3,
'1024': 3,
'1233': 3,
'1510': 13,
'1945': 4,
'1008': 4,
'2689': 2,
'1697': 6,
'1651': 2,
'1078': 2,
```

```
'1314': 13,
'856': 2,
'1621': 3,
'1484': 6,
'14000': 1,
'485': 2,
'1617': 1,
'1384': 3,
'1408': 5,
'2292': 3,
'2006': 2,
'1984': 3,
'1960': 3,
'3024': 2,
'1586': 5,
'2325': 3,
'1440 - 1884': 1,
'924': 6,
'1006': 3,
'2842': 1,
'1558.67': 1,
'1542': 2,
'2750': 9,
'3596': 10,
'1635': 6,
'1239': 2,
'1596': 4,
'1009': 5,
'1726': 3,
'925': 20,
'3356': 2,
'1395': 8,
'1121': 3,
'1606': 1,
'4634': 1,
'1232': 17,
'680': 8,
'3467.86': 1,
'1132': 7,
'1262': 7,
'1840': 7,
'1655': 16,
```

```
'1730': 13,
'2195': 1,
'1079': 4,
'1085': 15,
'3200': 21,
'1739': 2,
'4346': 2,
'935': 16,
'1178': 8,
'1065': 36,
'999': 3,
'1835': 8,
'2090': 1,
'1519': 3,
'1267': 5,
'1146': 8,
'3329': 1,
'785': 1,
'921': 4,
'2700': 33,
'1197': 16,
'716Sq. Meter': 1,
'3073': 1,
'755': 4,
'1213': 6,
'1654': 5,
'1367': 4,
'620': 10,
'982': 7,
'440': 6,
'1614': 6,
'1363': 2,
'1156': 9,
'1455': 12,
'1890': 12,
'2197': 2,
'3522': 2,
'764': 1,
'1381': 2,
'340': 1,
'2465': 1,
'547.34 - 827.31': 1,
```

```
'865': 5,
'1756': 9,
'628': 1,
'1765': 5,
'1532': 8,
'804.1': 1,
'1525': 18,
'1435': 12,
'1272': 7,
'1541': 6,
'1201': 4,
'1785': 6,
'1258': 4,
'1297': 5,
'1291': 4,
'1012': 18,
'2790': 9,
'580 - 650': 1,
'1373': 1,
'1052': 5,
'3680': 1,
'2863': 1,
'2424': 1,
'2710': 5,
'3040': 3,
'1294': 3,
'1808': 2,
'3516': 1,
'3850': 4,
'3770': 1,
'1537': 4,
'1512': 6,
'1378': 1,
'1277': 7,
'8000': 4,
'2550': 3,
'3198': 2,
'1690': 13,
'2572': 1,
'520': 6,
'1219': 4,
'1564': 7,
```

```
'3425 - 3435': 1,
'1482': 9,
'1469': 3,
'551': 2,
'1745': 9,
'1269.72': 1,
'1266': 1,
'1168': 5,
'1552': 2,
'1804 - 2273': 2,
'817': 1,
'2070': 4,
'1815': 1,
'654': 6,
'1181': 6,
'1616': 4,
'1306': 7,
'1626': 3,
'1585': 5,
'1370': 21,
'3250': 9,
'3630 - 3800': 3,
'660 - 670': 1,
'1193': 1,
'1678': 3,
'1500Sq. Meter': 1,
'2780': 2,
'2422': 3,
'882': 2,
'620 - 933': 1,
'1757': 2,
'142.61Sq. Meter': 2,
'2028': 1,
'2611': 1,
'1163': 8,
'1098': 5,
'1418': 16,
'2695 - 2940': 1,
'727': 1,
'2774': 6,
'1722': 1,
'2000 - 5634': 1,
```

```
'914': 4,
'1184': 5,
'927': 4,
'1488': 3,
'1699': 2,
'712': 2,
'1476': 8,
'1429': 1,
'1188': 5,
'1574Sq. Yards': 1,
'1026': 8,
'2690': 6,
'3606': 1,
'3630': 3,
'1494': 4,
'1758': 3,
'3450 - 3472': 1,
'3675': 2,
'2337': 1,
'5800': 2,
'682': 2,
'1632': 3,
'1041': 7,
'5080': 1,
'3012': 2,
'1072': 4,
'52272': 1,
'726': 2,
'993': 4,
'1743': 2,
'1204': 5,
'640': 5,
'972': 4,
'955': 4,
'2479.13': 1,
'3042': 1,
'1068': 3,
'1354': 5,
'1238': 4,
'2204': 1,
'1364': 5,
'1329': 5,
```

```
'916': 1,
'688': 1,
'1331': 2,
'2106': 3,
'1718': 7,
'1936': 6,
'2172.65': 1,
'990': 8,
'1685': 11,
'740': 5,
'1172': 3,
'896': 3,
'2460': 2,
'2025': 2,
'2475': 9,
'765': 2,
'3155': 3,
'1980': 4,
'3555': 1,
'585': 2,
'2045': 2,
'6150': 1,
'671': 1,
'2017': 3,
'1139': 7,
'581.91': 1,
'1158': 4,
'3951': 2,
'360': 4,
'1307': 3,
'1703': 8,
'1343': 4,
'1852': 5,
'1546': 2,
'1424': 5,
'1398': 3,
'1322': 8,
'432': 2,
'4750': 4,
'1250 - 1305': 1,
'1571': 6,
'1031': 5,
```

```
'1109': 5,
'2350': 10,
'1003': 2,
'824': 1,
'9600': 3,
'1111': 4,
'2519': 1,
'1837': 4,
'1673': 1,
'1112': 7,
'545': 5,
'670 - 980': 1,
'1754': 5,
'3584': 1,
'2168': 1,
'3125': 1,
'1095': 18,
'870': 8,
'1708': 4,
'1183': 7,
'3067': 3,
'1929': 4,
'1775': 1,
'1268': 6,
'1687': 1,
'3526': 3,
'983': 1,
'2039': 1,
'1316': 1,
'1005.03 - 1252.49': 1,
'1605': 9,
'1313': 6,
'1934': 2,
'877': 4,
'936': 6,
'1901': 1,
'1819': 3,
'3245': 1,
'1845': 3,
'2254': 5,
'515': 1,
'1893': 6,
```

```
'3025': 1,
'830': 6,
'2145': 8,
'1004 - 1204': 1,
'361.33Sq. Yards': 1,
'987': 1,
'2556': 2,
'946': 4,
'2121': 1,
'860': 6,
'1223': 9,
'1788': 3,
'1144': 3,
'1862': 5,
'1646': 1,
'1995': 2,
'1692': 3,
'6040': 1,
'2040': 6,
'1311': 3,
'4850': 1,
'645 - 936': 2,
'1942': 3,
'668': 1,
'1089': 8,
'960': 17,
'1583': 6,
'1018': 3,
'3400': 10,
'2710 - 3360': 1,
'395': 1,
'1259': 4,
'1216': 25,
'1428': 3,
'605': 6,
'1444': 5,
'1187': 8,
'1452.55': 1,
'2357': 3,
'1448': 3,
'296': 1,
'1058': 4,
```

```
'1411': 6,
'2249.81 - 4112.19': 3,
'2503': 3,
'2524': 2,
'1634': 4,
'1603': 3,
'714': 1,
'1573': 5,
'1465': 10,
'351': 1,
'2526': 1,
'3436 - 3643': 1,
'965': 11,
'1229': 2,
'2830 - 2882': 5,
'1055': 5,
'1304': 6,
'2720': 2,
'596 - 804': 1,
'1365': 19,
'1165': 30,
'1776.42': 1,
'1107': 6,
'2319': 1,
'1536': 3,
'1164': 3,
'11338': 1,
'30000': 1,
'3190': 1,
'1917': 7,
'1071': 3,
'4460': 1,
'3297': 1,
'693': 2,
'1021': 5,
'1608': 4,
'2289': 3,
'2257': 3,
'1263': 5,
'1255 - 1863': 1,
'1043': 3,
'1300 - 1405': 1,
```

```
'590': 1,
'1299': 8,
'3161': 1,
'1124': 3,
'871': 1,
'3515': 1,
'1118': 4,
'1051': 4,
'1478': 3,
'1226': 5,
'3366': 1,
'760': 10,
'1728': 2,
'2215': 13,
'1562': 1,
'1555': 10,
'2105': 1,
'6200': 3,
'606': 2,
'527': 2,
'2321': 2,
'1500 - 2400': 1,
'2167': 1,
'1274': 6,
'117Sq. Yards': 1,
'780': 9,
'7500': 7,
'904': 1,
'730': 2,
'977': 1,
'2540': 2,
'795': 4,
'1463': 2,
'2401': 1,
'2065': 3,
...})
```

In [17]:
```
data.shape
```

Out[17]: (13320, 9)

In [18]:
```
#preprocessing the total sqft cols as it has vivid entries
def preprocess_total_sqft(my_list):
    if len(my_list) == 1:

        try:
            return float(my_list[0])
        except:
            strings = ['Sq. Meter', 'Sq. Yards', 'Perch', 'Acres', 'Cents', 'Guntha', 'Grounds']
            split_list = re.split('(\d*.*\d)', my_list[0])[1:]
            area = float(split_list[0])
            type_of_area = split_list[1]

            if type_of_area == 'Sq. Meter':
                area_in_sqft = area * 10.7639
            elif type_of_area == 'Sq. Yards':
                area_in_sqft = area * 9.0
            elif type_of_area == 'Perch':
                area_in_sqft = area * 272.25
            elif type_of_area == 'Acres':
                area_in_sqft = area * 43560.0
            elif type_of_area == 'Cents':
                area_in_sqft = area * 435.61545
            elif type_of_area == 'Guntha':
                area_in_sqft = area * 1089.0
            elif type_of_area == 'Grounds':
                area_in_sqft = area * 2400.0
            return float(area_in_sqft)

    else:
        return (float(my_list[0]) + float(my_list[1]))/2.0
```

In [19]:
```
data['total_sqft'] = data.total_sqft.str.split('-').apply(preprocess_total_sqft)
```

In [20]:
```
#converting the categorical to numerical data - area_type
data.area_type.value_counts()
```

Out[20]:
```
Super built-up  Area    8790
Built-up  Area          2418
Plot  Area              2025
Carpet  Area              87
Name: area_type, dtype: int64
```

In [21]:
```
replace_area_type = {'Super built-up  Area': 0, 'Built-up  Area': 1, 'Plot  Area': 2, 'Carpet  Area': 3}
data['area_type'] = data.area_type.map(replace_area_type)
```

In [22]:
```
#converting the categorical to numerical data - availabilty
data.availability.value_counts()
```

Out[22]:
```
Ready To Move    10581
18-Dec             307
18-May             295
18-Apr             271
18-Aug             200
                   ...
14-Jul               1
15-Jun               1
16-Oct               1
15-Aug               1
14-Nov               1
Name: availability, Length: 81, dtype: int64
```

In [23]:
```
def replace_availabilty(my_string):
    if my_string == 'Ready To Move':
        return 0
    elif my_string == 'Immediate Possession':
        return 1
    else:
        return 2
```

In [24]:
```
data['availability'] = data.availability.apply(replace_availabilty)
```

In [25]:
```
#converting NaN in location
data['location'].isnull().sum()
```

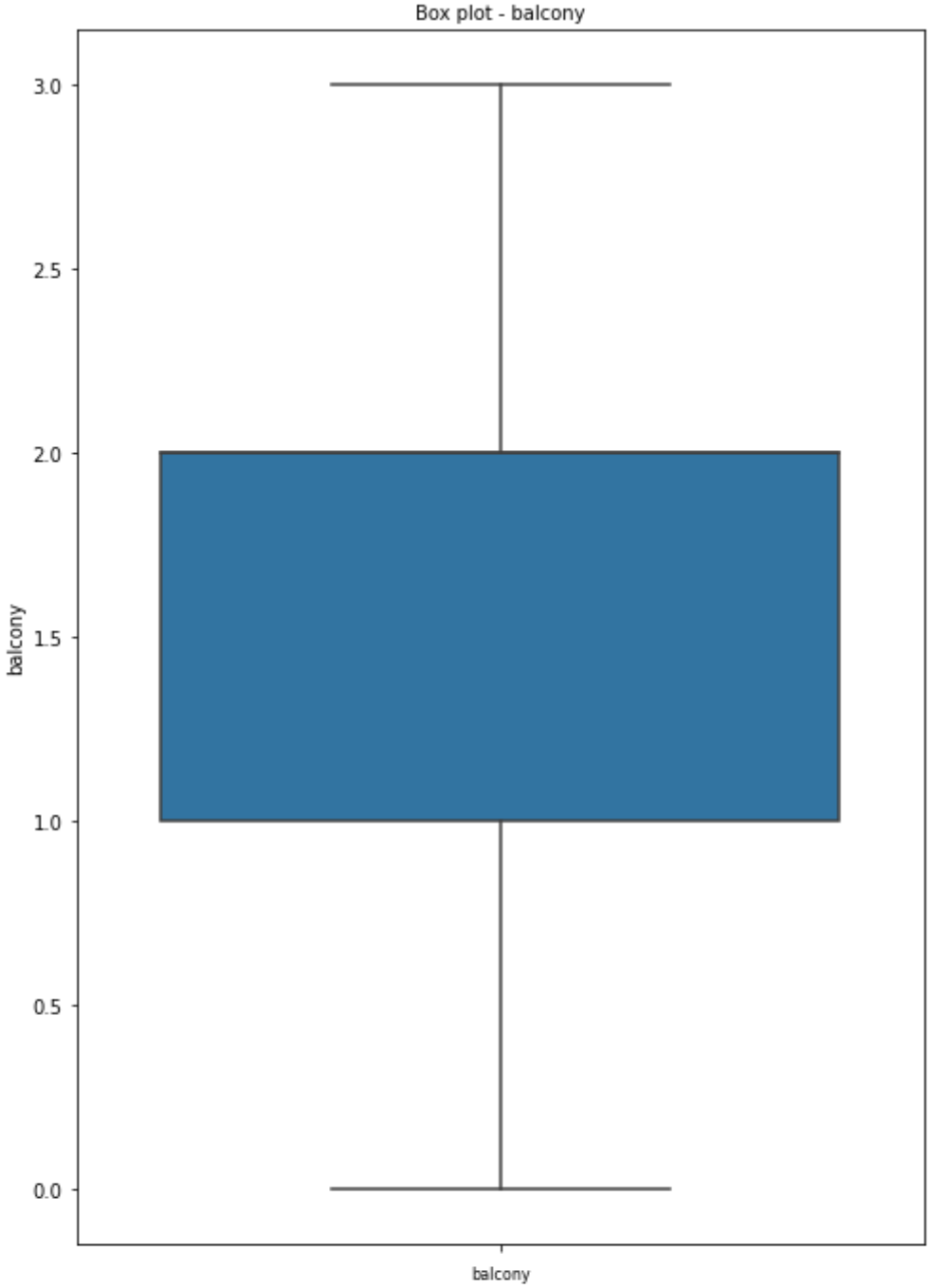
Out[25]: 1

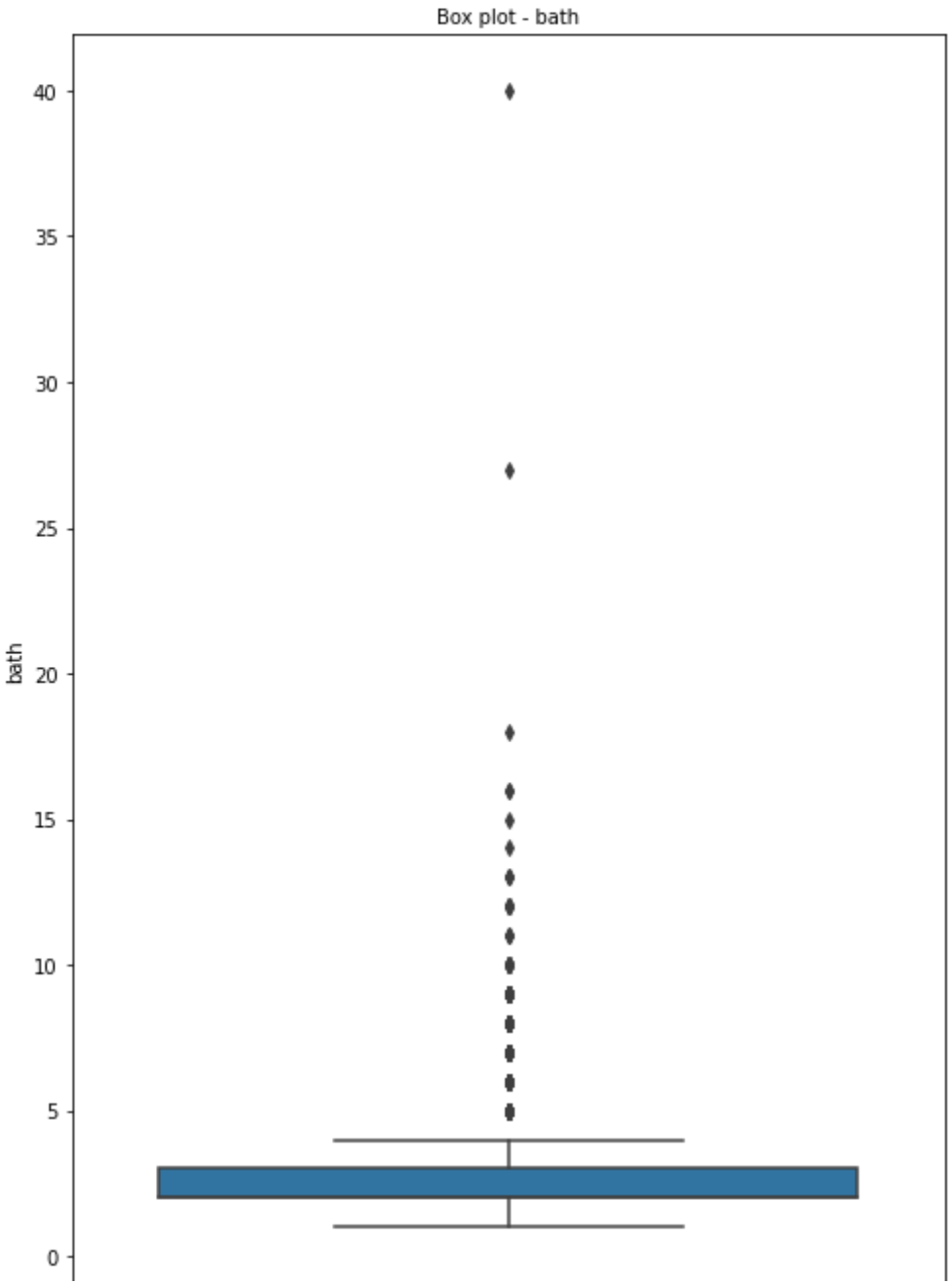
In [26]:
```
data['location'] = data['location'].fillna('No Location')
```

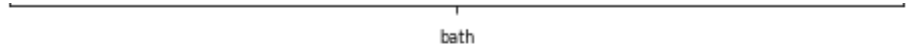
In [27]:
```
#converting the categorical to numerical data - size
Counter(data['size'])
```

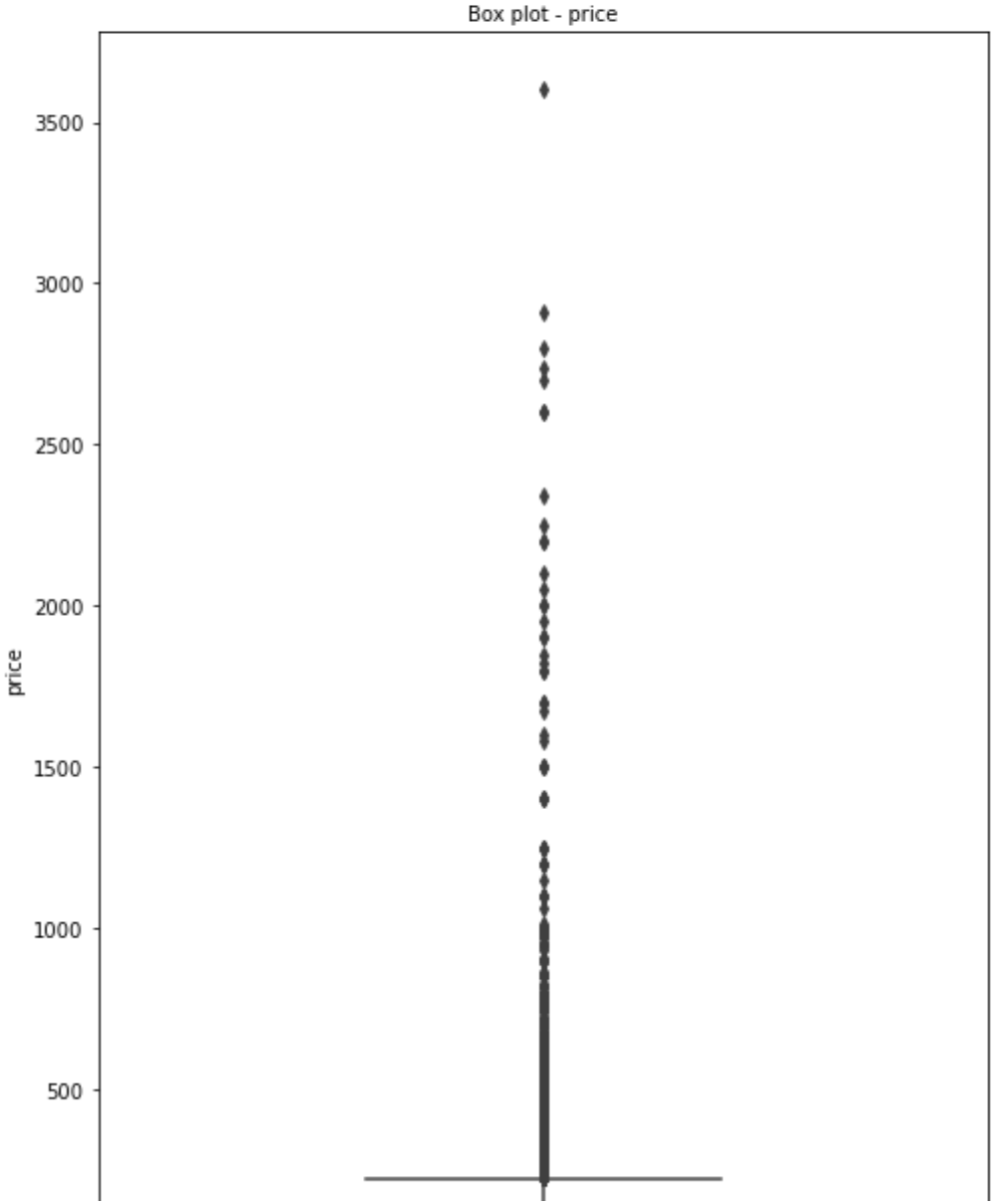
Out[27]:
```
Counter({'2 BHK': 5199,
         '4 Bedroom': 826,
         '3 BHK': 4310,
         '4 BHK': 591,
         '6 Bedroom': 191,
         '3 Bedroom': 547,
         '1 BHK': 538,
         '1 RK': 13,
         '1 Bedroom': 105,
         '8 Bedroom': 84,
         '2 Bedroom': 329,
         '7 Bedroom': 83,
         '5 BHK': 59,
         '7 BHK': 17,
         '6 BHK': 30,
         '5 Bedroom': 297,
         '11 BHK': 2,
         '9 BHK': 8,
         nan: 16,
         '9 Bedroom': 46,
         '27 BHK': 1,
         '10 Bedroom': 12,
         '11 Bedroom': 2,
         '10 BHK': 2,
         '19 BHK': 1,
         '16 BHK': 1,
         '43 Bedroom': 1,
         '14 BHK': 1,
         '8 BHK': 5,
         '12 Bedroom': 1,
         '13 BHK': 1,
         '18 Bedroom': 1})
```

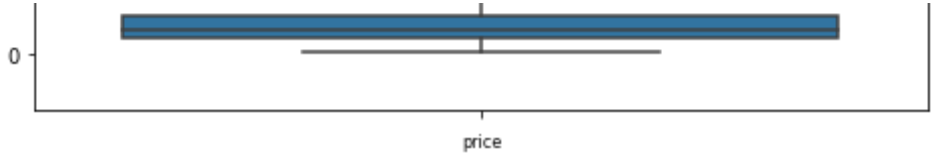
In [28]:
```
col_names = ['balcony','bath', 'price']

fig, ax = plt.subplots(len(col_names), figsize=(8,40))

for i, col_val in enumerate(col_names):

    sns.boxplot(y=data[col_val], ax=ax[i])
    ax[i].set_title('Box plot - '+col_val, fontsize=10)
    ax[i].set_xlabel(col_val, fontsize=8)

plt.show()
```


Box plot - balcony
3.0
2.5
2.0
1.5
1.0
0.5
0.0
balcony
balcony


Box plot - bath
bath
40
35
30
25
20
15
10
5
0


bath


Box plot - price
price
3500
3000
2500
2000
1500
1000
500



In [29]:
```
data.isnull().sum()
```

Out[29]:
```
area_type          0
availability       0
location           0
size              16
society         5502
total_sqft         0
bath              73
balcony          609
price              0
dtype: int64
```

In [30]:
```
data['size'].fillna('ffill',inplace=True)
```

In [31]:
```
data['society'].fillna('ffill',inplace=True)
data['bath'].fillna('ffill',inplace=True)
data['balcony'].fillna('ffill',inplace=True)
```

In [32]:
```
data.isnull().sum()
```

Out[32]:
```
area_type       0
availability    0
location        0
size            0
society         0
total_sqft      0
bath            0
balcony         0
price           0
dtype: int64
```

In [ ]:
```

```