In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
import re
```

Bussiness problem

Data mining or Data Understanding

In [2]:

```
df=pd.read_csv(r"C:\Users\moods\Downloads\Bengaluru_House_Data.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	area_type	availability	location	size	society	total_sqft	bath	balcony	pric
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.0
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.0
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.0
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.0
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.0

In [4]:

```
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
```

In [5]:

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

Out[5]:

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

Data PreProcessing

To find the Outliers Extensions

In [6]:

```
df.dtypes
```

Out[6]:

```
area_type        object
availability     object
location         object
size             object
society          object
total_sqft       object
bath            float64
balcony         float64
price           float64
dtype: object
```

In [7]:

```
df["total_sqft"].unique()
```

Out[7]:

```
array(['1056', '2600', '1440', ..., '1133 - 1384', '774', '4689'],
      dtype=object)
```

In [8]:

```
def cleaning(sento):
    try:
        u = int(sento)
        return u
    except:
        u=sento.split(" ")
        u=u[0]
        u=re.sub('[^0-9]',"",u)
        return u
```

In [9]:

```
df["total_sqft"]=df["total_sqft"].apply(lambda x:cleaning(x))
```

In [10]:

```
df.dtypes
```

Out[10]:

```
area_type        object
availability     object
location         object
size             object
society          object
total_sqft       object
bath            float64
balcony         float64
price           float64
dtype: object
```

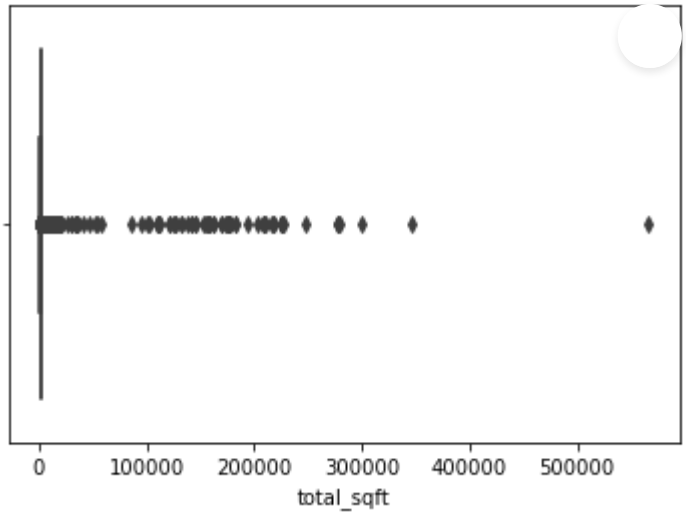
In [11]:

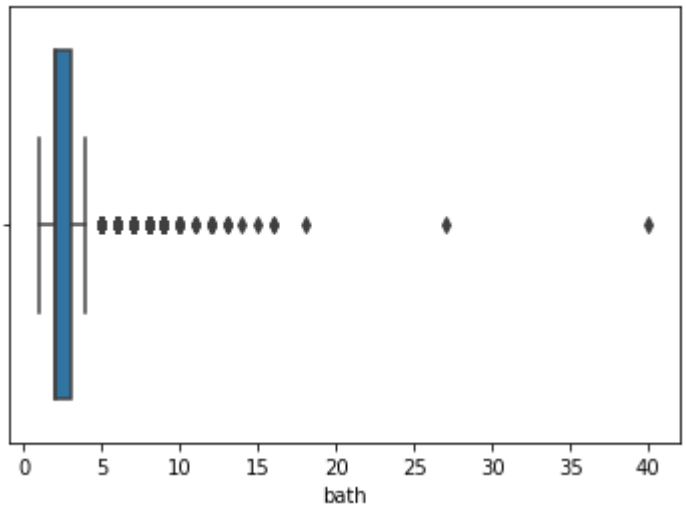
```
df["total_sqft"]=df["total_sqft"].astype("float64")
```

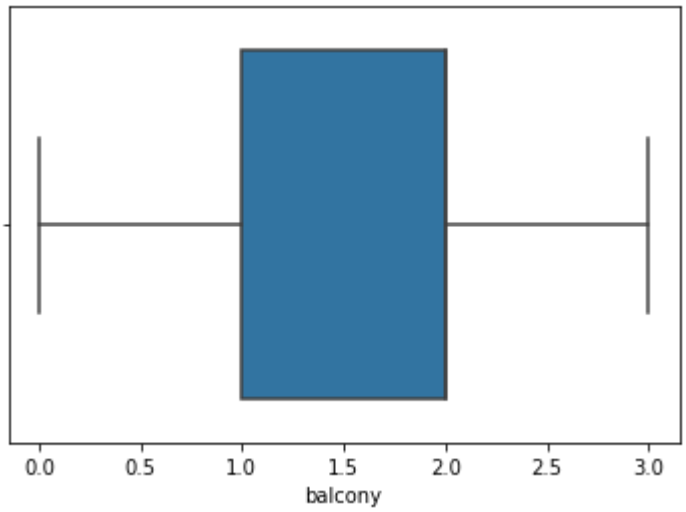
In [12]:

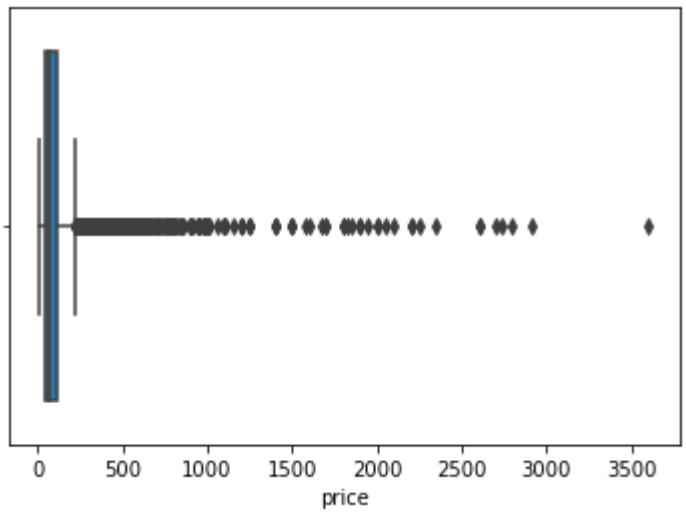
```
name=[]
cate=[]
for i in df.columns:
    try:
        sns.boxplot(df[i])
        plt.show()
        name.append(i)
    except:
        print("This is Categorical Data :{}".format(i))
        cate.append(i)
```

```
This is Categorical Data :area_type
This is Categorical Data :availability
This is Categorical Data :location
This is Categorical Data :size
This is Categorical Data :society
```



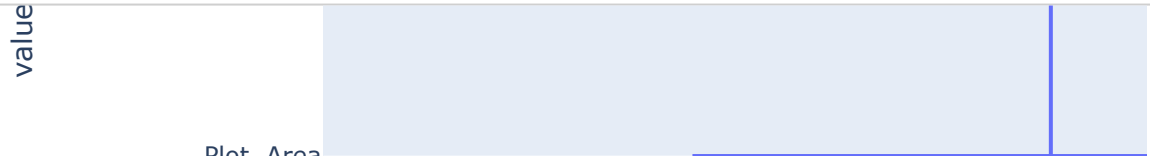






In [13]:

```
for i in df.columns:
    try:
        fig=px.box(df[i])
        fig.show()
    except:
        print("This is Categorial Data:{}".format(i))
```



This is Outliers in Bath fill Bath Value With Median & remaining Catgorial fill with mode

In [14]:

```
from sklearn.impute import SimpleImputer
```

In [15]:

```
median=SimpleImputer(missing_values=np.nan,strategy="median")
```

In [16]:

```
mode=SimpleImputer(missing_values=np.nan,strategy="most_frequent")
```

In [17]:

```
mean=SimpleImputer(missing_values=np.nan,strategy="mean")
```

In [18]:

```
for i in df.columns:
    if type(df[i][0])==str:
        if df[i].isnull().sum()==0:
            pass
        else:
            df[i]=mode.fit_transform(df[[i]])
    else:
        df[i]=mean.fit_transform(df[[i]])
```

In [19]:

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

Out[19]:

```
area_type       0
availability    0
location        0
size            0
society         0
total_sqft      0
bath            0
balcony         0
price           0
dtype: int64
```

Treating outliers

In [20]:

```
from feature_engine.outliers import Winsorizer
```

In [21]:

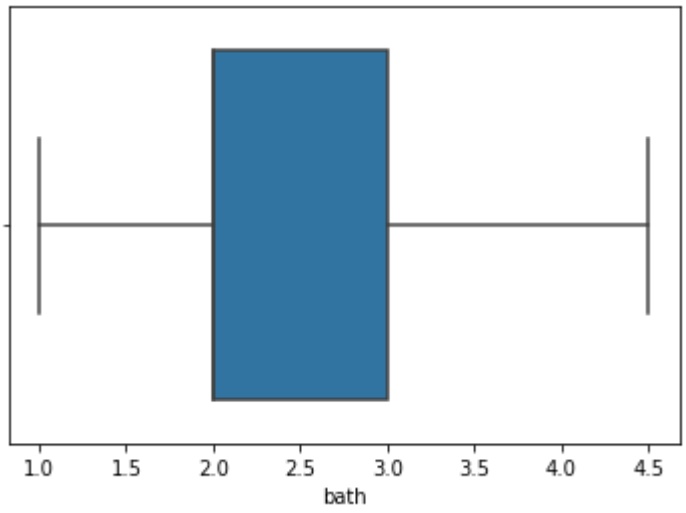
```
win=Winsorizer(capping_method='iqr',tail="both",fold=1.5,variables=["bath"])
```

In [22]:

```
df["bath"]=win.fit_transform(df[["bath"]])
```

In [23]:

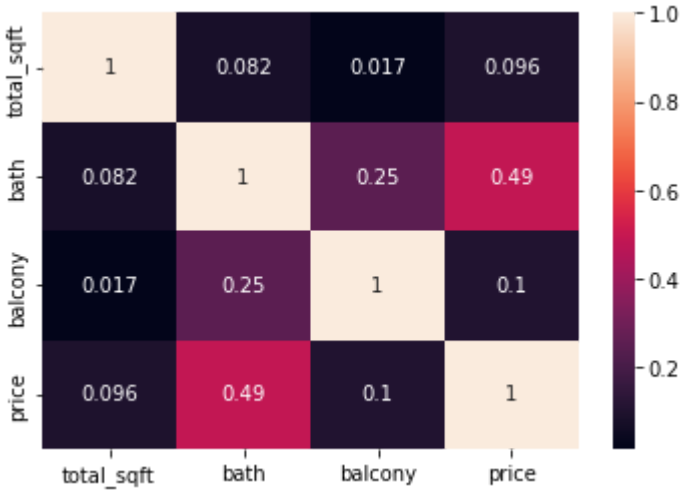
```
sns.boxplot(df["bath"])
plt.show()
```



outliers removed let's check for skewness to remove skewness check for co-relation

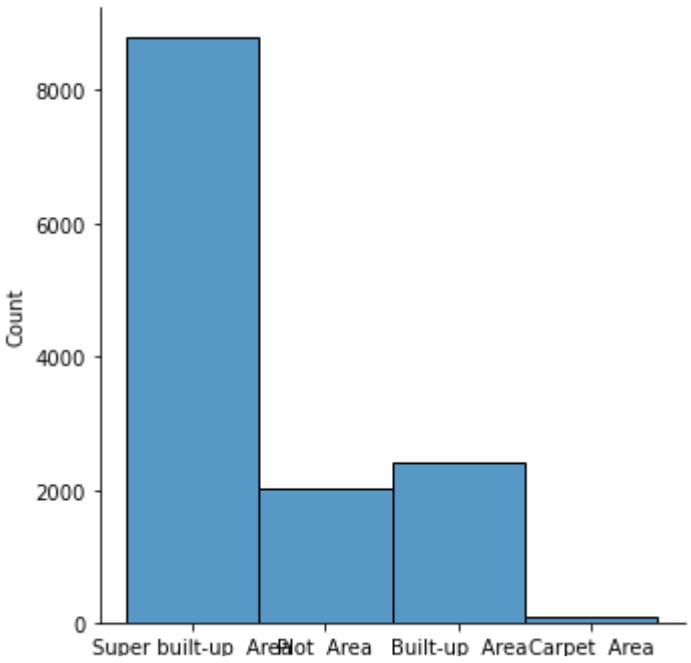
In [24]:

```
sns.heatmap(df.corr(),annot=True)
plt.show()
```



In [25]:

```
for i in df.columns:
    try:
        sns.displot(df[i])
        plt.show()
    except:
        pass
```



In [27]:

```
cate
```

Out[27]:

```
['area_type', 'availability', 'location', 'size', 'society']
```

In [28]:

```
df.head(3)
```

Out[28]:

	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	1056.0	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600.0	4.5	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	GrrvaGr	1440.0	2.0	3.0	62.00

In [32]:

```
for i in cate:
    x=df[i].nunique()
    print("{}The unique values in {}".format(x,i))
```

```
4The unique values in area_type
81The unique values in availability
1305The unique values in location
31The unique values in size
2688The unique values in society
```

Encoding

In [33]:

```
from sklearn.preprocessing import LabelEncoder,OneHotEncoder
```

In [39]:

```
Le=LabelEncoder()
Oe=OneHotEncoder(sparse=False,handle_unknown='error')
```

In [40]:

```
df["area_type"]= Le.fit_transform(df[["area_type"]])
```

In [41]:

```
df["size"]=Le.fit_transform(df[["size"]])
```

In [42]:

```
df.head()
```

Out[42]:

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	3	19-Dec	Electronic City Phase II	13	Coomee	1056.0	2.0	1.0	39.07
1	2	Ready To Move	Chikka Tirupathi	19	Theanmp	2600.0	4.5	3.0	120.00
2	0	Ready To Move	Uttarahalli	16	GrrvaGr	1440.0	2.0	3.0	62.00
3	3	Ready To Move	Lingadheeranahalli	16	Soiewre	1521.0	3.0	1.0	95.00
4	3	Ready To Move	Kothanur	13	GrrvaGr	1200.0	2.0	1.0	51.00

In [44]:

```
df.drop(columns=["availability","location","society"],inplace=True)
```

In [45]:

```
x=df.drop("price",axis=1)
```

In [46]:

```
y=df["price"]
```

In [47]:

```
from sklearn.model_selection import train_test_split
```

In [50]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

In [52]:

```
from sklearn .ensemble import RandomForestRegressor
```

In [57]:

```
ModellRRR=RandomForestRegressor()
```

In [58]:

```
ModellRRR.fit(x_train,y_train)
```

Out[58]:

```
RandomForestRegressor()
```

In [59]:

```
ModellRRR.score(x_train,y_train)
```

Out[59]:

```
0.9168904912575703
```

In [60]:

```
ModellRRR.score(x_test,y_test)
```

Out[60]:

```
0.5477949050113957
```

```
model is overfitted
```

In [62]:

```
y_pred=ModellRRR.predict(x_test)
```

In [63]:

```
y_pred
```

Out[63]:

```
array([ 91.47416667,  93.425     ,  58.9361848 , ...,  35.23965742,
        39.28725712, 186.25      ])
```

```
Evalution
```

In [65]:

```
from sklearn.metrics import mean_squared_error
from math import sqrt
```

In [66]:

```
mse=mean_squared_error(y_test,y_pred)
```

In [67]:

```
mse
```

Out[67]:

```
9899.438473770875
```

In [69]:

```
rmse=sqrt(mean_squared_error(y_test,y_pred))
rmse
```

Out[69]:

```
99.49592189517556
```

In [ ]:

```

```