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

df = pd.read_csv('House Price India.csv')
df.head()
```

```
           id   Date  number of bedrooms  number of bathrooms  living 
area  \
0  6762810145  42491                   5                 2.50         
3650   
1  6762810635  42491                   4                 2.50         
2920   
2  6762810998  42491                   5                 2.75         
2910   
3  6762812605  42491                   4                 2.50         
3310   
4  6762812919  42491                   3                 2.00         
2710   

   lot area  number of floors  waterfront present  number of views  \
0      9050               2.0                   0                4   
1      4000               1.5                   0                0   
2      9480               1.5                   0                0   
3     42998               2.0                   0                0   
4      4500               1.5                   0                0   

   condition of the house  ...  Built Year  Renovation Year  Postal 
Code  \
0                       5  ...        1921                0       
122003   
1                       5  ...        1909                0       
122004   
2                       3  ...        1939                0       
122004   
3                       3  ...        2001                0       
122005   
4                       4  ...        1929                0       
122006   

   Lattitude  Longitude  living_area_renov  lot_area_renov  \
0    52.8645   -114.557               2880            5400   
1    52.8878   -114.470               2470            4000   
2    52.8852   -114.468               2940            6600   
3    52.9532   -114.321               3350           42847   
4    52.9047   -114.485               2060            4500   

   Number of schools nearby  Distance from the airport    Price  
0                         2                         58  2380000  
```

```
1                         2                         51  1400000
2                         1                         53  1200000
3                         3                         76   838000
4                         1                         51   805000

[5 rows x 23 columns]
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14620 entries, 0 to 14619
Data columns (total 23 columns):
 #   Column                                 Non-Null Count  Dtype  
---  ------                                 --------------  -----  
 0   id                                     14620 non-null  int64  
 1   Date                                   14620 non-null  int64  
 2   number of bedrooms                     14620 non-null  int64  
 3   number of bathrooms                    14620 non-null  float64
 4   living area                            14620 non-null  int64  
 5   lot area                               14620 non-null  int64  
 6   number of floors                       14620 non-null  float64
 7   waterfront present                     14620 non-null  int64  
 8   number of views                        14620 non-null  int64  
 9   condition of the house                 14620 non-null  int64  
 10  grade of the house                     14620 non-null  int64  
 11  Area of the house(excluding basement)  14620 non-null  int64  
 12  Area of the basement                   14620 non-null  int64  
 13  Built Year                             14620 non-null  int64  
 14  Renovation Year                        14620 non-null  int64  
 15  Postal Code                            14620 non-null  int64  
 16  Lattitude                              14620 non-null  float64
 17  Longitude                              14620 non-null  float64
 18  living_area_renov                      14620 non-null  int64  
 19  lot_area_renov                         14620 non-null  int64  
 20  Number of schools nearby               14620 non-null  int64  
 21  Distance from the airport              14620 non-null  int64  
 22  Price                                  14620 non-null  int64  
dtypes: float64(4), int64(19)
memory usage: 2.6 MB
```

Handling Missing Values

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

```
number of bedrooms                       0
number of bathrooms                      0
living area                              0
lot area                                 0
number of floors                         0
waterfront present                       0
```

```
number of views                          0
condition of the house                   0
grade of the house                       0
Area of the house(excluding basement)    0
Area of the basement                     0
Built Year                               0
Renovation Year                          0
Postal Code                              0
Lattitude                                0
Longitude                                0
living_area_renov                        0
lot_area_renov                           0
Number of schools nearby                 0
Distance from the airport                0
Price                                    0
dtype: int64
```

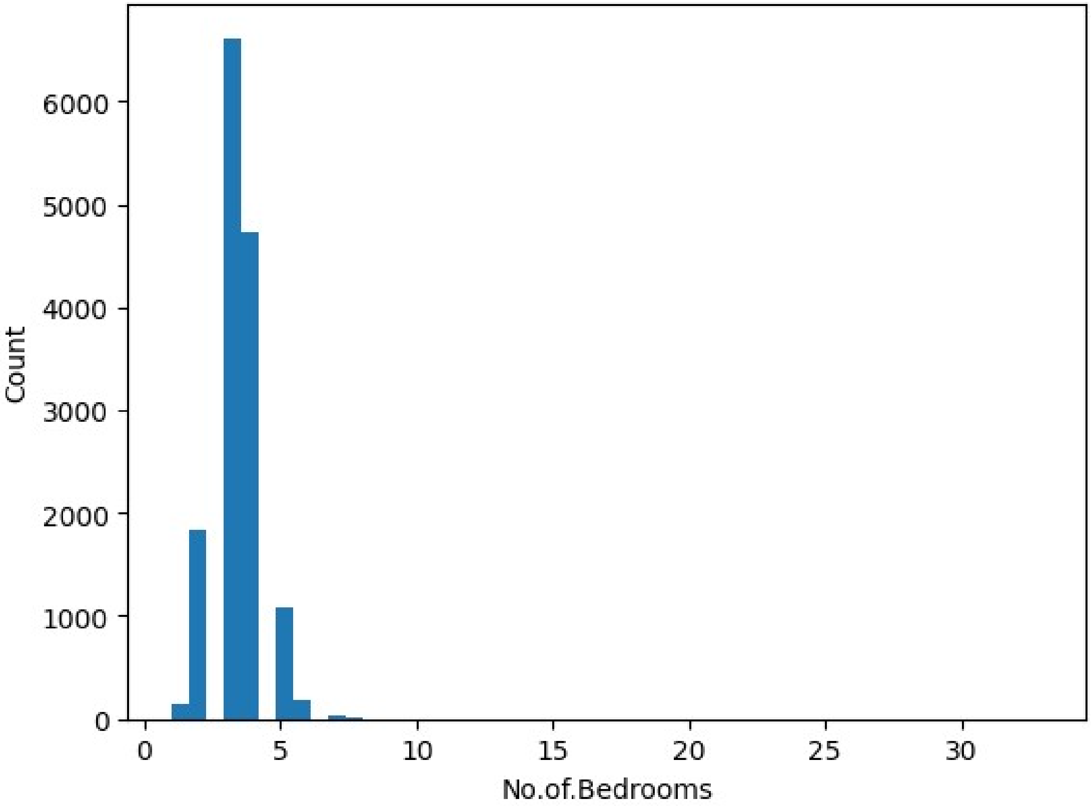
The above information shows that the none of the columns contains any null value in it. We don't need to perform any specific operations to handle the missing values.

Univariate Analysis

Histogram

```
plt.hist(df['number of bedrooms'],bins=50)
plt.xlabel("No.of.Bedrooms")
plt.ylabel("Count")
```

```
Text(0, 0.5, 'Count')
```



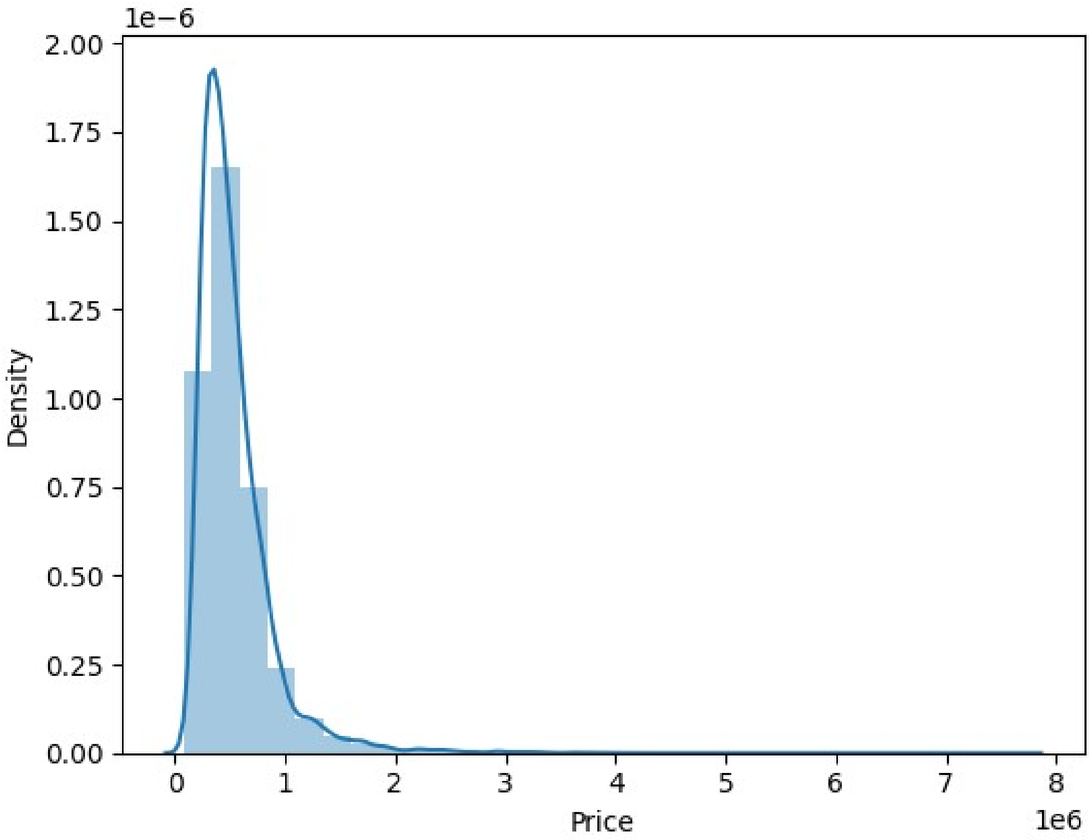
From the above graph we can clearly see that the peek count above 6000 is at range between 0 to 5. As the no.of.bedrooms increases after 5 the count values decreases tremoundously.

Distplot

```
sns.distplot(df['Price'],bins=30)
```

```
<ipython-input-38-9f4dfdc4bd19>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `histplot` (an axes-level function for
histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Price'],bins=30)
```

```
<Axes: xlabel='Price', ylabel='Density'>
```

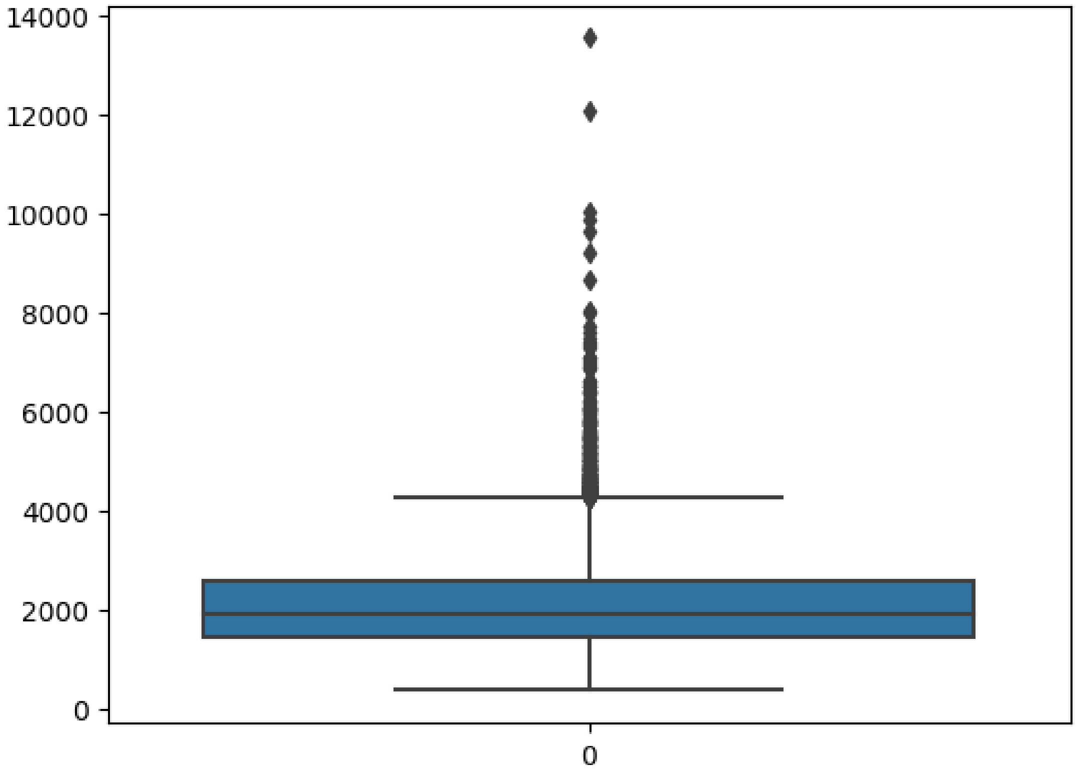


From the above distplot we came to know that the price distributes at peek between 0 and 1 related to density of the distribution.

Boxplot

```
sns.boxplot(df['living area'])
```

```
<Axes: >
```

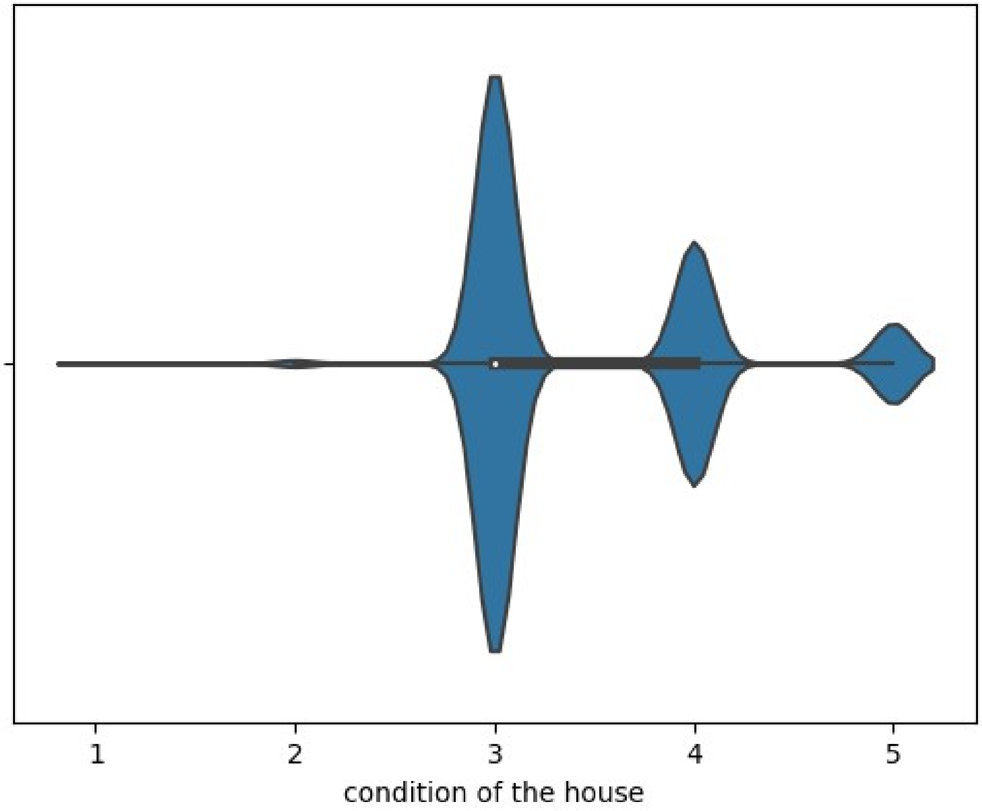


Boxplot is also used for detect the outlier in data set. It captures the summary of the data efficiently with a simple box and whiskers and allows us to compare easily across groups. Boxplot for living area and it contains many outliers and many outliers present in the features. The above one is a sample for detecting outliers.

Violinplot

```
sns.violinplot(x=df['condition of the house'])
```

```
<Axes: xlabel='condition of the house'>
```



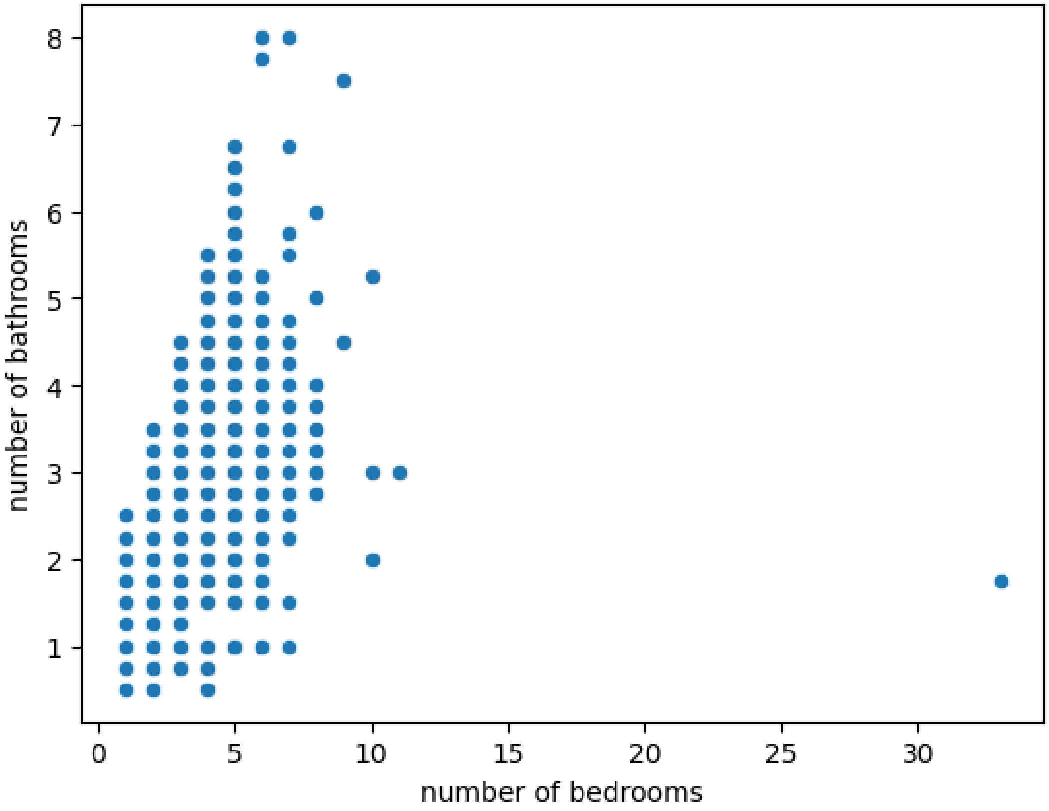
violinplot is used to vizualize the distribution numerical data and it shows the full distribution of data. The mean value of the variable "condition of the house" lies in 3 and the interquartile ranges between 3 to 4. The rest thin lines represents the rest distributions, except for the points that are determined to be the outliers. The higher probability lies in 3 and lowest probability lies above 5.

Bivariate Analysis

Scatterplot

```
sns.scatterplot(x=df['number of bedrooms'],y=df['number of
bathrooms'])
```

```
<Axes: xlabel='number of bedrooms', ylabel='number of bathrooms'>
```

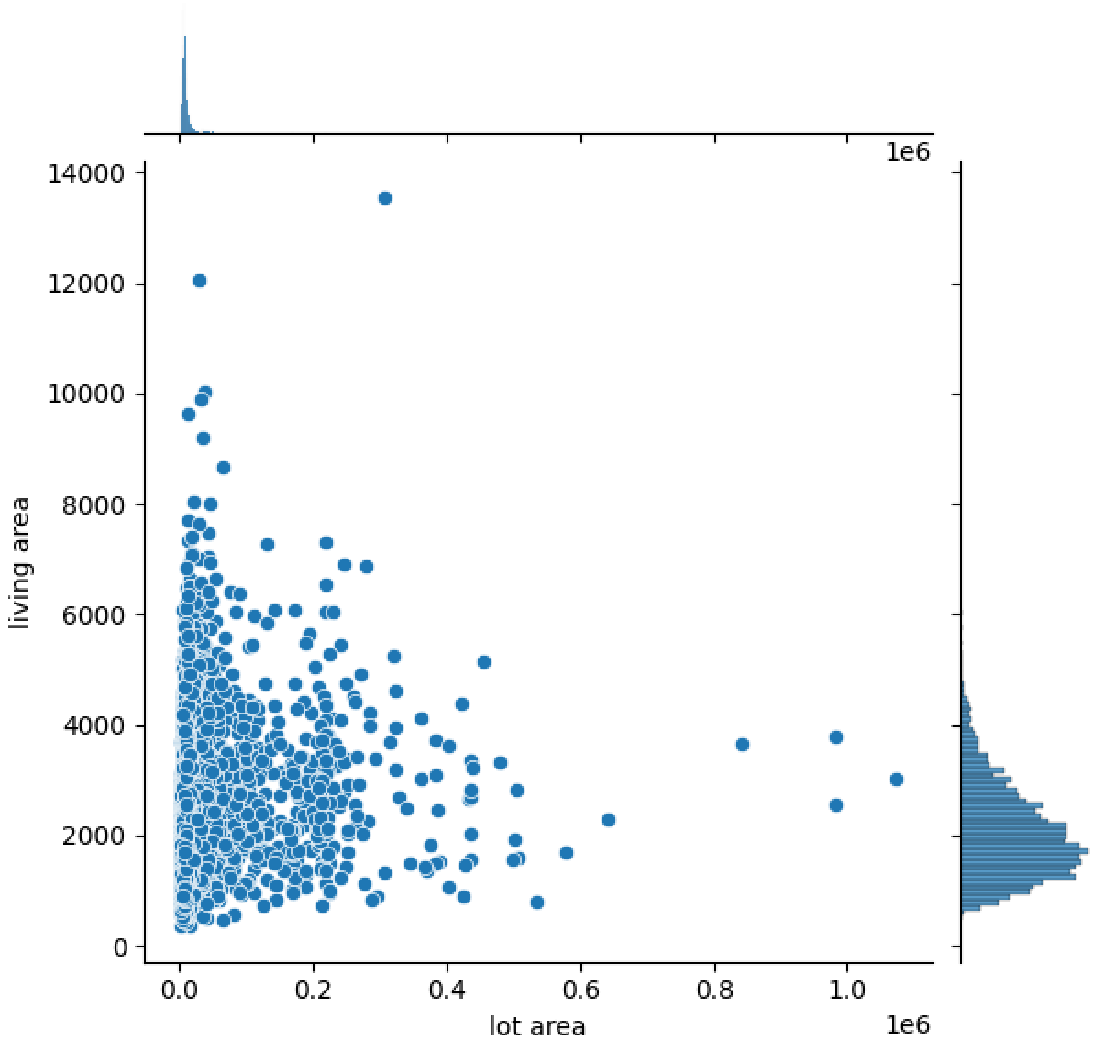


The scatterplot is used to show distributions between two variables. For no.of.bathrooms and no.of.bedrooms as far as the bathroom increases the bedroom number increases. And there are some outliers present in them.

Jointplot

```
sns.jointplot(data = df,x = 'lot area',y = 'living area')
```

```
<seaborn.axisgrid.JointGrid at 0x7f2b5c520a00>
```

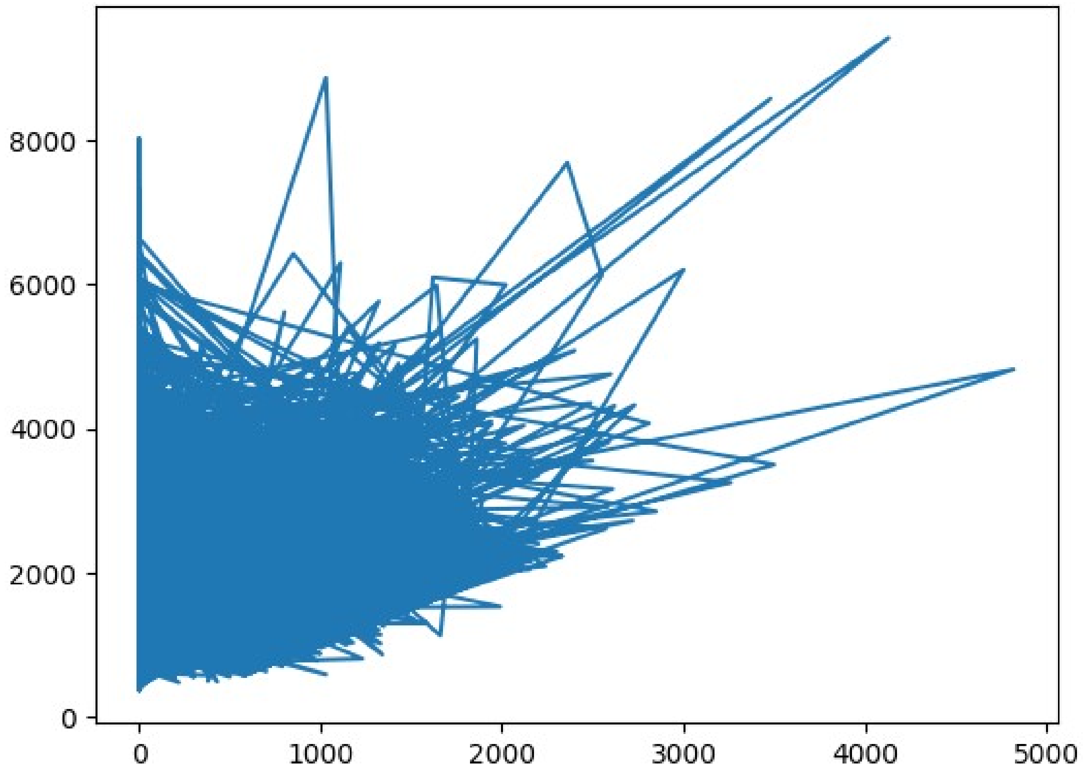


The relation between living area vs lot area and univariate of these has been shown. As far as the living area increases the lot area increases slighter and present many outliers between them. Univariate distribution of lot area remains same with slight increase in area but for living area the peak value is achieved at 2000 by gradual increase in it and then decreases until at a range of 5000.

Line plot

```
plt.plot(df['Area of the basement'],df['Area of the house(excluding
basement)'])
```

```
[<matplotlib.lines.Line2D at 0x7f2b5c61f4f0>]
```



Multivariate Analysis

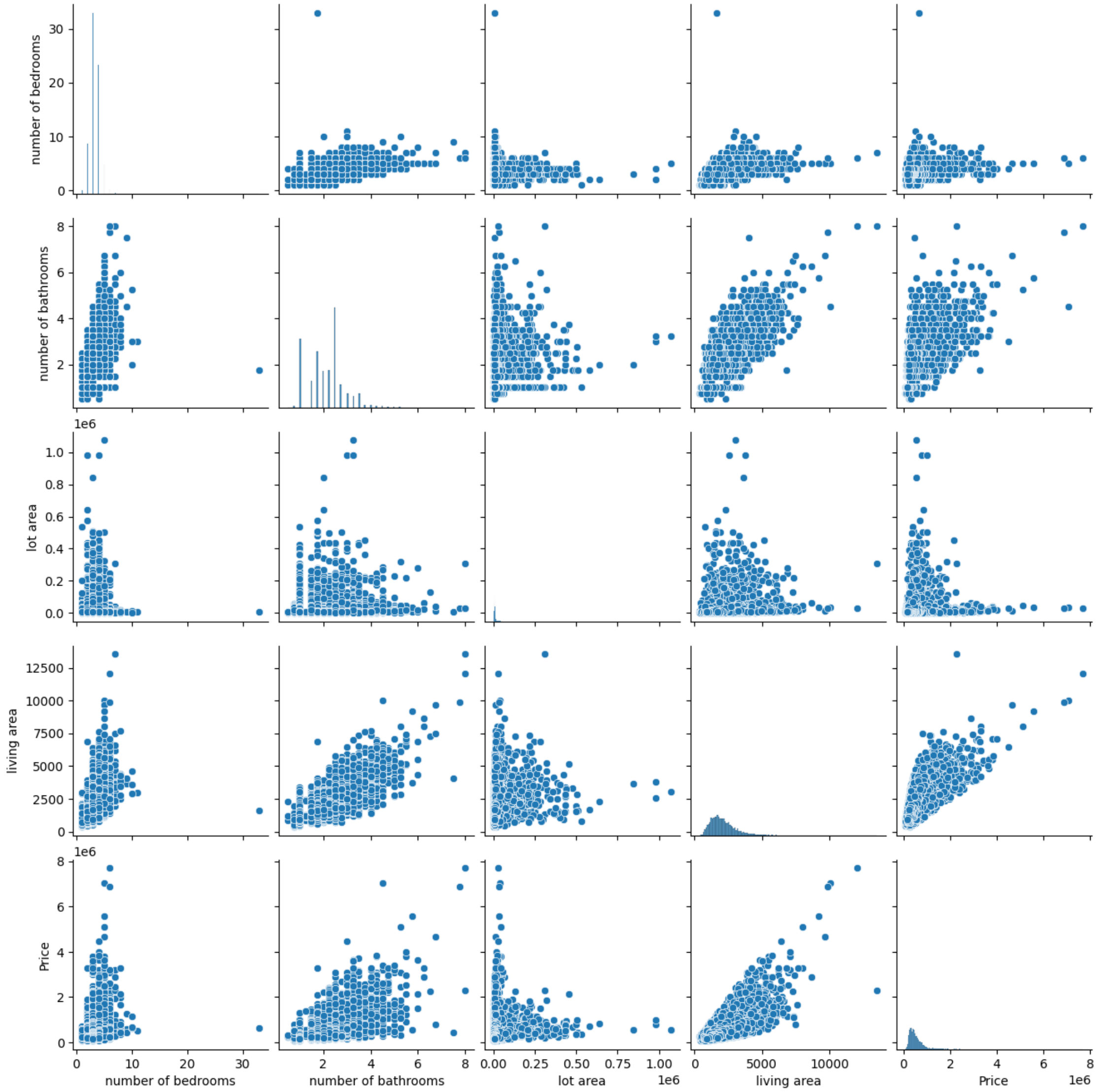
Pairplot

```
X = df[['number of bedrooms','number of bathrooms','lot area','living
area','Price']]
X

       number of bedrooms  number of bathrooms  lot area  living area
Price
0                       5                 2.50      9050         3650
2380000
1                       4                 2.50      4000         2920
1400000
2                       5                 2.75      9480         2910
1200000
3                       4                 2.50     42998         3310
838000
4                       3                 2.00      4500         2710
805000
...                   ...                  ...       ...          ...
...
14615                   2                 1.50     20000         1556
221700
14616                   3                 2.00      7000         1680
```

```
219200
14617                       2                 1.00      6120          1070
209000
14618                       4                 1.00      6621          1030
205000
14619                       3                 1.00      4770           900
146000

[14620 rows x 5 columns]
```

```
sns.pairplot(X,dropna=True)
```

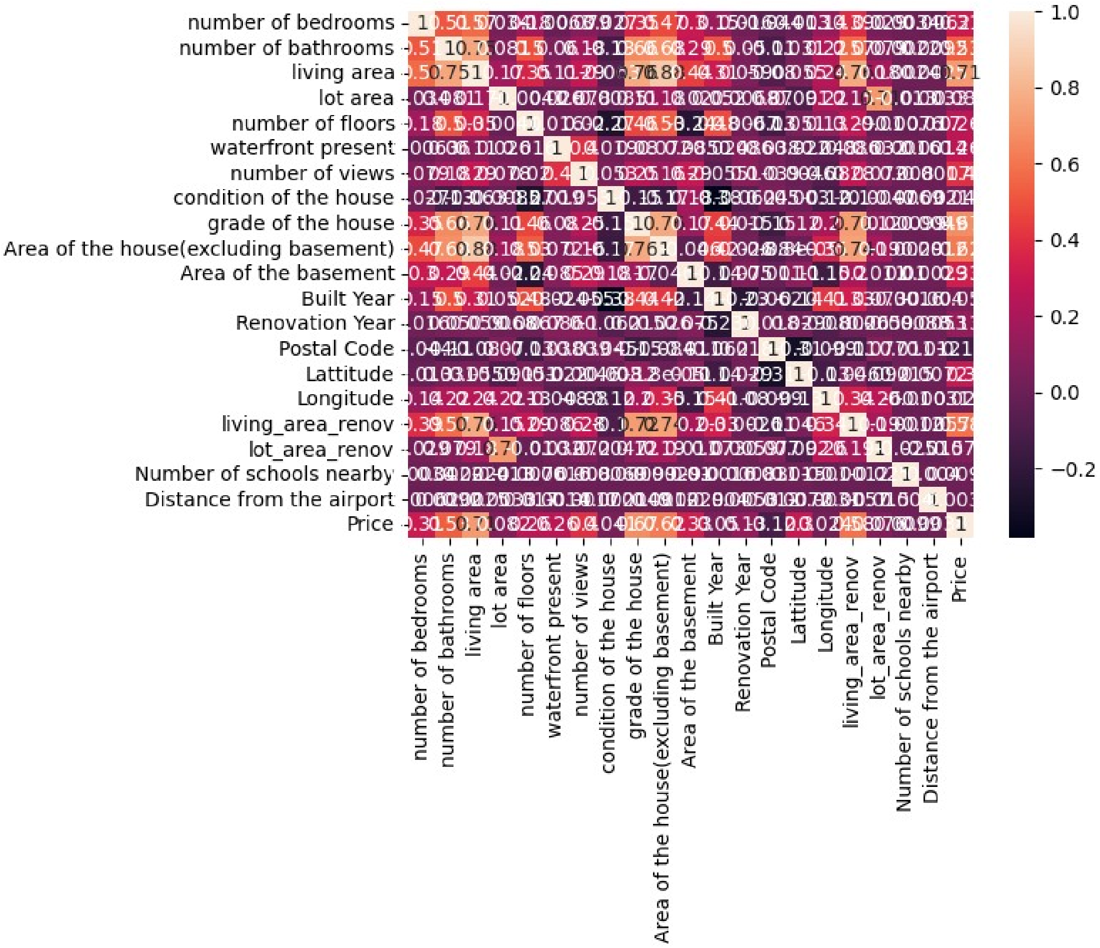
```
<seaborn.axisgrid.PairGrid at 0x7f2b560e3280>
```



From pairplot we can clearly see that some variable are linear to some variable and logistic to some variables. Most of the variables are linear to other variables. But in all variables outliers present in it.

```
df.drop(columns=['id','Date'],inplace=True)
sns.heatmap(df.corr(),annot=True)
```

```
<Axes: >
```


number of bedrooms
number of bathrooms
living area
lot area
number of floors
waterfront present
number of views
condition of the house
grade of the house
Area of the house(excluding basement)
Area of the basement
Built Year
Renovation Year
Postal Code
Lattitude
Longitude
living_area_renov
lot_area_renov
Number of schools nearby
Distance from the airport
Price