```
In [1]: import numpy as np
   import pandas as pd
   import matplotlib.pyplot as plt
   import seaborn as sns
```

## **Load Dataset**

In [23]: df = pd.read\_csv('House Price India.csv')
 df.head()

#### Out[23]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	COI
(	6762810145	42491	5	2.50	3650	9050	2.0	0	4	
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	

5 rows × 23 columns

localhost:8888/notebooks/AI Assignment 1.ipynb

```
In [3]: df.info()
```

<class 'pandas.core.frame.DataFrame'>

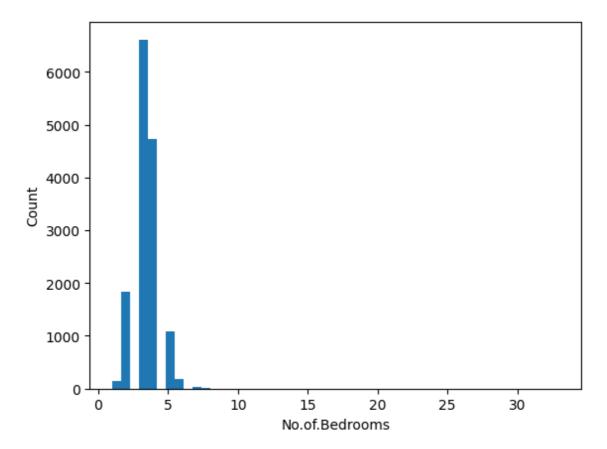
```
RangeIndex: 14620 entries, 0 to 14619
Data columns (total 23 columns):
     Column
                                           Non-Null Count Dtype
     -----
 0
     id
                                           14620 non-null int64
     Date
 1
                                           14620 non-null int64
     number of bedrooms
                                           14620 non-null int64
 2
 3
     number of bathrooms
                                           14620 non-null float64
    living area
                                           14620 non-null int64
 4
 5
    lot area
                                           14620 non-null int64
     number of floors
                                           14620 non-null float64
 6
 7
    waterfront present
                                           14620 non-null int64
    number of views
                                           14620 non-null int64
     condition of the house
 9
                                           14620 non-null int64
                                           14620 non-null int64
    grade of the house
 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
    Renovation Year
                                           14620 non-null int64
 15 Postal Code
                                           14620 non-null int64
 16 Lattitude
                                           14620 non-null float64
                                           14620 non-null float64
 17 Longitude
 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
```

## **Univariate Analysis**

Histogram

```
In [10]: plt.hist(df['number of bedrooms'],bins=50)
    plt.xlabel("No.of.Bedrooms")
    plt.ylabel("Count")
```

Out[10]: 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**

s).

In [11]: sns.distplot(df['Price'],bins=30)

C:\Users\ELCOT\AppData\Local\Temp\ipykernel\_70496\507312228.py:1: UserWarn
ing:

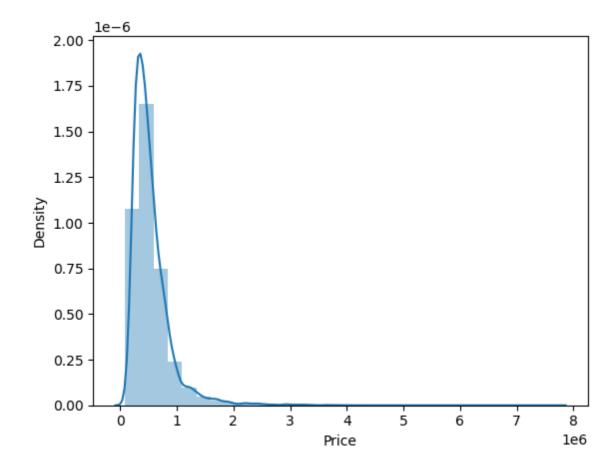
`distplot` is a deprecated function and will be removed in seaborn v0.14. 0.

Please adapt your code to use either `displot` (a figure-level function wi th similar flexibility) or `histplot` (an axes-level function for histogram

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

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

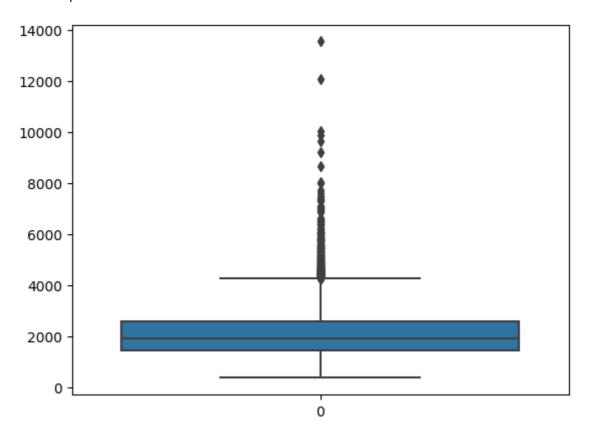
Out[11]: <AxesSubplot: xlabel='Price', ylabel='Density'>



## **Boxplot**

```
In [12]: sns.boxplot(df['living area'])
```

Out[12]: <AxesSubplot: >

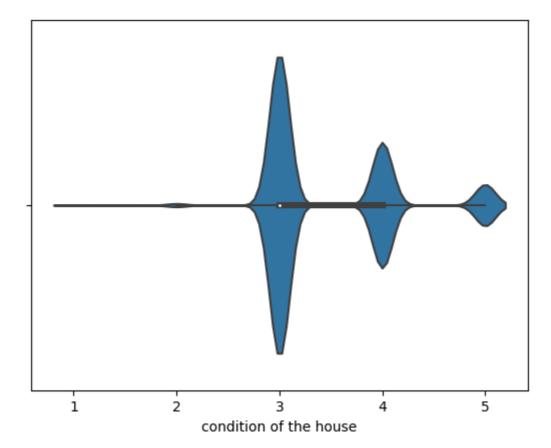


Boxplot is also used for detect the outlier in data set. It captures the summary of the data efficiently with a simple box and whiskers and allows us to compare easily across groups. Boxplot for living area and it contains many outliers and many outliers present in the features. The above one is a sample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

### **Violinplot**

```
In [13]: sns.violinplot(x=df['condition of the house'])
```

Out[13]: <AxesSubplot: xlabel='condition of the house'>



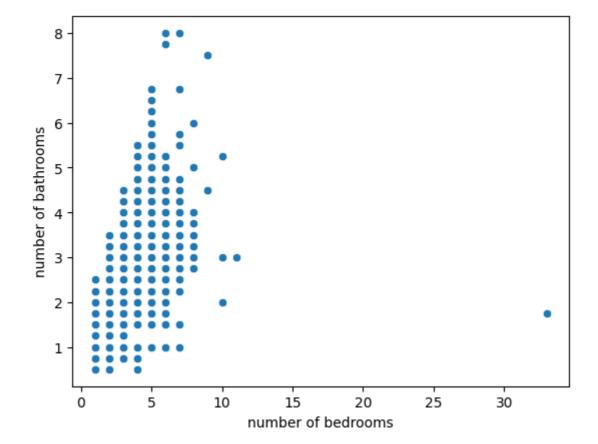
violinplot is used to vizualize the distribution numerical data and it shows the full distribution of data. The mean value of the variable "condition of the house" lies in 3 and the interquartile ranges between 3 to 4. The rest thin lines represents the rest distributions, except for the points that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

# **Bivariate Analysis**

### **Scatterplot**

In [14]: | sns.scatterplot(x=df['number of bedrooms'],y=df['number of bathrooms'])

Out[14]: <AxesSubplot: xlabel='number of bedrooms', ylabel='number of bathrooms'>

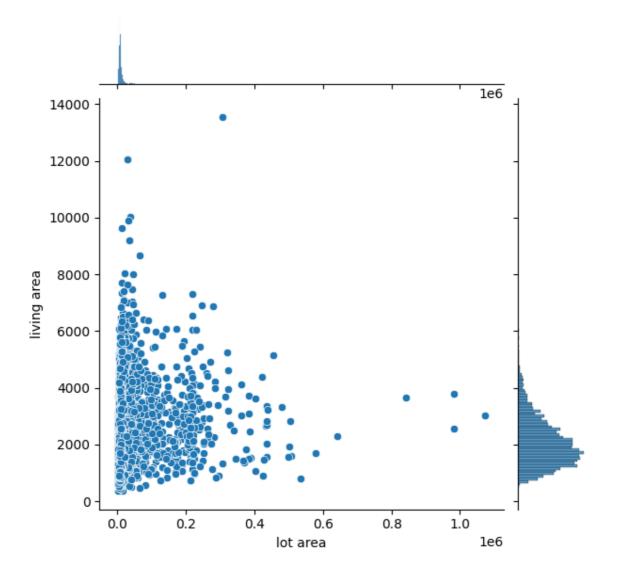


he scatterplot is used to show distributions between two variables. For no.of.bathrooms and no.of.bedrooms as far as the bathroom increases the bedroom number increases. And there are some outliers present in them.

### **Jointplot**

```
In [15]: sns.jointplot(data = df,x = 'lot area',y = 'living area')
```

Out[15]: <seaborn.axisgrid.JointGrid at 0x21867173ca0>

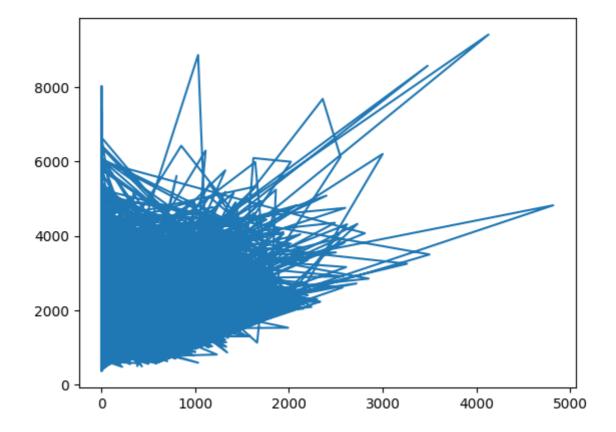


The relation between living area vs lot area and univariate of these has been shown. As far as the living area increases the lot area increases slighter and present many outliers between them. Univariate distribution of lot area remains same with slight increase in area but for living area the peak value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

## Line plot

In [16]: plt.plot(df['Area of the basement'],df['Area of the house(excluding basement

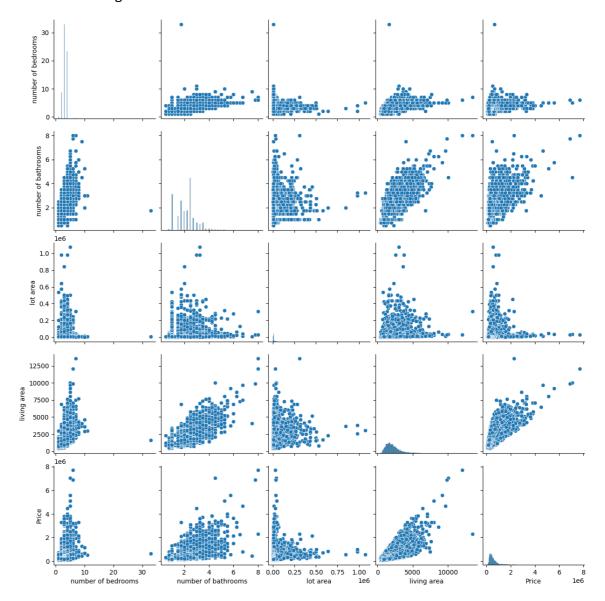
Out[16]: [<matplotlib.lines.Line2D at 0x218699cf2e0>]



# **Multivariate Analysis**

## Pair plot

Out[17]: <seaborn.axisgrid.PairGrid at 0x218672039a0>



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.

#### **Heat map**

```
In [18]:
                        df.drop(columns=['id','Date'],inplace=True)
                         sns.heatmap(df.corr(),annot=True)
Out[18]: <AxesSubplot: >
                                                                                                                                                                                                                   - 1.0
                                                            number of bedrooms - 10.50.5070 841 80 0609.9 207335 470. B. D590 694.40314. 8990290 304063
                                                          number of bathrooms -1.5 10.70.081.50.06.18.1366.68.290.50.96.0.13120.570790.220
living area -1.50.7510.107.35.10.2900.7680.48.310.590805524.70.0.80.24
                                                                                                                                                                                                                   - 0.8
                                                                                   lot area - 03.4811 0.00.402.607.60850.18.0205006.807.09.20.19.0.0130308
                                                                number of floors -), 1 80, 5-6 500 10, 0 15692, 2746, 5-6, 2742, 6 6 6 70 30 50 113, 2 6 00 10 2750 272 waterfront present -0 05666, 10 10 26 11 0, 46 0 15 00007, 126 50 2 486, 636 22 20 4868, 60 3200, 126 12 number of views - 0 75918, 2250 750 12, 41 0 53325, 166, 2250 5551, 40 3964, 60 8 2507, 20 80 00 22
                                                                                                                                                                                                                   - 0.6
                                                          condition of the house - 0.207-0.3060908.0.700.95 1 ). 105. 1071-8,-0.8006.04.50-0.31-201900.400.69.2014
                           grade of the house -1.35.60.70.10.46.08.25.1 10.70.10.09046151512.2.70010309046
Area of the house(excluding basement) -1.40.60.80.18.5307216.17761 .0960202.88846030.70.0900202
                                                                                                                                                                                                                     0.4
                                                           Area of the basement -0.B.29.44.02.24.8529.18.10704 1 ).D490500110.15.20D10.D023
                                                                    Built Year -0.1 50.5.300 5243602.4 553844442.1 1 1 0.20306216440.53307.601660.40 Renovation Year -0.0 56050530.6860736.0 .005021.502.6 7.52 2 .001.8 2030.801606690333.3 Postal Code -.0 4410.0 8007.0 303.803.9 45-0 5038.41.005021 1 0.2010-901.0 7.001.0 1021
                                                                                                                                                                                                                     0.2
                                                                                 Lattitude -.001.03.06.509.06.30200460-8.28-e90151.040-293 1 0.0.394.609.00150 702
                                                                                                                                                                                                                     0.0
                                                                               Longitude -). 14. 20. 24. 20. 1380 4893. 1 0. 2). 36. 1540 -080 991 1 0. 34. 26 00 0 0 302
area_renov --). 39. 5 0. 70. 16. 290 862 80. 0<mark>. 70. 70. 2</mark>-8 30 902 6. 0 4663 4 1 0-0 900 12657
                                                                  living area renov -
                                                                       lot area renov - 02.9 7091 8 70 . 0.103.207/2/004 702 . D9001.00708005907.70 9.226.1 9 1 . 92.501.67
                                                                                                                                                                                                                       -0.2
                                                   Number of schools nearby -00.B40292910006016080099929300006060203260.D0102. 1 .0000
                                                    Price -0.30.50). 70.0 8022 65.2 60.04.0 40
                                                                                                             living area –
lot area –
number of floors –
                                                                                                        number of bathrooms
                                                                                                                           waterfront present number of views
                                                                                                                                         grade of the house grade of the house. Area of the house excluding basement)
                                                                                                                                                                      Lattitude
Longitude
                                                                                                                                                                 Postal Code
                                                                                                   number of bedrooms
                                                                                                                                     condition of the house
                                                                                                                                                   Area of the basement
                                                                                                                                                       Built Year
                                                                                                                                                             Renovation Year
                                                                                                                                                                                         Number of schools nearby
                                                                                                                                                                                 living_area_renov
                                                                                                                                                                                     lot area renov
                                                                                                                                                                                              Distance from the airport
```

## Statistical analysis

In [5]: df.describe()

Out[5]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area
count	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.000000	1.462000e+04
mean	6.762821e+09	42604.538646	3.379343	2.129583	2098.262996	1.509328e+04
std	6.237575e+03	67.347991	0.938719	0.769934	928.275721	3.791962e+04
min	6.762810e+09	42491.000000	1.000000	0.500000	370.000000	5.200000e+02
25%	6.762815e+09	42546.000000	3.000000	1.750000	1440.000000	5.010750e+03
50%	6.762821e+09	42600.000000	3.000000	2.250000	1930.000000	7.620000e+03
75%	6.762826e+09	42662.000000	4.000000	2.500000	2570.000000	1.080000e+04
max	6.762832e+09	42734.000000	33.000000	8.000000	13540.000000	1.074218e+06
8 rows × 23 columns						
4						<b>&gt;</b>

Descriptive statistics to summerize the data by computing mean, median, mode, standard derivation and likewise other informations of data.

# Handling missing values

In [21]:	df.isnull().any()	
Out[21]:	number of bedrooms	False
	number of bathrooms	False
	living area	False
	lot area	False
	number of floors	False
	waterfront present	False
	number of views	False
	condition of the house	False
	grade of the house	False
	Area of the house(excluding basement)	False
	Area of the basement	False
	Built Year	False
	Renovation Year	False
	Postal Code	False
	Lattitude	False
	Longitude	False
	living_area_renov	False
	lot_area_renov	False
	Number of schools nearby	False
	Distance from the airport	False
	Price	False
	dtype: bool	

```
In [22]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14620 entries, 0 to 14619
Data columns (total 21 columns):
```

#	Column	Non-Null Count	Dtype			
0	number of bedrooms	14620 non-null				
1	number of bathrooms	14620 non-null	float64			
2	living area	14620 non-null	int64			
3	lot area	14620 non-null	int64			
4	number of floors	14620 non-null	float64			
5	waterfront present	14620 non-null	int64			
6	number of views	14620 non-null	int64			
7	condition of the house	14620 non-null	int64			
8	grade of the house	14620 non-null	int64			
9	Area of the house(excluding basement)	14620 non-null	int64			
10	Area of the basement	14620 non-null	int64			
11	Built Year	14620 non-null	int64			
12	Renovation Year	14620 non-null	int64			
13	Postal Code	14620 non-null	int64			
14	Lattitude	14620 non-null	float64			
15	Longitude	14620 non-null	float64			
16	living_area_renov	14620 non-null	int64			
17	lot_area_renov	14620 non-null	int64			
18	Number of schools nearby	14620 non-null	int64			
19	Distance from the airport	14620 non-null	int64			
20	Price	14620 non-null	int64			
dtypes: float64(4), int64(17)						
memory usage: 2.3 MB						

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.

```
In [ ]:
```