

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

df = pd.read_csv("/content/House Price India.csv")
df
```

df

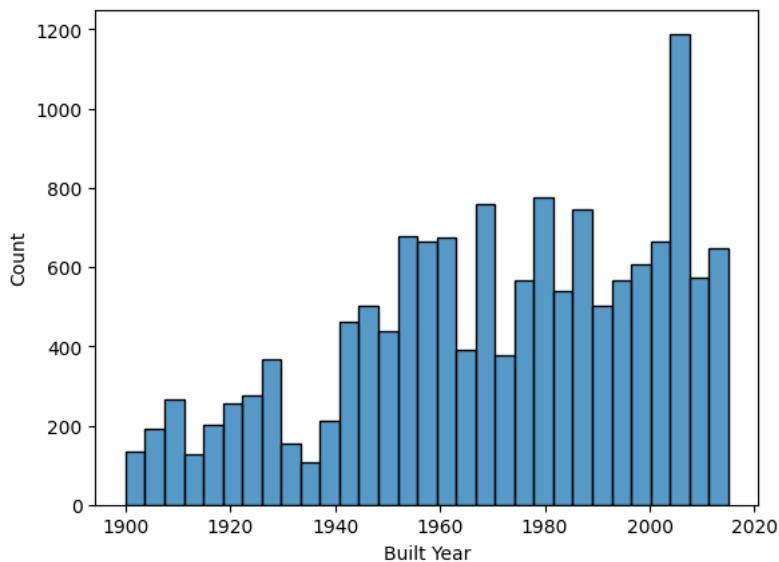
	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	num vi
0	6762810145	42491	5	2.50	3650	9050	2.0	0	
1	6762810635	42491	4	2.50	2920	4000	1.5	0	
2	6762810998	42491	5	2.75	2910	9480	1.5	0	
3	6762812605	42491	4	2.50	3310	42998	2.0	0	
4	6762812919	42491	3	2.00	2710	4500	1.5	0	
...
14615	6762830250	42734	2	1.50	1556	20000	1.0	0	
14616	6762830339	42734	3	2.00	1680	7000	1.5	0	
14617	6762830618	42734	2	1.00	1070	6120	1.0	0	
14618	6762830709	42734	4	1.00	1030	6621	1.0	0	
14619	6762831463	42734	3	1.00	900	4770	1.0	0	

14620 rows × 23 columns

◀ ▶

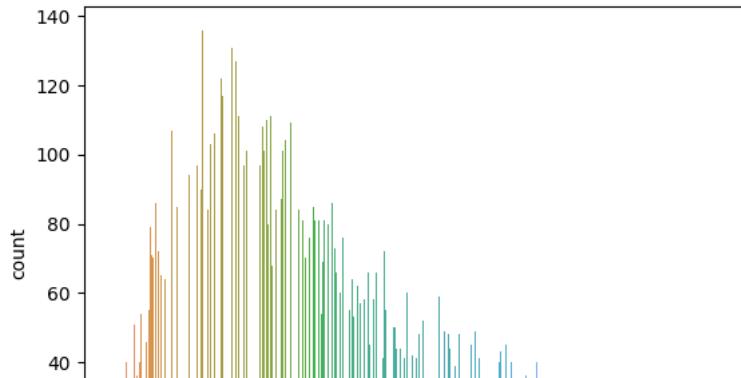
```
# Univariate analysis
# Using histogram plot function
sb.histplot(df['Built Year'])
```

<Axes: xlabel='Built Year', ylabel='Count'>



```
## Bar chart
sb.countplot(x=df['living_area_renov'])
```

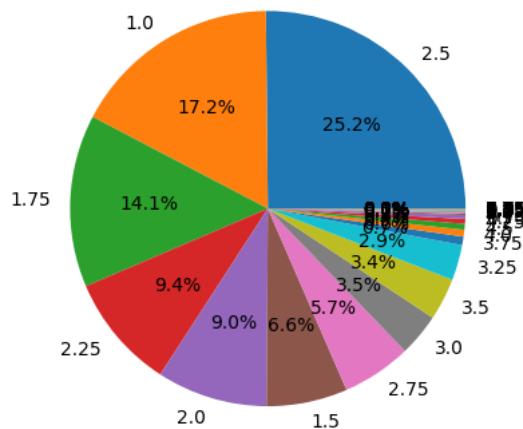
```
<Axes: xlabel='living_area_renov', ylabel='count'>
```



```
## pie chart
import matplotlib.pyplot as plt

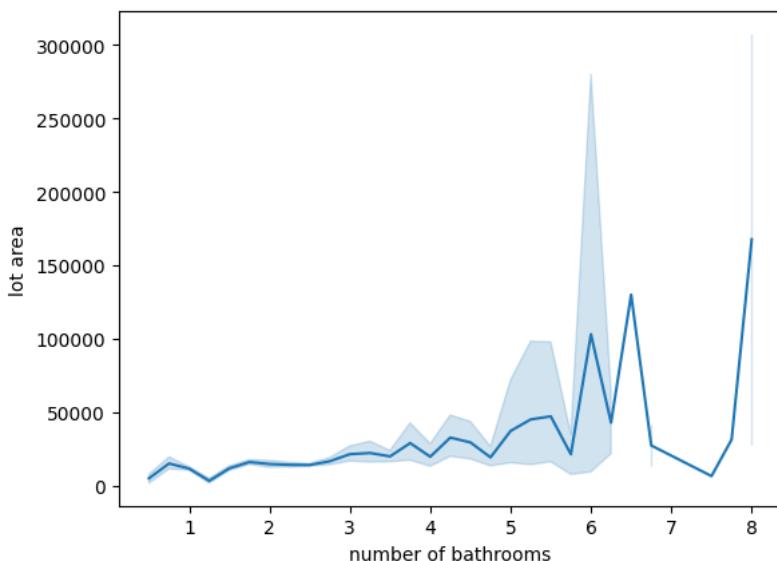
x = df['number of bathrooms'].value_counts()

plt.pie(x.values, labels=x.index, autopct='%1.1f%%')
plt.show()
```



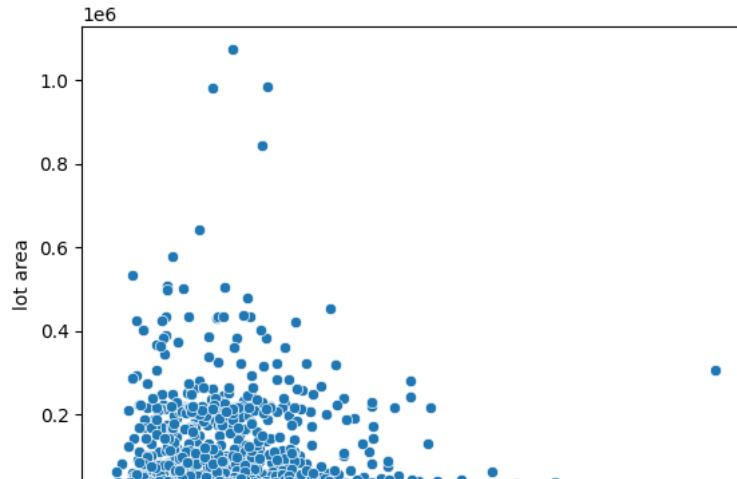
```
# Bivariate analysis
sb.lineplot(x = df['number of bathrooms'], y=df['lot area'])
```

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



```
sb.scatterplot(x=df['living area'], y=df['lot area'])
```

<Axes: xlabel='living area', ylabel='lot area'>

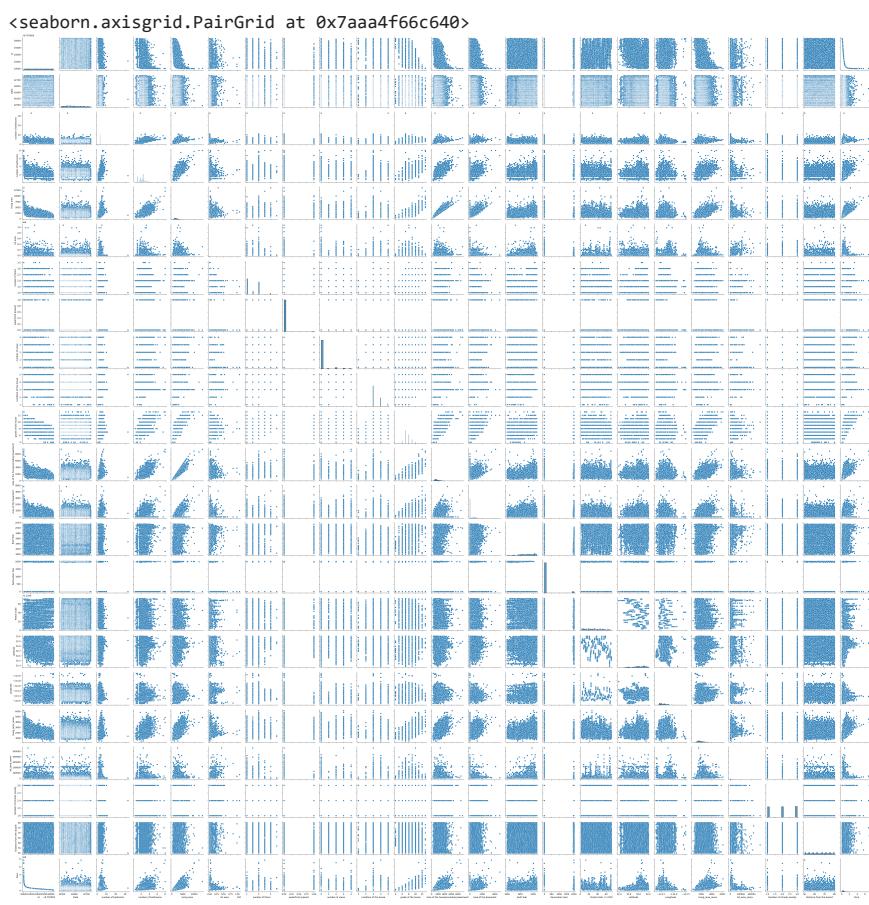


sb.heatmap(df.corr(), annot=True)

<Axes: >



sb.pairplot(df)



```
sb.heatmap(df.corr(), annot=True)
```

<Axes: >



Descriptive statistics

Number of schools nearby: 0.0000000000000000

df.describe()

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

8 rows × 23 columns

Handling Missing Values

df.isnull().any()

id	False
Date	False
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
Latitude	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	