Assignment 2

Name: K Naga Sai Krishna

Perform the Below Tasks to complete the assignment:-

Tasks:-

1. Download the dataset: Dataset
2. Load the dataset.
3. Perform the Below Visualizations.
    - Univariate Analysis
    - Bi - Variate Analysis
    - Multivariate Analysis
4. Perform descriptive statistics on the dataset.
5. Handle the Missing values.

1. , 2.

```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
import seaborn as sns
```

```
df2 = pd.read_csv("D:\\SmartBridge\\House Price India.csv\\House Price India.csv")
```

```
[ ]  df2.shape
```

```
(14620, 23)
```

```
[ ]  df2.columns
```

```
Index(['id', 'Date', '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'],
      dtype='object')
```

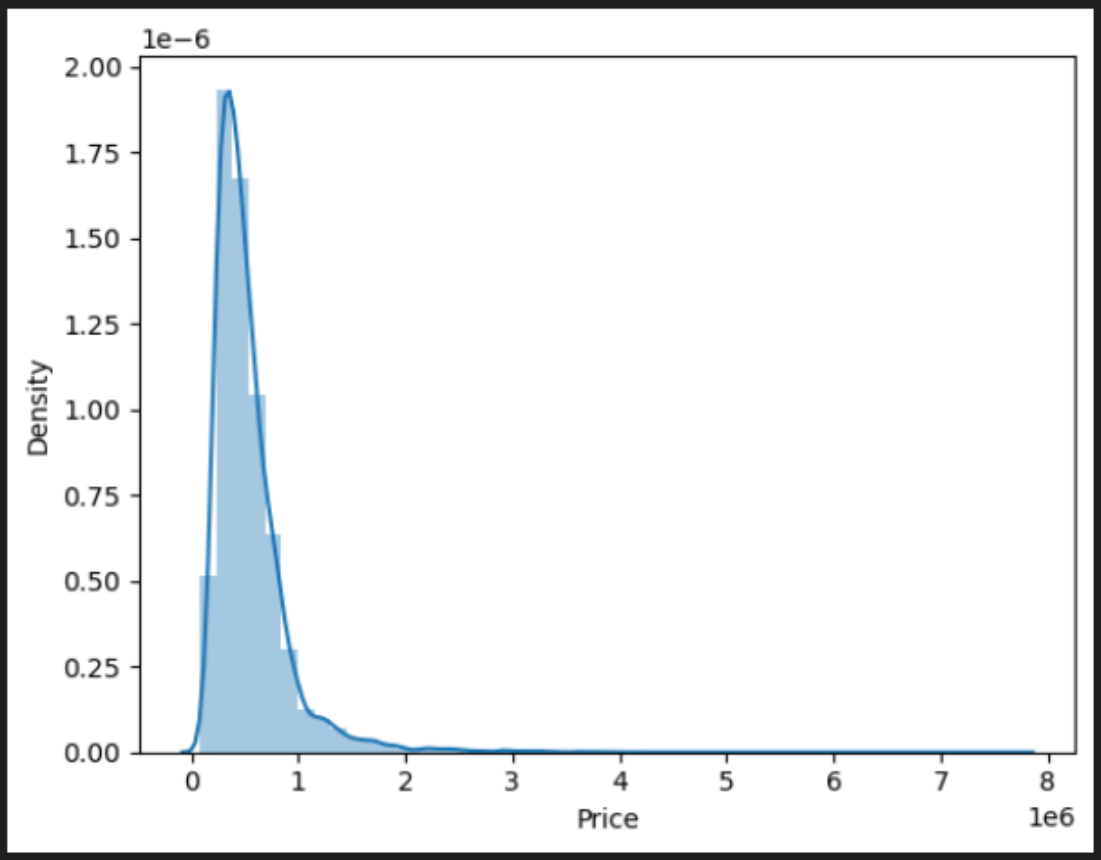
3.

Univariate analysis

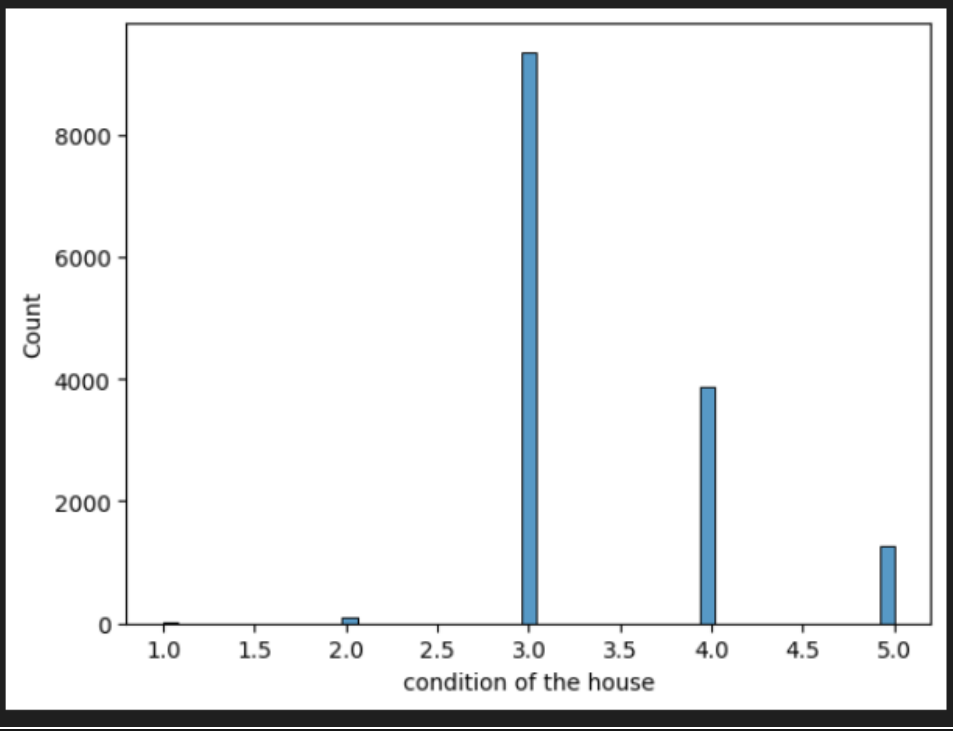
```
sns.distplot(df2['Price'])
plt.show()
```

✓ 0.7s

```
C:\Users\K N S Krishna\AppData\Local\Temp\ipykernel_21824\3388071935.py: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(df2['Price'])
```

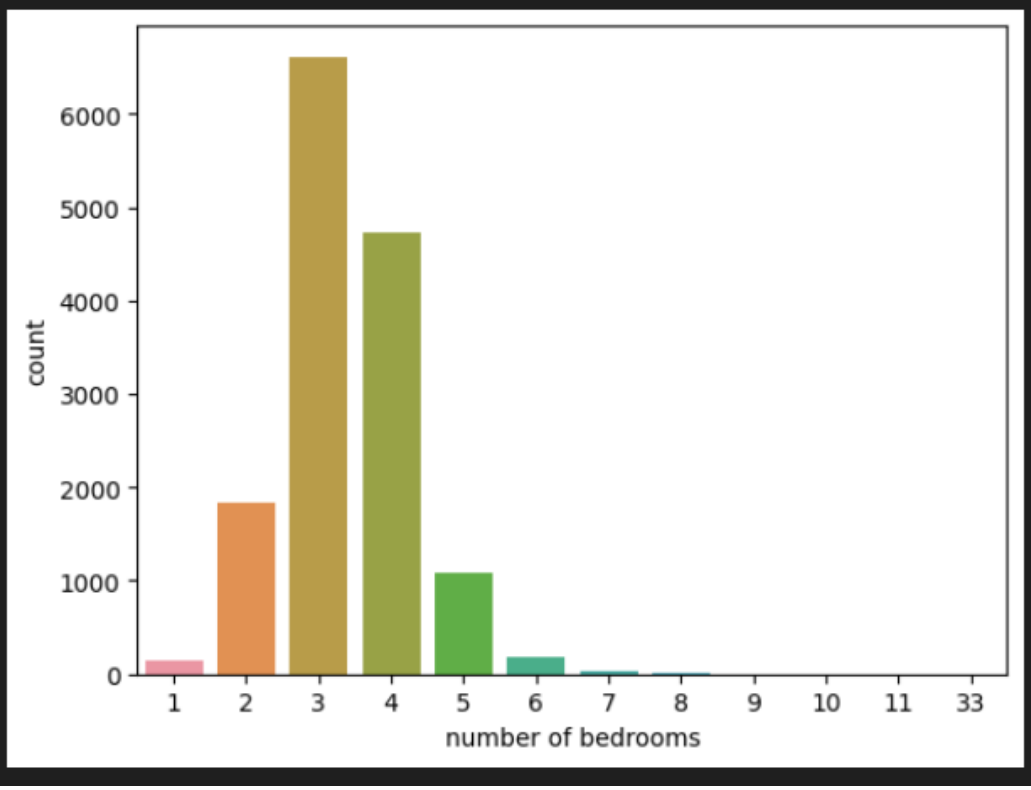


```
sns.histplot(df2['condition of the house'])
plt.show()
#This shows most of the houses are given 3 out of 5 rating for the condition of the house
```



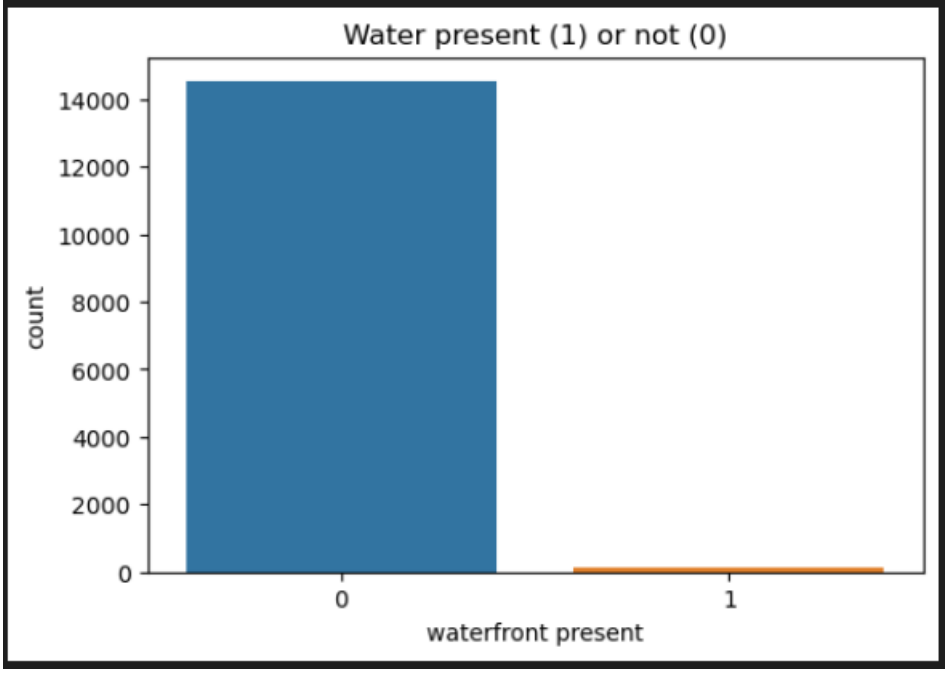

```
sns.countplot(x=df2['number of bedrooms'])
#This shows the distribution of no of bedrooms in houses
```

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

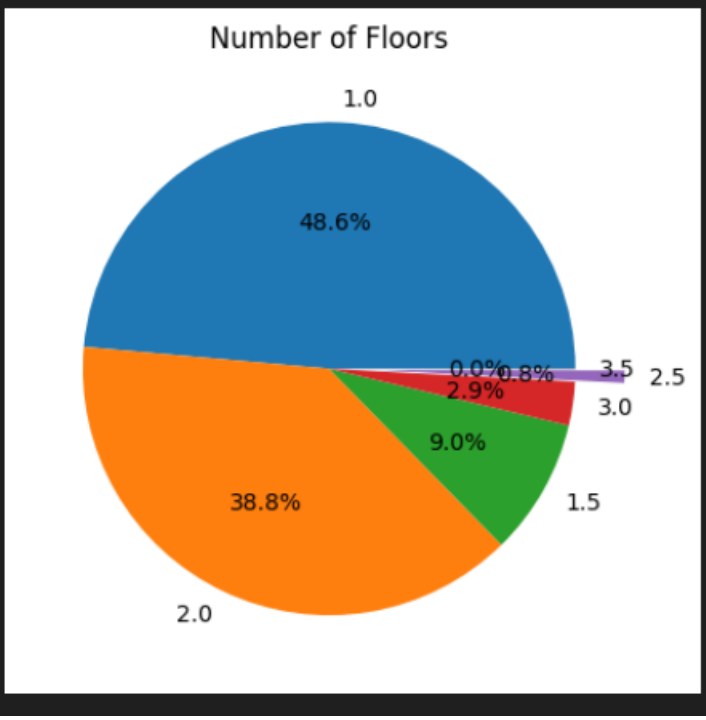


```
plt.figure(figsize=(6,4))
plt.title('Water present (1) or not (0)')
sns.countplot(data=df2, x='waterfront present')
plt.show()
#This shows the count of waterfront
```

✓ 0.3s



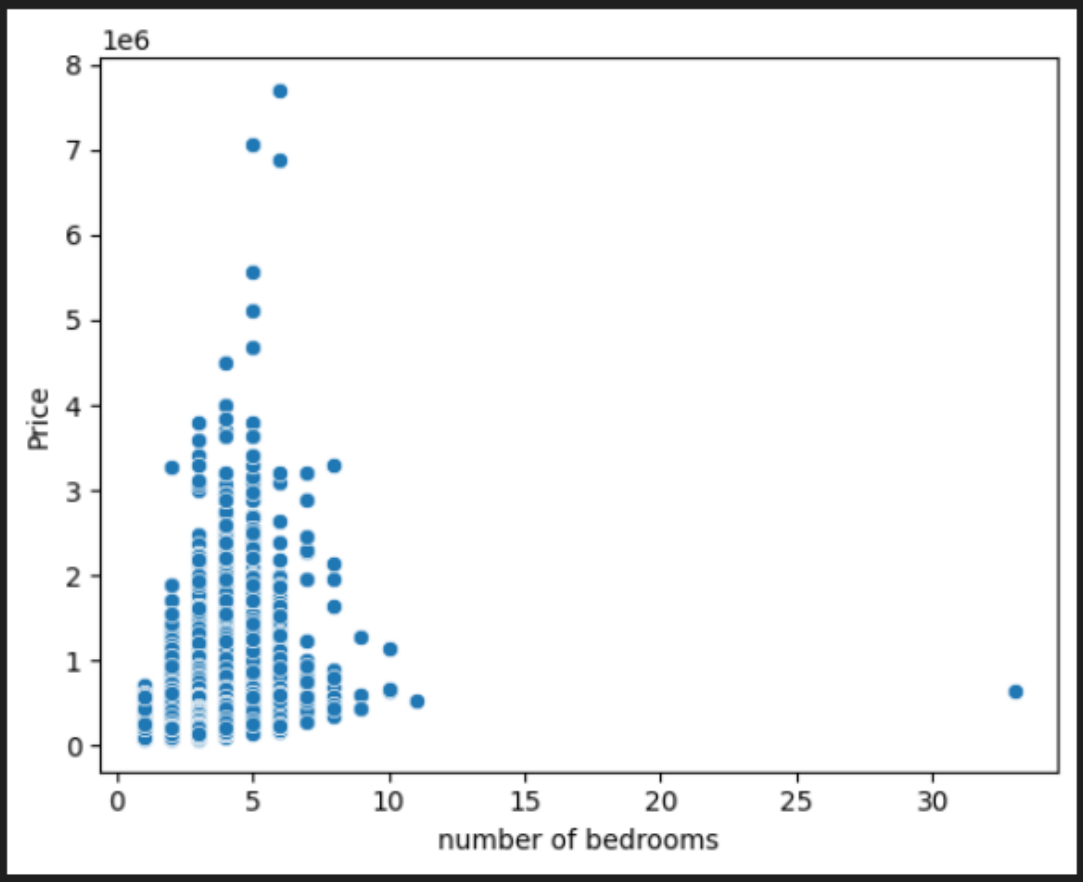

```
x = df2['number of floors'].value_counts()
plt.pie(x.values,[0,0,0,0,0.2,0],labels=x.index,autopct='%1.1f%%')
plt.title('Number of Floors')
plt.show()
#This shows the percentage of no of floors in buildings
```



bivariate analysis

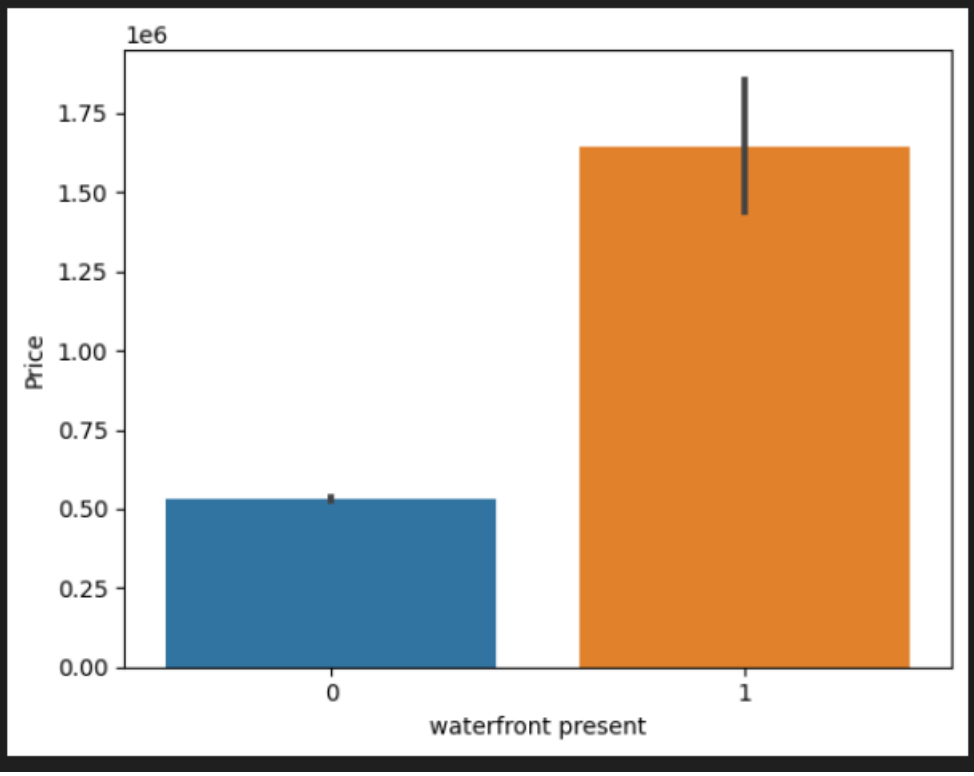
```
sns.scatterplot(x=df2['number of bedrooms'],
                y=df2['Price'])
#This shows a outlier in the plot
```

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

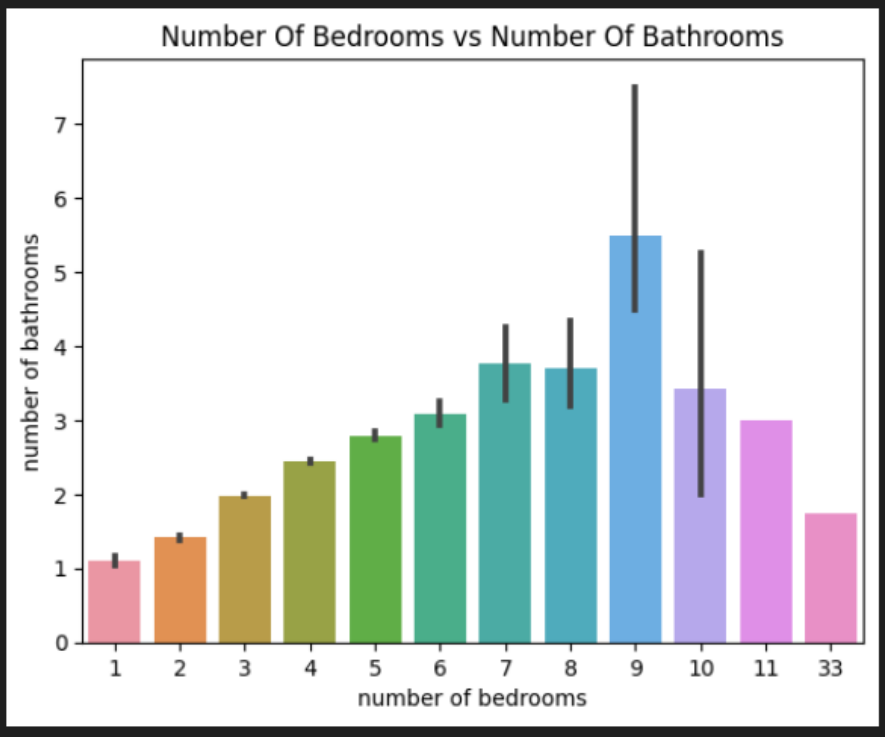


```
sns.barplot(x=df2['waterfront present'],y=df2['Price'])
```

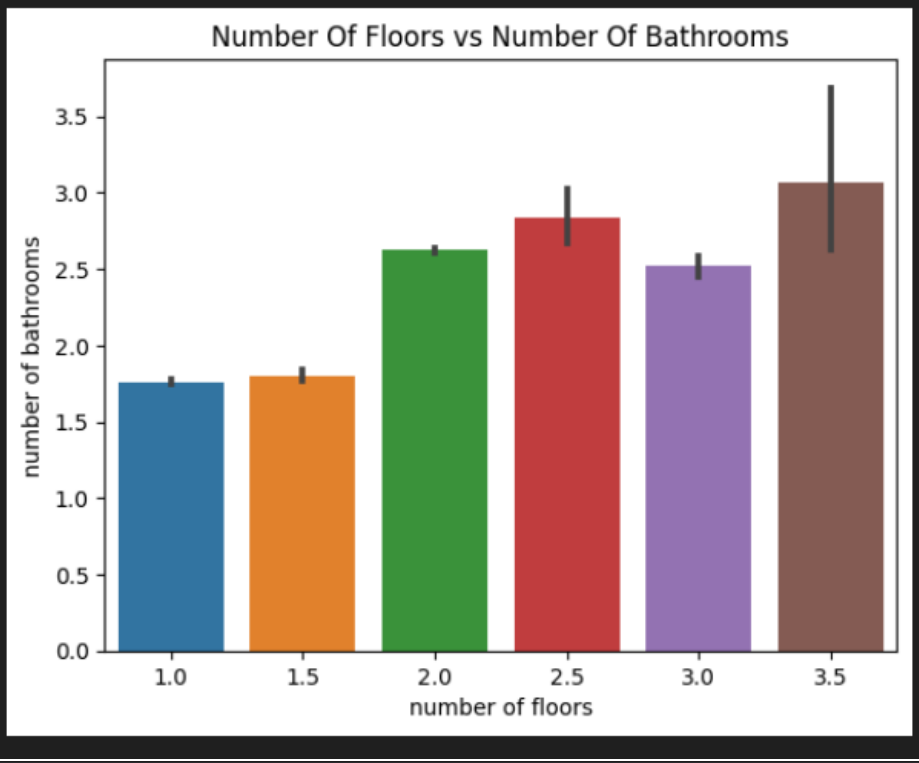
```
<Axes: xlabel='waterfront present', ylabel='Price'>
```



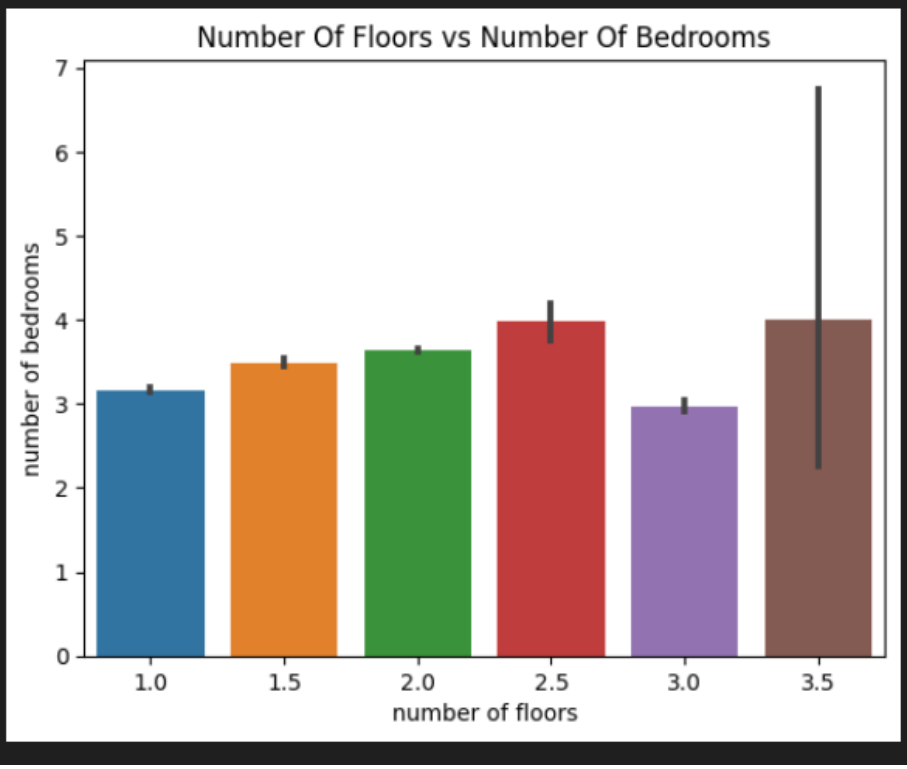

```
sns.barplot(data=df2, x='number of bedrooms', y='number of bathrooms')
plt.title('number of bedrooms'.title() + ' vs ' + 'number of bathrooms'.title())
plt.show()
```



```
sns.barplot(data=df2, x='number of floors', y='number of bathrooms')
plt.title('number of floors'.title() + ' vs ' + 'number of bathrooms'.title())
plt.show()
```




```
sns.barplot(data=df2, x='number of floors', y='number of bedrooms')
plt.title('number of floors'.title() + ' vs ' + 'number of bedrooms'.title())
plt.show()
```



Multivariate analysis

```
df2.corr()
```

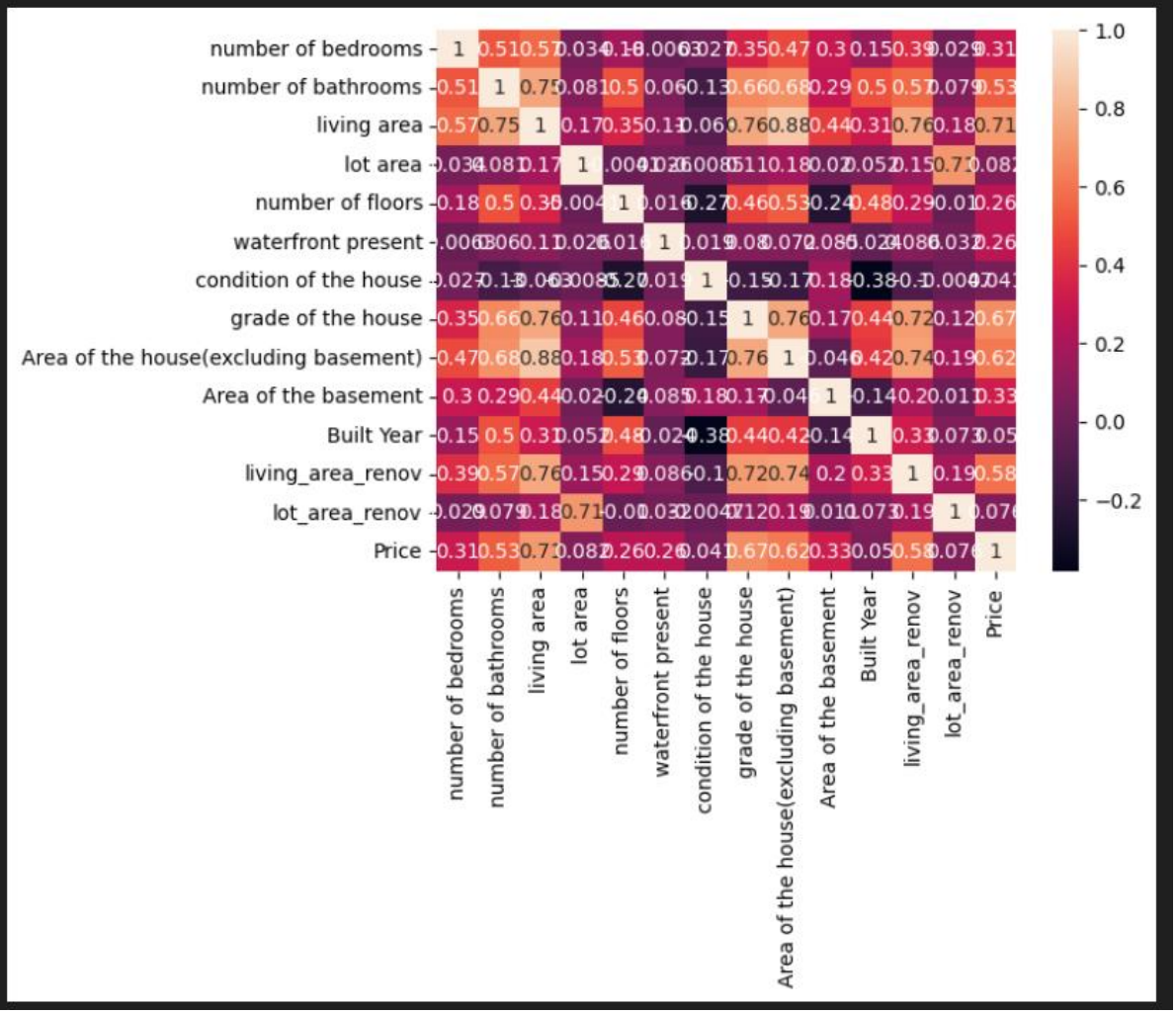
	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condition of the house	...	Built Year	Renovation Year	Postal Code	Lattitude	Longitude
id	1.000000	0.045966	-0.329034	-0.516909	-0.648127	-0.100269	-0.312305	-0.112937	-0.293004	-0.045061	...	-0.068645	-0.109155	0.294709	-0.479334	-0.070841
Date	0.045966	1.000000	-0.015663	-0.026485	-0.021958	0.004392	-0.010335	0.012006	-0.004782	-0.027402	...	-0.005869	-0.011636	0.018243	-0.023327	-0.018231
number of bedrooms	-0.329034	-0.015663	1.000000	0.509784	0.570526	0.034416	0.177294	-0.006257	0.078665	0.026597	...	0.152954	0.016132	-0.044156	-0.013163	0.135712
number of bathrooms	-0.516909	-0.026485	0.509784	1.000000	0.753517	0.080806	0.502924	0.060104	0.183789	-0.128232	...	0.498127	0.049669	-0.105546	0.031156	0.223904
living area	-0.648127	-0.021958	0.570526	0.753517	1.000000	0.174420	0.354743	0.105837	0.287728	-0.063358	...	0.309602	0.059400	-0.080303	0.054518	0.240208
lot area	-0.100269	0.004392	0.034416	0.080806	0.174420	1.000000	-0.004138	0.026282	0.078308	-0.008548	...	0.051615	0.006848	0.070131	-0.090983	0.221432
number of floors	-0.312305	-0.010335	0.177294	0.502924	0.354743	-0.004138	1.000000	0.016316	0.020153	-0.269928	...	0.481565	0.006705	-0.129788	0.050731	0.127550
waterfront present	-0.112937	0.012006	-0.006257	0.060104	0.105837	0.026282	0.016316	1.000000	0.400206	0.018644	...	-0.024226	0.085865	0.038318	-0.021795	-0.047791
number of views	-0.293004	-0.004782	0.078665	0.183789	0.287728	0.078308	0.020153	0.400206	1.000000	0.052533	...	-0.055357	0.102944	0.039268	-0.004555	-0.079706
condition of the house	-0.045061	-0.027402	0.026597	-0.128232	-0.063358	-0.008548	-0.269928	0.018644	0.052533	1.000000	...	-0.381718	-0.062126	0.045334	-0.002998	-0.121189
grade of the house	-0.673448	-0.033097	0.352945	0.663054	0.761835	0.110546	0.463082	0.079831	0.254532	-0.152530	...	0.440358	0.014501	-0.146342	0.115256	0.203754
Area of the house(excluding basement)	-0.565116	-0.015994	0.473599	0.684391	0.875793	0.183553	0.525643	0.071865	0.162672	-0.167695	...	0.419369	0.025727	-0.083730	-0.000088	0.345899
Area of the basement	-0.290806	-0.015711	0.300332	0.287190	0.441491	0.019755	-0.242976	0.085441	0.293062	0.180609	...	-0.138843	0.075104	-0.010542	0.112989	-0.145879
Built Year	-0.068645	-0.005869	0.152954	0.498127	0.309602	0.051615	0.481565	-0.024226	-0.055357	-0.381718	...	1.000000	-0.233683	-0.062349	-0.143153	0.414591
Renovation Year	-0.109155	-0.011636	0.016132	0.049669	0.059400	0.006848	0.006705	0.085865	0.102944	-0.062126	...	-0.233683	1.000000	0.018006	0.028908	-0.080050
Postal Code	0.294709	0.018243	-0.044156	-0.105546	-0.080303	0.070131	-0.129788	0.038318	0.039268	0.045334	...	-0.062349	0.018006	1.000000	-0.310172	-0.099003
Lattitude	-0.479334	-0.023327	-0.013163	0.031156	0.054518	-0.090983	0.050731	-0.021795	-0.004555	-0.002998	...	-0.143153	0.028908	-0.310172	1.000000	-0.131472

```
df3=df2
```

```
#dropping the unnessesary columns
df3.drop(['id','Date','Lattitude','Longitude','Postal Code',
'number of views','Renovation Year','Number of schools nearby',
'Distance from the airport'],axis=1,inplace=True)
```

```
sns.heatmap(df3.corr(), annot=True)
```

```
<Axes: >
```



4.

```
df2.iloc[:,2:].describe()
```

✓ 0.2s

	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)	...
count	14620.000000	14620.000000	14620.000000	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.000000	14620.000000	14620.000000	...
mean	3.379343	2.129583	2098.262996	1.509328e+04	1.502360	0.007661	0.233105	3.430506	7.682421	1801.783926	...
std	0.938719	0.769934	928.275721	3.791962e+04	0.540239	0.087193	0.766259	0.664151	1.175033	833.809963	...
min	1.000000	0.500000	370.000000	5.200000e+02	1.000000	0.000000	0.000000	1.000000	4.000000	370.000000	...
25%	3.000000	1.750000	1440.000000	5.010750e+03	1.000000	0.000000	0.000000	3.000000	7.000000	1200.000000	...
50%	3.000000	2.250000	1930.000000	7.620000e+03	1.500000	0.000000	0.000000	3.000000	7.000000	1580.000000	...
75%	4.000000	2.500000	2570.000000	1.080000e+04	2.000000	0.000000	0.000000	4.000000	8.000000	2240.000000	...
max	33.000000	8.000000	13540.000000	1.074218e+06	3.500000	1.000000	4.000000	5.000000	13.000000	9410.000000	...

8 rows × 21 columns

```
df2.iloc[:,2:].describe()
```

✓ 0.2s Python

Area of the (excluding basement)	...	Built Year	Renovation Year	Postal Code	Lattitude	Longitude	living_area_renov	lot_area_renov	Number of schools nearby	Distance from the airport	Price
520.000000	...	14620.000000	14620.000000	14620.000000	14620.000000	14620.000000	14620.000000	14620.000000	14620.000000	14620.000000	1.462000e+04
301.783926	...	1970.926402	90.924008	122033.062244	52.792848	-114.404007	1996.702257	12753.500068	2.012244	64.950958	5.389322e+05
333.809963	...	29.493625	416.216661	19.082418	0.137522	0.141326	691.093366	26058.414467	0.817284	8.936008	3.675324e+05
370.000000	...	1900.000000	0.000000	122003.000000	52.385900	-114.709000	460.000000	651.000000	1.000000	50.000000	7.800000e+04
200.000000	...	1951.000000	0.000000	122017.000000	52.707600	-114.519000	1490.000000	5097.750000	1.000000	57.000000	3.200000e+05
580.000000	...	1975.000000	0.000000	122032.000000	52.806400	-114.421000	1850.000000	7620.000000	2.000000	65.000000	4.500000e+05
240.000000	...	1997.000000	0.000000	122048.000000	52.908900	-114.315000	2380.000000	10125.000000	3.000000	73.000000	6.450000e+05
410.000000	...	2015.000000	2015.000000	122072.000000	53.007600	-113.505000	6110.000000	560617.000000	3.000000	80.000000	7.700000e+06

5.

```
#checking if NaN values are present
df2.isna().sum()
```

```
id                                      0
Date                                    0
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
```