

```
from google.colab import files
uploaded = files.upload()
```

Choose Files

No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving House Price India.csv to House Price India.csv

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import io
df = pd.read_csv(io.BytesIO(uploaded['House Price India.csv']))

df.head()
```

	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	Latitude
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5	...	1921	0	122003	52.864
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5	...	1909	0	122004	52.887
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3	...	1939	0	122004	52.885
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3	...	2001	0	122005	52.953
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	4	...	1929	0	122006	52.904

5 rows × 23 columns

```
df.tail()
```

	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	Latitude
14615	6762830250	42734	2	1.5	1556	20000	1.0	0	0	4	...	1957	0	122066	52
14616	6762830339	42734	3	2.0	1680	7000	1.5	0	0	4	...	1968	0	122072	52
14617	6762830618	42734	2	1.0	1070	6120	1.0	0	0	3	...	1962	0	122056	52
14618	6762830709	42734	4	1.0	1030	6621	1.0	0	0	4	...	1955	0	122042	52
14619	6762831463	42734	3	1.0	900	4770	1.0	0	0	3	...	1969	2009	122018	52

5 rows × 23 columns

```
df
```

	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	Latt
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5	...	1921	0	122003	52
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5	...	1909	0	122004	52
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3	...	1939	0	122004	52
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3	...	2001	0	122005	52
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	4	...	1929	0	122006	52
...

df.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')
```

df.dtypes

```
id                int64  
Date              int64  
number of bedrooms    int64  
number of bathrooms   float64  
living area          int64  
lot area             int64  
number of floors      float64  
waterfront present    int64  
number of views       int64  
condition of the house int64  
grade of the house    int64  
Area of the house(excluding basement) int64  
Area of the basement  int64  
Built Year           int64  
Renovation Year      int64  
Postal Code          int64  
Lattitude            float64  
Longitude             float64  
living_area_renov     int64  
lot_area_renov        int64  
Number of schools nearby int64  
Distance from the airport int64  
Price                int64  
dtype: object
```

df.info

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

	number of views	condition of the house	...	Built Year	\
0	4	5	...	1921	
1	0	5	...	1909	
2	0	3	...	1939	
3	0	3	...	2001	
4	0	4	...	1929	
...	
14615	0	4	...	1957	
14616	0	4	...	1968	
14617	0	3	...	1962	
14618	0	4	...	1955	
14619	0	3	...	1969	

	Renovation Year	Postal Code	Lattitude	Longitude	living_area_renov	\
0	0	122003	52.8645	-114.557	2880	
1	0	122004	52.8878	-114.470	2470	
2	0	122004	52.8852	-114.468	2940	
3	0	122005	52.9532	-114.321	3350	
4	0	122006	52.9047	-114.485	2060	
...	
14615	0	122066	52.6191	-114.472	2250	
14616	0	122072	52.5075	-114.393	1540	
14617	0	122056	52.7289	-114.507	1130	
14618	0	122042	52.7157	-114.411	1420	
14619	2009	122018	52.5338	-114.552	900	

	lot_area_renov	Number of schools nearby	Distance from the airport	\
0	5400	2	58	
1	4000	2	51	
2	6600	1	53	
3	42847	3	76	
4	15000	1	51	

```
df.shape
```

```
(14620, 23)
```

Univariate Analysis

```
print(df.describe())
```

	id	Date	number of bedrooms	number of bathrooms	\
count	1.462000e+04	14620.000000	14620.000000	14620.000000	
mean	6.762821e+09	42604.538646	3.379343	2.129583	
std	6.237575e+03	67.347991	0.938719	0.769934	
min	6.762810e+09	42491.000000	1.000000	0.500000	
25%	6.762815e+09	42546.000000	3.000000	1.750000	
50%	6.762821e+09	42600.000000	3.000000	2.250000	
75%	6.762826e+09	42662.000000	4.000000	2.500000	
max	6.762832e+09	42734.000000	33.000000	8.000000	

	living area	lot area	number of floors	waterfront present	\
count	14620.000000	1.462000e+04	14620.000000	14620.000000	
mean	2098.262996	1.509328e+04	1.502360	0.007661	
std	928.275721	3.791962e+04	0.540239	0.087193	
min	370.000000	5.200000e+02	1.000000	0.000000	
25%	1440.000000	5.010750e+03	1.000000	0.000000	
50%	1930.000000	7.620000e+03	1.500000	0.000000	
75%	2570.000000	1.080000e+04	2.000000	0.000000	
max	13540.000000	1.074218e+06	3.500000	1.000000	

	number of views	condition of the house	...	Built Year	\
count	14620.000000	14620.000000	...	14620.000000	
mean	0.233105	3.430506	...	1970.926402	
std	0.766259	0.664151	...	29.493625	
min	0.000000	1.000000	...	1900.000000	
25%	0.000000	3.000000	...	1951.000000	
50%	0.000000	3.000000	...	1975.000000	
75%	0.000000	4.000000	...	1997.000000	
max	4.000000	5.000000	...	2015.000000	

	Renovation Year	Postal Code	Lattitude	Longitude	\
count	14620.000000	14620.000000	14620.000000	14620.000000	
mean	90.924008	122033.062244	52.792848	-114.404007	
std	416.216661	19.082418	0.137522	0.141326	
min	0.000000	122003.000000	52.385900	-114.709000	
25%	0.000000	122017.000000	52.707600	-114.519000	
50%	0.000000	122032.000000	52.806400	-114.421000	
75%	0.000000	122048.000000	52.908900	-114.315000	
max	2015.000000	122072.000000	53.007600	-113.505000	

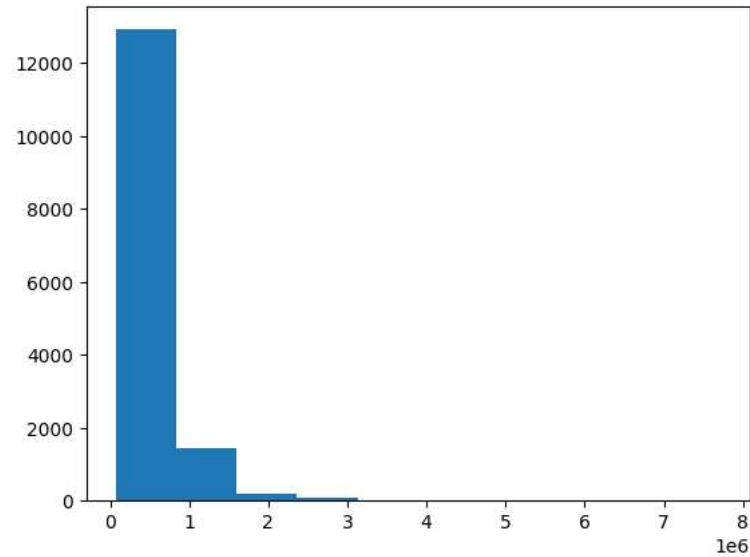
```
living_area_renov lot_area_renov Number of schools nearby \
```

count	14620.000000	14620.000000	14620.000000
mean	1996.702257	12753.500068	2.012244
std	691.093366	26058.414467	0.817284
min	460.000000	651.000000	1.000000
25%	1490.000000	5097.750000	1.000000
50%	1850.000000	7620.000000	2.000000
75%	2380.000000	10125.000000	3.000000
max	6110.000000	560617.000000	3.000000

	Distance from the airport	Price
count	14620.000000	1.462000e+04
mean	64.950958	5.389322e+05
std	8.936008	3.675324e+05
min	50.000000	7.800000e+04
25%	57.000000	3.200000e+05
50%	65.000000	4.500000e+05
75%	72.000000	6.450000e+05

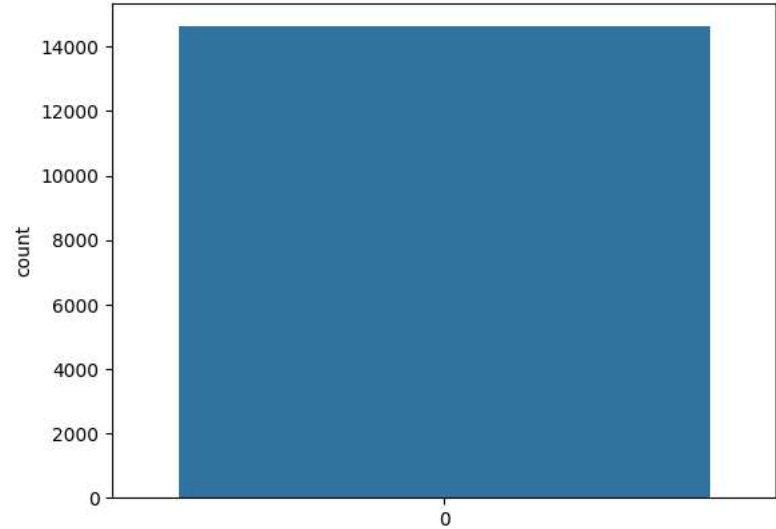
```
plt.hist(df['Price'])
```

```
(array([1.2916e+04, 1.4260e+03, 1.9100e+02, 6.0000e+01, 1.9000e+01,
        2.0000e+00, 2.0000e+00, 1.0000e+00, 1.0000e+00, 2.0000e+00]),
 array([ 78000., 840200., 1602400., 2364600., 3126800., 3889000.,
        4651200., 5413400., 6175600., 6937800., 7700000.]),
 <BarContainer object of 10 artists>)
```



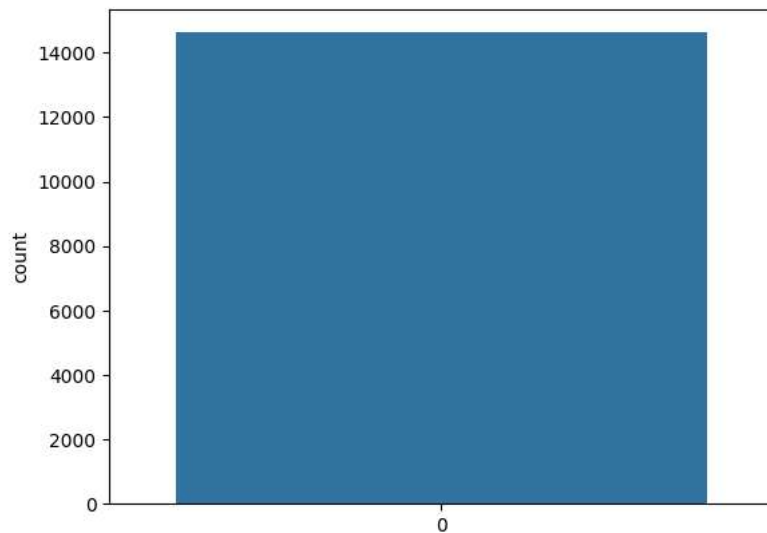
```
sns.countplot(df['number of views'])
```

<Axes: ylabel='count'>



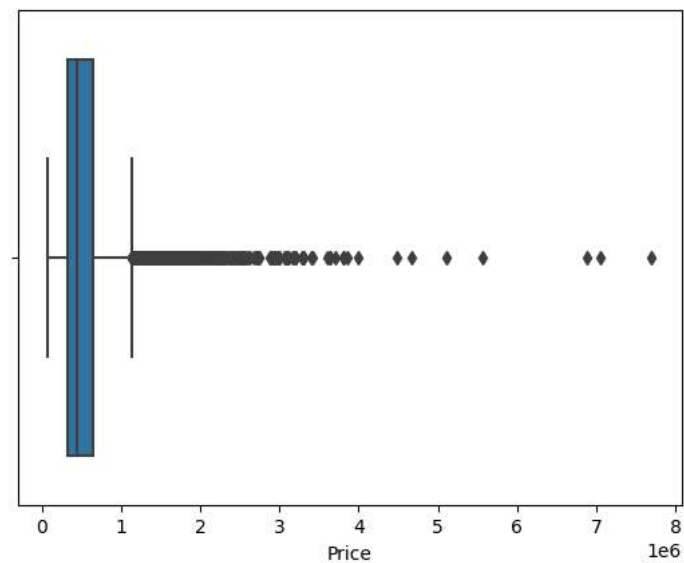
```
sns.countplot(df['number of floors'])
```

<Axes: ylabel='count'>



```
sns.boxplot(x=df['Price'])
```

<Axes: xlabel='Price'>



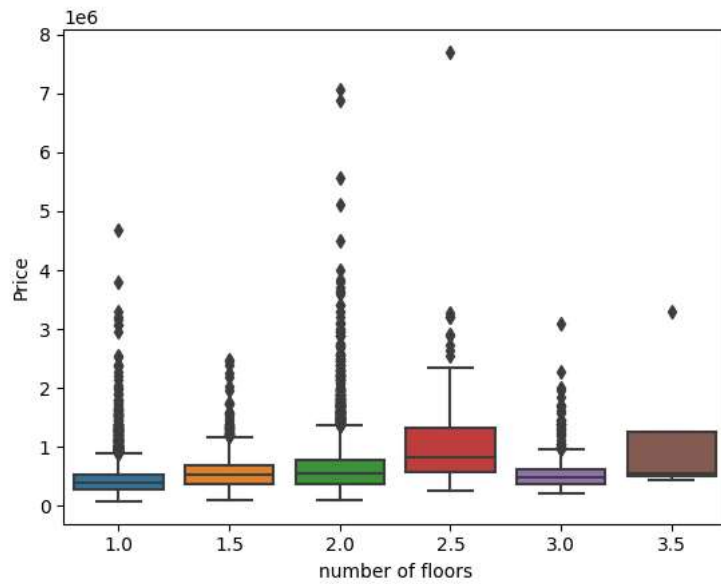
```
sns.boxplot(x=df['number of floors'],y=df['Price'])
```

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

```
1e6
```

```
sns.boxplot(x=df['number of floors'],y=df['Price'])
```

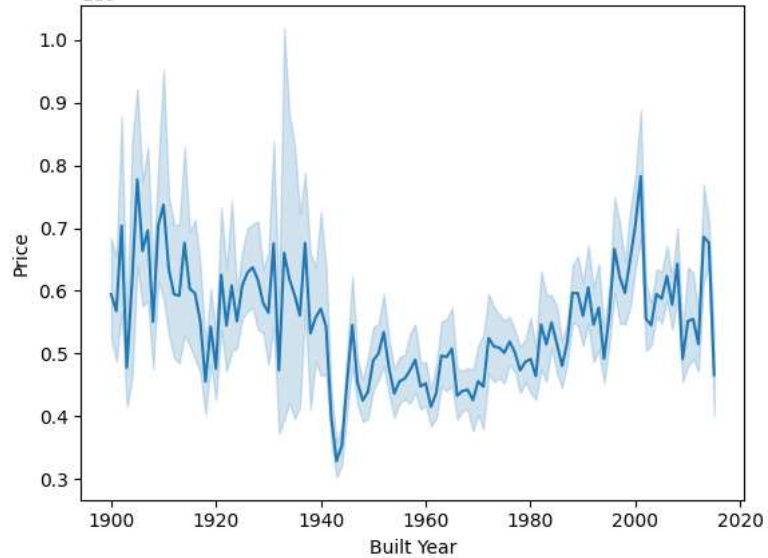
```
<Axes: xlabel='number of floors', ylabel='Price'>
```



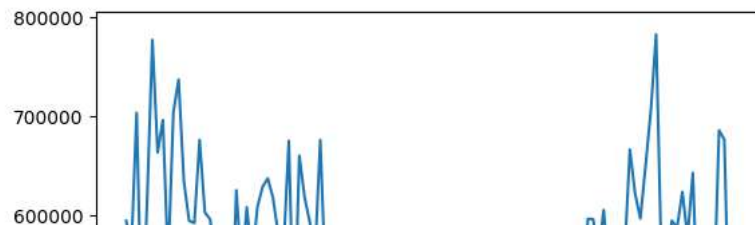
```
sns.lineplot(x=df['Built Year'],y=df['Price'])
```

```
<Axes: xlabel='Built Year', ylabel='Price'>
```

```
1e6
```

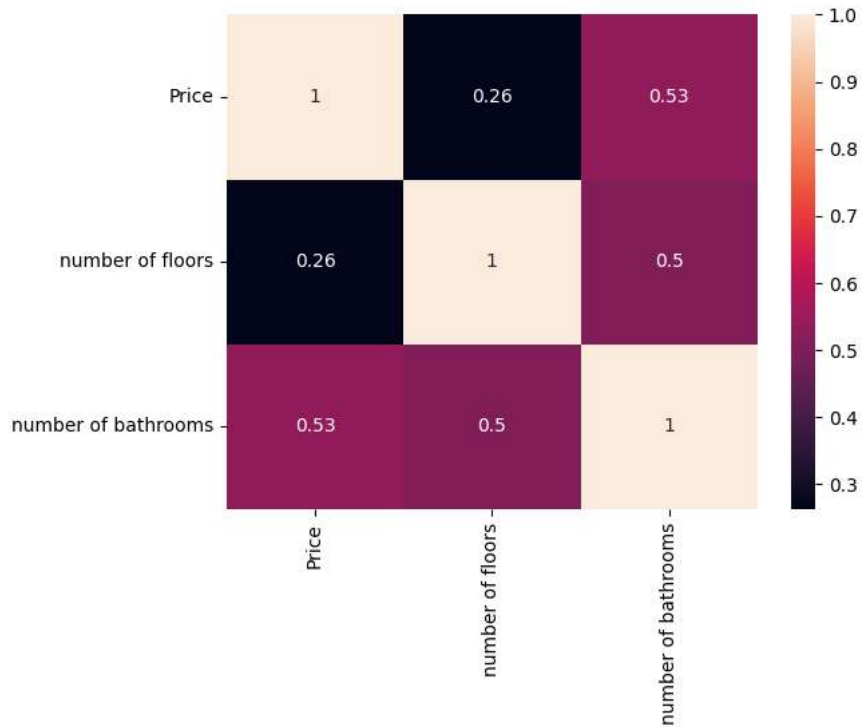


```
sns.lineplot(x=df.groupby('Built Year').mean().index, y=df.groupby('Built Year').mean()['Price'])
plt.show()
```



```
sns.heatmap(df[['Price', 'number of floors', 'number of bathrooms']].corr(), annot=True)
```

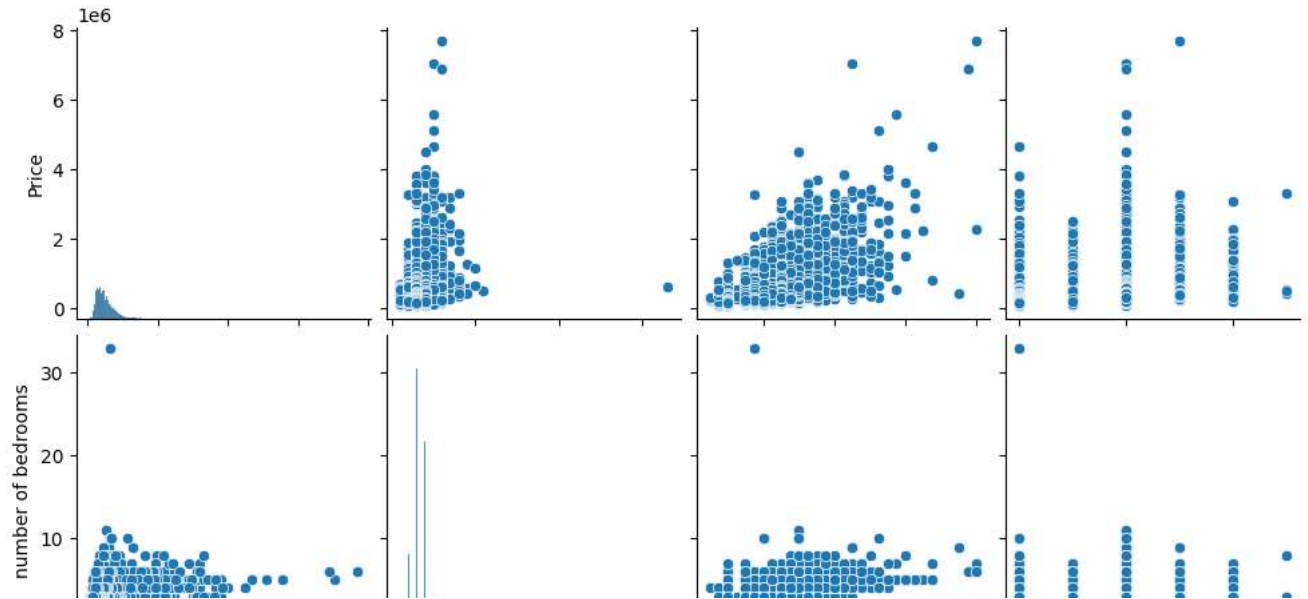
<Axes: >



Multivariate Analysis

```
sns.pairplot(df[['Price', 'number of bedrooms', 'number of bathrooms', 'number of floors']])
```

```
<seaborn.axisgrid.PairGrid at 0x7a6cd53b1990>
```



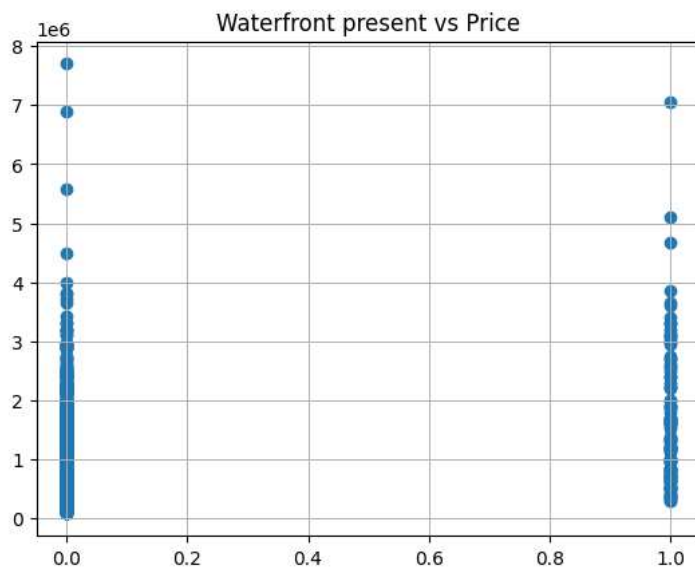
```
df.duplicated().sum()
```

```
0
```

```
plt.scatter(df['waterfront present'], df['Price'])
```

```
plt.title("Waterfront present vs Price")
```

```
plt.grid(linestyle='-', linewidth=0.7)
```

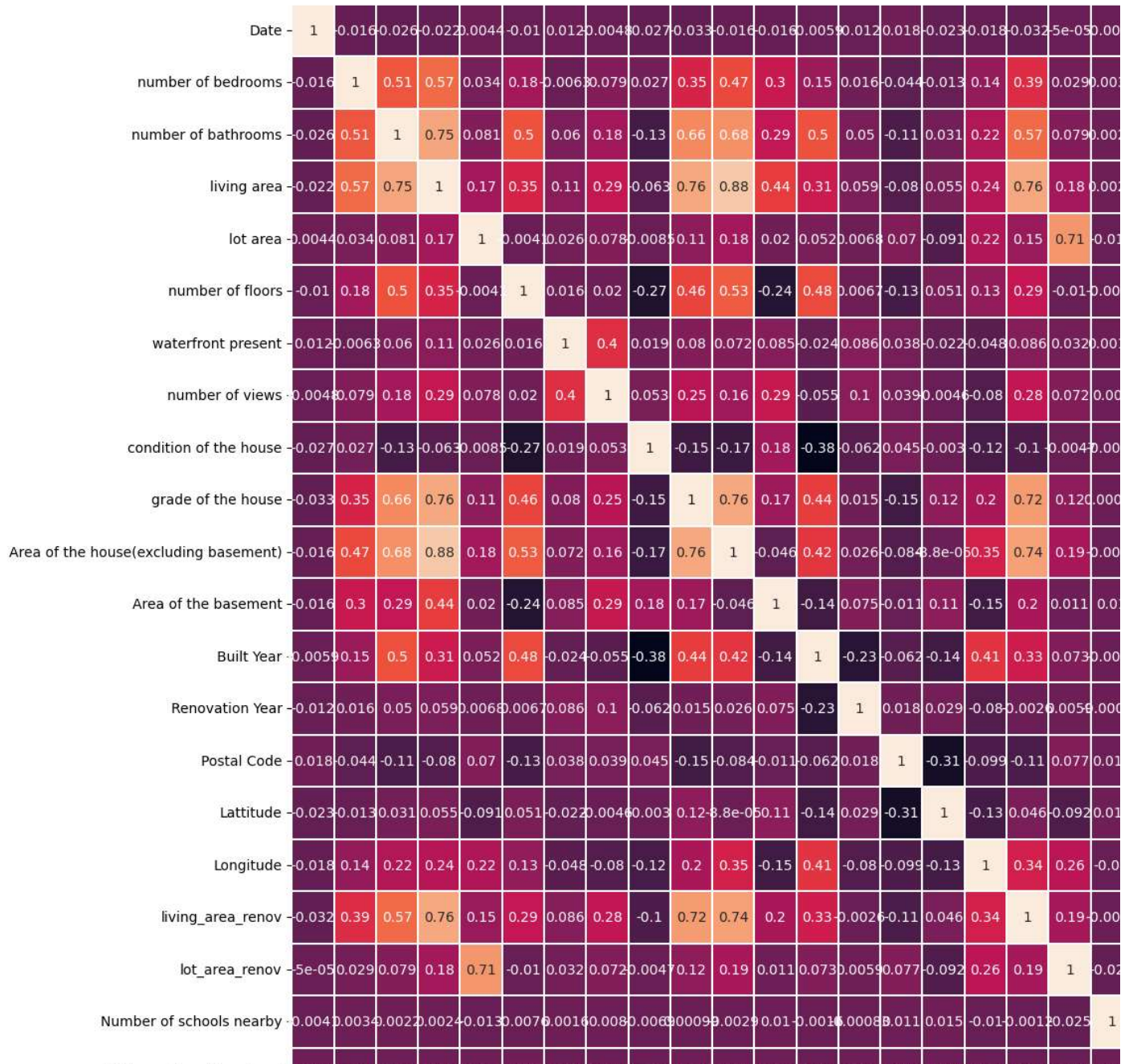


```
sns.scatterplot(df['number of bedrooms'])
```

```
plt.grid(linestyle='-', linewidth=0.5)
```




```
plt.subplots(figsize=(15, 15))
sns.heatmap(df.drop(['id'], axis=1).corr(), linewidth=0.3, annot=True)
plt.show()
```



```
print(df.describe())
```

	id	Date	number of bedrooms	number of bathrooms
count	1.462000e+04	14620.000000	14620.000000	14620.000000
mean	6.762821e+09	42604.538646	3.379343	2.129583
std	6.237575e+03	67.347991	0.938719	0.769934
min	6.762810e+09	42491.000000	1.000000	0.500000
25%	6.762815e+09	42546.000000	3.000000	1.750000
50%	6.762821e+09	42600.000000	3.000000	2.250000
75%	6.762826e+09	42662.000000	4.000000	2.500000
max	6.762832e+09	42734.000000	33.000000	8.000000

	living area	lot area	number of floors	waterfront present \
count	14620.000000	1.462000e+04	14620.000000	14620.000000
mean	2098.262996	1.509328e+04	1.502360	0.007661
std	928.275721	3.791962e+04	0.540239	0.087193
min	370.000000	5.200000e+02	1.000000	0.000000
25%	1440.000000	5.010750e+03	1.000000	0.000000
50%	1930.000000	7.620000e+03	1.500000	0.000000
75%	2570.000000	1.080000e+04	2.000000	0.000000
max	13540.000000	1.074218e+06	3.500000	1.000000

	number of views	condition of the house	...	Built Year \
count	14620.000000	14620.000000	...	14620.000000
mean	0.233105	3.430506	...	1970.926402
std	0.766259	0.664151	...	29.493625
min	0.000000	1.000000	...	1900.000000
25%	0.000000	3.000000	...	1951.000000
50%	0.000000	3.000000	...	1975.000000
75%	0.000000	4.000000	...	1997.000000
max	4.000000	5.000000	...	2015.000000

	Renovation Year	Postal Code	Latitude	Longitude \
count	14620.000000	14620.000000	14620.000000	14620.000000
mean	90.924008	122033.062244	52.792848	-114.404007
std	416.216661	19.082418	0.137522	0.141326
min	0.000000	122003.000000	52.385900	-114.709000
25%	0.000000	122017.000000	52.707600	-114.519000
50%	0.000000	122032.000000	52.806400	-114.421000
75%	0.000000	122048.000000	52.908900	-114.315000
max	2015.000000	122072.000000	53.007600	-113.505000

	living_area_renov	lot_area_renov	Number of schools nearby \
count	14620.000000	14620.000000	14620.000000
mean	1996.702257	12753.500068	2.012244
std	691.093366	26058.414467	0.817284
min	460.000000	651.000000	1.000000
25%	1490.000000	5097.750000	1.000000
50%	1850.000000	7620.000000	2.000000
75%	2380.000000	10125.000000	3.000000
max	6110.000000	560617.000000	3.000000

	Distance from the airport	Price
count	14620.000000	1.462000e+04
mean	64.950958	5.389322e+05
std	8.936008	3.675324e+05
min	50.000000	7.800000e+04
25%	57.000000	3.200000e+05
50%	65.000000	4.500000e+05
75%	73.000000	6.450000e+05

```
print(df.count())
```

id	14620
Date	14620
number of bedrooms	14620
number of bathrooms	14620
living area	14620
lot area	14620
number of floors	14620
waterfront present	14620
number of views	14620
condition of the house	14620
grade of the house	14620
Area of the house(excluding basement)	14620
Area of the basement	14620
Built Year	14620
Renovation Year	14620
Postal Code	14620
Latitude	14620
Longitude	14620
living_area_renov	14620
lot_area_renov	14620
Number of schools nearby	14620
Distance from the airport	14620
Price	14620
dtype: int64	

```
print(df.corr())
```

	id	Date	number of bedrooms \
id	1.000000	0.045966	-0.329034
Date	0.045966	1.000000	-0.015663

number of bedrooms	-0.329034	-0.015663	1.000000
number of bathrooms	-0.516909	-0.026485	0.509784
living area	-0.648127	-0.021958	0.570526
lot area	-0.100269	0.004392	0.034416
number of floors	-0.312305	-0.010335	0.177294
waterfront present	-0.112937	0.012006	-0.006257
number of views	-0.293004	-0.004782	0.078665
condition of the house	-0.045061	-0.027402	0.026597
grade of the house	-0.673448	-0.033097	0.352945
Area of the house(excluding basement)	-0.565116	-0.015994	0.473599
Area of the basement	-0.290806	-0.015711	0.300332
Built Year	-0.068645	-0.005869	0.152954
Renovation Year	-0.109155	-0.011636	0.016132
Postal Code	0.294709	0.018243	-0.044156
Latitude	-0.479334	-0.023327	-0.013163
Longitude	-0.070841	-0.018231	0.135712
living_area_renov	-0.599900	-0.032495	0.389855
lot_area_renov	-0.089604	-0.000050	0.029400
Number of schools nearby	-0.004821	-0.004071	0.003397
Distance from the airport	-0.004542	0.011457	-0.006157
Price	-0.773114	-0.027919	0.308460

	number of bathrooms	living area \
id	-0.516909	-0.648127
Date	-0.026485	-0.021958
number of bedrooms	0.509784	0.570526
number of bathrooms	1.000000	0.753517
living area	0.753517	1.000000
lot area	0.080806	0.174420
number of floors	0.502924	0.354743
waterfront present	0.060104	0.105837
number of views	0.183789	0.287728
condition of the house	-0.128232	-0.063358
grade of the house	0.663054	0.761835
Area of the house(excluding basement)	0.684391	0.875793
Area of the basement	0.287190	0.441491
Built Year	0.498127	0.309602
Renovation Year	0.049669	0.059400
Postal Code	-0.105546	-0.080303
Latitude	0.031156	0.054518
Longitude	0.223904	0.240208
living_area_renov	0.570530	0.757571
lot_area_renov	0.078627	0.180312
Number of schools nearby	0.002180	0.002370
Distance from the airport	0.009206	0.002511
Price	0.531735	0.712169

	lot area	number of floors \
id	-0.100269	-0.312305
Date	0.004392	-0.010335
number of bedrooms	0.034416	0.177294
number of bathrooms	0.080806	0.502924
living area	0.174420	0.354743
lot area	1.000000	-0.004138
number of floors	-0.004138	1.000000

```
print(df['Number of schools nearby'].value_counts())
```

```
3    4973
2    4853
1    4794
Name: Number of schools nearby, dtype: int64
```

Handle the missing values

```
print(df.isnull().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
Latitude            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

```

```
df.dropna(inplace=True)
```

```
df.fillna(0, inplace=True)
```

```
df.interpolate(inplace=True)
```

```

from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler

```

```

x=df.drop(['Price', 'Date'],axis=1)
x.set_index(['id'],inplace=True)
y=df[['id', 'Price']]

```

```
x.head()
```

	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	Renovatio Yea
id													
6762810145	5	2.50	3650	9050	2.0	0	4	5	10	3370	280	1921	
6762810635	4	2.50	2920	4000	1.5	0	0	5	8	1910	1010	1909	
6762810998	5	2.75	2910	9480	1.5	0	0	3	8	2910	0	1939	
6762812605	4	2.50	3310	42998	2.0	0	0	3	9	3310	0	2001	
6762812919	3	2.00	2710	4500	1.5	0	0	4	8	1880	830	1929	

```

sc=StandardScaler()
sc=MinMaxScaler()
x=pd.DataFrame(sc.fit_transform(x),columns=x.columns.values)
x.head()

```

	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	Renovatio Yea
0	0.12500	0.266667	0.249051	0.007945	0.4	0.0	1.0	1.00	0.666667	0.331858	0.058091	0.182609	0.
1	0.09375	0.266667	0.193622	0.003241	0.2	0.0	0.0	1.00	0.444444	0.170354	0.209544	0.078261	0.
2	0.12500	0.300000	0.192863	0.008345	0.2	0.0	0.0	0.50	0.444444	0.280973	0.000000	0.339130	0.
3	0.09375	0.266667	0.223235	0.039562	0.4	0.0	0.0	0.50	0.555556	0.325221	0.000000	0.878261	0.
4	0.06250	0.200000	0.177677	0.003707	0.2	0.0	0.0	0.75	0.444444	0.167035	0.172199	0.252174	0.

```
y.head()
```

	id	Price
0	6762810145	2380000
1	6762810635	1400000
2	6762810998	1200000

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import r2_score
```

```
x_train,x_test,y_train,y_test = train_test_split(x , y['Price'],test_size =0.1,random_state=2)
model = GradientBoostingRegressor(n_estimators= 400,max_depth=5,min_samples_split=2,learning_rate=0.1)
model.fit(x_train,y_train)
```

```

v      GradientBoostingRegressor
GradientBoostingRegressor(max_depth=5, n_estimators=400)
```

```
y_pred = model.predict(x_test)
model.score(x_test,y_test)
```

```
0.9118862866954918
```

```
r2_score(y_pred,y_test)
```

```
0.9012749137875955
```

```
y_pred
```

```
array([497766.12740438, 244495.3776842 , 293819.40063242, ...,
       698495.60350629, 297006.00386358, 245881.76921871])
```

```
y_pred_list = y['id'][-len(y_pred):].tolist()
```

```
y_pred_df=pd.DataFrame(y_pred_list,columns=['ID'])
y_pred_df['Predicted Price'] = y_pred.round(2)
```

```
y_pred_df
```

	ID	Predicted Price
0	6762811233	497766.13
1	6762811403	244495.38
2	6762811775	293819.40
3	6762811861	397555.35
4	6762812009	474843.29
...
1457	6762830250	1041014.57
1458	6762830339	317512.59
1459	6762830618	698495.60
1460	6762830709	297006.00
1461	6762831463	245881.77

```
1462 rows × 2 columns
```

