SYAM KRISHNA REDDY PULAGAM

REG NO: 21BAI1725 VIT CHENNAI CAMPUS

AI&ML ASSIGNMENT-2

Importing all the neccesary libraries

In [1]:

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

Importing dataset from csv file

In [2]:

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

Out[2]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views
0	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
...	...	...	...	...	...	...	...	...	..
14615	6762830250	42734	2	1.50	1556	20000	1.0	0	0
14616	6762830339	42734	3	2.00	1680	7000	1.5	0	0
14617	6762830618	42734	2	1.00	1070	6120	1.0	0	0
14618	6762830709	42734	4	1.00	1030	6621	1.0	0	0
14619	6762831463	42734	3	1.00	900	4770	1.0	0	0

14620 rows × 23 columns

In [3]:

```
df.head()
```

Out[3]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	co
0	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

In [4]:

```
df.shape
```

Out[4]:

```
(14620, 23)
```

In [5]:

```
columns=df.columns
columns
```

Out[5]:

```
Index(['id', 'Date', 'number of bedrooms', 'number of bathrooms',
       'living area', 'lot area', 'number of floors', 'waterfront presen
t',
       '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')
```

In [6]:

```
type(columns)
```

Out[6]:

```
pandas.core.indexes.base.Index
```

In [7]:

```
for i in columns:
    print(f"'{i}'---> ",end="")
    print(df[i].dtype)
    print()
```

```
'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
```

In [8]:

```
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
```

Univariate Analysis

In [9]:

```
if len(df['id'])==len(df['id'].unique()):
    print(len(df['id']))
    print("it has all unique values. so it shows no relation with price column")
else:
    print("it has duplicates")
```

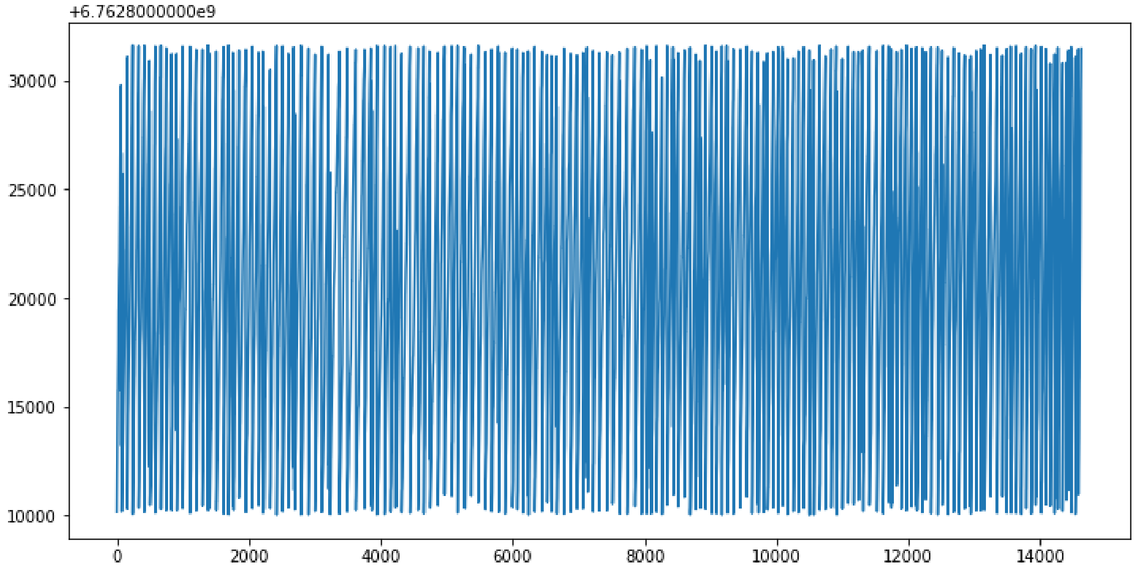
```
14620
it has all unique values. so it shows no relation with price column
```

In [10]:

```
plt.figure(figsize=(12,6))
plt.plot(df['id'])
```

Out[10]:

```
[<matplotlib.lines.Line2D at 0x1b1b33cab80>]
```

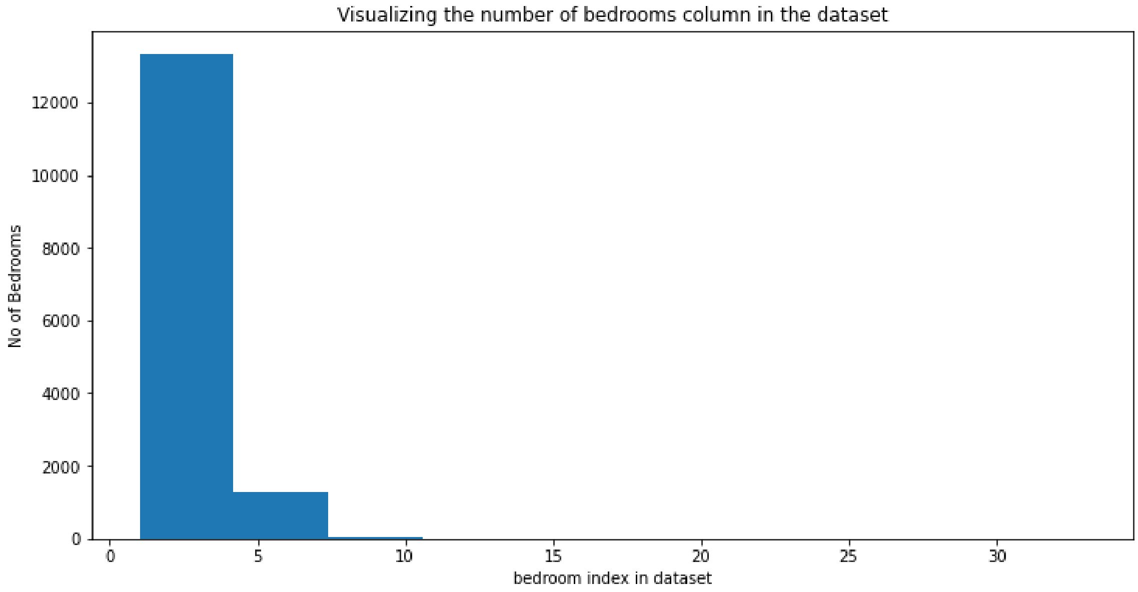


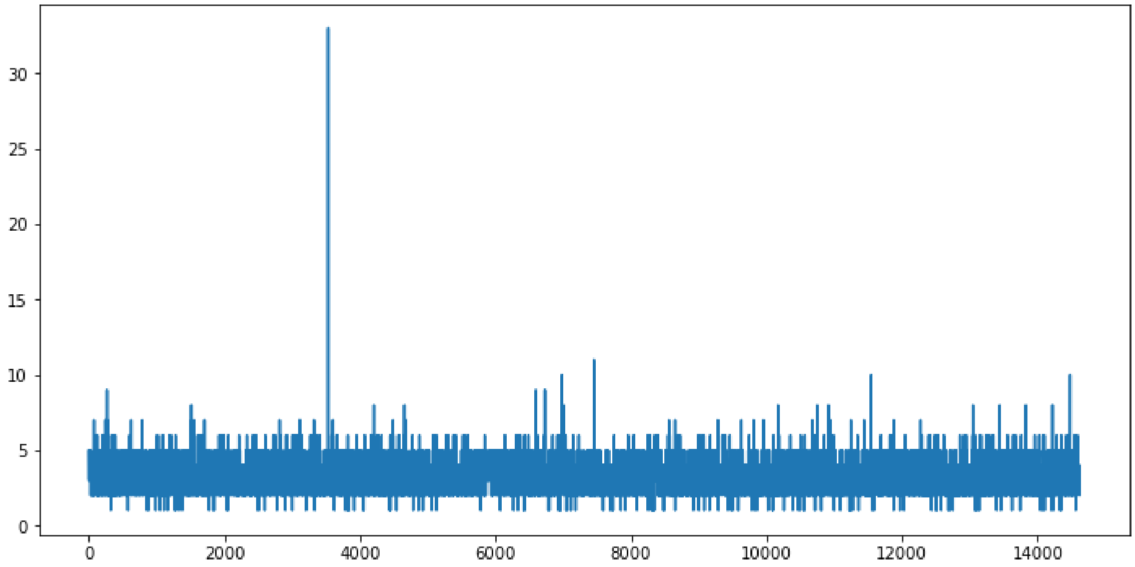
In [11]:

```
plt.figure(figsize=(12,6))
plt.hist(df['number of bedrooms'])
plt.ylabel("No of Bedrooms")
plt.xlabel("bedroom index in dataset")
plt.title("Visualizing the number of bedrooms column in the dataset")
plt.figure(figsize=(12,6))
plt.plot(df['number of bedrooms'])
```

Out[11]:

```
[<matplotlib.lines.Line2D at 0x1b1b3d9fd30>]
```



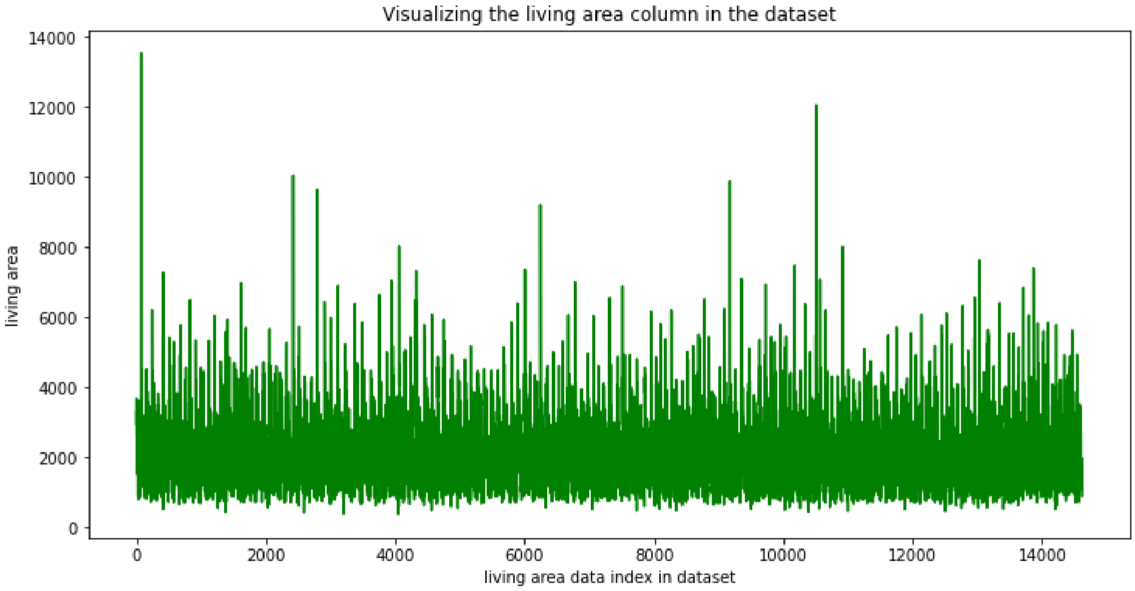


In [12]:

```
plt.figure(figsize=(12,6))
plt.plot(df['living area'],'g')
plt.ylabel("living area")
plt.xlabel("living area data index in dataset")
plt.title("Visualizing the living area column in the dataset")
```

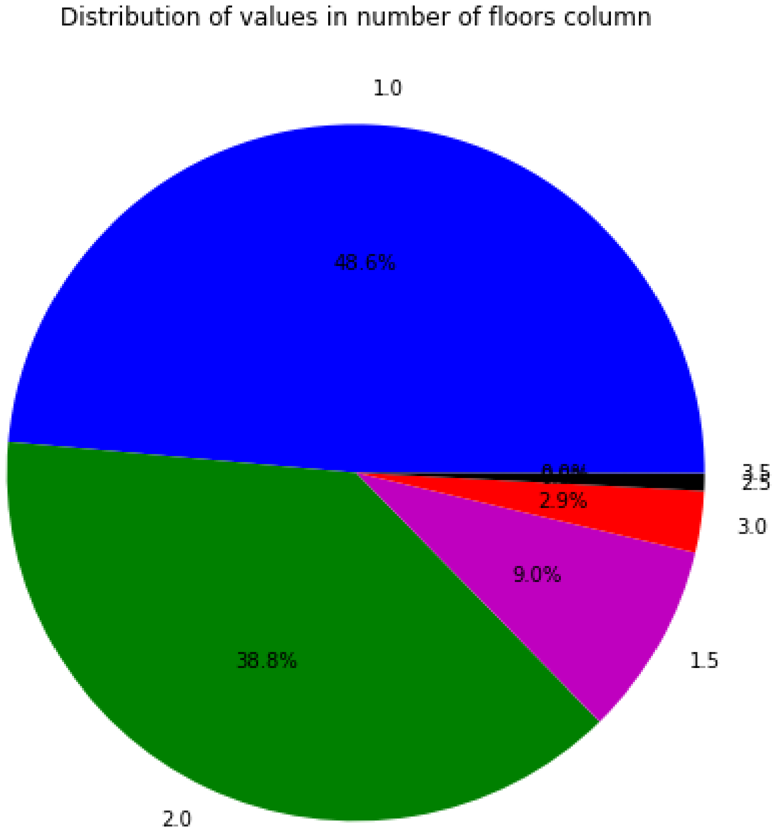
Out[12]:

```
Text(0.5, 1.0, 'Visualizing the living area column in the dataset')
```



In [13]:

```
plt.figure(figsize=(15,8))
a=[1.0,2.0,1.5,3.0,2.5,3.5]
colors=['b','g','m','r','k']
plt.pie(df['number of floors'].value_counts(),labels=a,colors=colors,autopct = "%1.1f%%"
plt.title("Distribution of values in number of floors column")
plt.show()
```

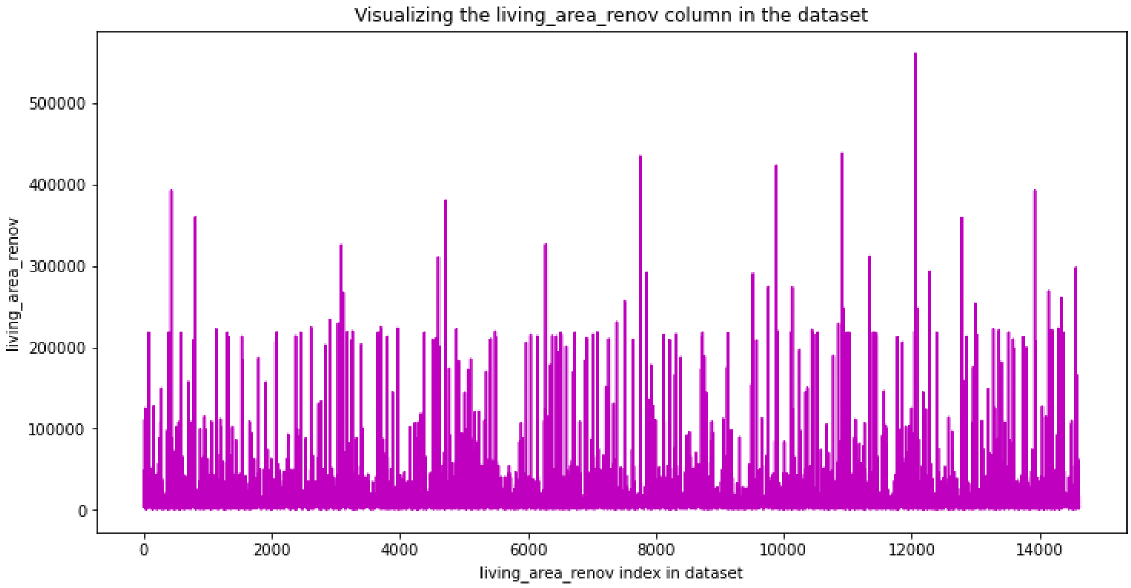


In [14]:

```
plt.figure(figsize=(12,6))
plt.plot(df['lot_area_renov'],'m')
plt.ylabel("living_area_renov")
plt.xlabel("living_area_renov index in dataset")
plt.title("Visualizing the living_area_renov column in the dataset")
```

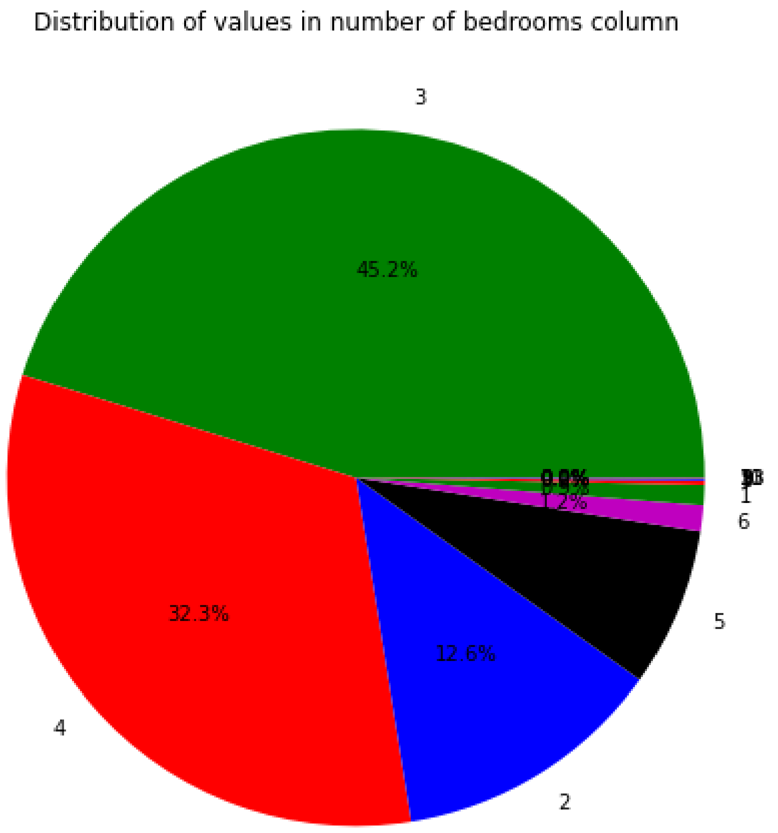
Out[14]:

```
Text(0.5, 1.0, 'Visualizing the living_area_renov column in the dataset')
```



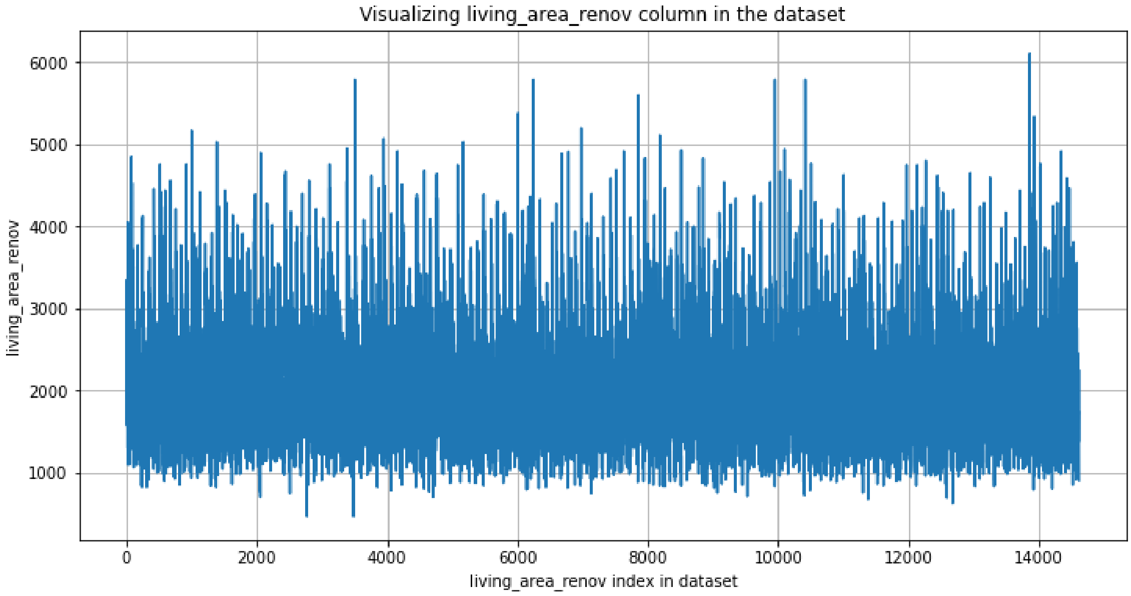
In [15]:

```
plt.figure(figsize=(15,8))
label=[3,4,2,5,6,1,7,8,9,10,33,11]
colors=['g','r','b','k','m']
plt.pie(df['number of bedrooms'].value_counts(),labels=label,colors=colors,autopct = "%1
plt.title("Distribution of values in number of bedrooms column")
plt.show()
```



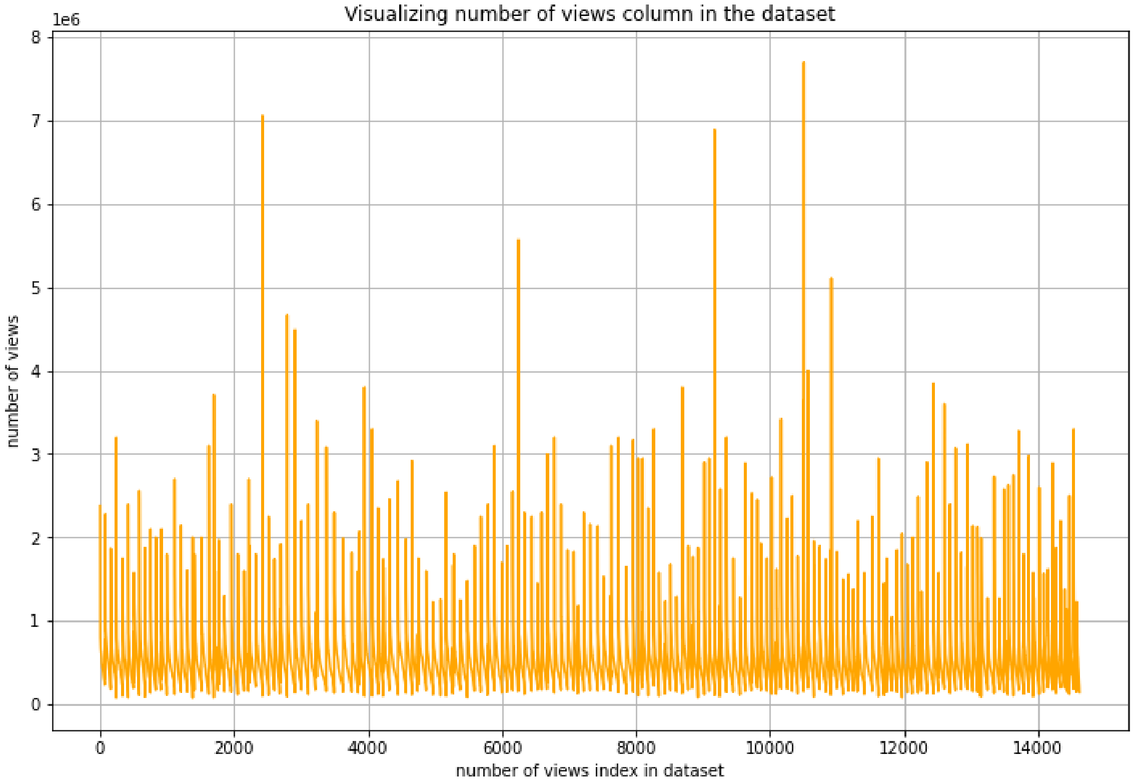
In [16]:

```
plt.figure(figsize=(12,6))
plt.plot(df['living_area_renov'],)
plt.ylabel("living_area_renov")
plt.xlabel("living_area_renov index in dataset")
plt.title("Visualizing living_area_renov column in the dataset")
plt.grid(True)
```



In [17]:

```
plt.figure(figsize=(12,8))
plt.plot(df['Price'],color='orange')
plt.ylabel("number of views")
plt.xlabel("number of views index in dataset")
plt.title("Visualizing number of views column in the dataset")
plt.grid(True)
```



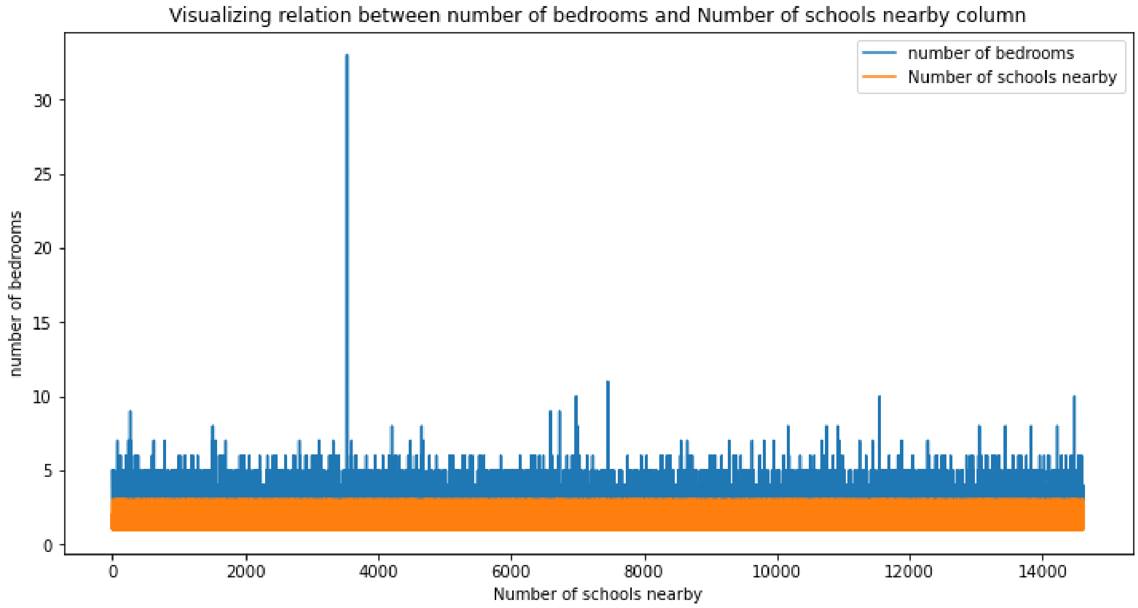
Bi - Variate Analysis

In [18]:

```
plt.figure(figsize=(12,6))
plt.plot(df['number of bedrooms'])
plt.plot(df['Number of schools nearby'])
plt.xlabel("Number of schools nearby")
plt.ylabel("number of bedrooms")
plt.title("Visualizing relation between number of bedrooms and Number of schools nearby
plt.legend(['number of bedrooms','Number of schools nearby'])
```

Out[18]:

```
<matplotlib.legend.Legend at 0x1b1b5879f70>
```

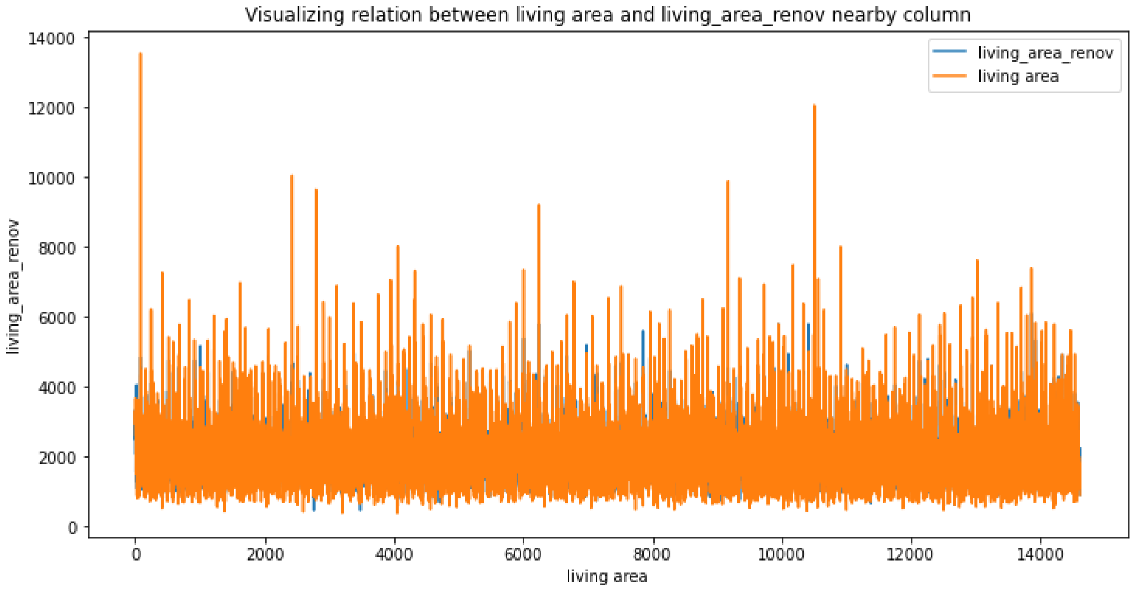


In [19]:

```
plt.figure(figsize=(12,6))
plt.plot(df['living_area_renov'])
plt.plot(df['living area'])
plt.xlabel("living area")
plt.ylabel("living_area_renov")
plt.title("Visualizing relation between living area and living_area_renov nearby column"
plt.legend(['living_area_renov','living area'])
```

Out[19]:

```
<matplotlib.legend.Legend at 0x1b1b60c8760>
```

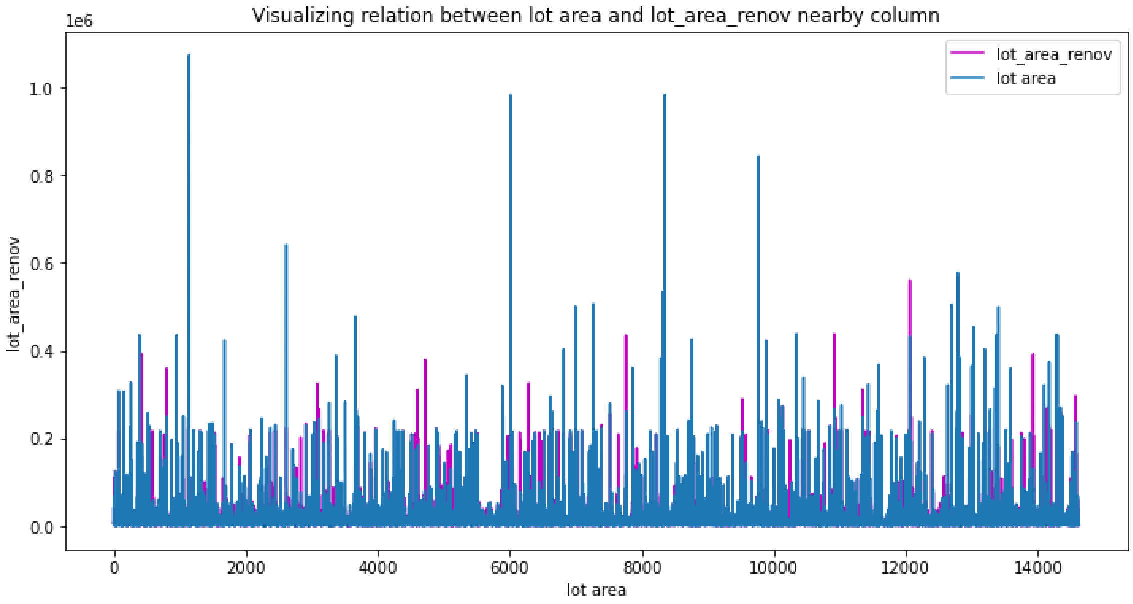


In [20]:

```
plt.figure(figsize=(12,6))
plt.plot(df['lot_area_renov'],'m')
plt.plot(df['lot area'])
plt.xlabel("lot area")
plt.ylabel("lot_area_renov")
plt.title("Visualizing relation between lot area and lot_area_renov nearby column")
plt.legend(['lot_area_renov','lot area'])
```

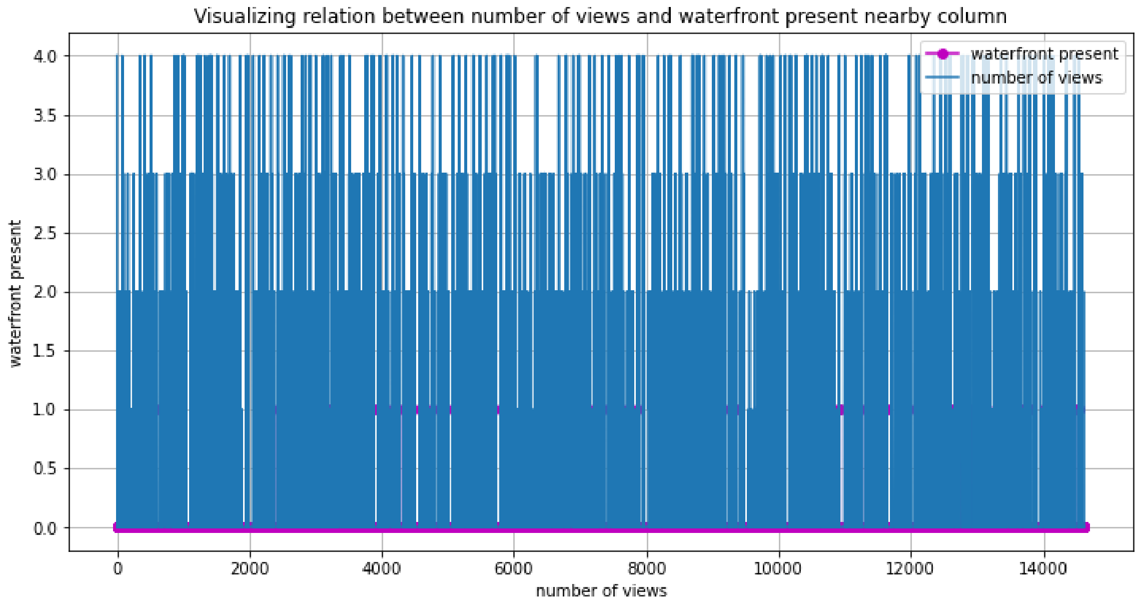
Out[20]:

```
<matplotlib.legend.Legend at 0x1b1b5f6ebe0>
```



In [21]:

```
plt.figure(figsize=(12,6))
plt.plot(df['waterfront present'],'o-m')
plt.plot(df['number of views'])
plt.xlabel("number of views")
plt.ylabel("waterfront present")
plt.title("Visualizing relation between number of views and waterfront present nearby co
plt.legend(['waterfront present','number of views'])
plt.grid(True)
```

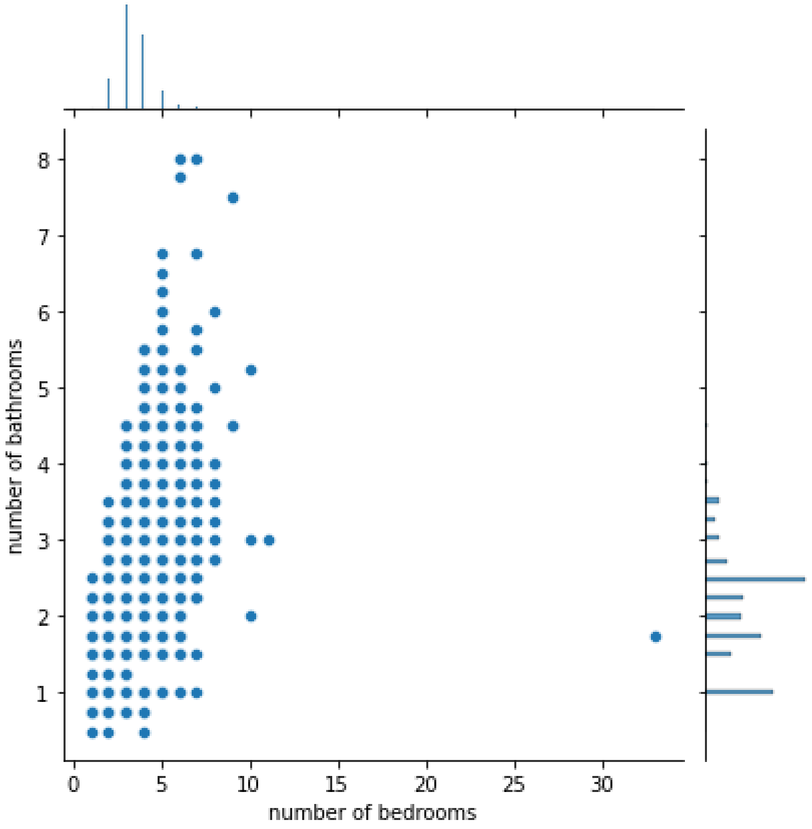


In [35]:

```
sns.jointplot(x= 'number of bedrooms',y ='number of bathrooms',data=df)
```

Out[35]:

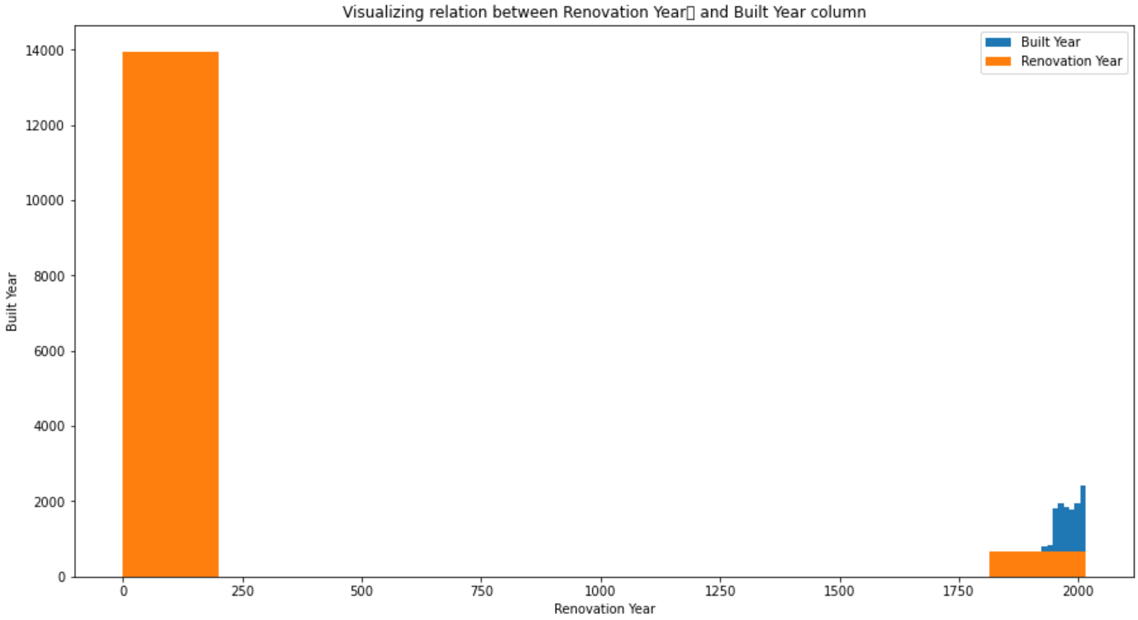
```
<seaborn.axisgrid.JointGrid at 0x1b1bdd9ea30>
```

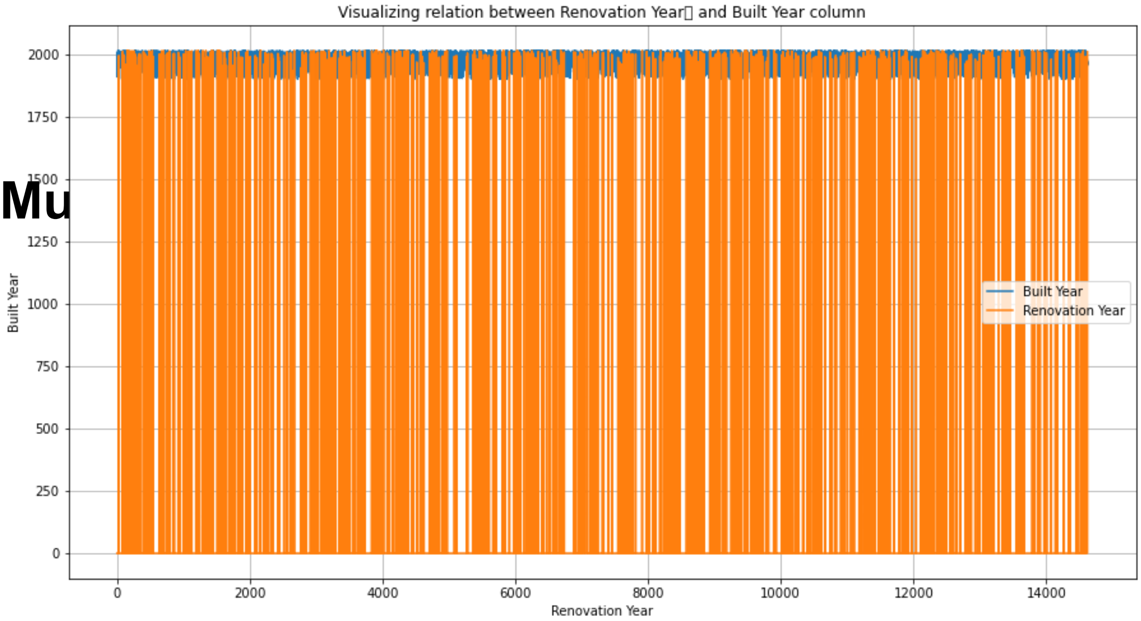


In [22]:

```
#using hist
plt.figure(figsize=(15,8))
plt.hist(df['Built Year'])
plt.hist(df['Renovation Year'])
plt.xlabel("Renovation Year")
plt.ylabel("Built Year")
plt.title("Visualizing relation between Renovation Year  and Built Year column")
plt.legend(['Built Year','Renovation Year'])
#using plot
plt.figure(figsize=(15,8))
plt.plot(df['Built Year'])
plt.plot(df['Renovation Year'])
plt.xlabel("Renovation Year")
plt.ylabel("Built Year")
plt.title("Visualizing relation between Renovation Year  and Built Year column")
plt.legend(['Built Year','Renovation Year'])
plt.grid(True)
```

```
C:\Users\HP\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.p
y:240: RuntimeWarning: Glyph 9 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\HP\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.p
y:203: RuntimeWarning: Glyph 9 missing from current font.
  font.set_text(s, 0, flags=flags)
```




Visualizing relation between Renovation Year and Built Year column
Built Year
Renovation Year
Built Year
Renovation Year
2000
1750
1500
1250
1000
750
500
250
0
0
2000
4000
6000
8000
10000
12000
14000

Mu

In [40]:

```
df.corr()
```

Out[40]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	numbe of floor
id	1.000000	0.045966	-0.329034	-0.516909	-0.648127	-0.100269	-0.31230
Date	0.045966	1.000000	-0.015663	-0.026485	-0.021958	0.004392	-0.01033
number of bedrooms	-0.329034	-0.015663	1.000000	0.509784	0.570526	0.034416	0.17729
number of bathrooms	-0.516909	-0.026485	0.509784	1.000000	0.753517	0.080806	0.50292
living area	-0.648127	-0.021958	0.570526	0.753517	1.000000	0.174420	0.35474
lot area	-0.100269	0.004392	0.034416	0.080806	0.174420	1.000000	-0.00413
number of floors	-0.312305	-0.010335	0.177294	0.502924	0.354743	-0.004138	1.00000
waterfront present	-0.112937	0.012006	-0.006257	0.060104	0.105837	0.026282	0.01631
number of views	-0.293004	-0.004782	0.078665	0.183789	0.287728	0.078308	0.02015
condition of the house	-0.045061	-0.027402	0.026597	-0.128232	-0.063358	-0.008548	-0.26992
grade of the house	-0.673448	-0.033097	0.352945	0.663054	0.761835	0.110546	0.46308
Area of the house(excluding basement)	-0.565116	-0.015994	0.473599	0.684391	0.875793	0.183553	0.52564
Area of the basement	-0.290806	-0.015711	0.300332	0.287190	0.441491	0.019755	-0.24297
Built Year	-0.068645	-0.005869	0.152954	0.498127	0.309602	0.051615	0.48156
Renovation Year	-0.109155	-0.011636	0.016132	0.049669	0.059400	0.006848	0.00670
Postal Code	0.294709	0.018243	-0.044156	-0.105546	-0.080303	0.070131	-0.12978
Lattitude	-0.479334	-0.023327	-0.013163	0.031156	0.054518	-0.090983	0.05073
Longitude	-0.070841	-0.018231	0.135712	0.223904	0.240208	0.221432	0.12755
living_area_renov	-0.599900	-0.032495	0.389855	0.570530	0.757571	0.149744	0.28509
lot_area_renov	-0.089604	-0.000050	0.029400	0.078627	0.180312	0.706812	-0.01012
Number of schools nearby	-0.004821	-0.004071	0.003397	0.002180	0.002370	-0.012671	-0.00757
Distance from the airport	-0.004542	0.011457	-0.006157	0.009206	0.002511	0.003291	0.01656
Price	-0.773114	-0.027919	0.308460	0.531735	0.712169	0.081992	0.26273

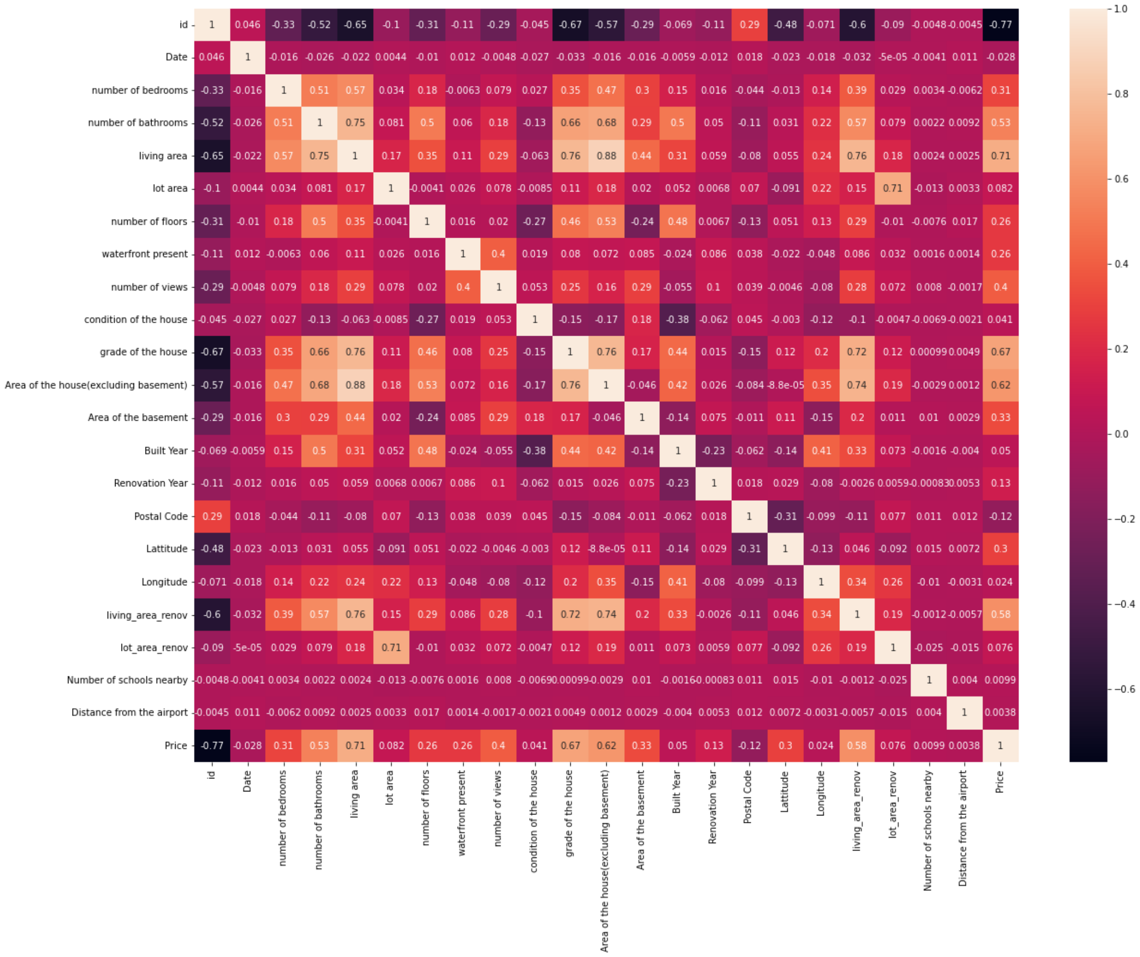
23 rows × 23 columns

In [23]:

```
plt.figure(figsize=(20,15))
sns.heatmap(df.corr(),annot=True)
```

Out[23]:

```
<AxesSubplot:>
```

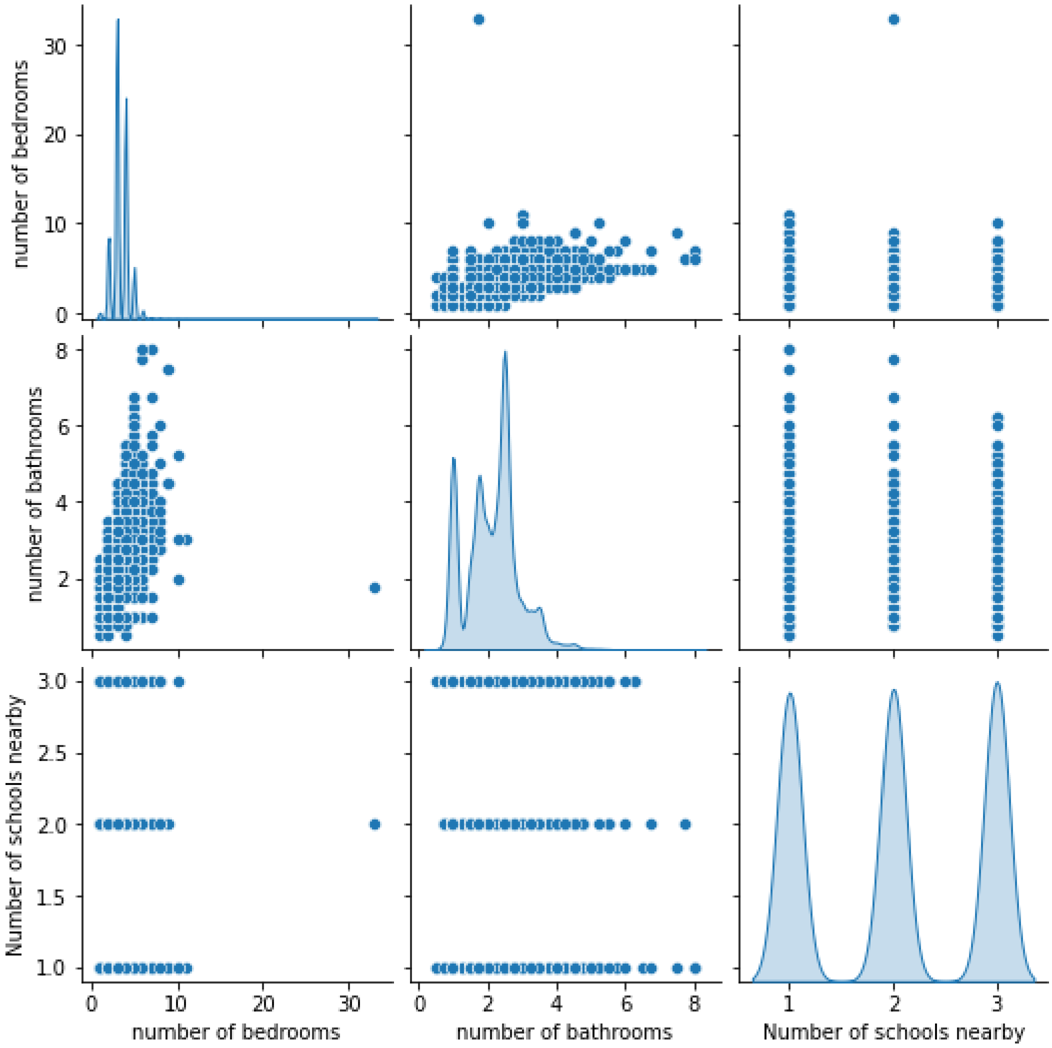


In [32]:

```
sns.pairplot(
    df,
    x_vars=["number of bedrooms", "number of bathrooms", "Number of schools nearby"],
    y_vars=["number of bedrooms", "number of bathrooms", "Number of schools nearby"],
    kind='scatter',
    diag_kind='kde'
)
```

Out[32]:

```
<seaborn.axisgrid.PairGrid at 0x1b1bcf16c10>
```

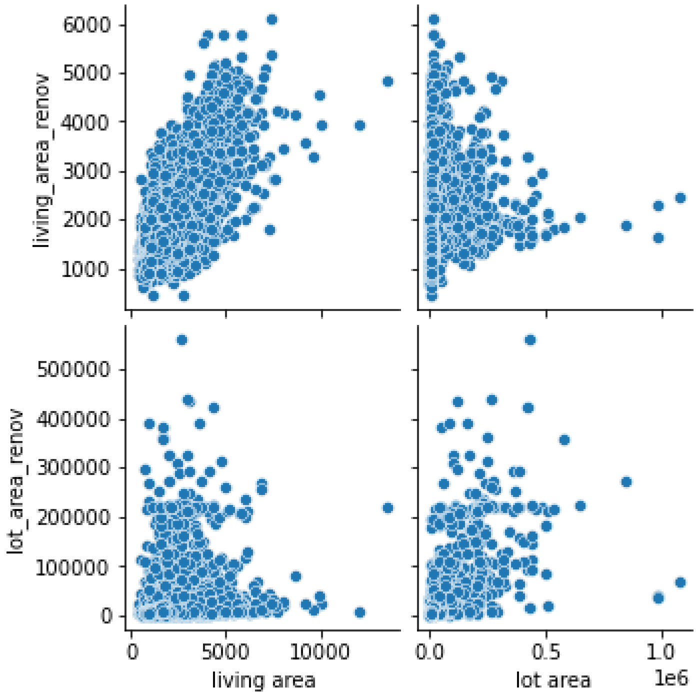


In [39]:

```
sns.pairplot(
    df,
    x_vars=["living area", "lot area"],
    y_vars=["living_area_renov", "lot_area_renov"],
    diag_kind='kde'
)
```

Out[39]:

```
<seaborn.axisgrid.PairGrid at 0x1b1be9d9040>
```



Descriptive statistics on the dataset

In [25]:

```
df.describe()
```

Out[25]:

mber of views	condition of the house	...	Built Year	Renovation Year	Postal Code	Lattitude	Longit
.000000	14620.000000	...	14620.000000	14620.000000	14620.000000	14620.000000	14620.000
.233105	3.430506	...	1970.926402	90.924008	122033.062244	52.792848	-114.404
.766259	0.664151	...	29.493625	416.216661	19.082418	0.137522	0.141
.000000	1.000000	...	1900.000000	0.000000	122003.000000	52.385900	-114.709
.000000	3.000000	...	1951.000000	0.000000	122017.000000	52.707600	-114.519
.000000	3.000000	...	1975.000000	0.000000	122032.000000	52.806400	-114.421
.000000	4.000000	...	1997.000000	0.000000	122048.000000	52.908900	-114.315
.000000	5.000000	...	2015.000000	2015.000000	122072.000000	53.007600	-113.505

Handling the Missing values

In [26]:

```
df.isnull().any()
```

Out[26]:

```
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
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 [27]:

```
df.isnull().sum() #no null values in the dataset
```

Out[27]:

```
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
```