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

```
df = pd.read_csv('/content/sample_data/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	Renovatio Yea
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5	...	1921	
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5	...	1909	
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3	...	1939	
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3	...	2001	
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	4	...	1929	

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	Renov
14615	6762830250	42734	2	1.5	1556	20000	1.0	0	0	4	...	1957	
14616	6762830339	42734	3	2.0	1680	7000	1.5	0	0	4	...	1968	
14617	6762830618	42734	2	1.0	1070	6120	1.0	0	0	3	...	1962	
14618	6762830709	42734	4	1.0	1030	6621	1.0	0	0	4	...	1955	
14619	6762831463	42734	3	1.0	900	4770	1.0	0	0	3	...	1969	

5 rows × 23 columns

```
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
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
```

```
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
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
```

```
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
```

```
df.describe()
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	
count	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.000000	1.462000e+04	14620.000000	14620.000000	1
mean	6.762821e+09	42604.538646	3.379343	2.129583	2098.262996	1.509328e+04	1.502360	0.007661	
std	6.237575e+03	67.347991	0.938719	0.769934	928.275721	3.791962e+04	0.540239	0.087193	
min	6.762810e+09	42491.000000	1.000000	0.500000	370.000000	5.200000e+02	1.000000	0.000000	
25%	6.762815e+09	42546.000000	3.000000	1.750000	1440.000000	5.010750e+03	1.000000	0.000000	
50%	6.762821e+09	42600.000000	3.000000	2.250000	1930.000000	7.620000e+03	1.500000	0.000000	
75%	6.762826e+09	42662.000000	4.000000	2.500000	2570.000000	1.080000e+04	2.000000	0.000000	
max	6.762832e+09	42734.000000	33.000000	8.000000	13540.000000	1.074218e+06	3.500000	1.000000	

8 rows × 23 columns

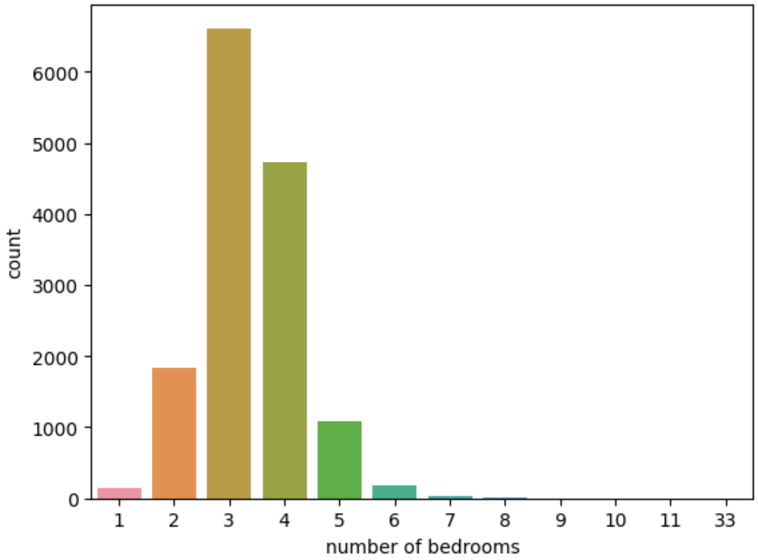
```
df.describe(include='all')
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	
count	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.000000	1.462000e+04	14620.000000	14620.000000	1
mean	6.762821e+09	42604.538646	3.379343	2.129583	2098.262996	1.509328e+04	1.502360	0.007661	
std	6.237575e+03	67.347991	0.938719	0.769934	928.275721	3.791962e+04	0.540239	0.087193	
min	6.762810e+09	42491.000000	1.000000	0.500000	370.000000	5.200000e+02	1.000000	0.000000	
25%	6.762815e+09	42546.000000	3.000000	1.750000	1440.000000	5.010750e+03	1.000000	0.000000	
50%	6.762821e+09	42600.000000	3.000000	2.250000	1930.000000	7.620000e+03	1.500000	0.000000	
75%	6.762826e+09	42662.000000	4.000000	2.500000	2570.000000	1.080000e+04	2.000000	0.000000	
max	6.762832e+09	42734.000000	33.000000	8.000000	13540.000000	1.074218e+06	3.500000	1.000000	

8 rows × 23 columns

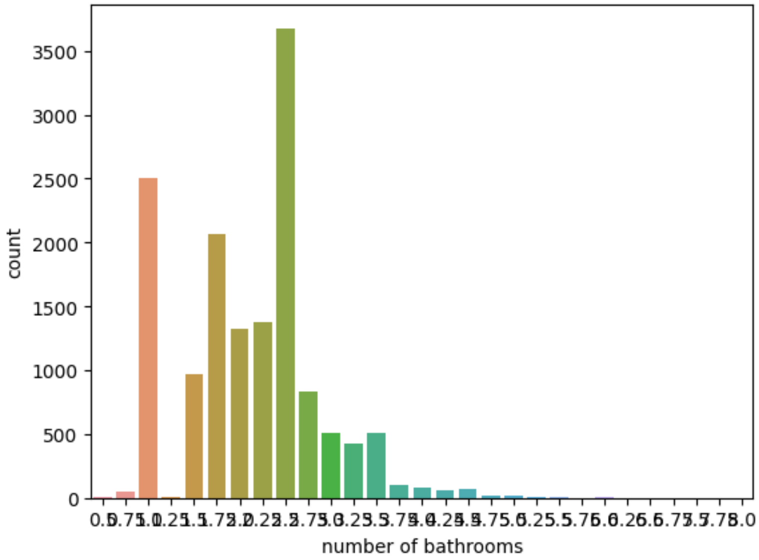
```
sns.countplot(data=df,x='number of bedrooms')
```

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



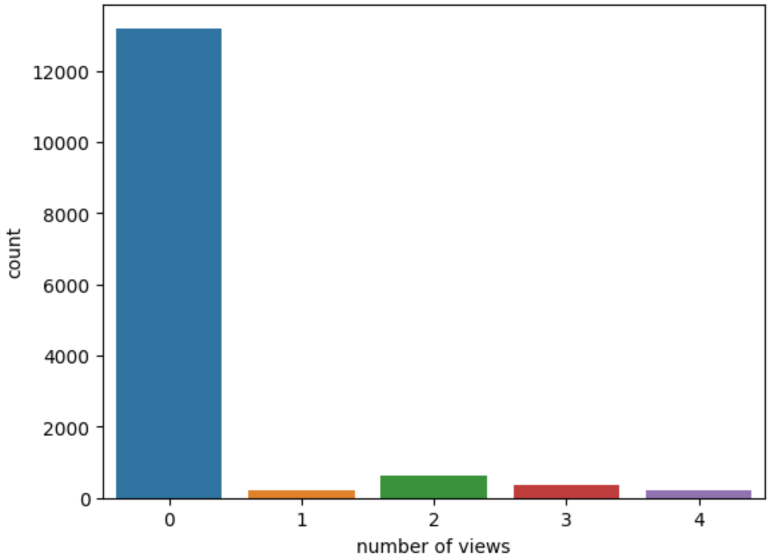
```
sns.countplot(data=df,x='number of bathrooms')
```

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



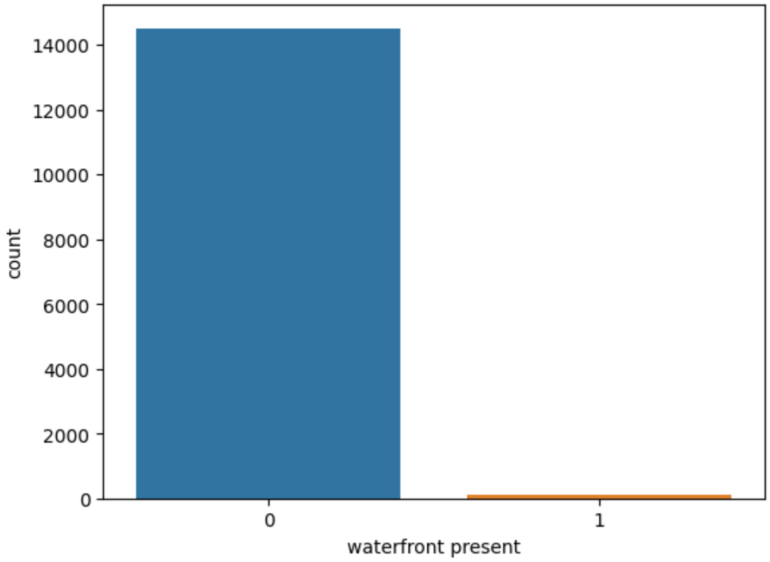
```
sns.countplot(data=df,x='number of views')
```

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



```
sns.countplot(data=df,x='waterfront present')
```

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

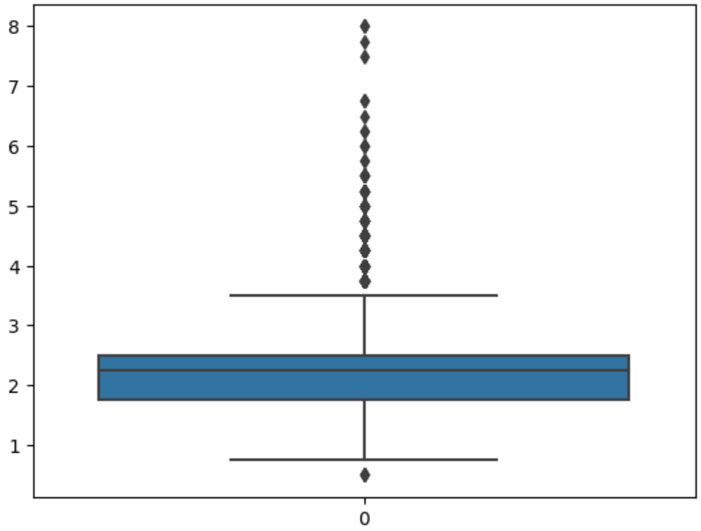


```
sns.boxplot(df['number of bedrooms'])
```

```
<Axes: >
```

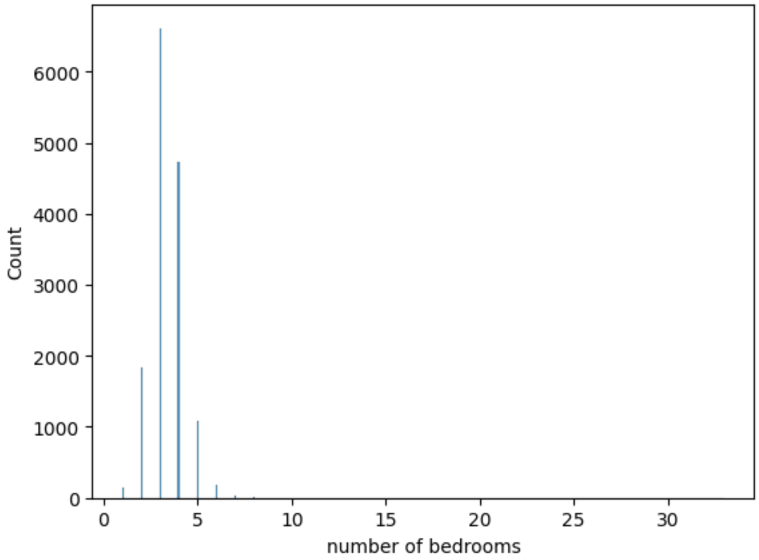
```
sns.boxplot(df['number of bathrooms'])
```

```
<Axes: >
```



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

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

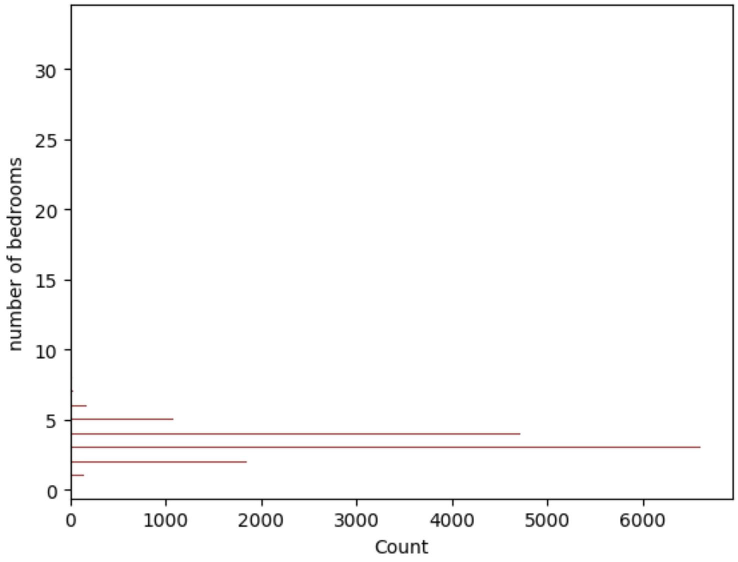


```
sns.histplot(df['number of bathrooms'])
```

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

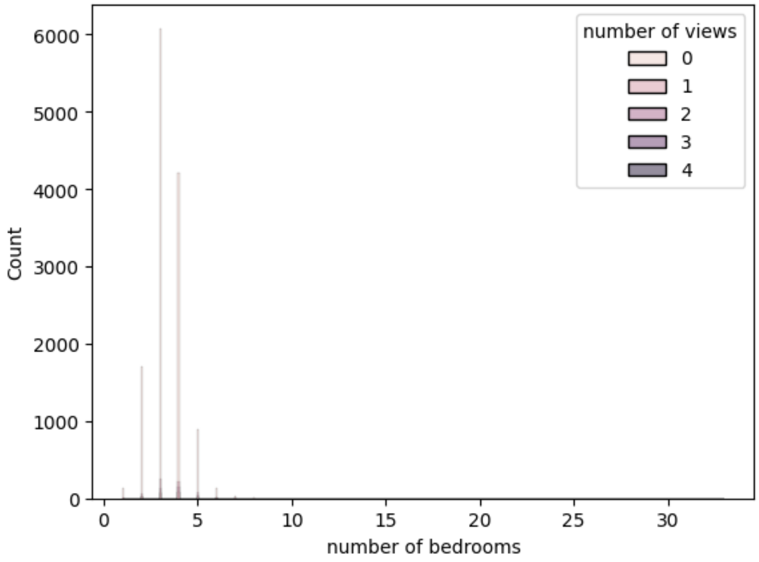
```
sns.histplot(y='number of bedrooms',data=df,color='darkred')
```

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



```
sns.histplot(x='number of bedrooms',data=df,hue=df['number of views'])
```

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



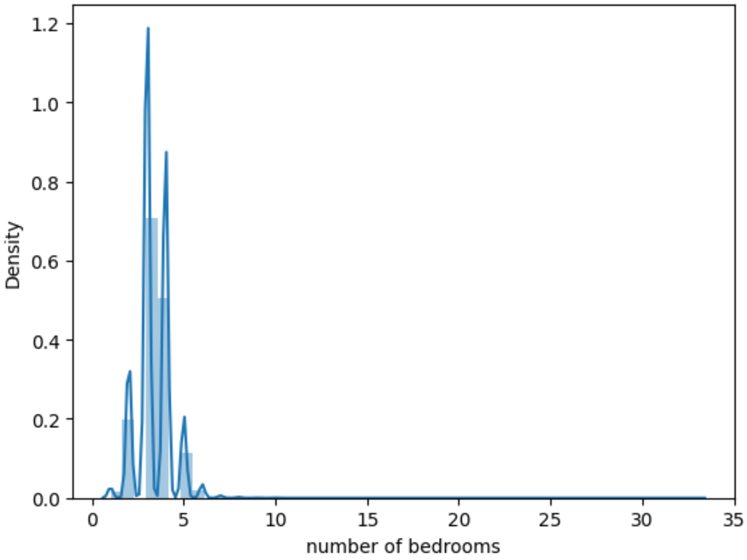
```
sns.histplot(df['waterfront present'])
```

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

14000
12000
10000
8000
nt

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

```
<ipython-input-147-1d9da7a0f54b>: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(df['number of bedrooms'])
<Axes: xlabel='number of bedrooms', ylabel='Density'>
```



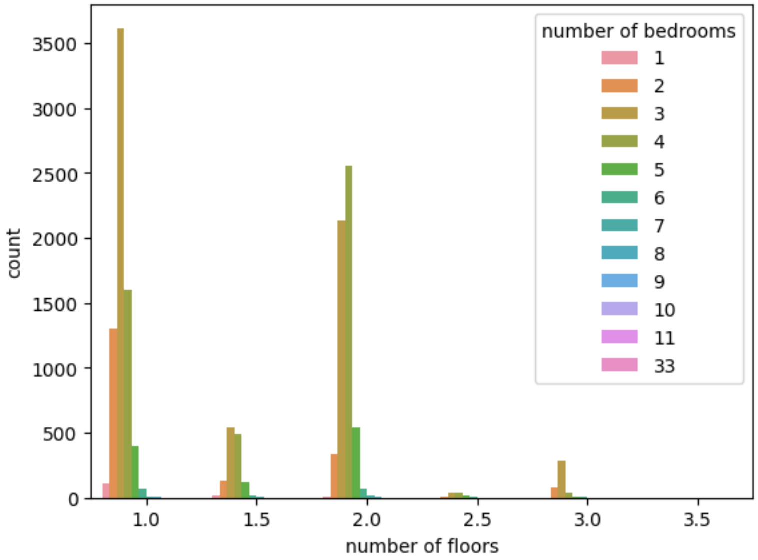
```
sns.distplot(df["number of bedrooms"],hist=False,color='pink')
```

```
<ipython-input-148-5fbd2920968f>: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 `kdeplot` (an axes-level function for kernel density plots).

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

```
# biveriate plot
```

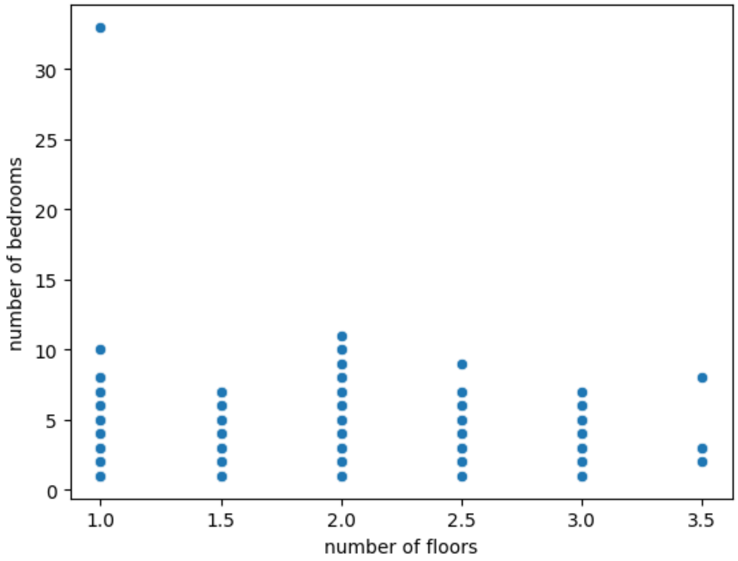
```
sns.countplot(data=df,x='number of floors',hue='number of bedrooms')
```

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



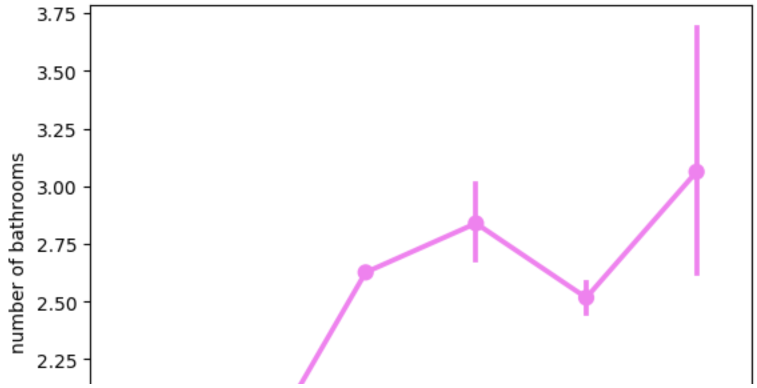
```
sns.scatterplot(data=df,x='number of floors',y='number of bedrooms')
```

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



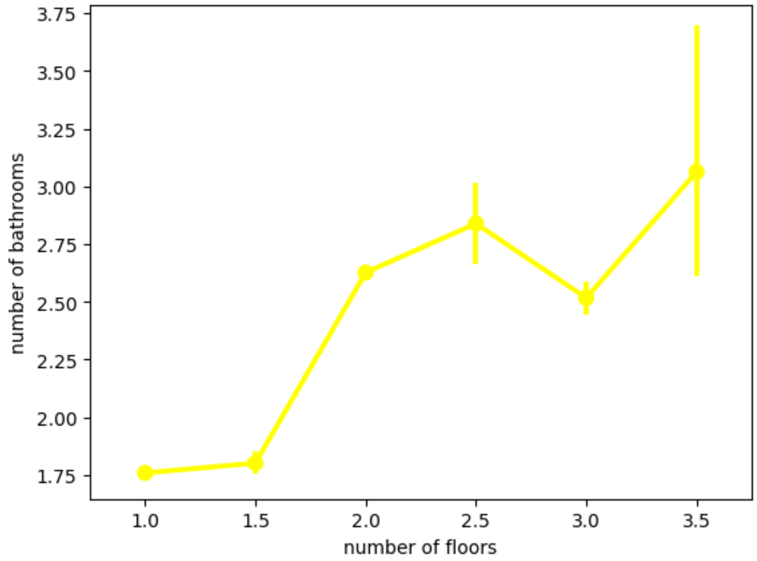
```
sns.pointplot(x='number of floors',y='number of bathrooms',data=df,color='violet')
```

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

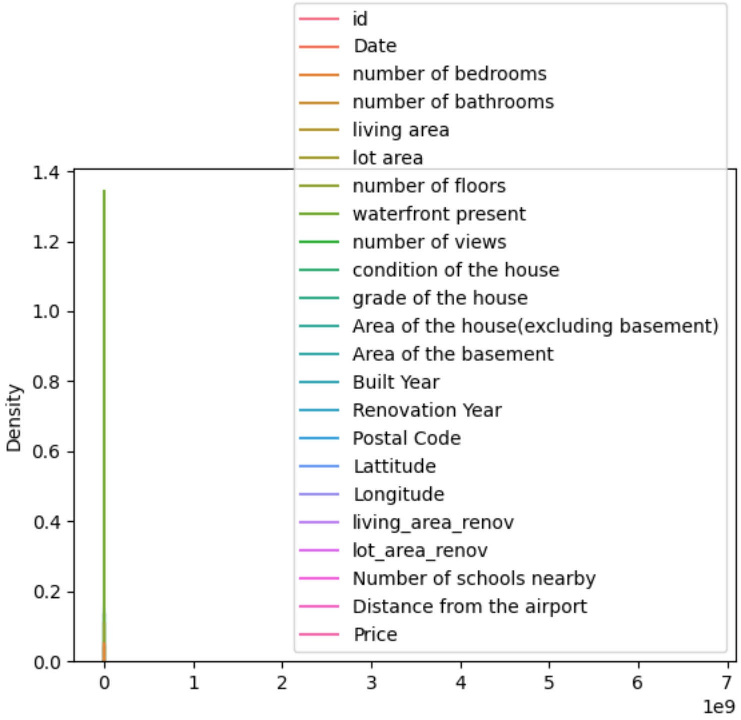


```
sns.pointplot(x='number of floors',y='number of bathrooms',data=df,color='yellow')
```

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

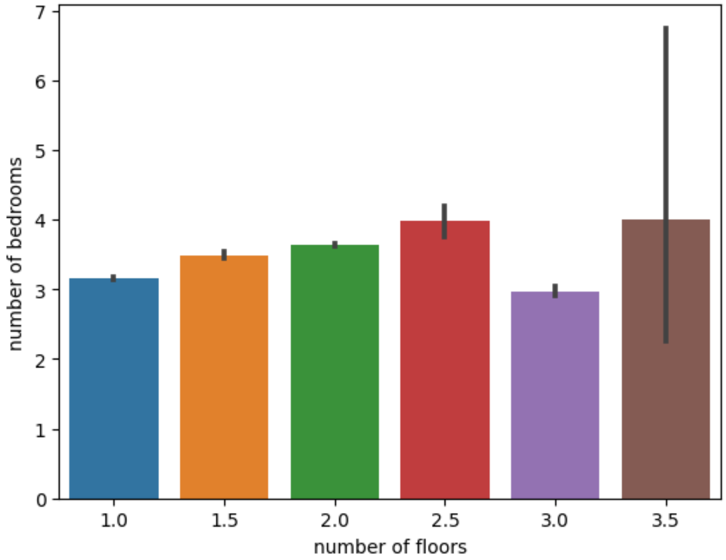


```
sns.kdeplot(df);
```



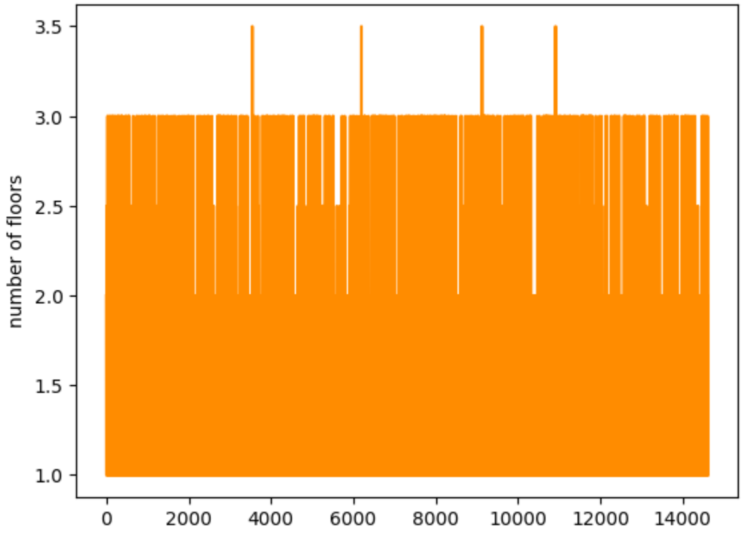
```
sns.barplot(x='number of floors',y='number of bedrooms',data=df)
```

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



```
sns.lineplot(df['number of floors'],color='darkorange')
```

```
<Axes: ylabel='number of floors'>
```



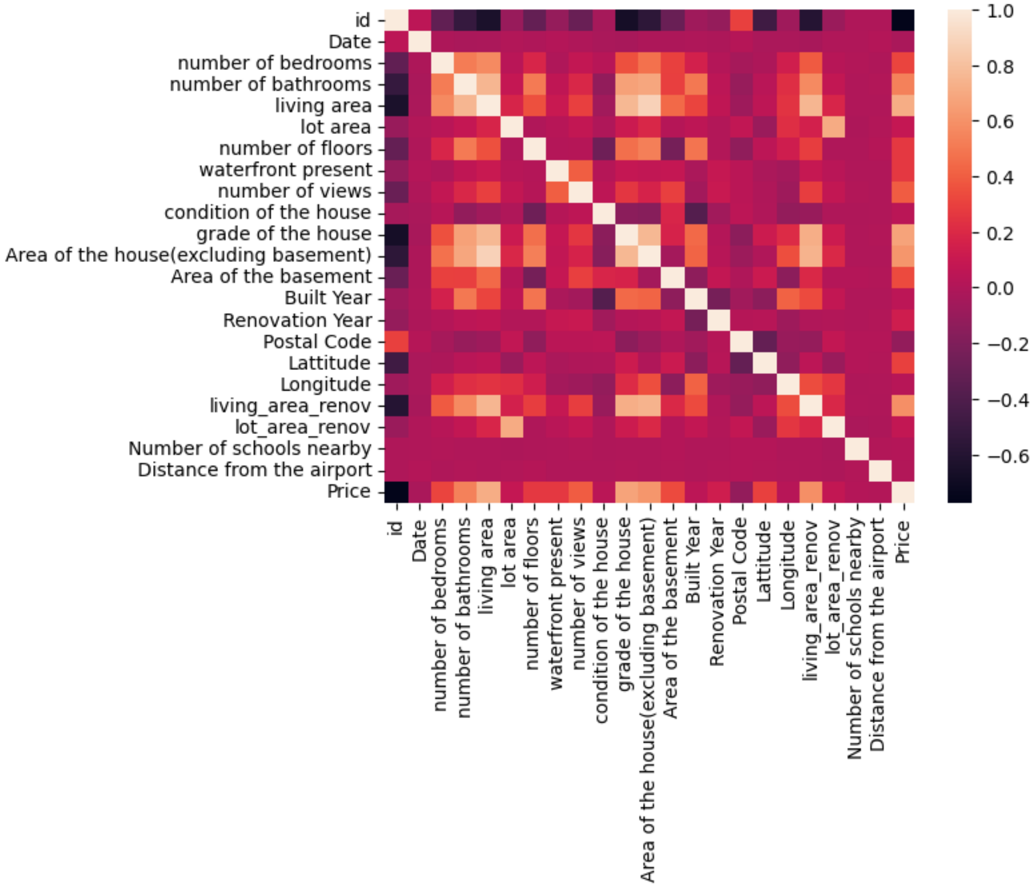
```
sns.regplot(x='number of floors',y='number of bedrooms',data=df)
```

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

```
# Multivariate
```

```
sns.heatmap(df.corr())
```

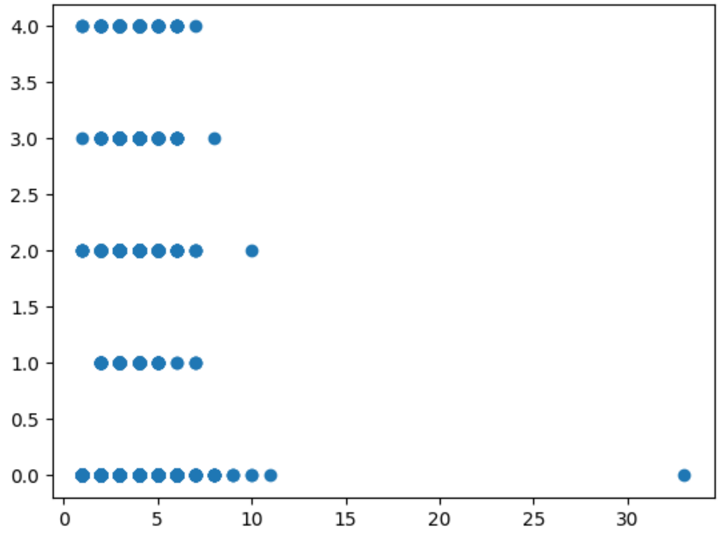
```
<Axes: >
```



```
#matplot lib
```

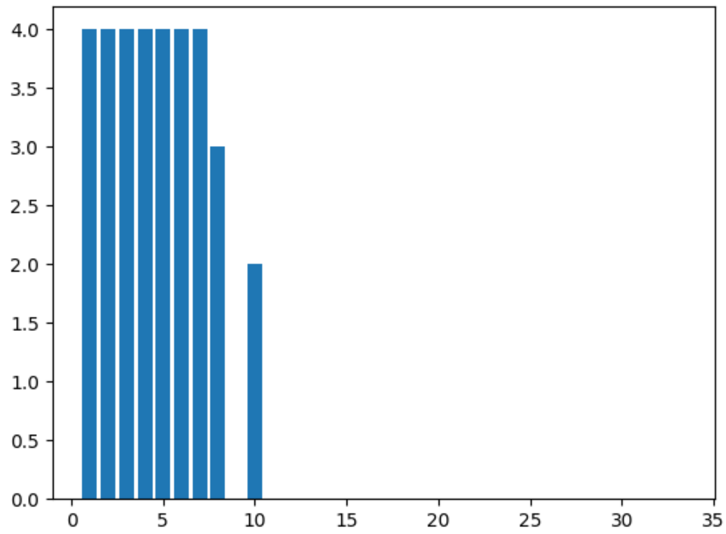
```
plt.scatter(df['number of bedrooms'],df['number of views'])
```

```
<matplotlib.collections.PathCollection at 0x7f641d6dc9d0>
```



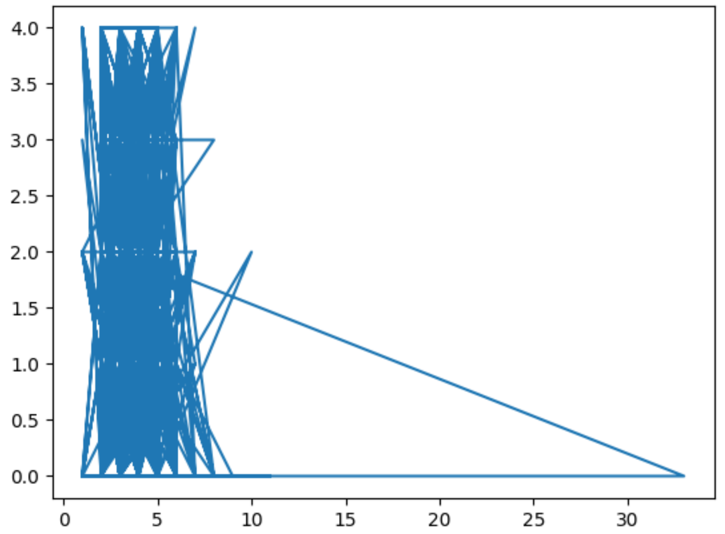
```
plt.bar(df['number of bedrooms'],df['number of views'])
```

```
<BarContainer object of 14620 artists>
```



```
plt.plot(df['number of bedrooms'],df['number of views'])
```

```
[<matplotlib.lines.Line2D at 0x7f641d6deb60>]
```



```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

● ×