Assignment 3

Assignment 3

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

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

```
hr.head()
```

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

```
hr.tail(10)
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	numbe of view
14610	6762828349	42734	4	2.75	1810	7350	1.0	0	
14611	6762828783	42734	3	1.75	1350	7686	1.0	0	
14612	6762828856	42734	3	1.00	1180	5350	1.5	0	
14613	6762829600	42734	3	1.00	1400	10425	1.0	0	
14614	6762829669	42734	3	1.75	1590	7931	1.0	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	

10 rows × 23 columns

```
hr.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
```

```
hr.isnull()
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condi of h
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	
14615	False	False	False	False	False	False	False	False	False	
14616	False	False	False	False	False	False	False	False	False	
14617	False	False	False	False	False	False	False	False	False	
14618	False	False	False	False	False	False	False	False	False	
14619	False	False	False	False	False	False	False	False	False	

14620 rows × 23 columns

```
hr.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
```

```
hr.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
```

```
hr.describe()
```

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

8 rows × 23 columns

```
hr.head()
```

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

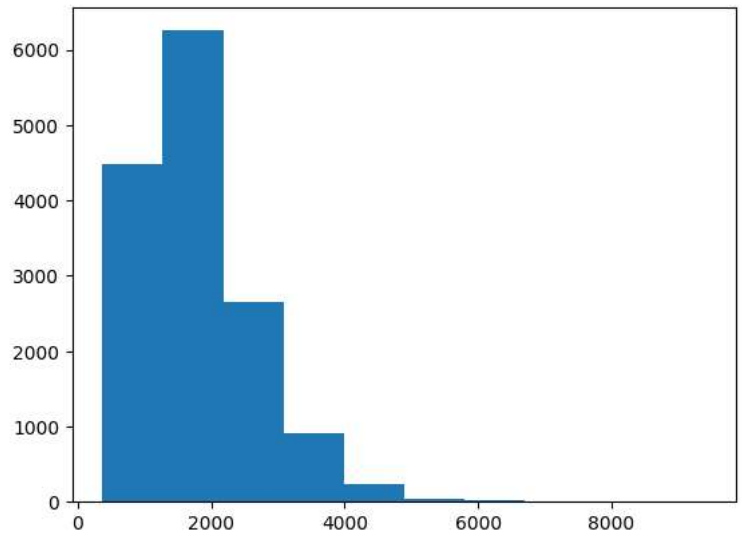
```
print(hr.describe())
```

```
                 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%                    73.000000  6.450000e+05
```

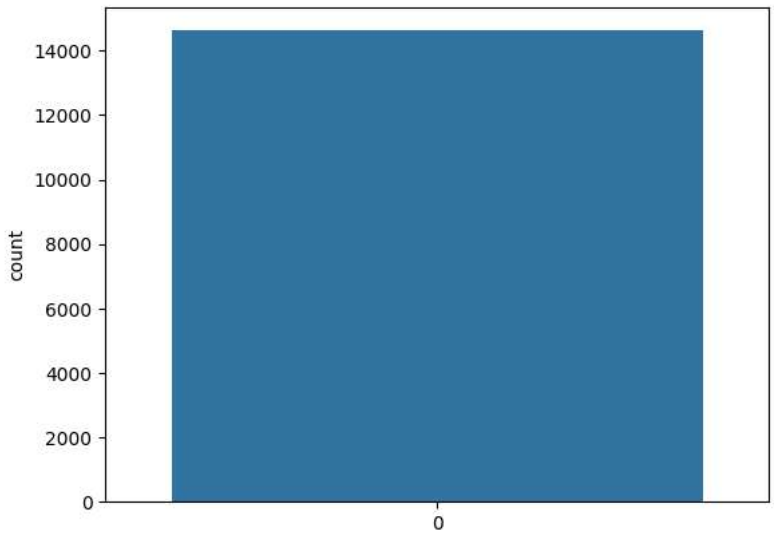
```
plt.hist(hr['Area of the house(excluding basement)'])
```

```
(array([4.479e+03, 6.255e+03, 2.653e+03, 9.190e+02, 2.440e+02, 4.600e+01,
        1.800e+01, 1.000e+00, 2.000e+00, 3.000e+00]),
 array([ 370., 1274., 2178., 3082., 3986., 4890., 5794., 6698., 7602.,
        8506., 9410.]),
 <BarContainer object of 10 artists>)
```



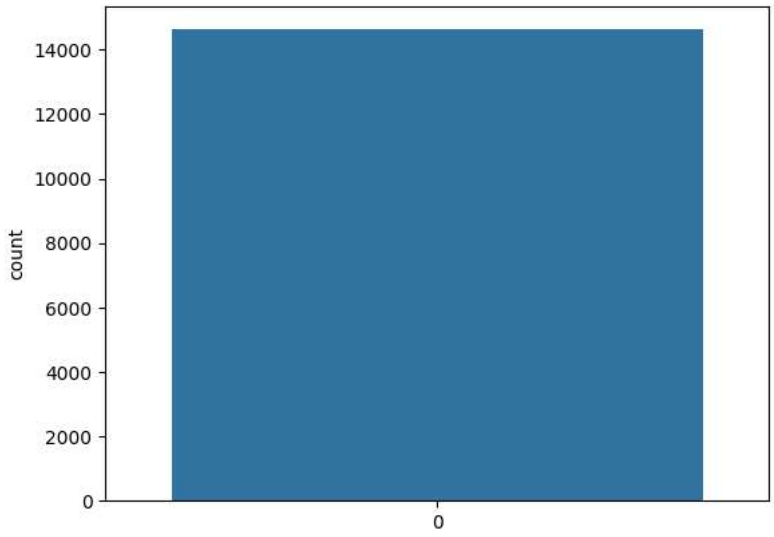

```
sns.countplot(hr['lot area'])
```

```
<Axes: ylabel='count'>
```




```
sns.countplot(hr['number of bedrooms'])
```

```
<Axes: ylabel='count'>
```

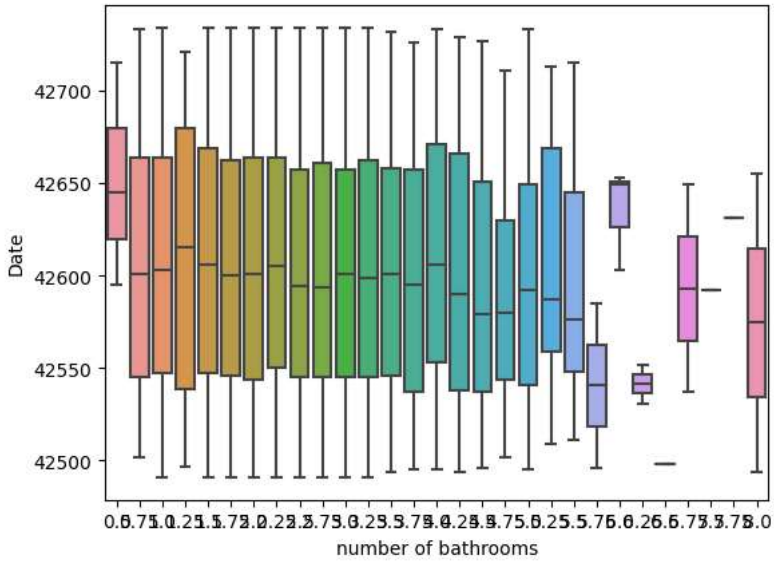



```
sns.boxplot(x=hr['id'])
```

```
<Axes: xlabel='id'>
```

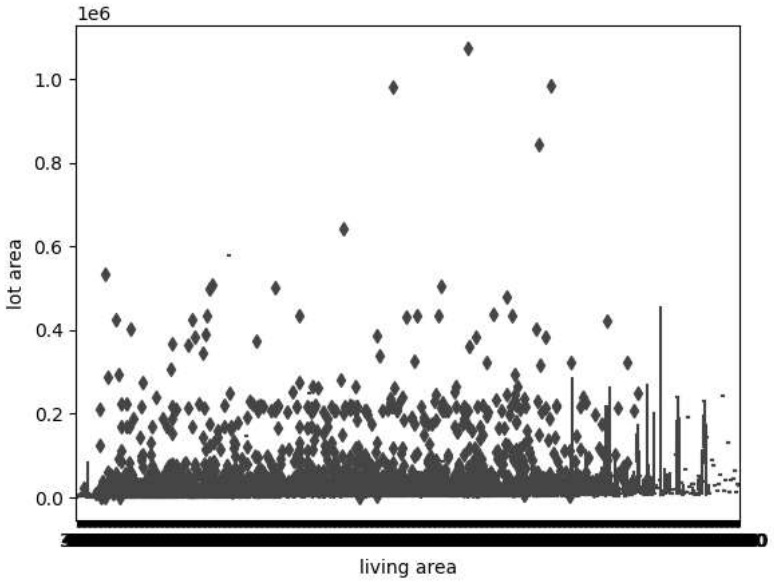
```
sns.boxplot(x=hr['number of bathrooms'],y=hr['Date'])
```

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




```
sns.boxplot(x=hr['living area'],y=hr['lot area'])
```

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

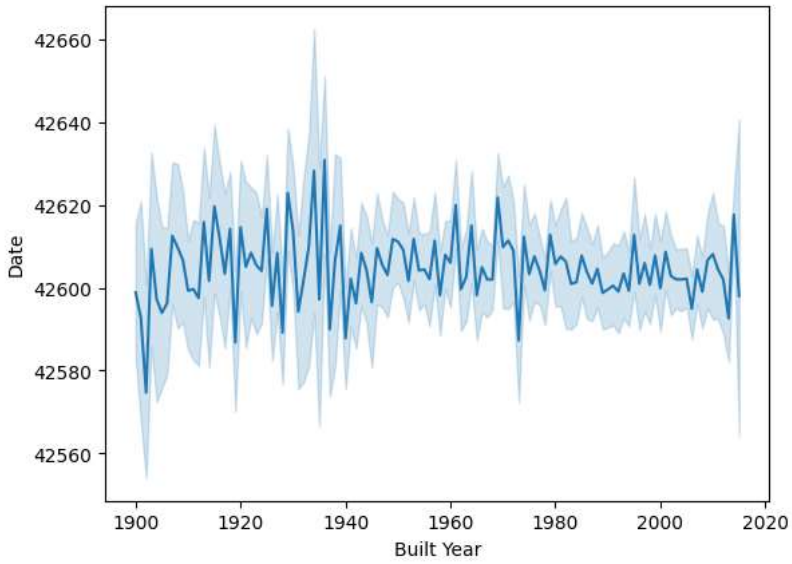



```
sns.lineplot(x=hr['Built Year'],y=hr['Date'])
```

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


```
sns.lineplot(x=hr['Built Year'],y=hr['Date'])
```

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




```
(hr[['Price','number of views','grade of the house','condition of the house']])
```

	Price	number of views	grade of the house	condition of the house
0	2380000	4	10	5
1	1400000	0	8	5
2	1200000	0	8	3
3	838000	0	9	3
4	805000	0	8	4
...	...	...	...	...
14615	221700	0	7	4
14616	219200	0	7	4
14617	209000	0	6	3
14618	205000	0	6	4
14619	146000	0	6	3

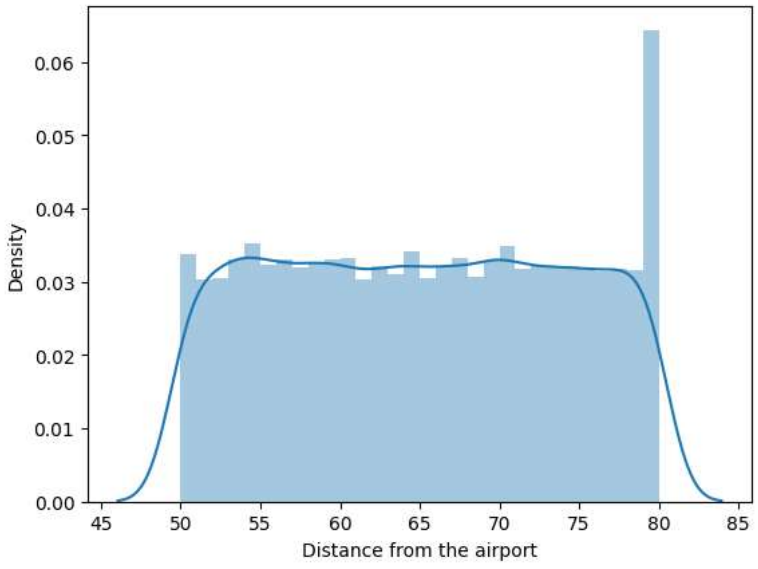
14620 rows × 4 columns

```
plt.hist(hr['number of bedrooms'],bins=50)
```

```
(array([1.360e+02, 1.844e+03, 0.000e+00, 6.612e+03, 4.724e+03, 0.000e+00,
        1.079e+03, 1.760e+02, 0.000e+00, 3.000e+01, 1.100e+01, 0.000e+00,
        3.000e+00, 0.000e+00, 3.000e+00, 1.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 1.000e+00]),
 array([ 1.  ,  1.64,  2.28,  2.92,  3.56,  4.2 ,  4.84,  5.48,  6.12,
         6.76,  7.4 ,  8.04,  8.68,  9.32,  9.96, 10.6 , 11.24, 11.88,
        12.52, 13.16, 13.8 , 14.44, 15.08, 15.72, 16.36, 17.  , 17.64,
        18.28, 18.92, 19.56, 20.2 , 20.84, 21.48, 22.12, 22.76, 23.4 ,
        24.04, 24.68, 25.32, 25.96, 26.6 , 27.24, 27.88, 28.52, 29.16,
        29.8 , 30.44, 31.08, 31.72, 32.36, 33.  ]),
 <BarContainer object of 50 artists>)
```

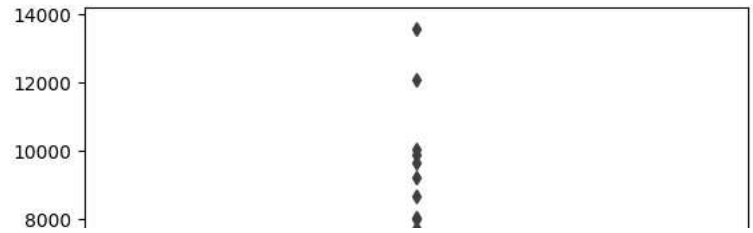

```
sns.distplot(hr['Distance from the airport'],bins=30)
```

```
<ipython-input-26-9951cfa0f999>: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(hr['Distance from the airport'],bins=30)
<Axes: xlabel='Distance from the airport', ylabel='Density'>
```



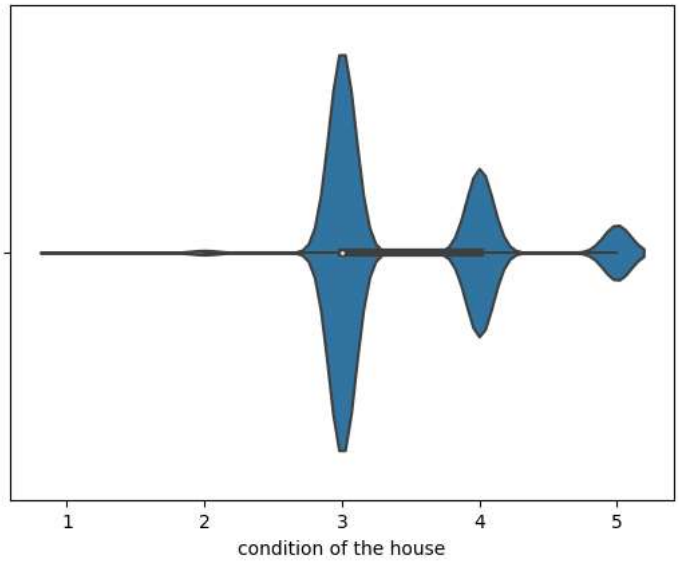

```
sns.boxplot(hr['living area'])
```

```
<Axes: >
```



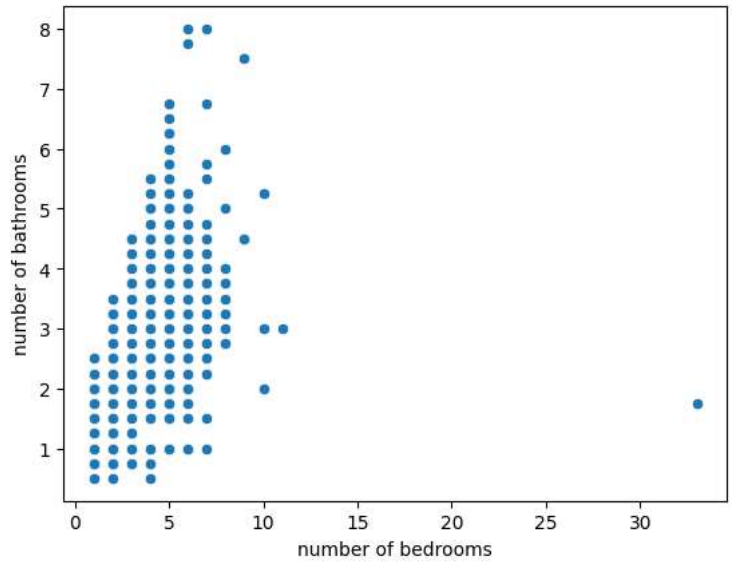

```
sns.violinplot(x=hr['condition of the house'])
```

```
<Axes: xlabel='condition of the house'>
```



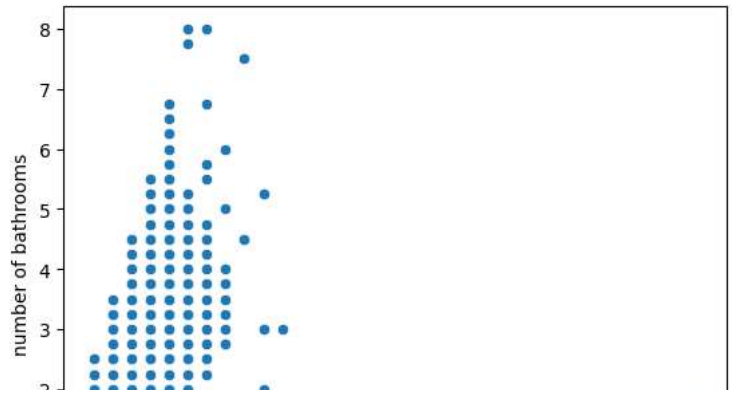

```
sns.scatterplot(x=hr['number of bedrooms'],y=hr['number of bathrooms'])
```

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



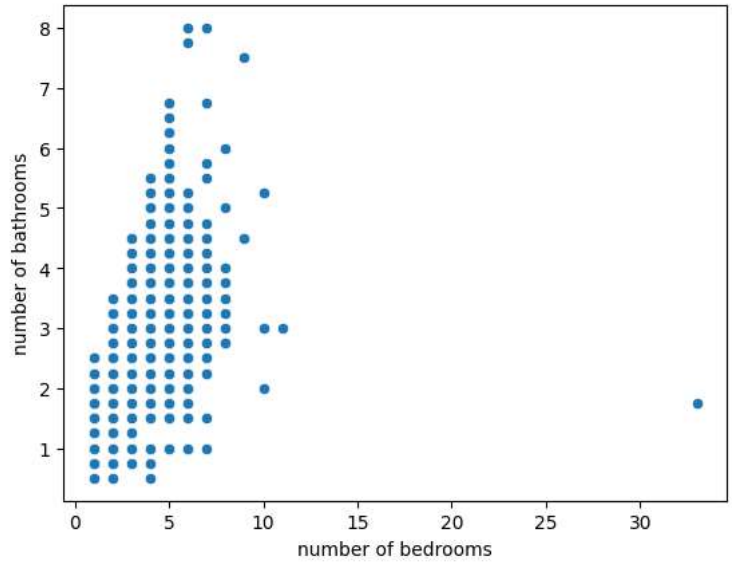

```
sns.scatterplot(x=hr['number of bedrooms'],y=hr['number of bathrooms'])
```

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



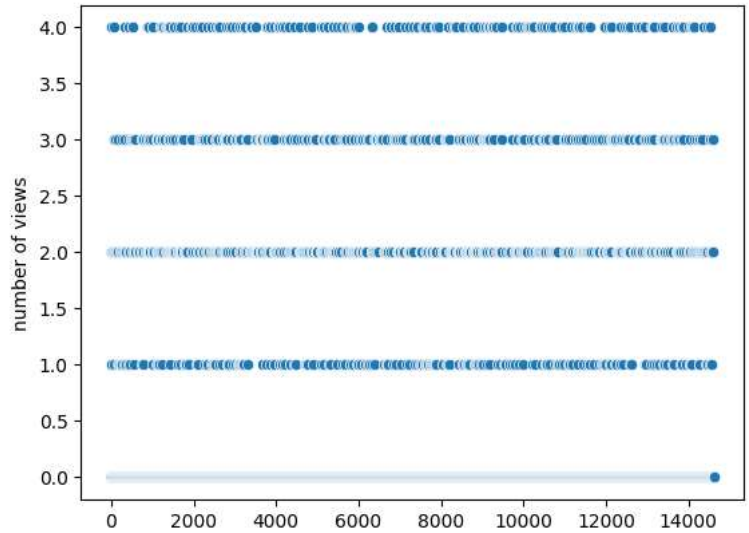

```
sns.scatterplot(x=hr['number of bedrooms'],y=hr['number of bathrooms'])
```

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



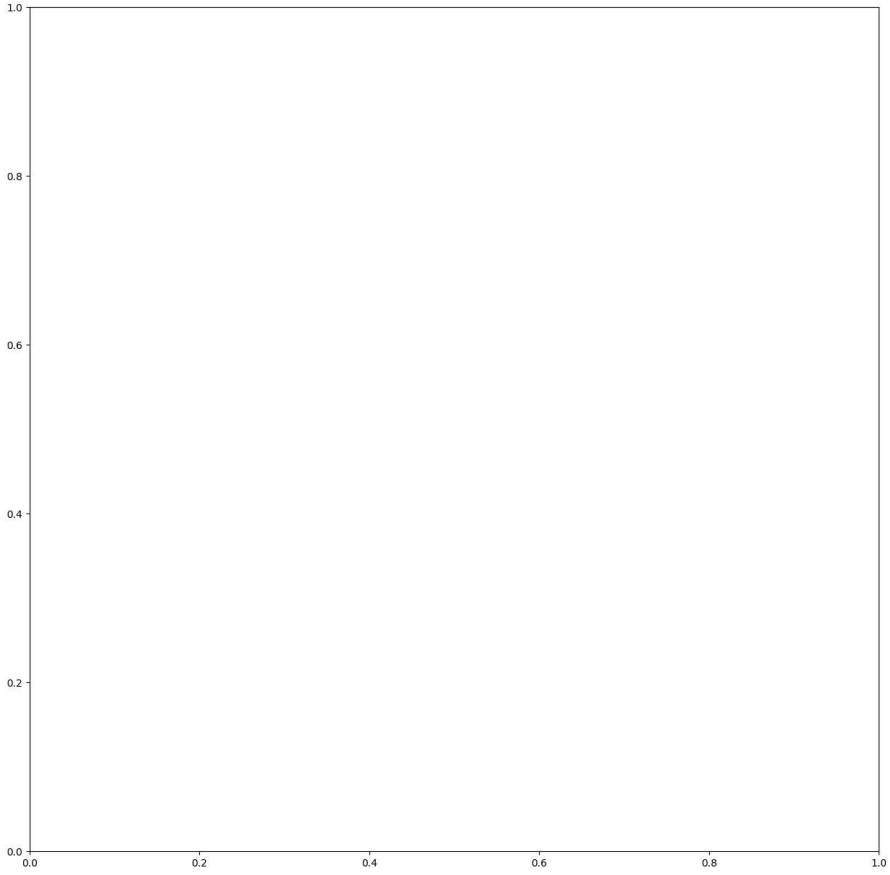

```
sns.scatterplot(hr['number of views'])
```

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



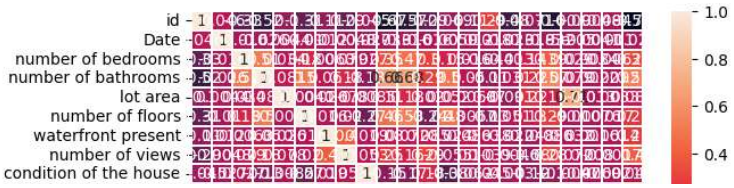

```
plt.subplots(figsize=(15,15))
```

```
(<Figure size 1500x1500 with 1 Axes>, <Axes: >)
```




```
sns.heatmap(hr.drop(['living area'],axis=1).corr(),linewidth=0.3,annot=True)
```

```
<Axes: >
```




```
plt.show()
```

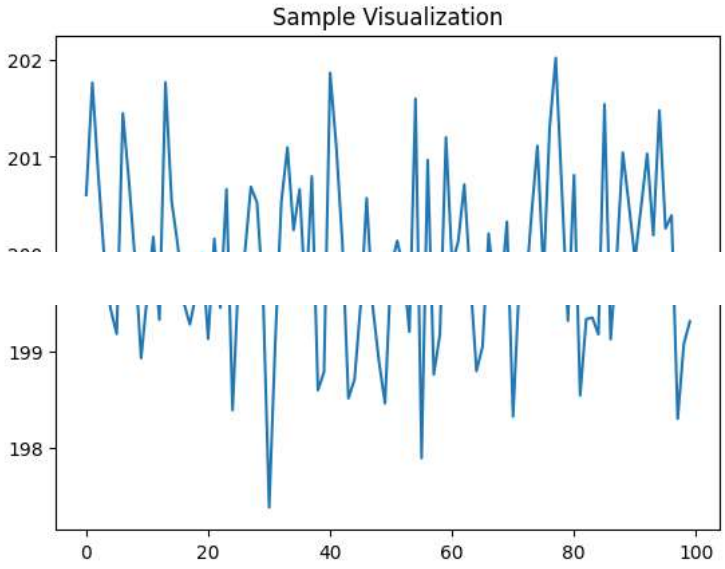

```
print(hr.count())
```

```
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
Lattitude                             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(hr['number of bedrooms'].value_counts())
```

```
3     6612
4     4724
2     1844
5     1079
6      176
1      136
7       30
8       11
9        3
10       3
33       1
11       1
Name: number of bedrooms, dtype: int64
```

```
ys = 200 + np.random.randn(100)
x = [x for x in range(len(ys))]
plt.plot(x, ys, '-')
plt.fill_between(x, ys, 195, where=(ys < 195), facecolor='b', alpha=0.6)
plt.title("Sample Visualization")
plt.show()
```


Sample Visualization
202
201
199
198
0
20
40
60
80
100