AI Assignment - 1

```
import pandas as pd
import matplotlib.pyplot as mtl
import numpy as np
import seaborn as sb
ds=pd.read_csv("/dataset.csv")
print(ds.head())
```

```
---------------------------------------------------------------------------
-----
ModuleNotFoundError                       Traceback (most recent call
last)
<ipython-input-1-34c5ca2a9982> in <module>
----> 1 import pandas as pd
      2 import matplotlib.pyplot as mtl
      3 import numpy as np
      4 import seaborn as sb
      5 ds=pd.read_csv("/dataset.csv")

ModuleNotFoundError: No module named 'pandas'
```

1.Visualizations

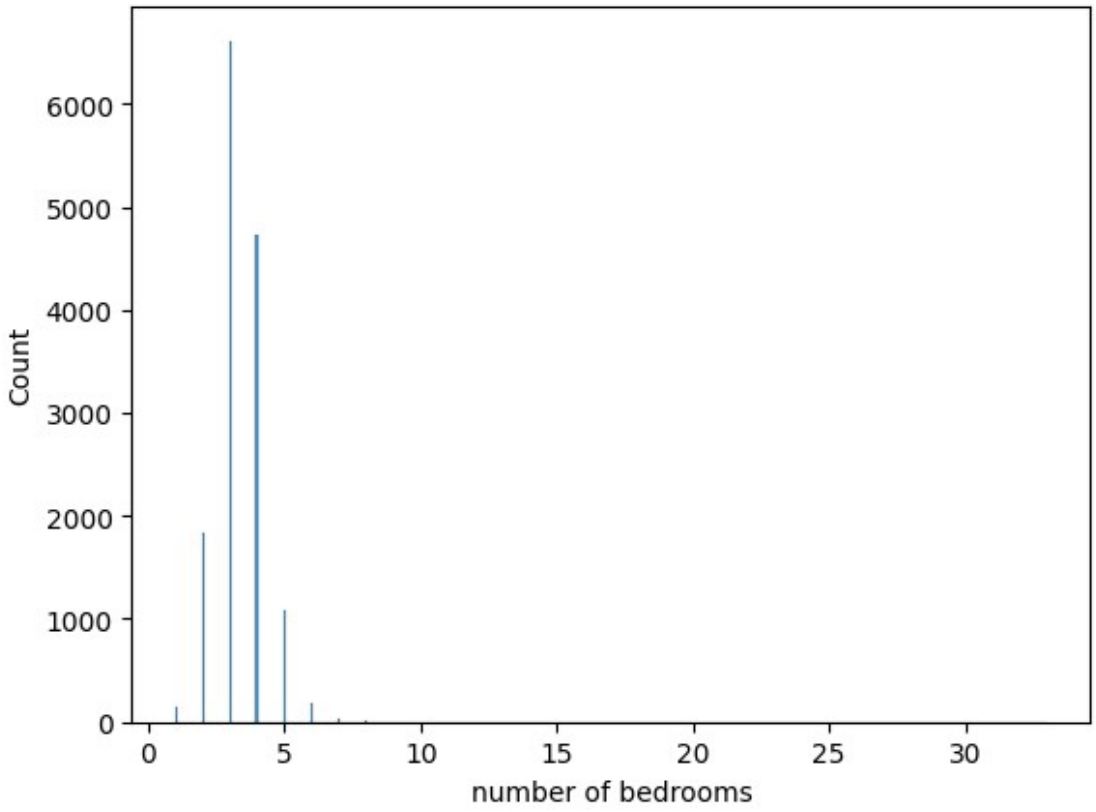
1A. Univariate analysis:

Analize the attributes of the given dataset individually based on the count of values and see the repetition of similar values and see the repeatition of the values in the total data given in the dataset

Analyzing based on the number of bedrooms:

```
sb.histplot(ds['number of bedrooms'])
```

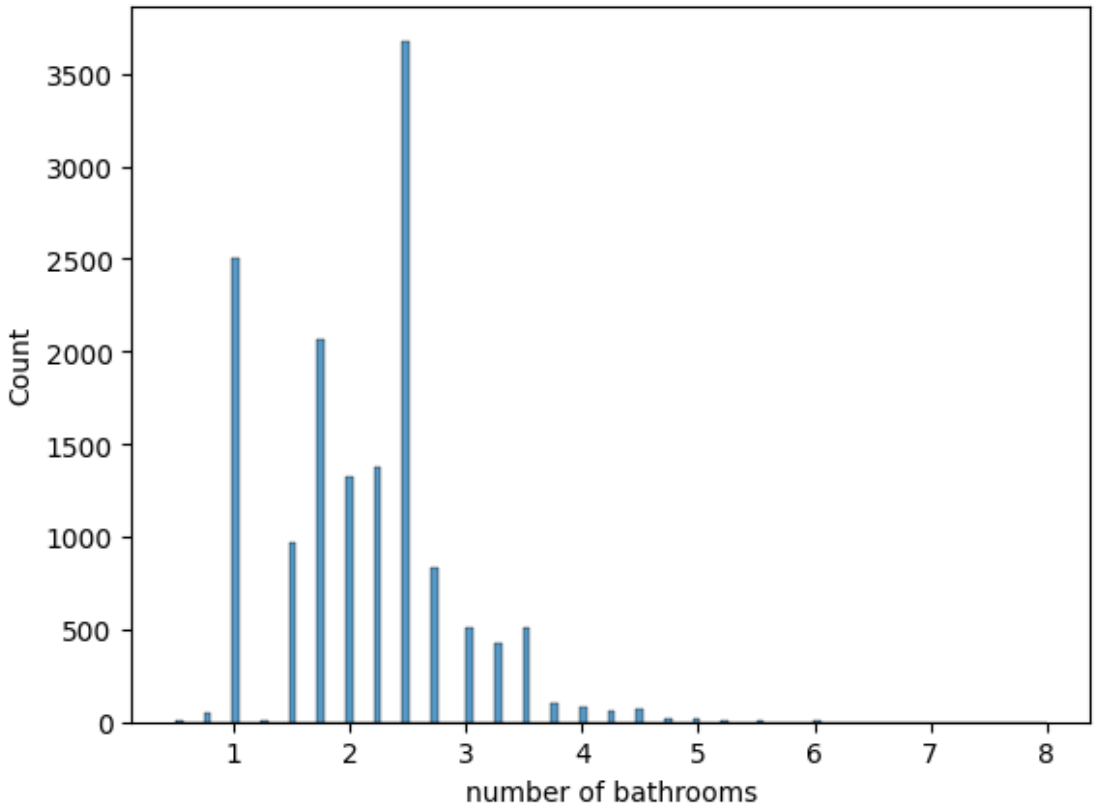
```
<Axes: xlabel='number of bedrooms', ylabel='Count'>
```



Analyzing based on the number of bathrooms:

```
sb.histplot(ds['number of bathrooms'])
```

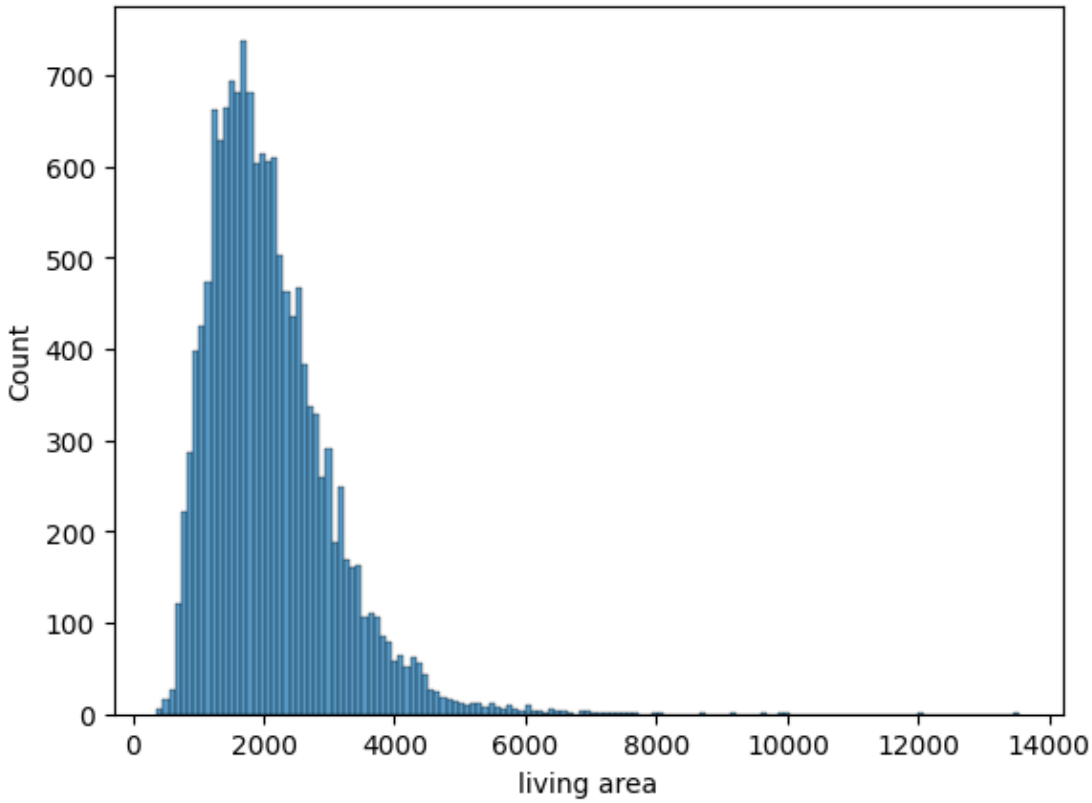
```
<Axes: xlabel='number of bathrooms', ylabel='Count'>
```



Analyzing based on the living area:

```
sb.histplot(ds['living area'])
```

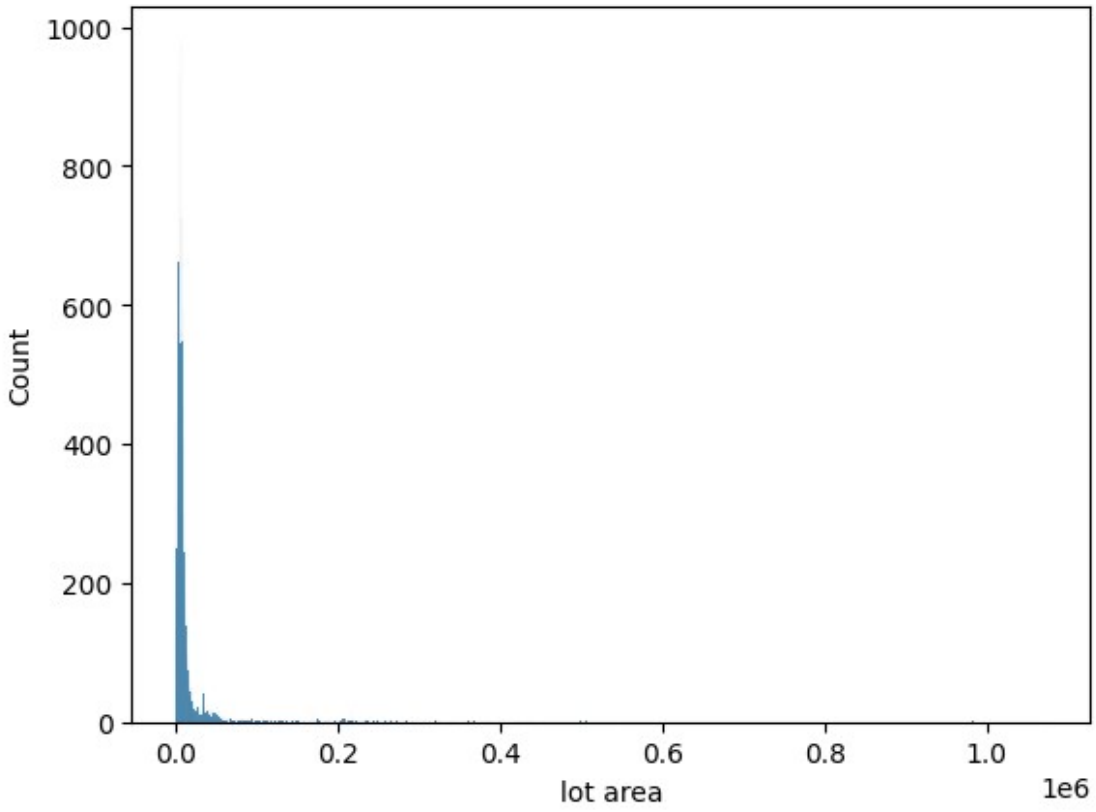
```
<Axes: xlabel='living area', ylabel='Count'>
```



Analyzing based on the lot area:

```
sb.histplot(ds['lot area'])
```

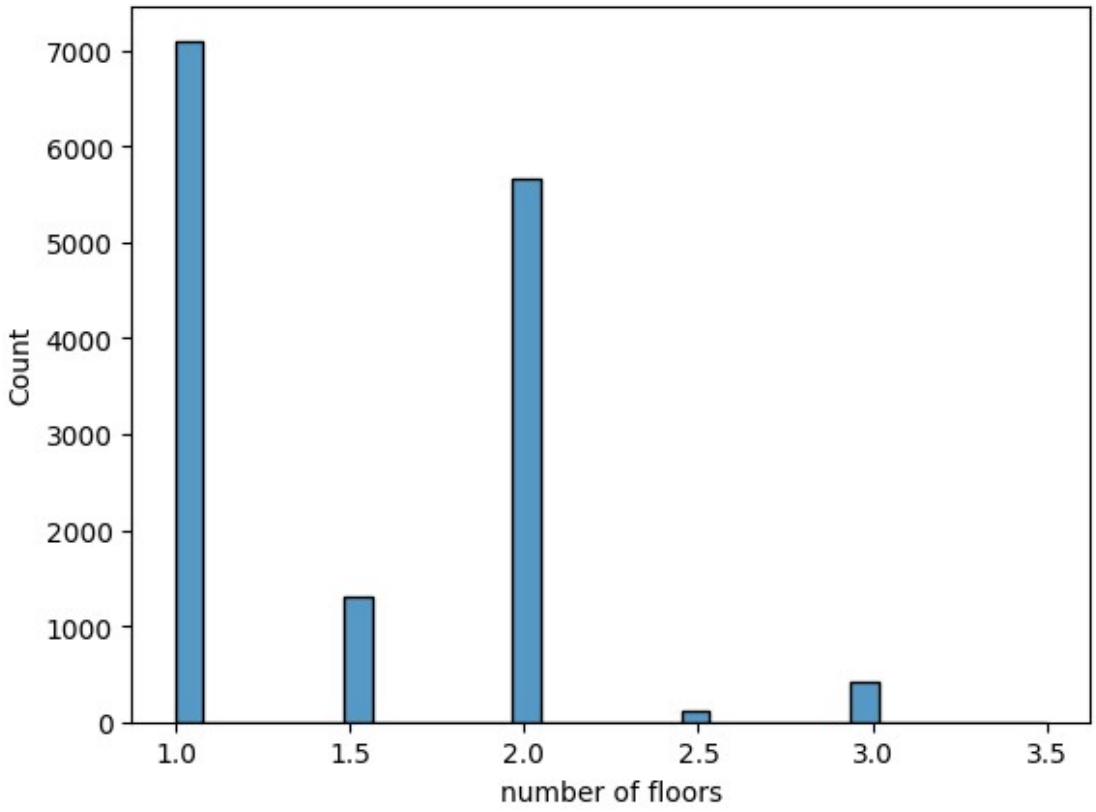
```
<Axes: xlabel='lot area', ylabel='Count'>
```



Analyzing based on the number of floors :

```
sb.histplot(ds['number of floors'])
```

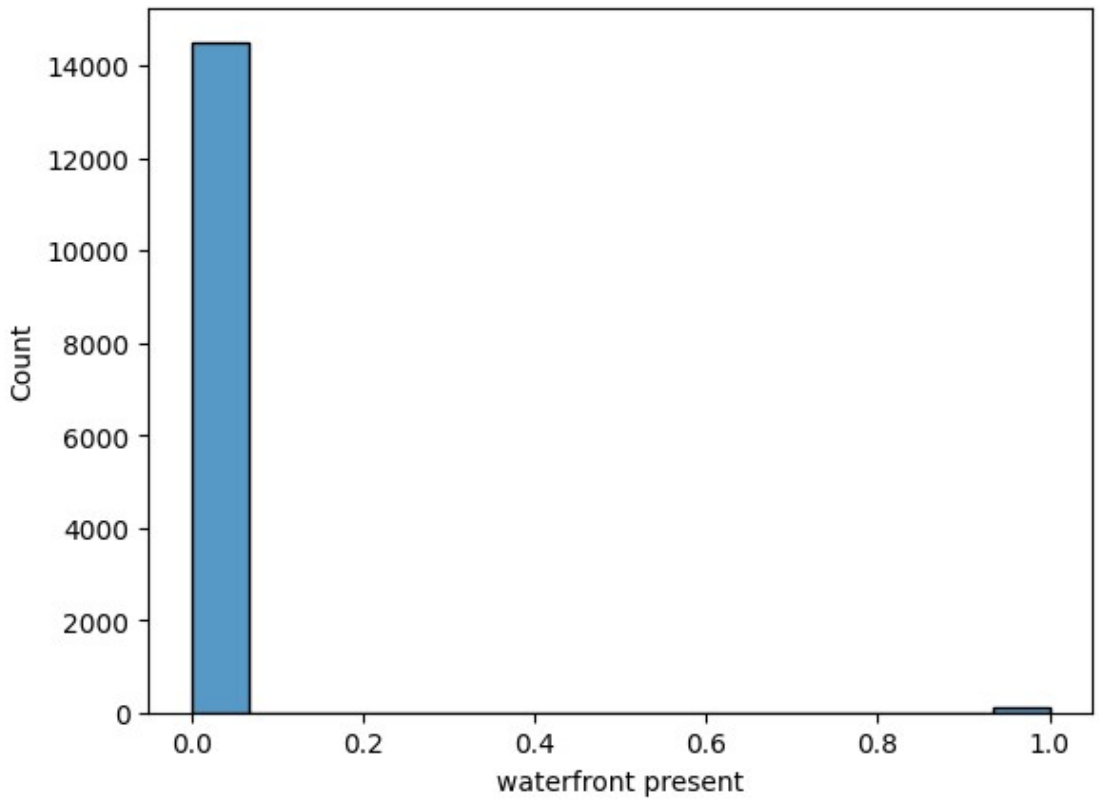
```
<Axes: xlabel='number of floors', ylabel='Count'>
```



Analyzing based on the waterfront present :

```
sb.histplot(ds['waterfront present'])
```

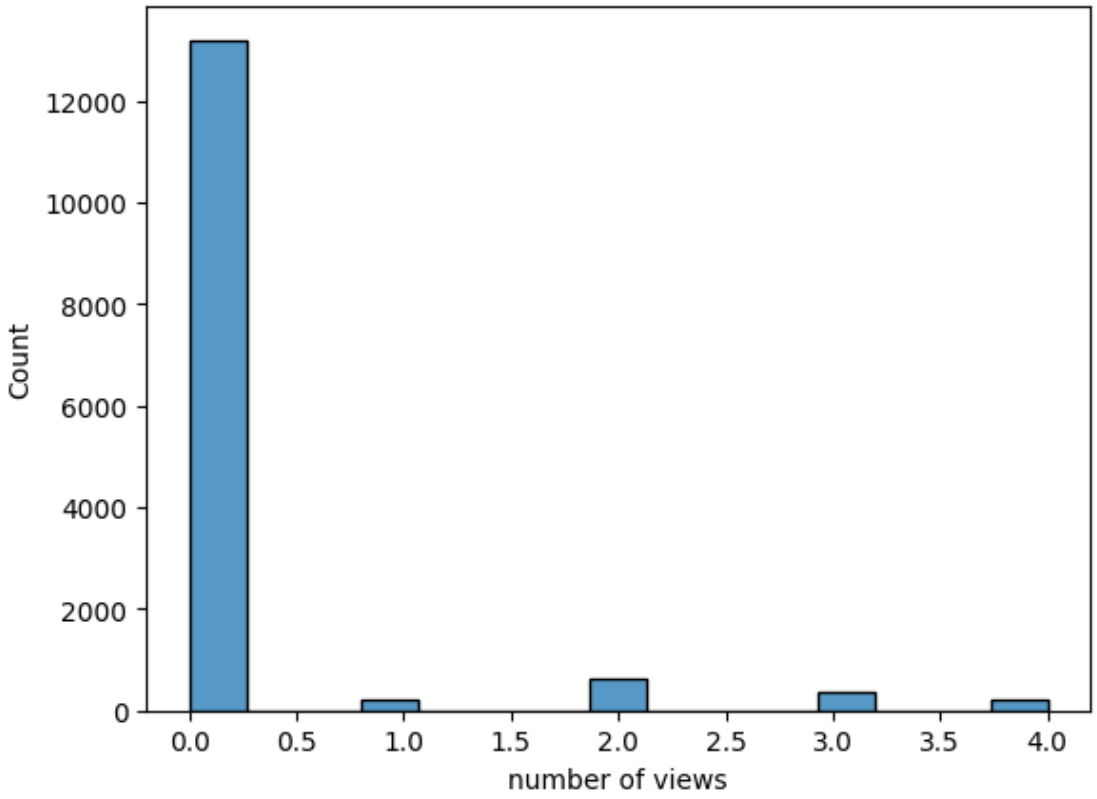
```
<Axes: xlabel='waterfront present', ylabel='Count'>
```



Analyzing based on the number of views:

```
sb.histplot(ds['number of views'])
```

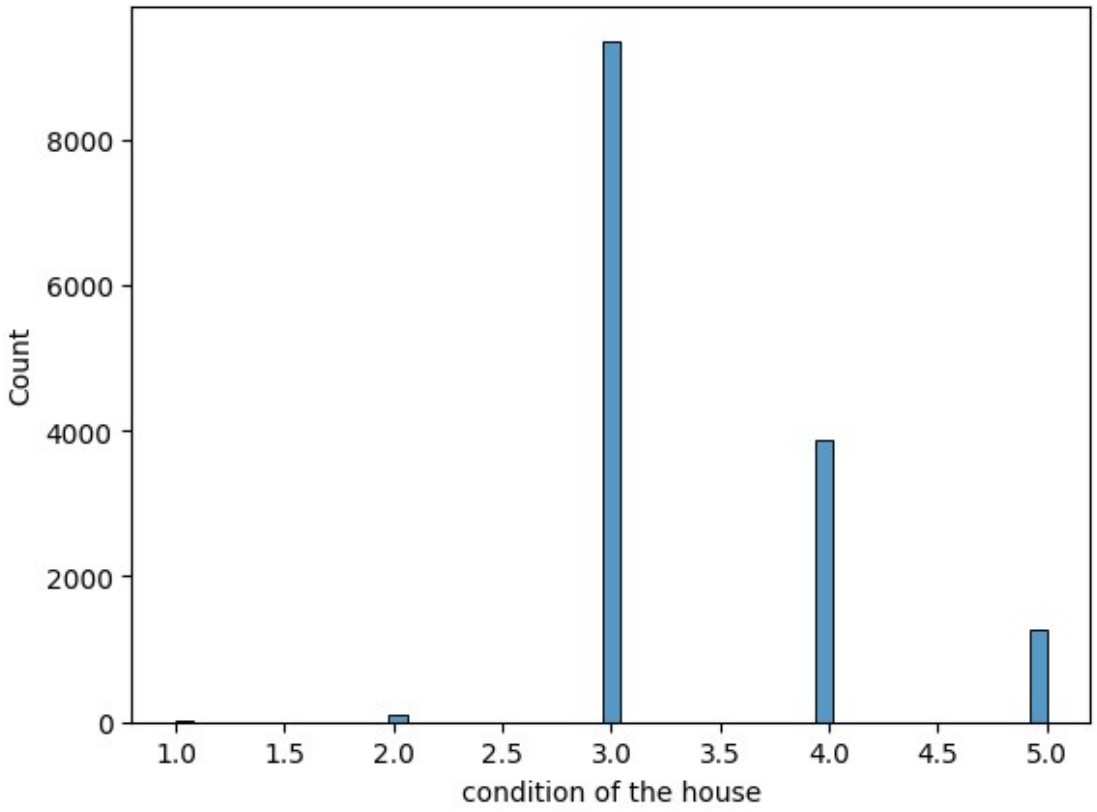
```
<Axes: xlabel='number of views', ylabel='Count'>
```



Analyzing based on the condition of the house :

```
sb.histplot(ds['condition of the house'])
```

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



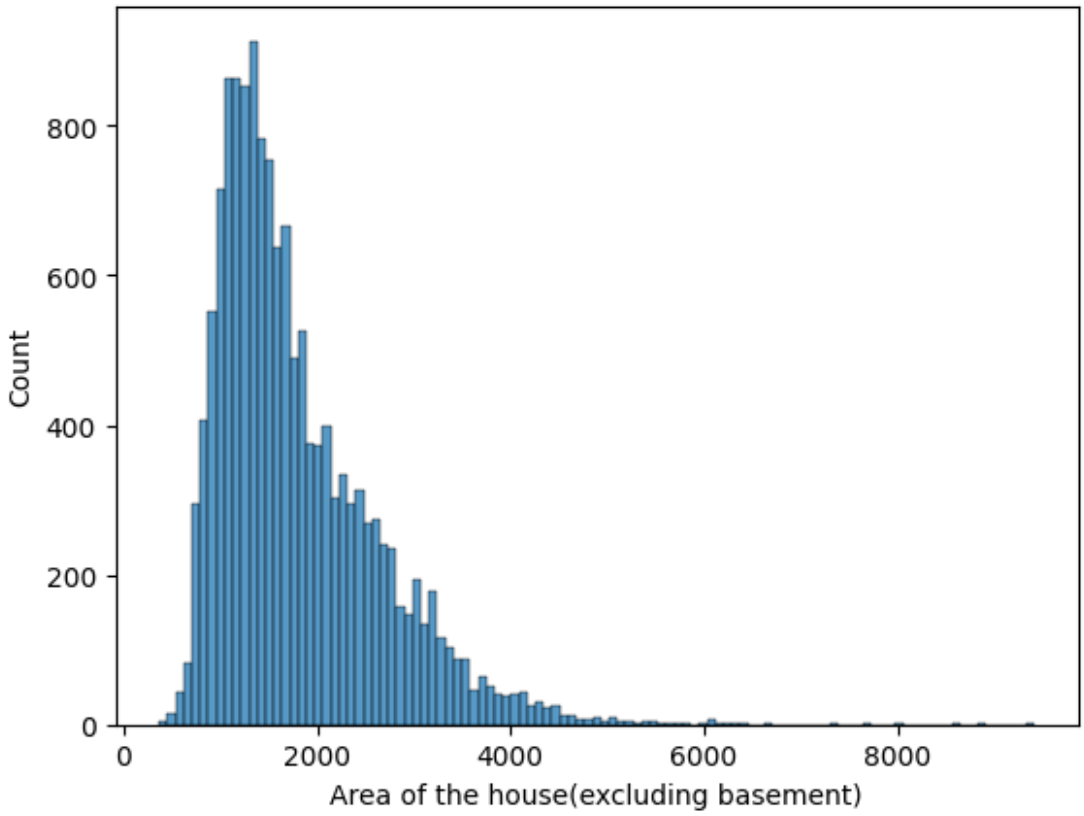
Analyzing based on the grade of the house :

```
sb.histplot(ds['grade of the house'])
```

Analyzing based on the Area of the house(excluding basement) :

```
sb.histplot(ds['Area of the house(excluding basement)'])
```

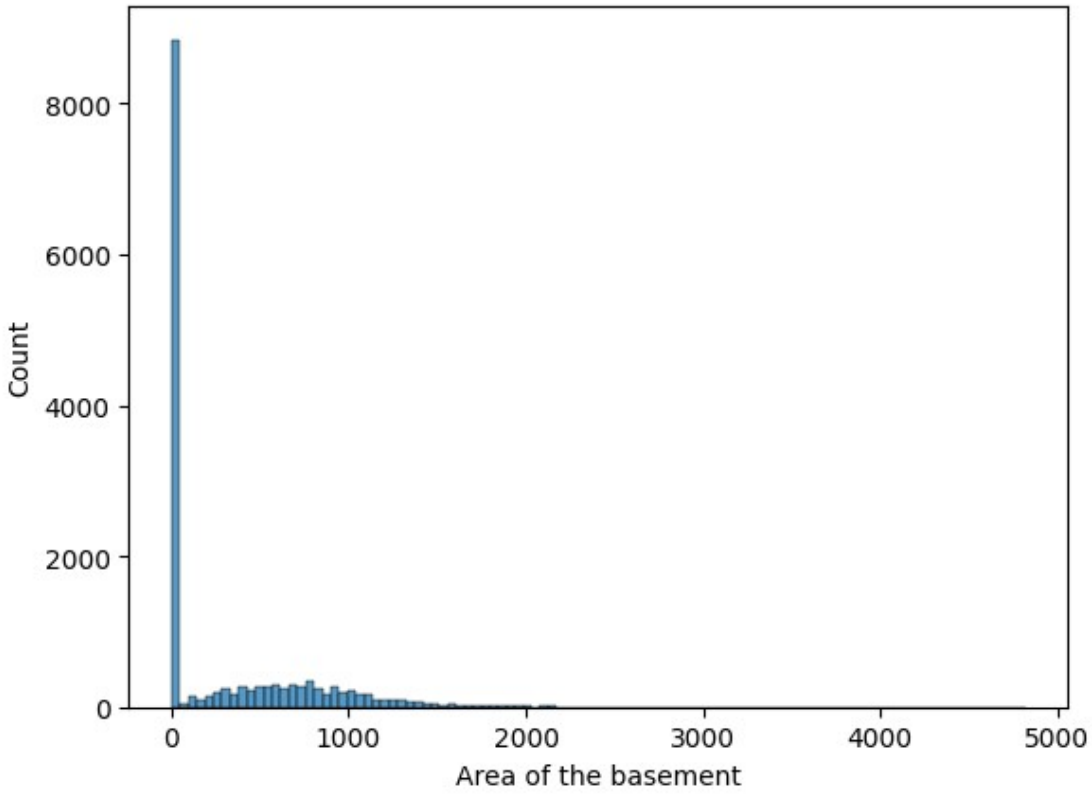
```
<Axes: xlabel='Area of the house(excluding basement)', ylabel='Count'>
```



Analyzing based on the Area of the basement :

```
sb.histplot(ds['Area of the basement'])
```

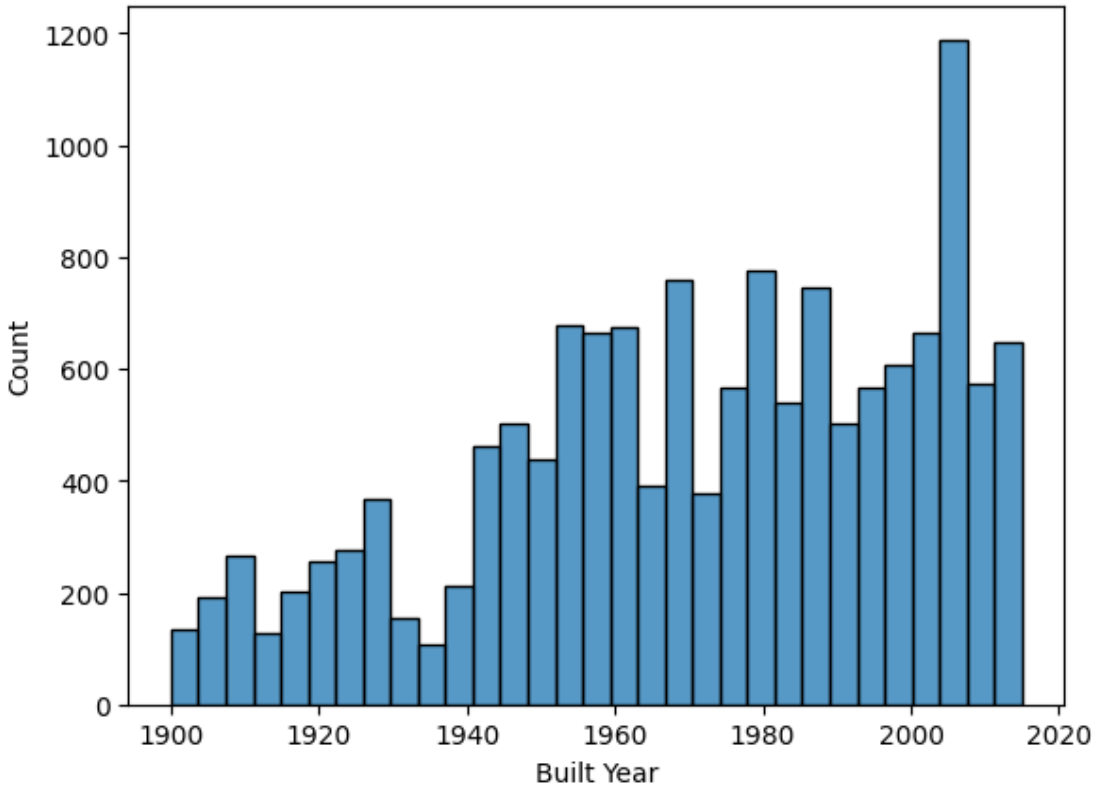
```
<Axes: xlabel='Area of the basement', ylabel='Count'>
```



Analyzing based on the Built Year :

```
sb.histplot(ds['Built Year'])
```

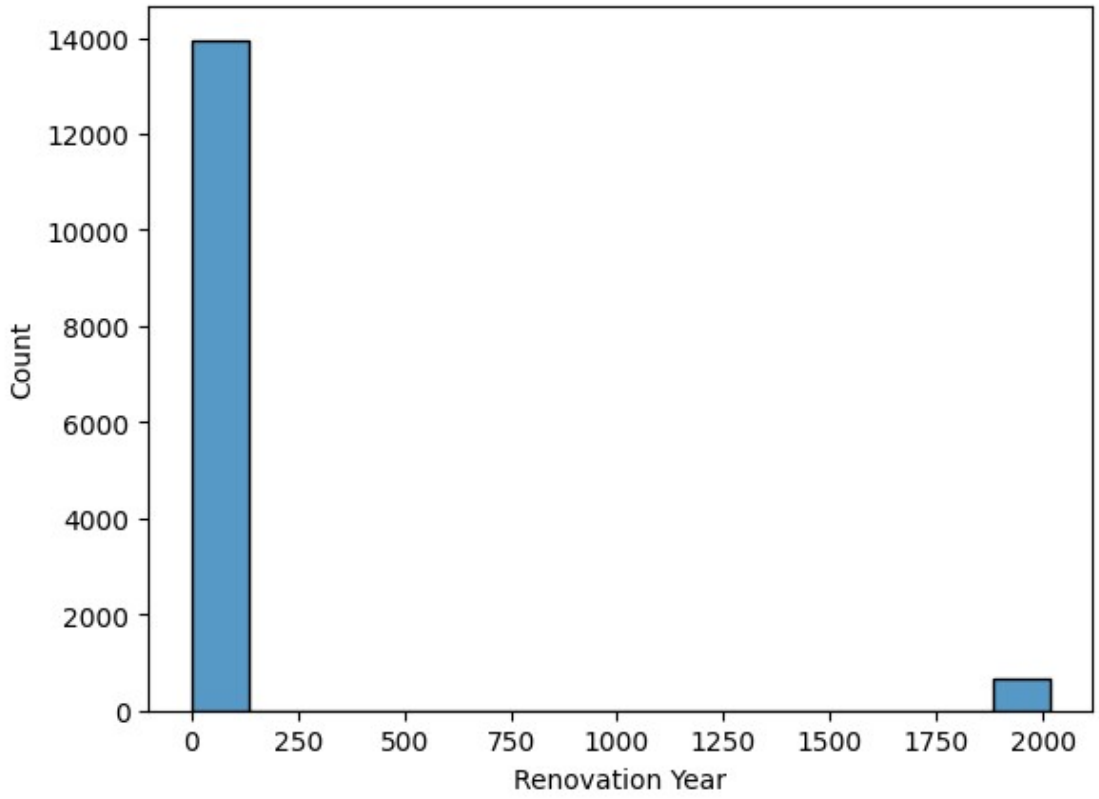
```
<Axes: xlabel='Built Year', ylabel='Count'>
```



Analyzing based on the Renovation Year :

```
sb.histplot(ds['Renovation Year'])
```

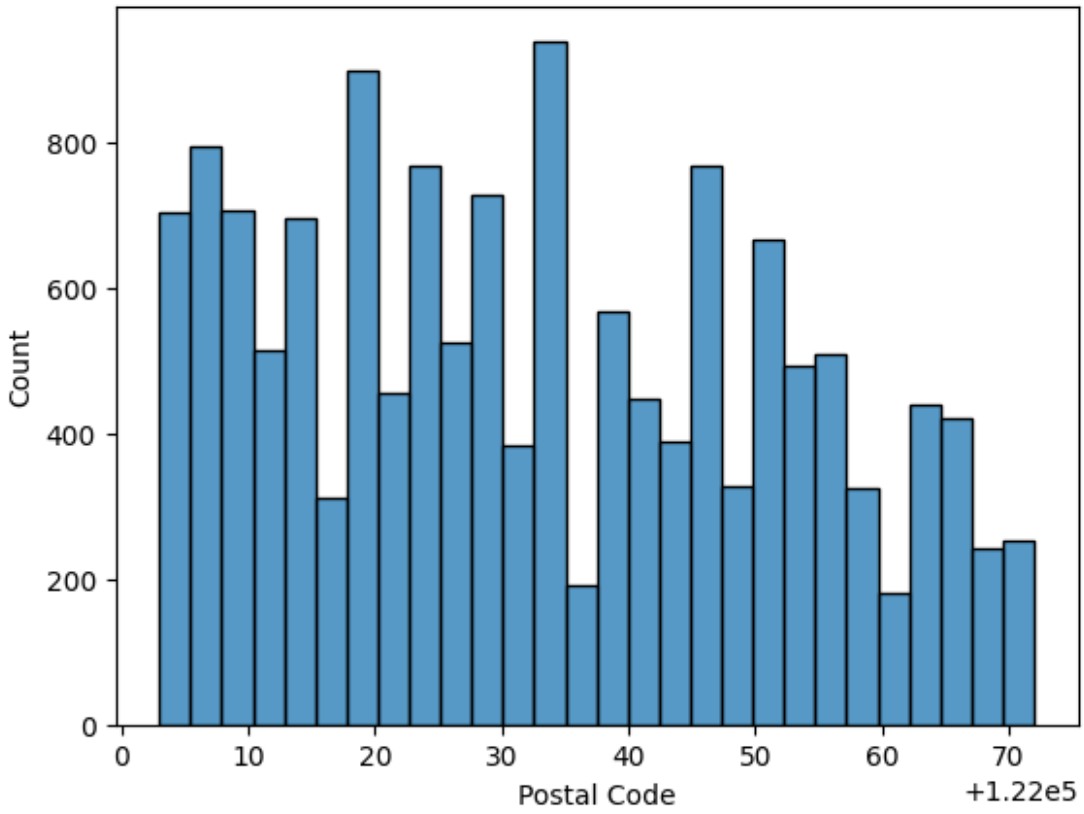
```
<Axes: xlabel='Renovation Year', ylabel='Count'>
```



Analyzing based on the Postal Code :

```
sb.histplot(ds['Postal Code'])
```

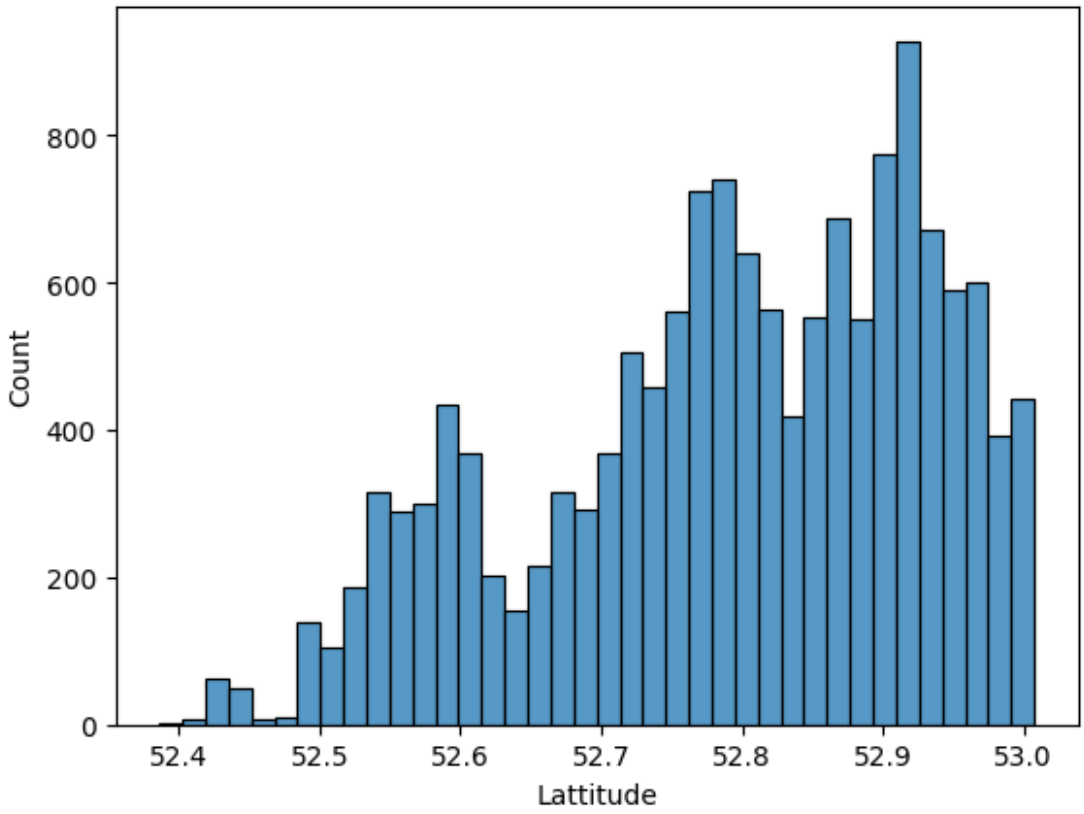
```
<Axes: xlabel='Postal Code', ylabel='Count'>
```



Analyzing based on the Lattitude :

```
sb.histplot(ds['Lattitude'])
```

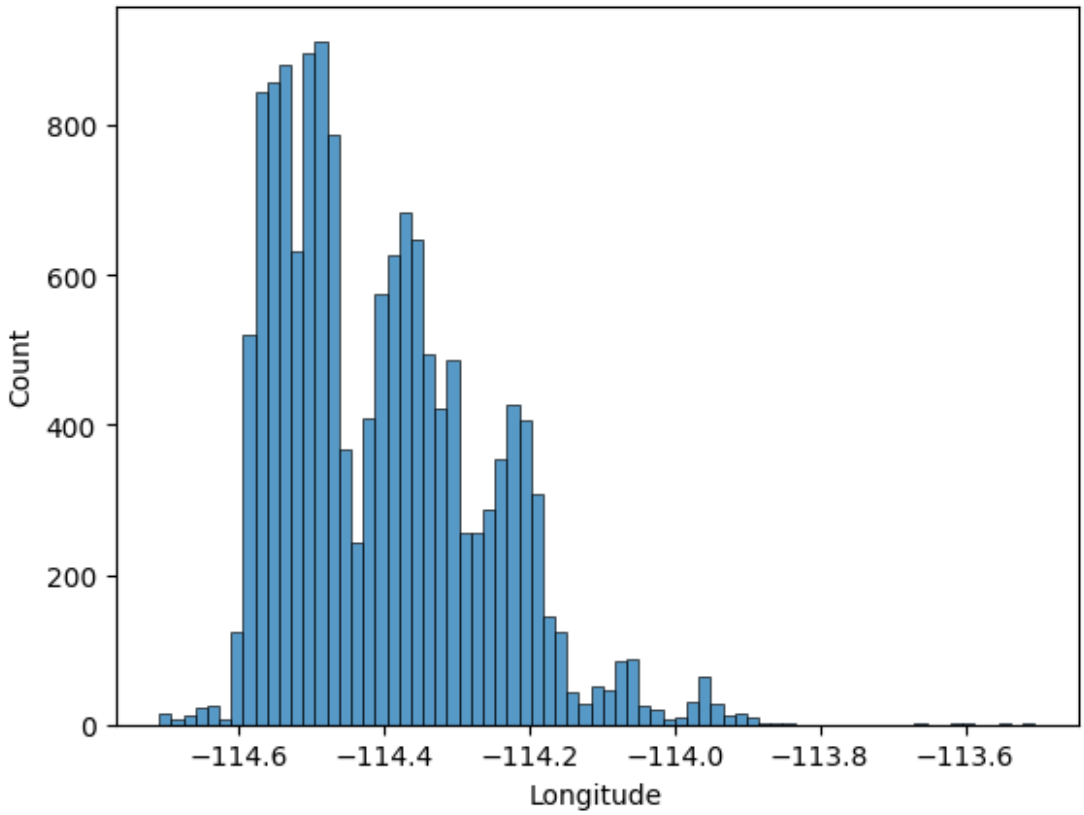
```
<Axes: xlabel='Lattitude', ylabel='Count'>
```



Analyzing based on the Longitude :

```
sb.histplot(ds['Longitude'])
```

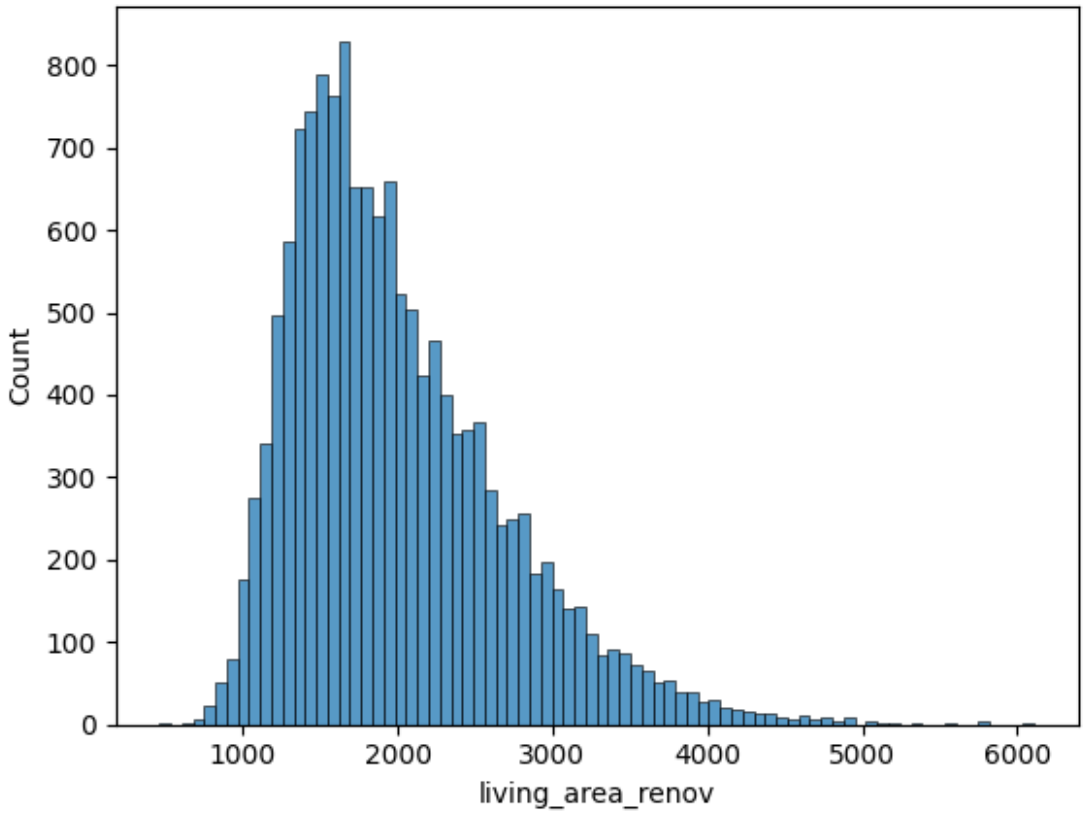
```
<Axes: xlabel='Longitude', ylabel='Count'>
```



Analyzing based on the living_area_renov :

```
sb.histplot(ds['living_area_renov'])
```

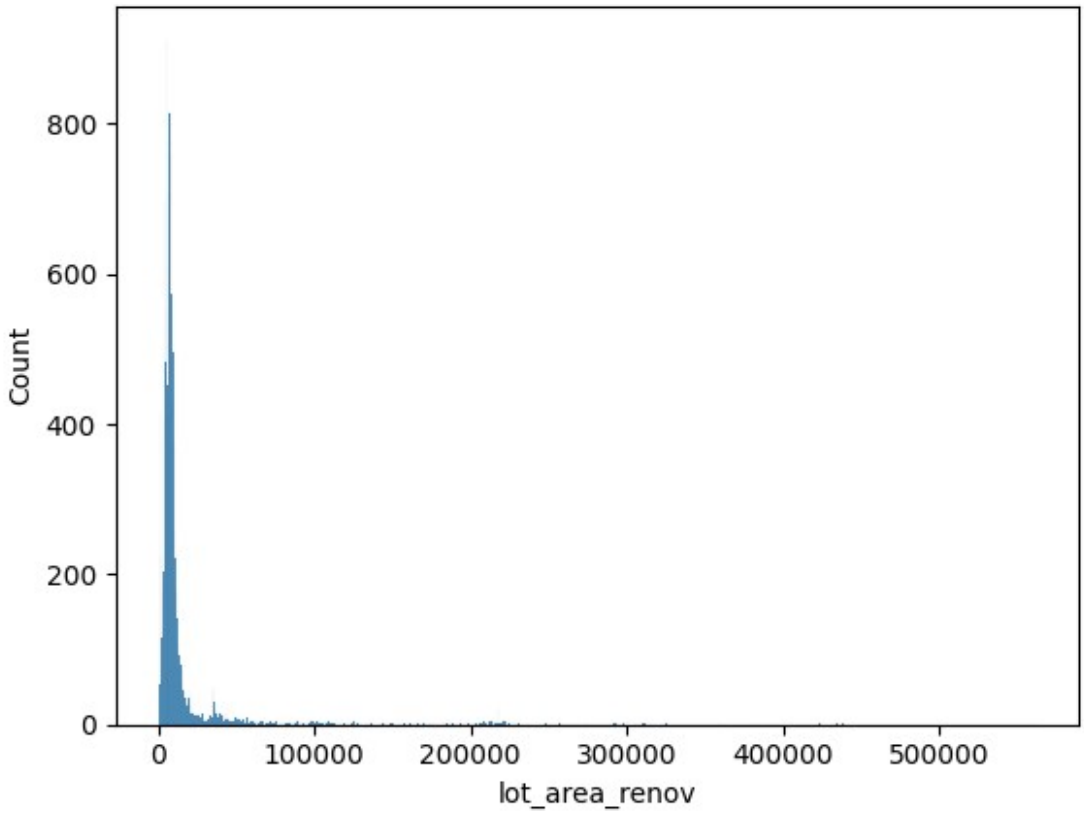
```
<Axes: xlabel='living_area_renov', ylabel='Count'>
```



Analyzing based on the lot_area_renov :

```
sb.histplot(ds['lot_area_renov'])
```

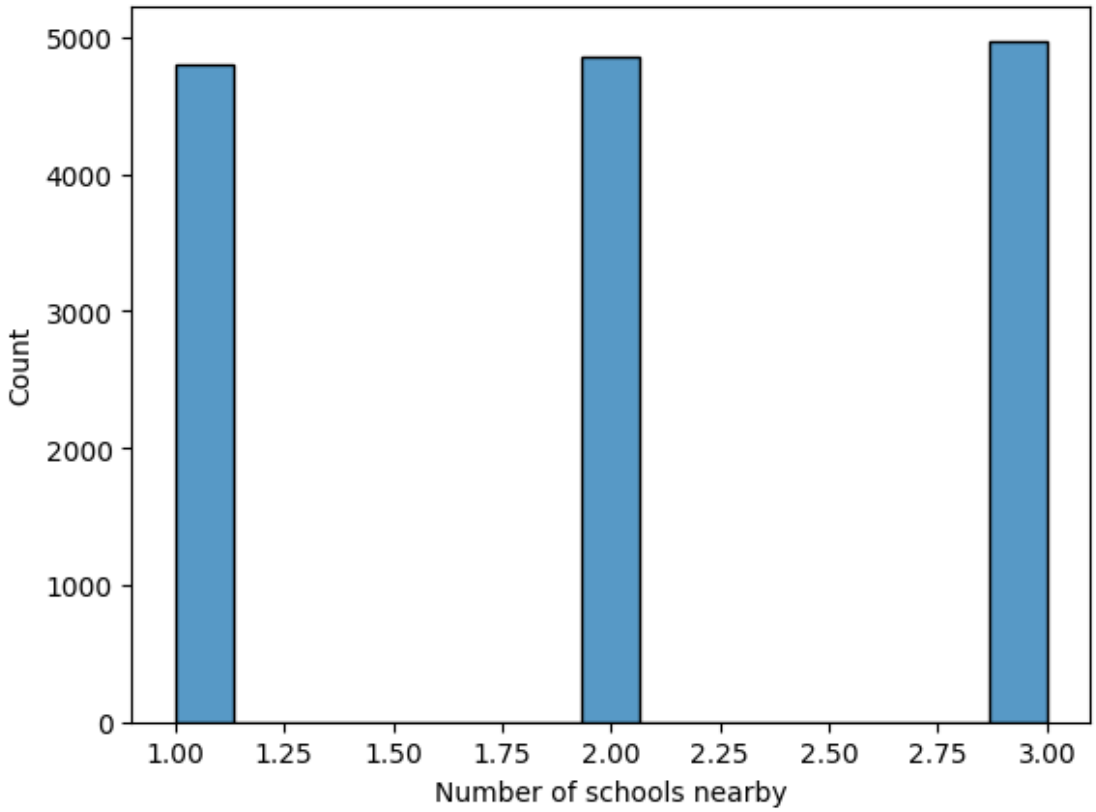
```
<Axes: xlabel='lot_area_renov', ylabel='Count'>
```



Analyzing based on the Number of schools nearby :

```
sb.histplot(ds['Number of schools nearby'])
```

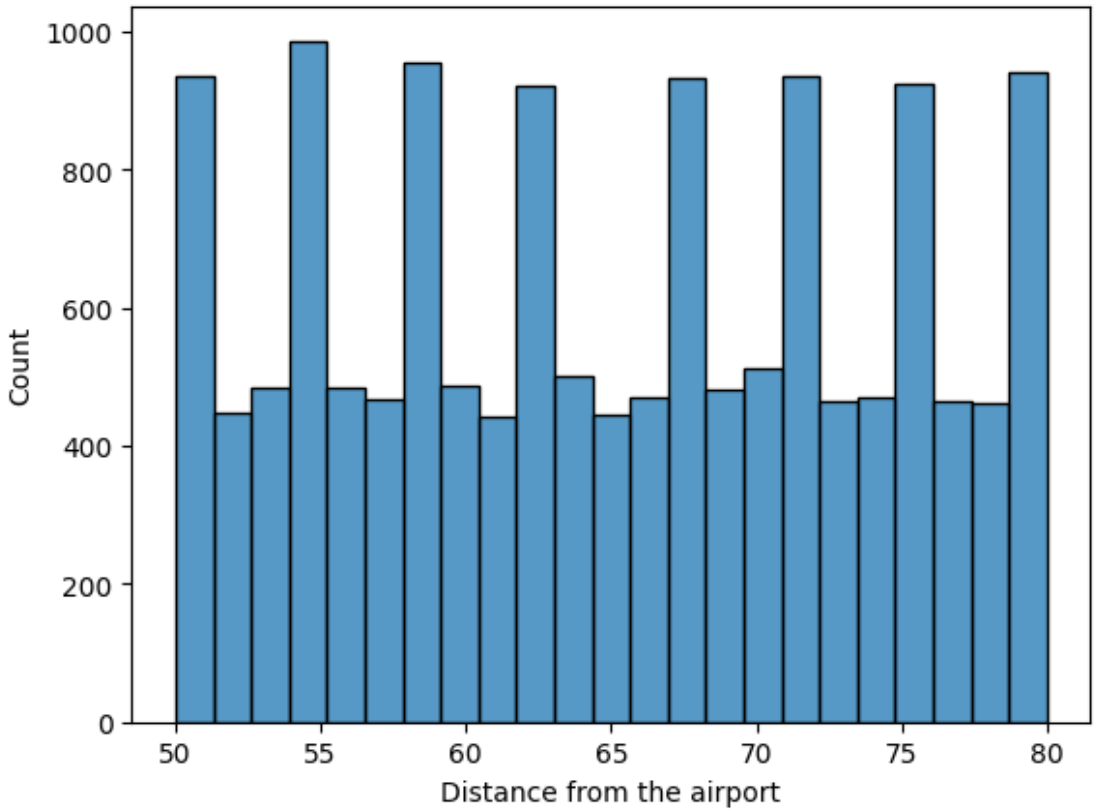
```
<Axes: xlabel='Number of schools nearby', ylabel='Count'>
```



Analyzing based on the Distance from the airport :

```
sb.histplot(ds['Distance from the airport'])
```

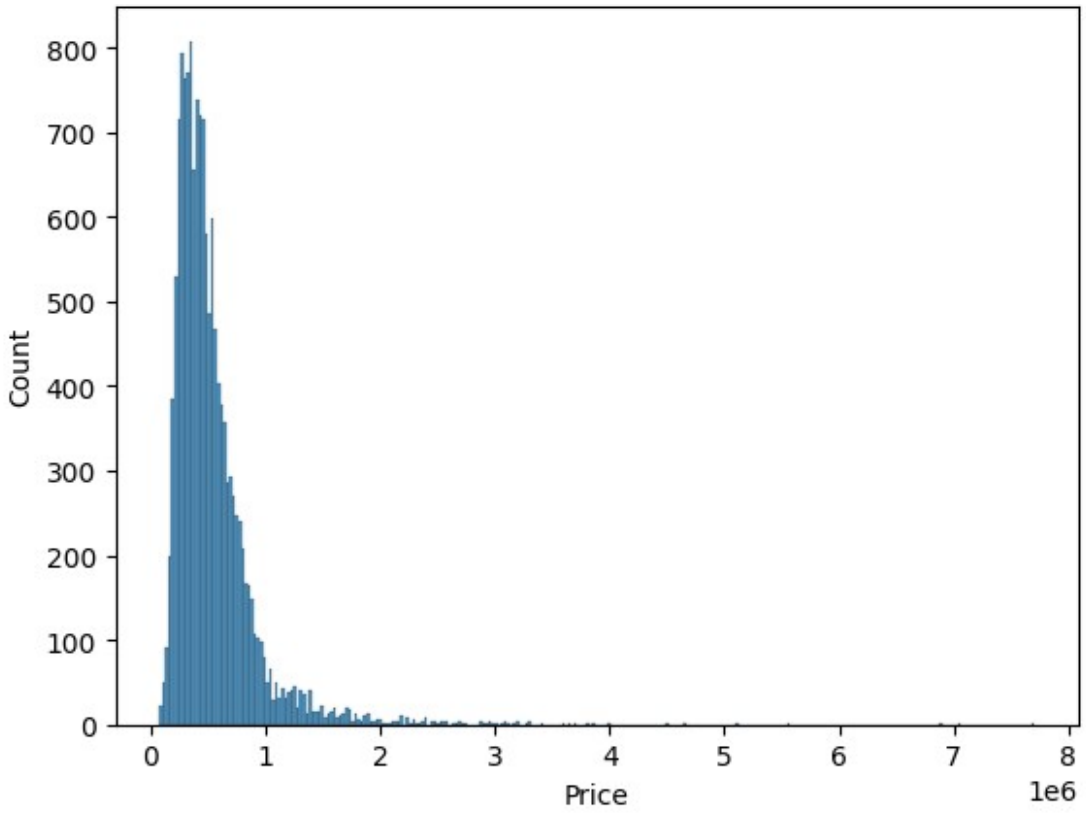
```
<Axes: xlabel='Distance from the airport', ylabel='Count'>
```



Analyzing based on the Price :

```
sb.histplot(ds['Price'])
```

```
<Axes: xlabel='Price', ylabel='Count'>
```



We can see the variations in the data attributes in the given dataset clearly from the above performed operations(graphs). These variations describe the properties of the dataset based on the values.

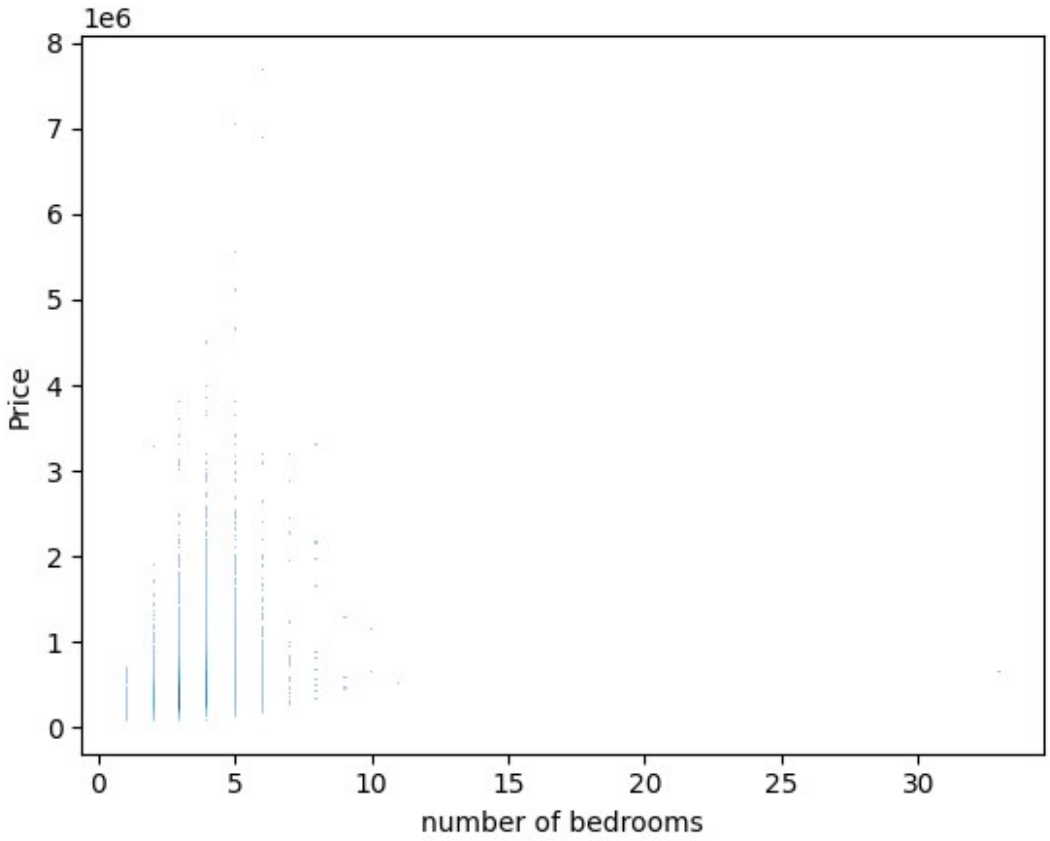
1B. Bivariate analysis

Now lets perform bivariate analysis comparing the price of the houses with the other attributes given in the dataset. Since the price of the house is the most relatable attribute with the other attributes, it is compared to all the other attributes of the dataset.

Analyzing the relation between number of bedrooms and the price

```
sb.histplot(x=ds["number of bedrooms"],y=ds['Price'])
```

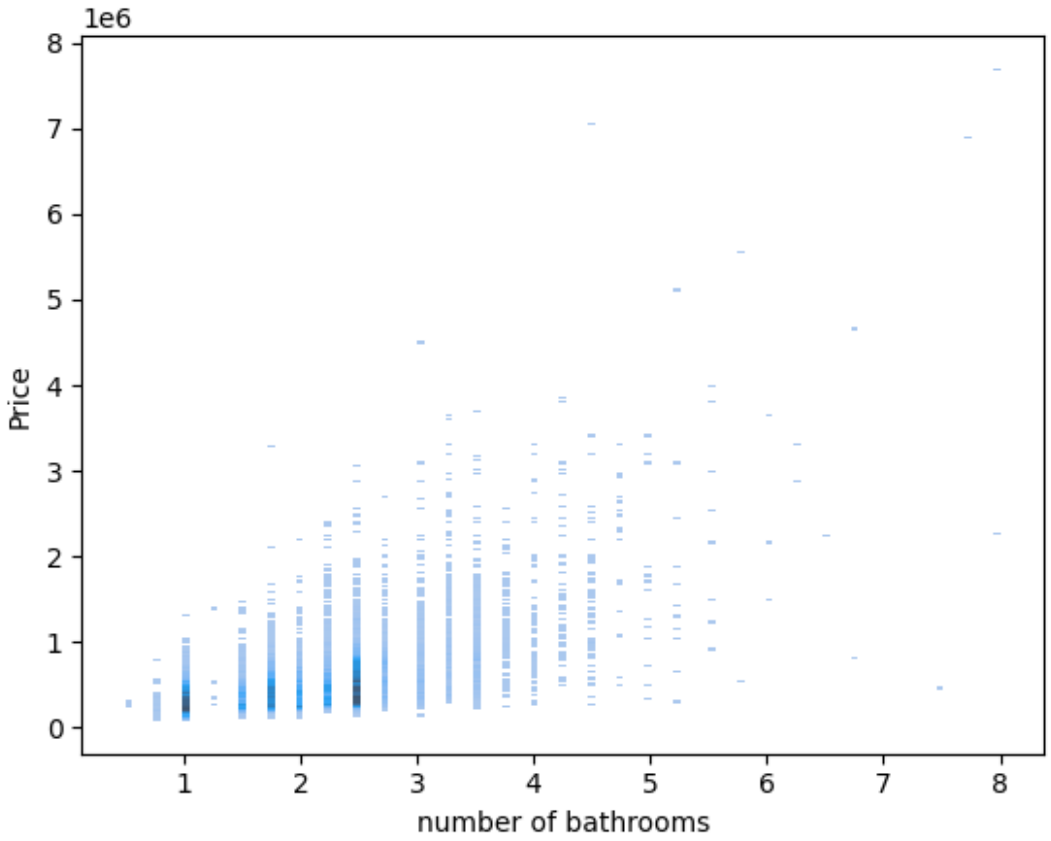
```
<Axes: xlabel='number of bedrooms', ylabel='Price'>
```



Analyzing the relation between number of bathrooms and the price

```
sb.histplot(x=ds["number of bathrooms"],y=ds['Price'])
```

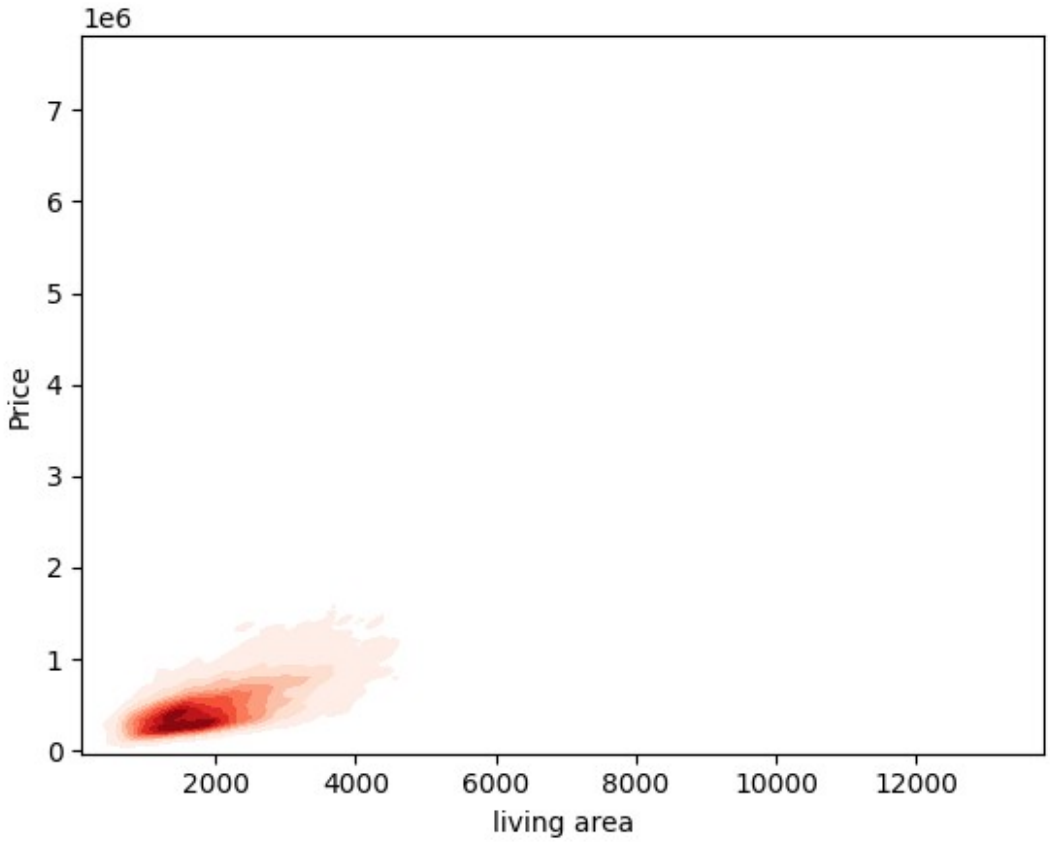
```
<Axes: xlabel='number of bathrooms', ylabel='Price'>
```



Analyzing the relation between living area and the price

```
sb.kdeplot(x=ds["living
area"],y=ds['Price'],cmap="Reds",fill=True,bw_adjust=.5)
```

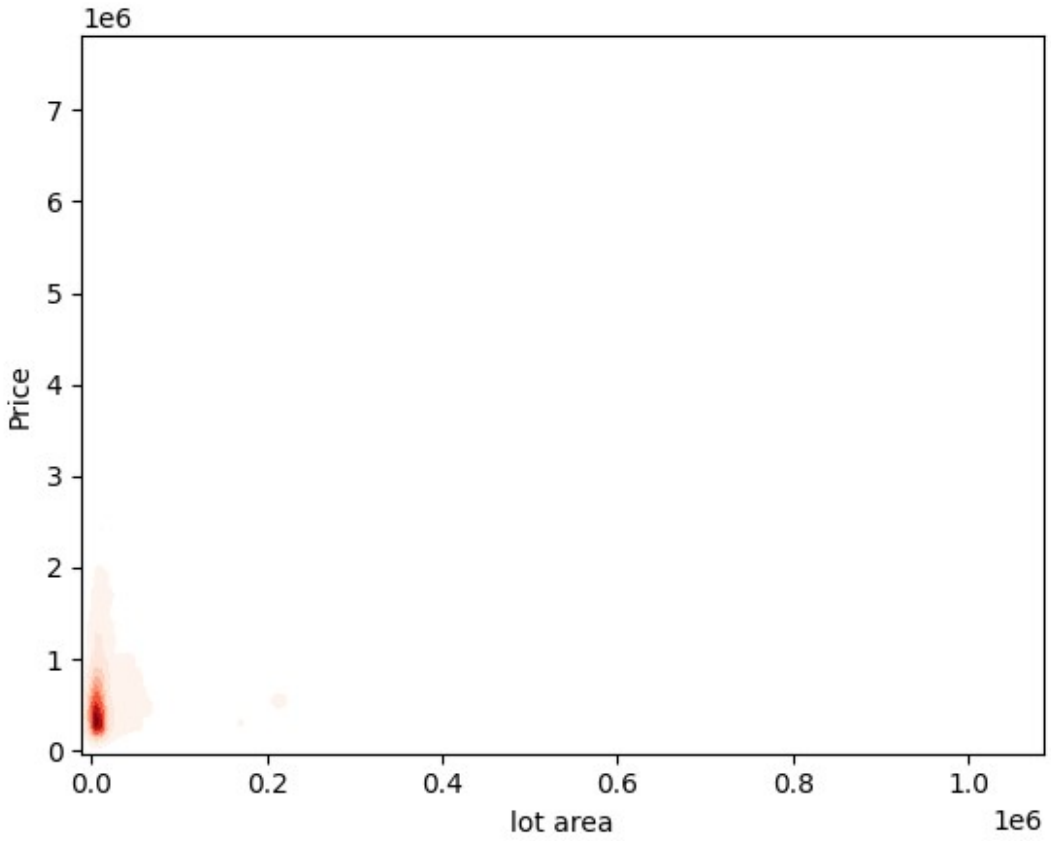
```
<Axes: xlabel='living area', ylabel='Price'>
```



Analyzing the relation between lot area and the price

```
sb.kdeplot(x=ds["lot
area"],y=ds['Price'],cmap="Reds",fill=True,bw_adjust=.5)
```

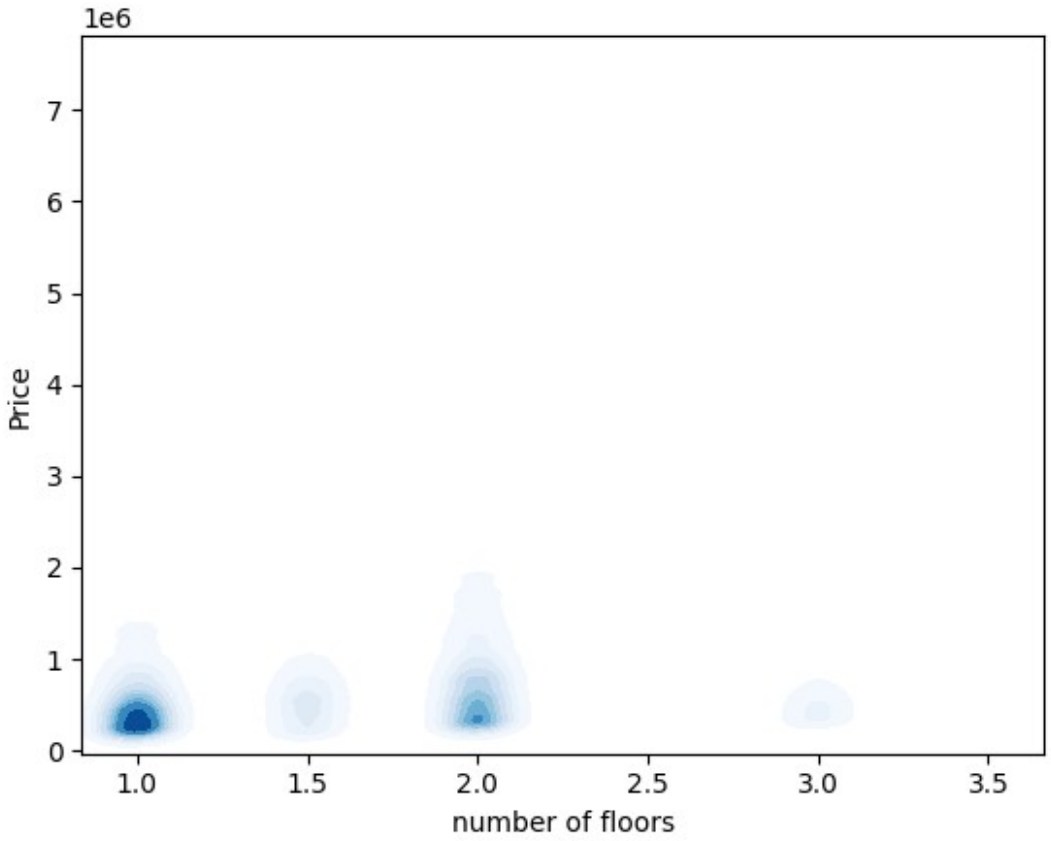
```
<Axes: xlabel='lot area', ylabel='Price'>
```



Analyzing the relation between number of floors and the price

```
sb.kdeplot(x=ds["number of
floors"],y=ds['Price'],cmap="Blues",fill=True,bw_adjust=.5)
```

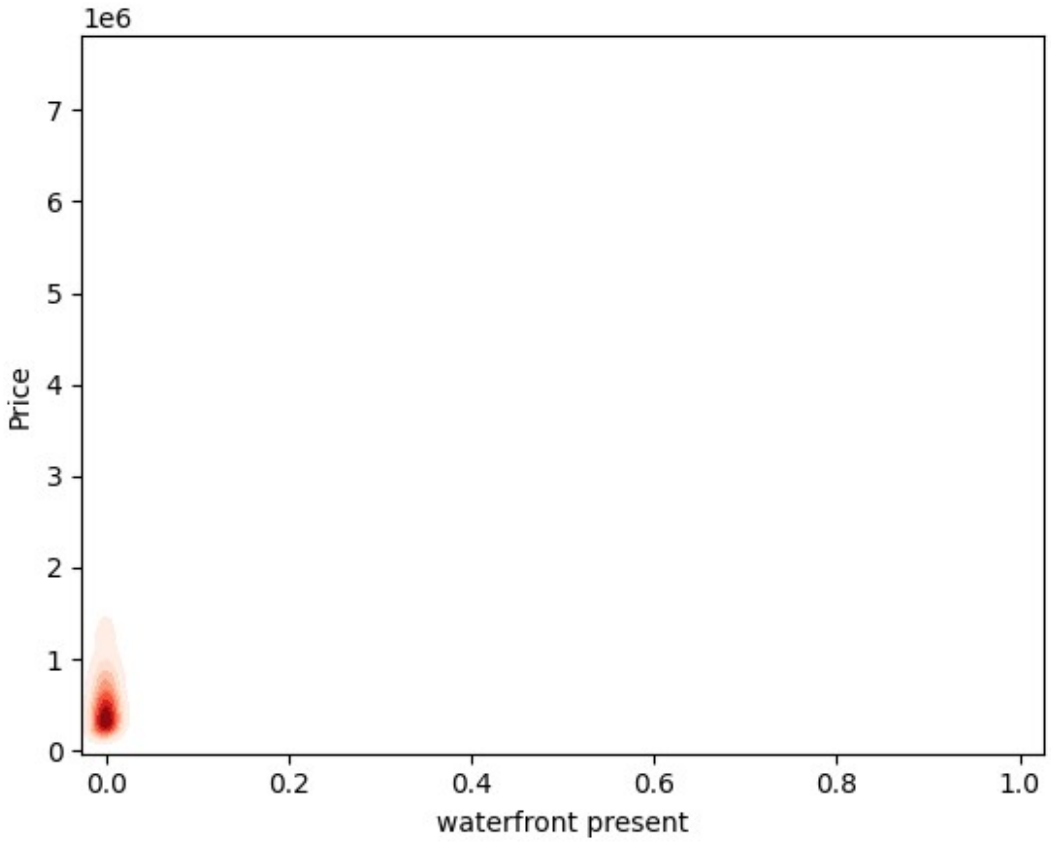
```
<Axes: xlabel='number of floors', ylabel='Price'>
```



Analyzing the relation between waterfront availability and the price

```
sb.kdeplot(x=ds["waterfront
present"],y=ds['Price'],cmap="Reds",fill=True,bw_adjust=.5)
```

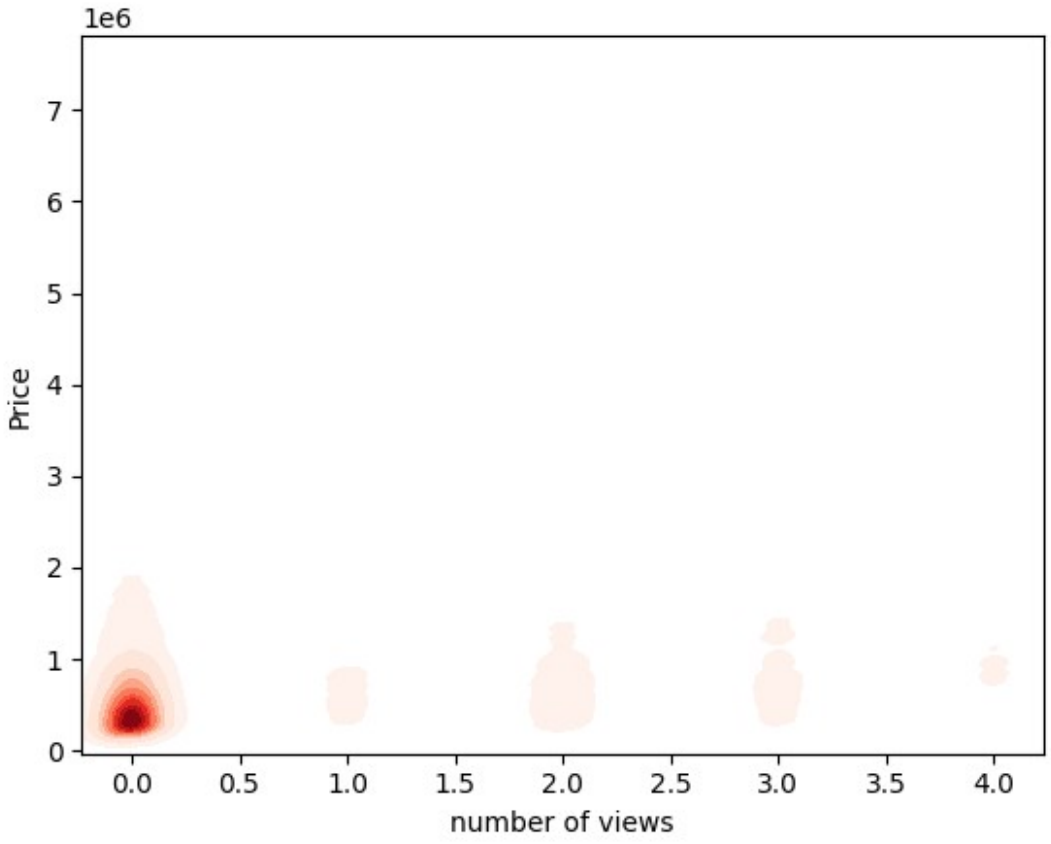
```
<Axes: xlabel='waterfront present', ylabel='Price'>
```



Analyzing the relation between number of views and the price

```
sb.kdeplot(x=ds["number of
views"],y=ds['Price'],cmap="Reds",fill=True,bw_adjust=.5)
```

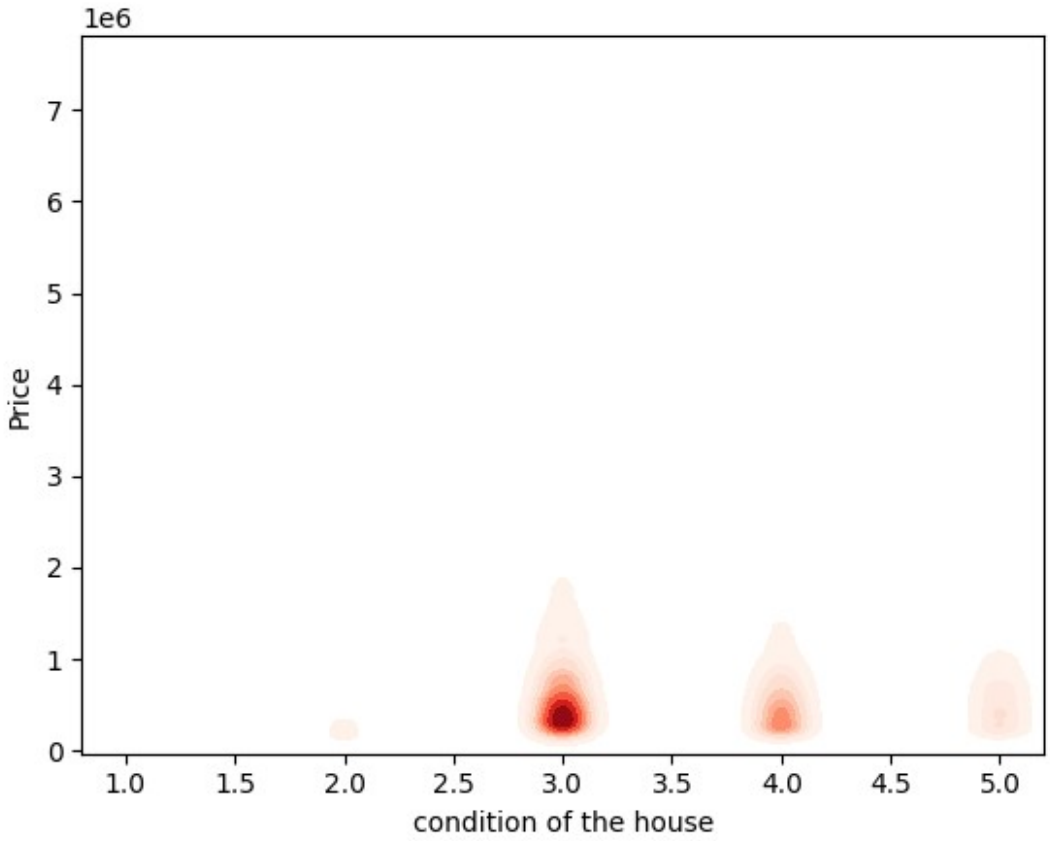
```
<Axes: xlabel='number of views', ylabel='Price'>
```



Analyzing the relation between condition of the house and the price

```
sb.kdeplot(x=ds["condition of the
house"],y=ds['Price'],cmap="Reds",fill=True,bw_adjust=.5)
```

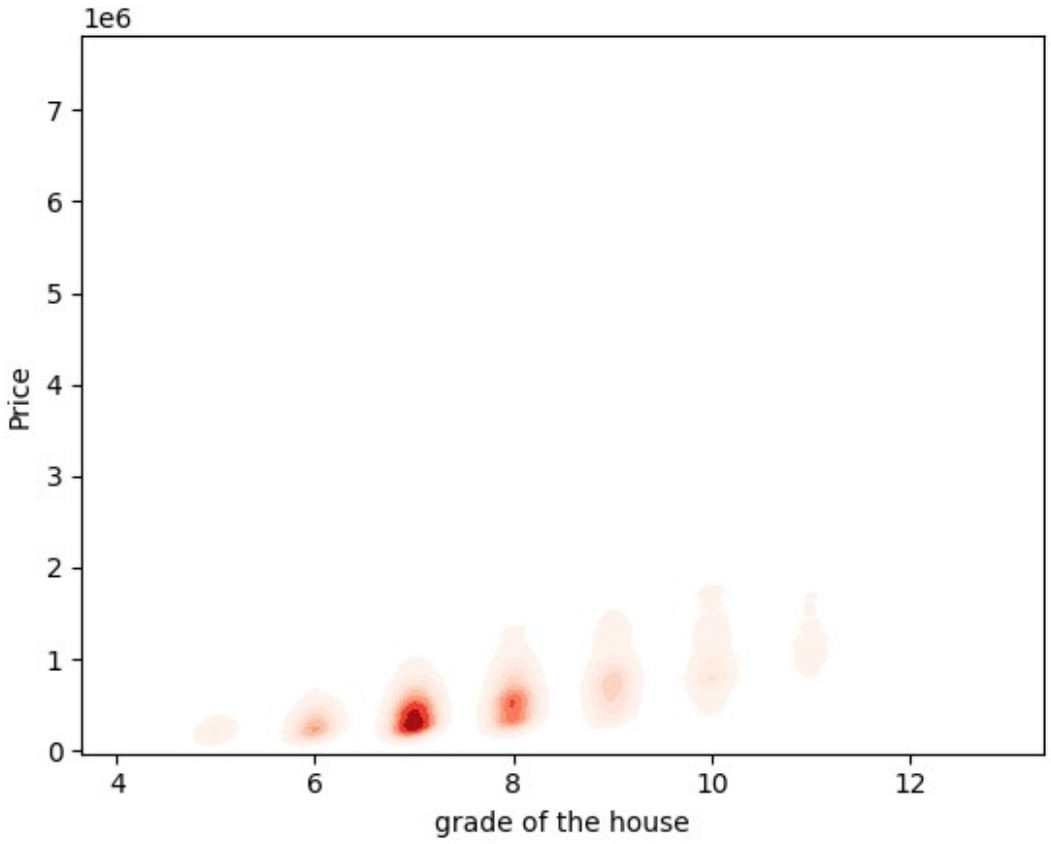
```
<Axes: xlabel='condition of the house', ylabel='Price'>
```



Analyzing the relation between grade of the house and the price

```
sb.kdeplot(x=ds["grade of the
house"],y=ds['Price'],cmap="Reds",fill=True,bw_adjust=.5)
```

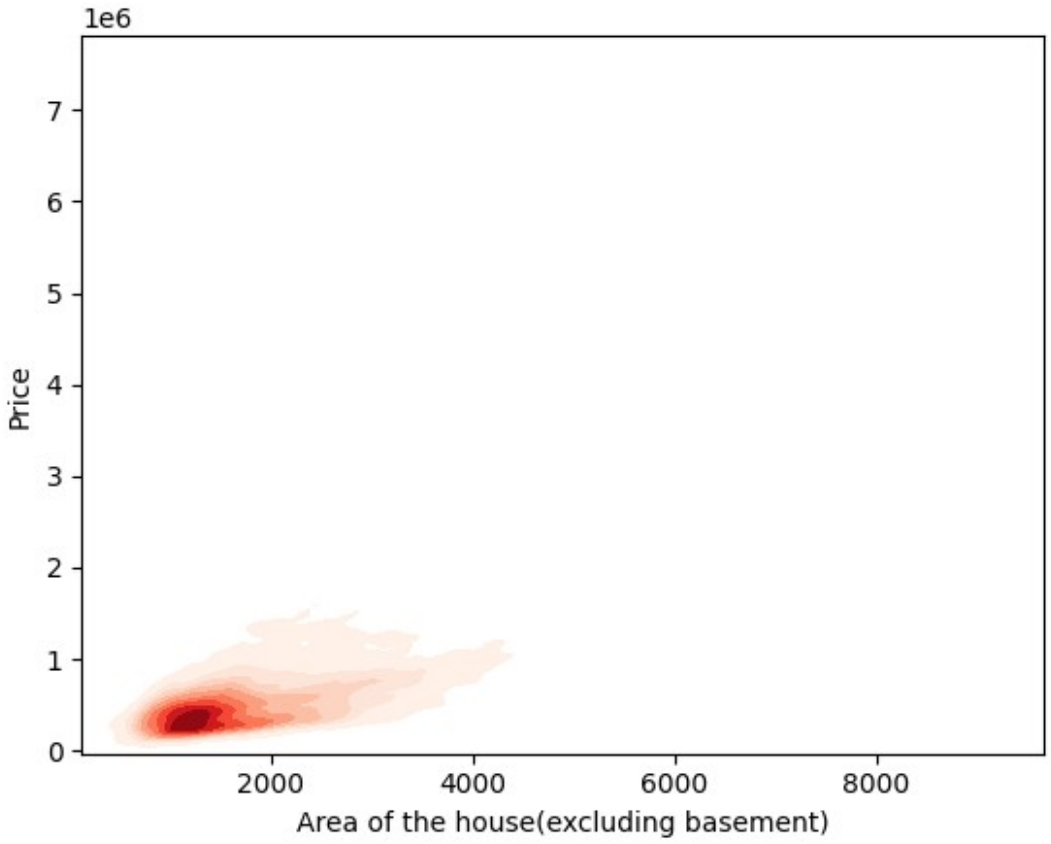
```
<Axes: xlabel='grade of the house', ylabel='Price'>
```



Analyzing the relation between area of the house and the price

```
sb.kdeplot(x=ds["Area of the house(excluding
basement)"],y=ds['Price'],cmap="Reds",fill=True,bw_adjust=.5)
```

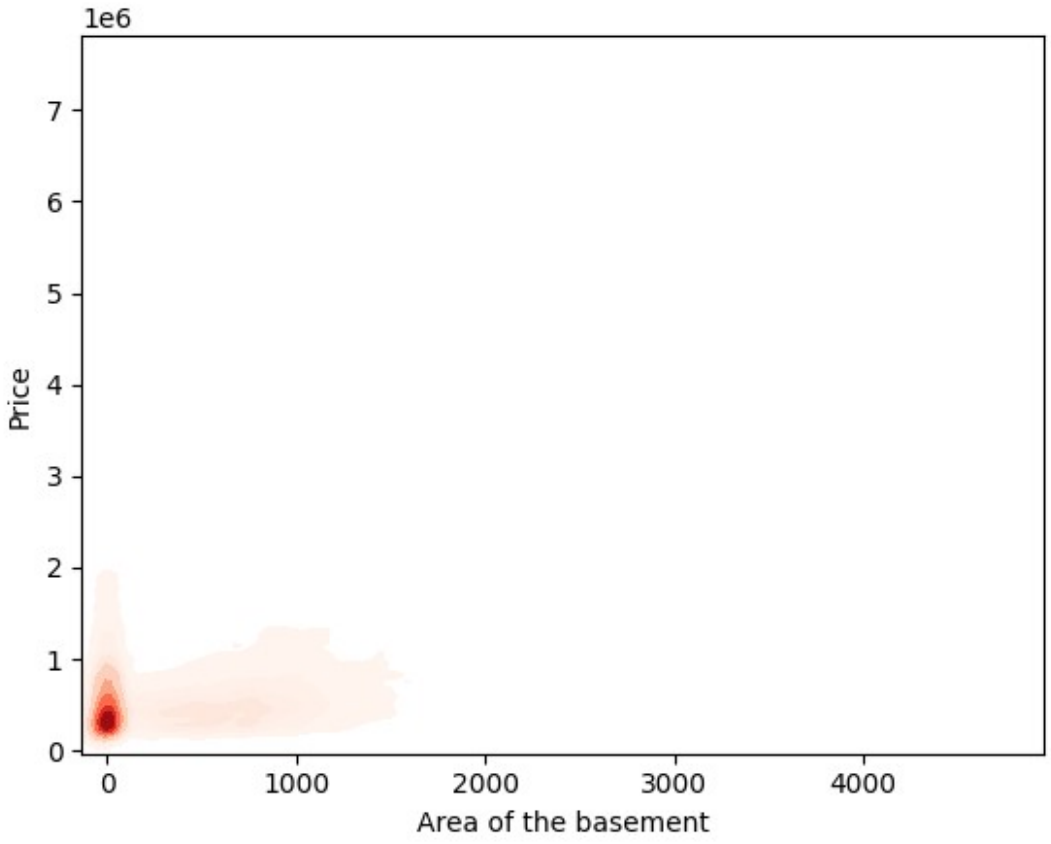
```
<Axes: xlabel='Area of the house(excluding basement)', ylabel='Price'>
```



Analyzing the relation between area of the basement and the price

```
sb.kdeplot(x=ds["Area of the
basement"],y=ds['Price'],cmap="Reds",fill=True,bw_adjust=.5)
```

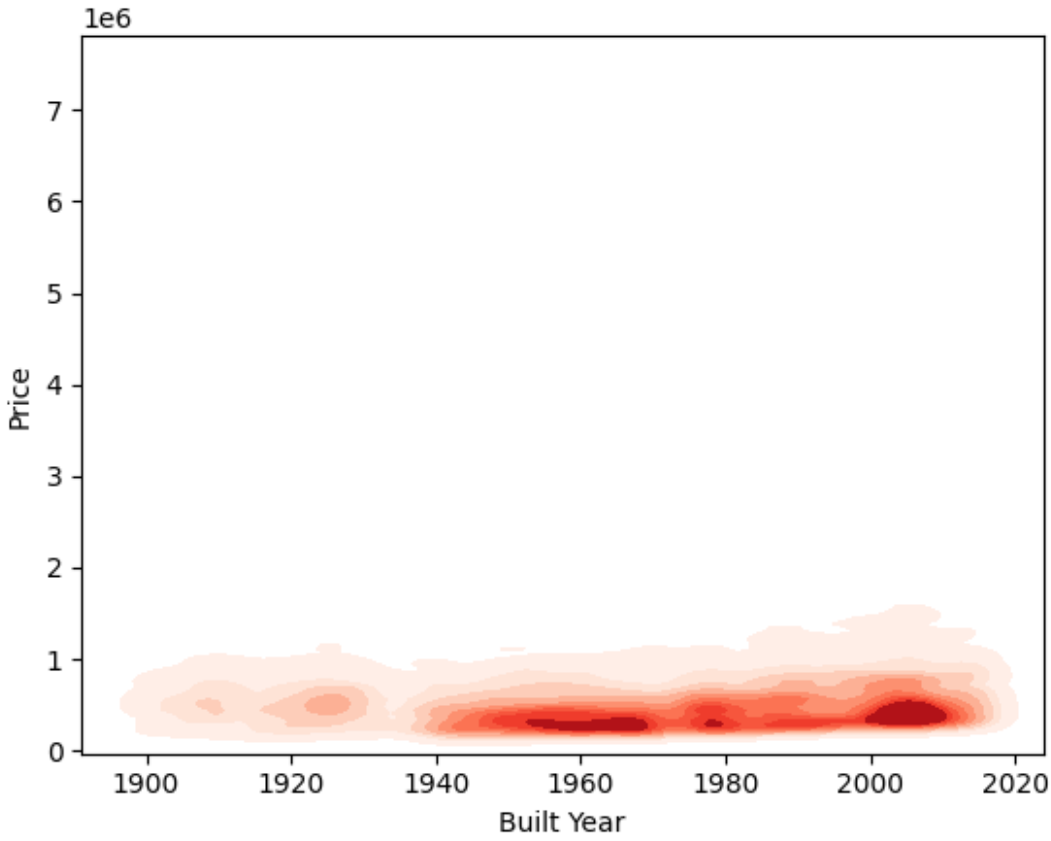
```
<Axes: xlabel='Area of the basement', ylabel='Price'>
```



Analyzing the relation between built year and the price

```
sb.kdeplot(x=ds["Built
Year"],y=ds['Price'],cmap="Reds",fill=True,bw_adjust=.5)
```

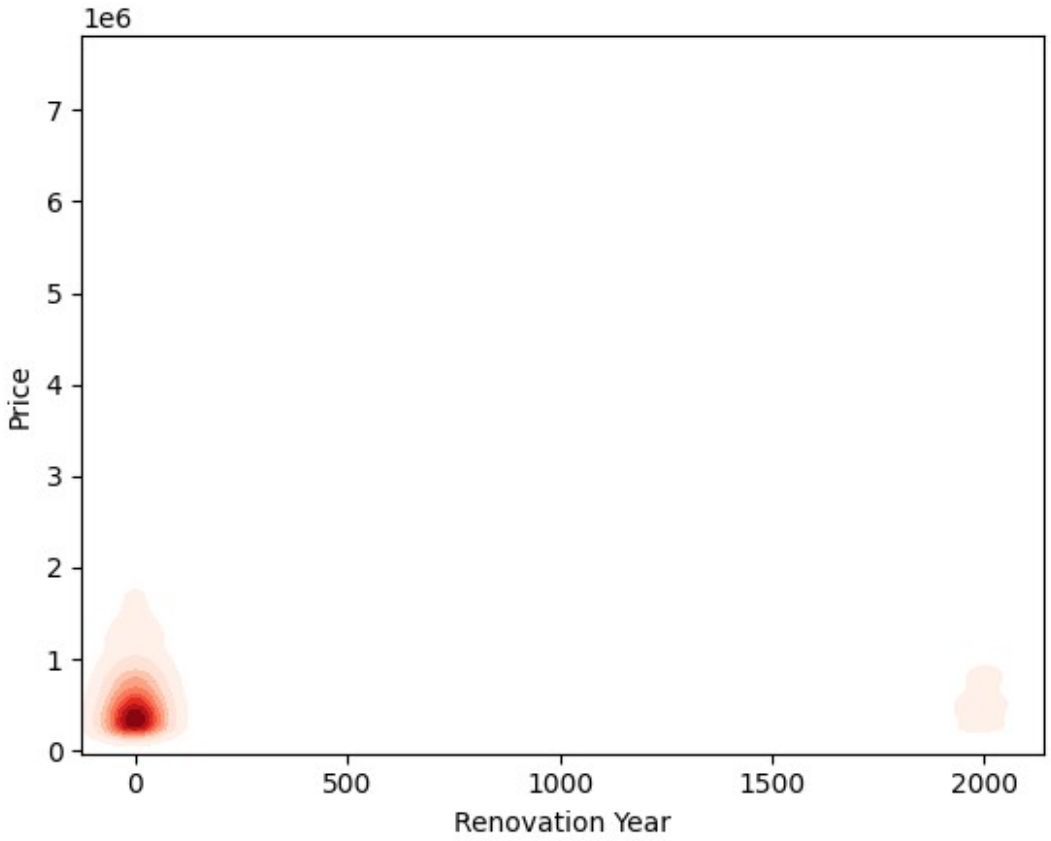
```
<Axes: xlabel='Built Year', ylabel='Price'>
```



Analyzing the relation between renovation year and the price

```
sb.kdeplot(x=ds["Renovation
Year"],y=ds['Price'],cmap="Reds",fill=True,bw_adjust=.5)
```

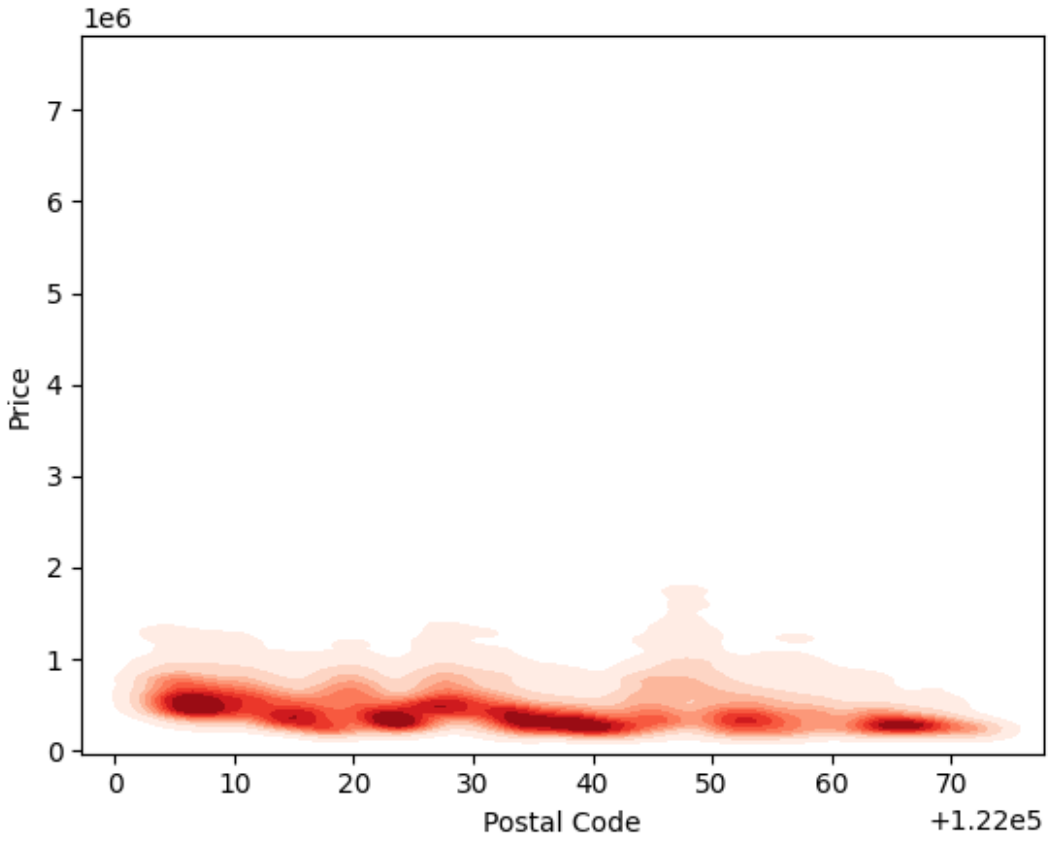
```
<Axes: xlabel='Renovation Year', ylabel='Price'>
```



Analyzing the relation between postal code and the price

```
sb.kdeplot(x=ds["Postal
Code"],y=ds['Price'],cmap="Reds",fill=True,bw_adjust=.5)
```

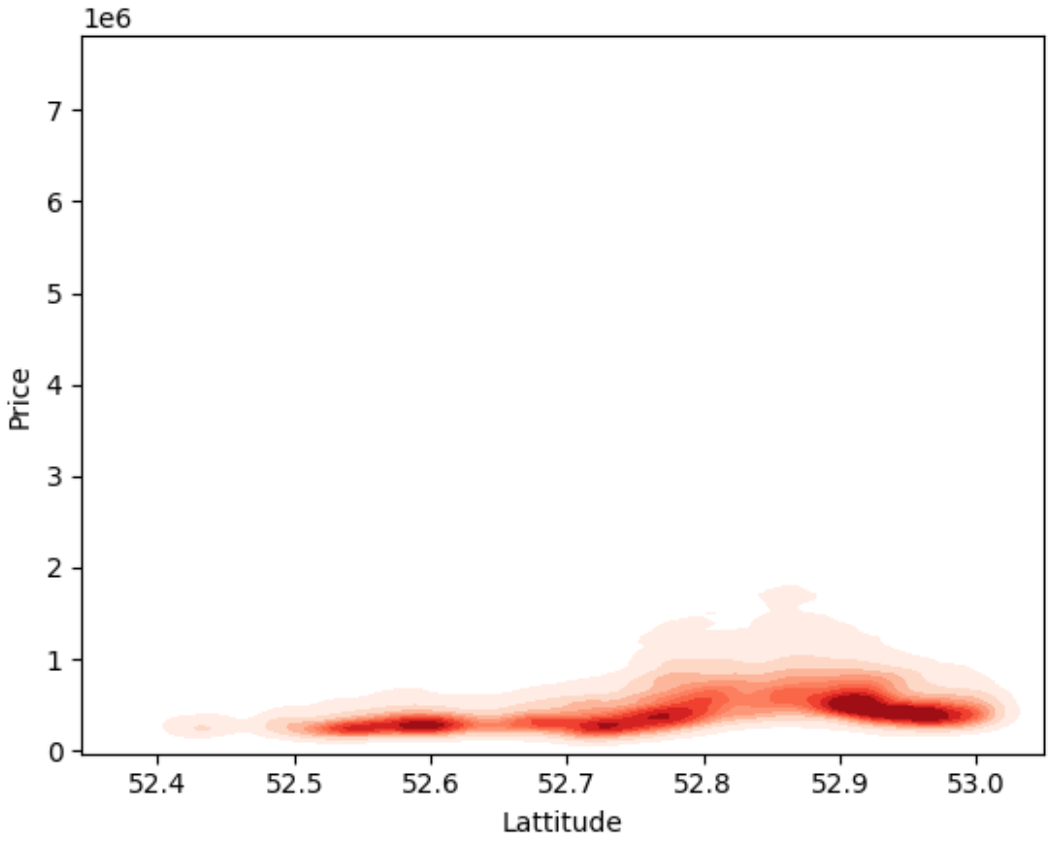
```
<Axes: xlabel='Postal Code', ylabel='Price'>
```



Analyzing the relation between lattitude and the price

```
sb.kdeplot(x=ds["Lattitude"],y=ds['Price'],cmap="Reds",fill=True,bw_adjust=.5)
```

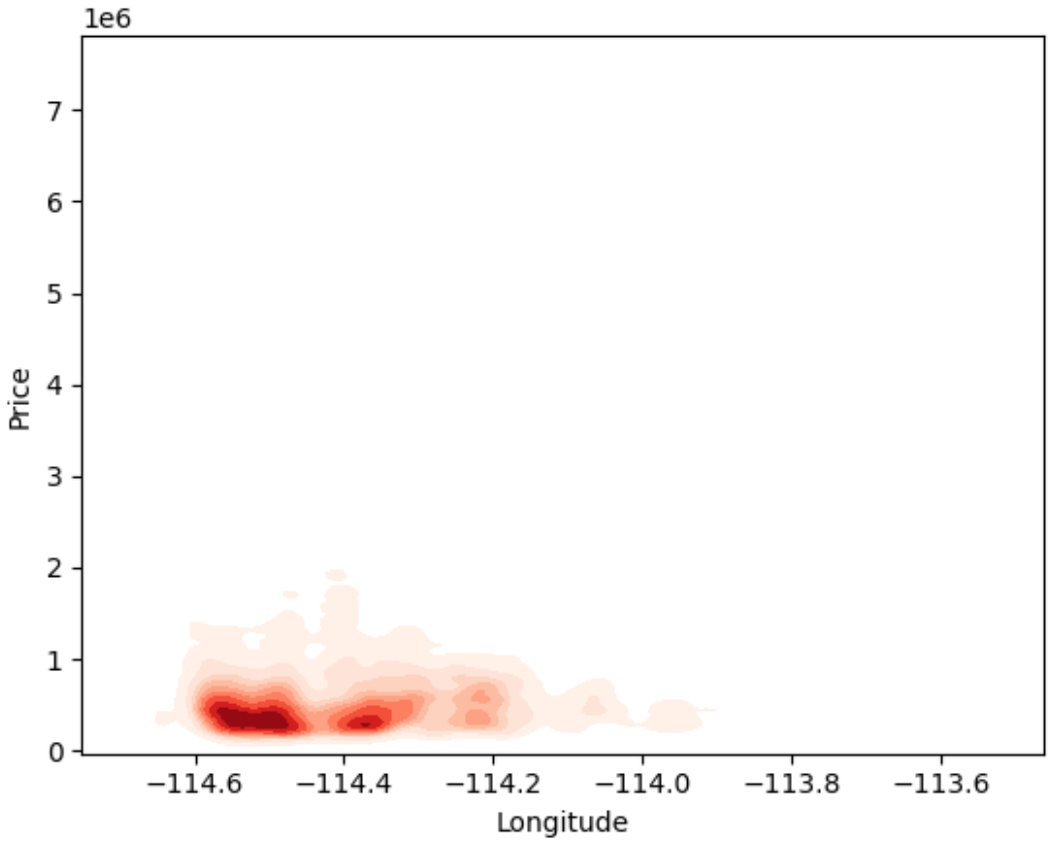
```
<Axes: xlabel='Lattitude', ylabel='Price'>
```



Analyzing the relation between longitude and the price

```
sb.kdeplot(x=ds["Longitude"],y=ds['Price'],cmap="Reds",fill=True,bw_adjust=.5)
```

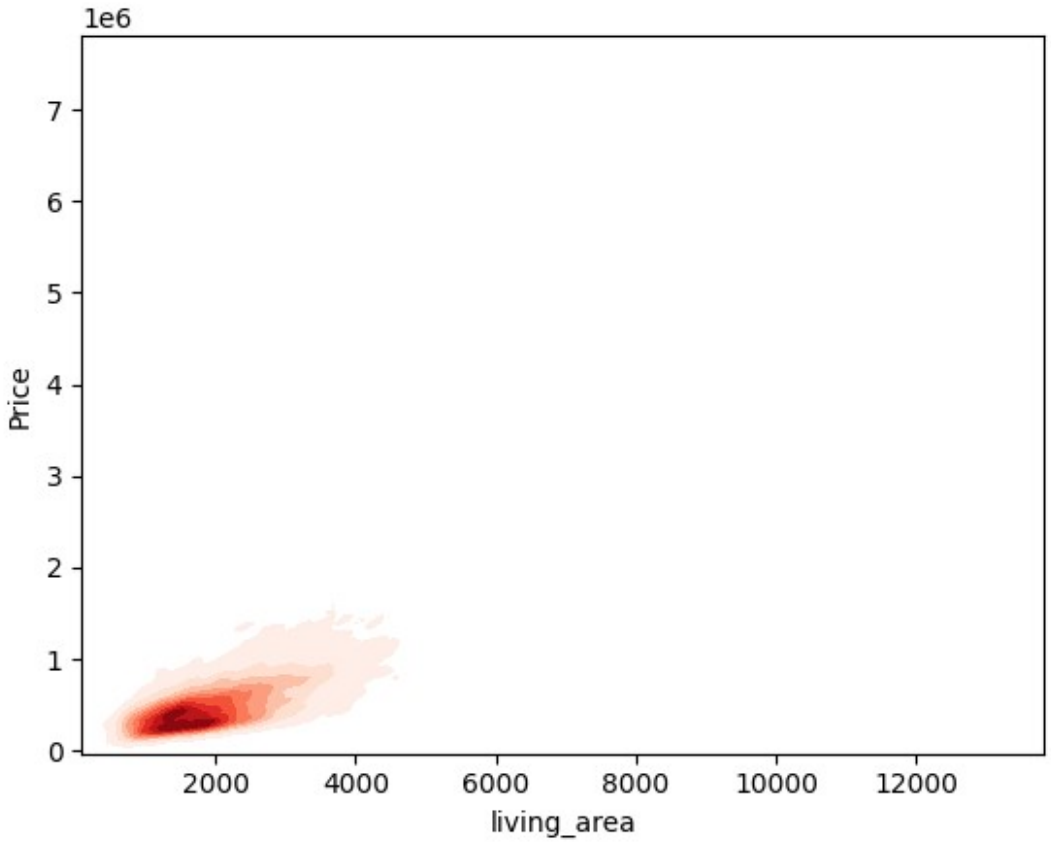
```
<Axes: xlabel='Longitude', ylabel='Price'>
```



Analyzing the relation between living area and the price

```
sb.kdeplot(x=ds["living_area"],y=ds['Price'],cmap="Reds",fill=True,bw_
adjust=.5)
```

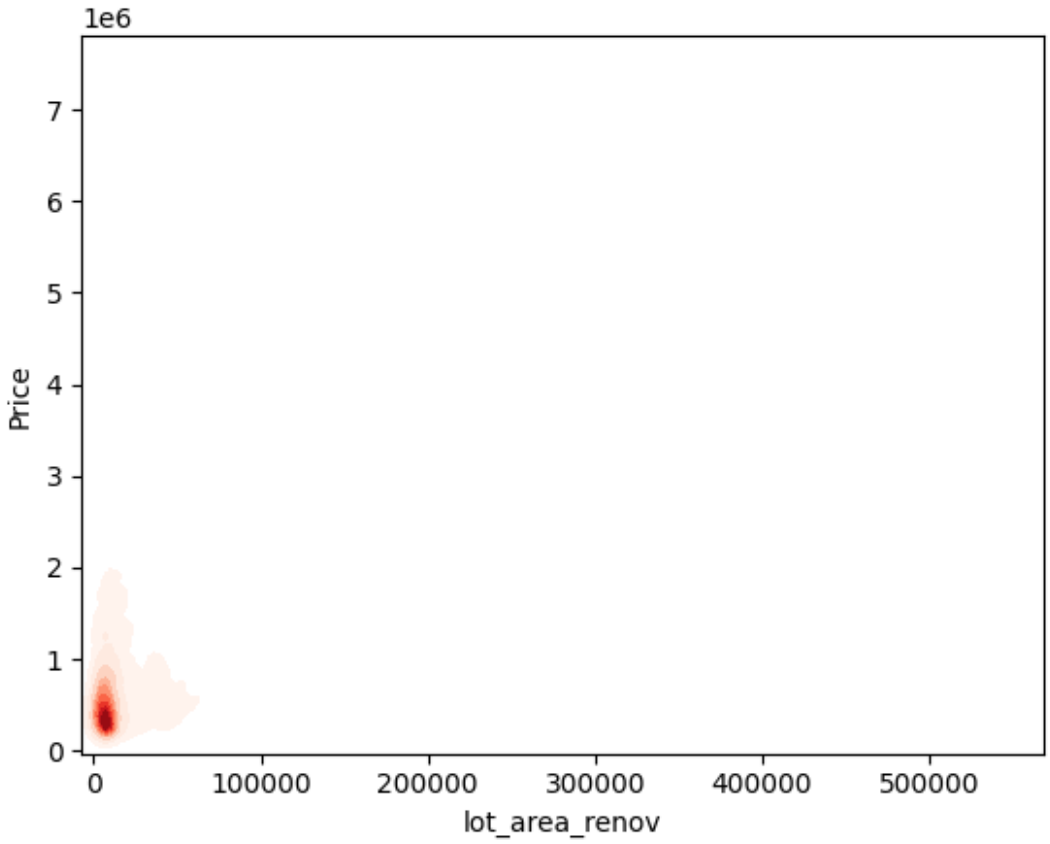
```
<Axes: xlabel='living_area', ylabel='Price'>
```



Analyzing the relation between lot area renovation and the price

```
sb.kdeplot(x=ds["lot_area_renov"],y=ds['Price'],cmap="Reds",fill=True,
bw_adjust=.5)
```

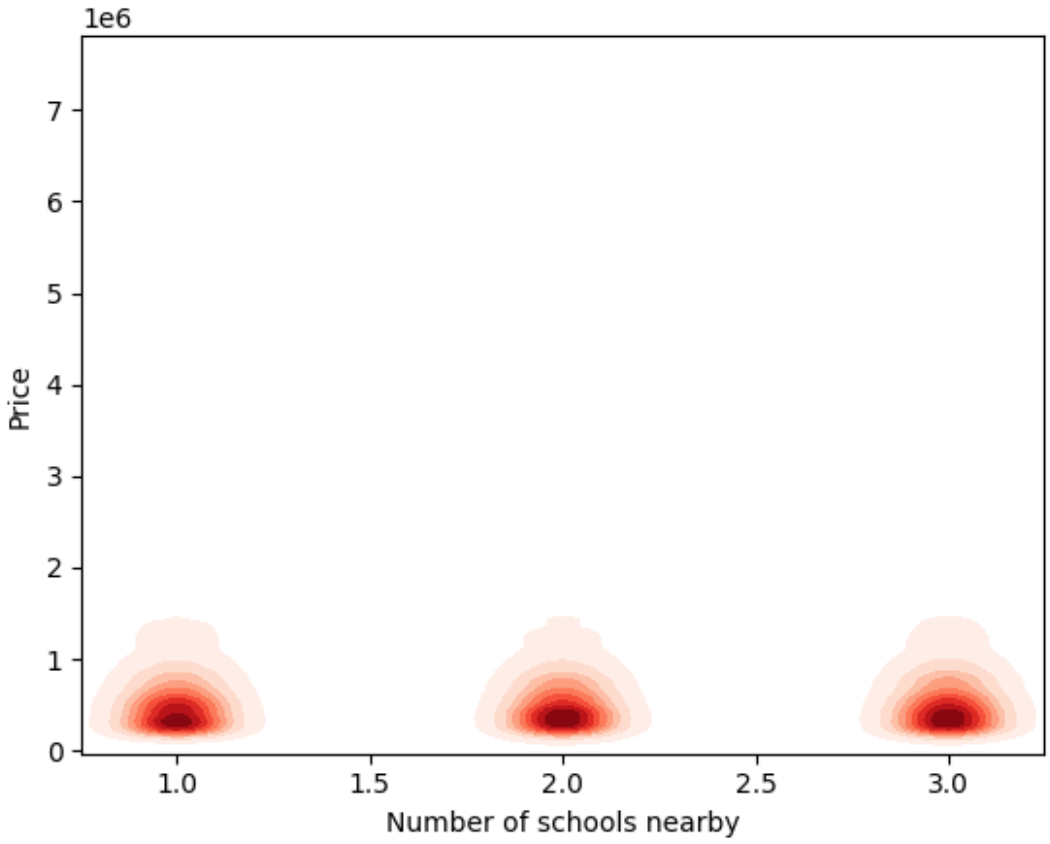
```
<Axes: xlabel='lot_area_renov', ylabel='Price'>
```



Analyzing the relation between number of schools nearby and the price

```
sb.kdeplot(x=ds["Number of schools
nearby"],y=ds['Price'],cmap="Reds",fill=True,bw_adjust=.5)
```

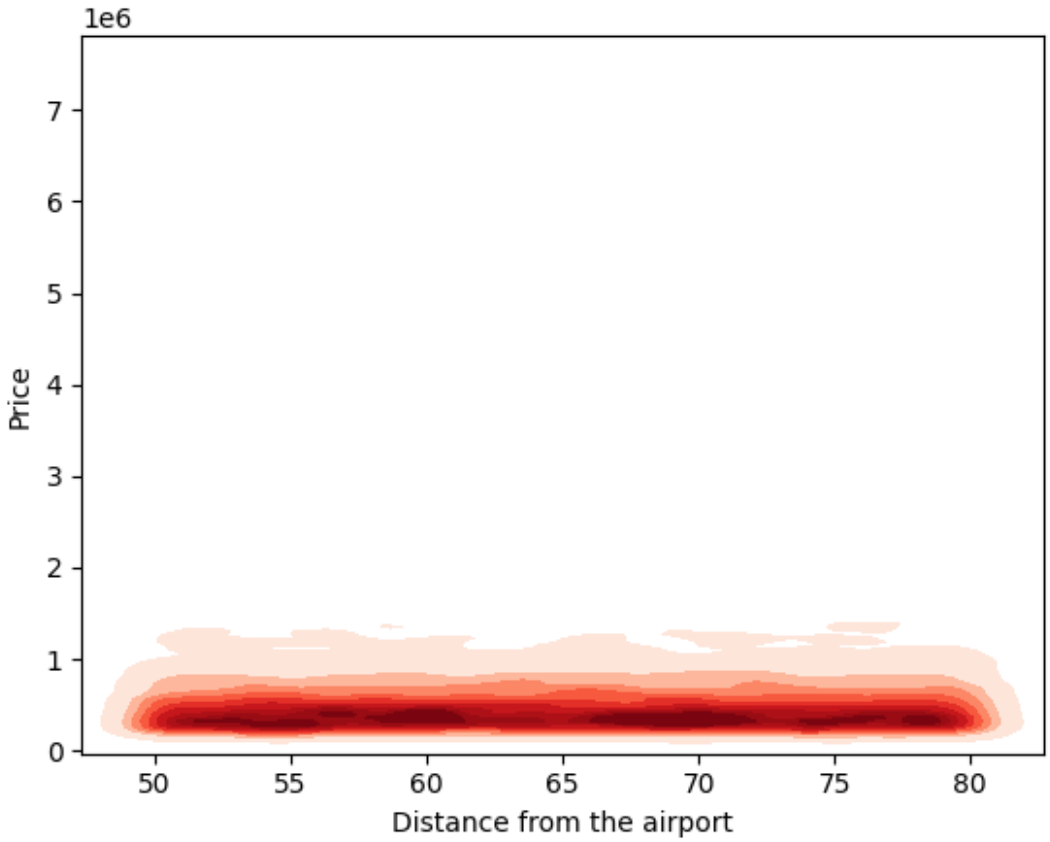
```
<Axes: xlabel='Number of schools nearby', ylabel='Price'>
```



Analyzing the relation between distance from the airport and the price

```
sb.kdeplot(x=ds["Distance from the
airport"],y=ds['Price'],cmap="Reds",fill=True,bw_adjust=.5)
```

```
<Axes: xlabel='Distance from the airport', ylabel='Price'>
```



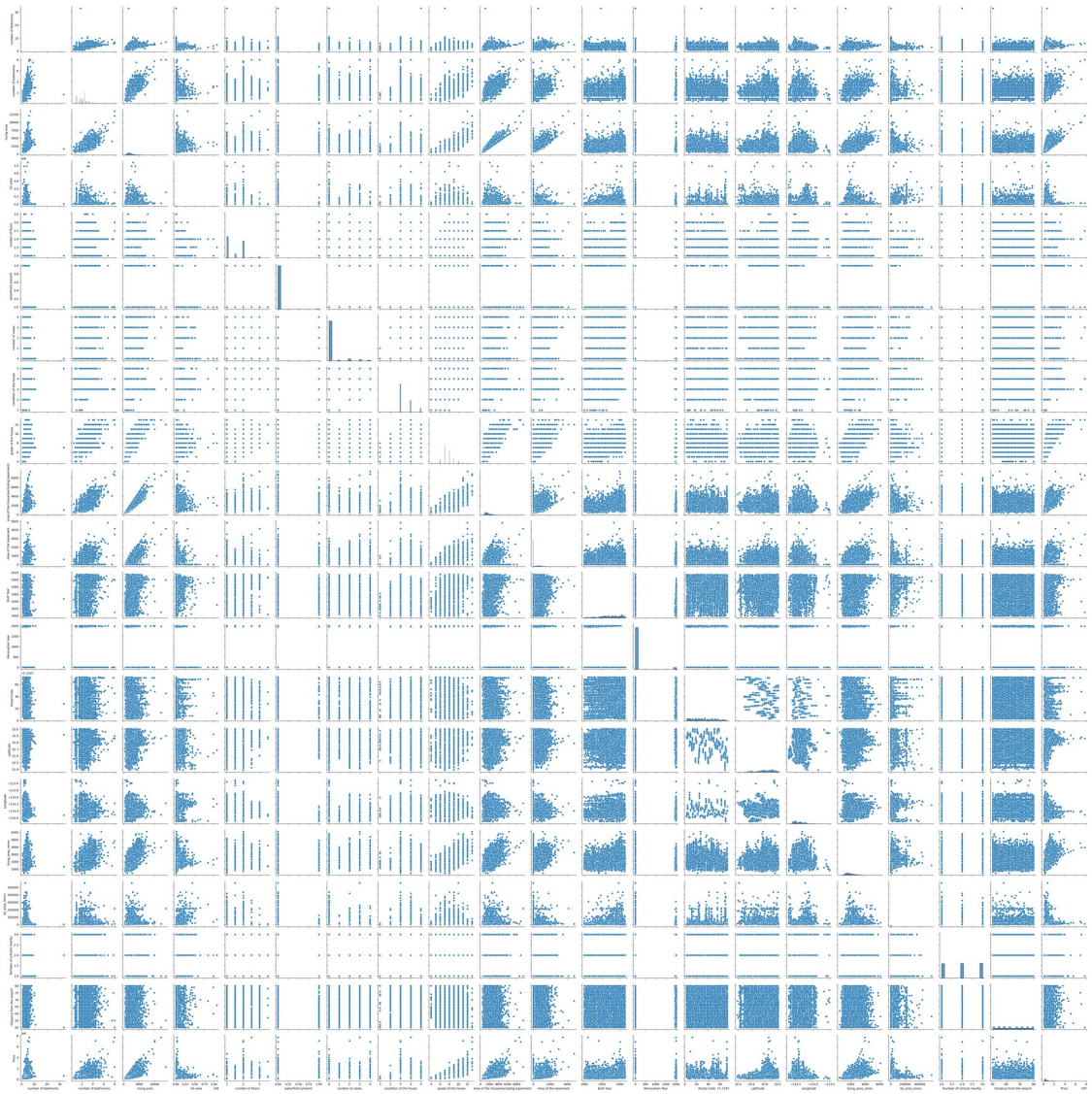
The results when the attributes are compared to the price of the house are as in the above charts. we can see that some houses share more common attributes with each other and they are represented in darker colour. The shade on the graph represents the density of the houses with similar attributes and price points.

1C. Multivariate analysis:

Now lets perform multivariate analysis for the given dataset of the houses.

```
sb.pairplot(data=ds[["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","Nu
mber of schools nearby","Distance from the airport","Price"]])
```

```
<seaborn.axisgrid.PairGrid at 0x7fa114352640>
```



In the above picture, we can see multiple graphs that represent the relations os the attrributes with each other. Each of the attributes are compared to all the other attributes using a matrix of graphs or charts. Those charts are as represented in the above output. w can observe that some of the attributes share more similarities than the other attributes. Such attributes are related more closely than the other scattered attributes.

2. Descriptive statistics

Lets get the descriptive statistics now

```
ds.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
max                    80.000000  7.700000e+06

[8 rows x 23 columns]
```

We can see the statistics of the given dataset from the above table. Now lets perform analysis with graphs and charts

3. Missing value

now lets check for null values and data types

In each of the columns, the non-null count is same which implies that no column contains null values which implies there are no empty cells.

```
print(ds.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
None
```