

Name: Naga Venkata Prabhu Shobith Paripalli

Registration Number: 21BAI1722

Campus: Chennai

G-mail: shobith.paripalli2021@vitstudent.ac.in

```
In [ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [ ]: df = pd.read_csv(r"C:\Users\shobi\Downloads\archive\House Price India.csv")
df.head()
```

Out[]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condit of ho
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 [ ]: df.describe()
```

Out[]:

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

8 rows × 23 columns

◀ ▶

In []: `df.isnull().sum()`

Out[]:

<code>id</code>	0
<code>Date</code>	0
<code>number of bedrooms</code>	0
<code>number of bathrooms</code>	0
<code>living area</code>	0
<code>lot area</code>	0
<code>number of floors</code>	0
<code>waterfront present</code>	0
<code>number of views</code>	0
<code>condition of the house</code>	0
<code>grade of the house</code>	0
<code>Area of the house(excluding basement)</code>	0
<code>Area of the basement</code>	0
<code>Built Year</code>	0
<code>Renovation Year</code>	0
<code>Postal Code</code>	0
<code>Latitude</code>	0
<code>Longitude</code>	0
<code>living_area_renov</code>	0
<code>lot_area_renov</code>	0
<code>Number of schools nearby</code>	0
<code>Distance from the airport</code>	0
<code>Price</code>	0
<code>dtype: int64</code>	

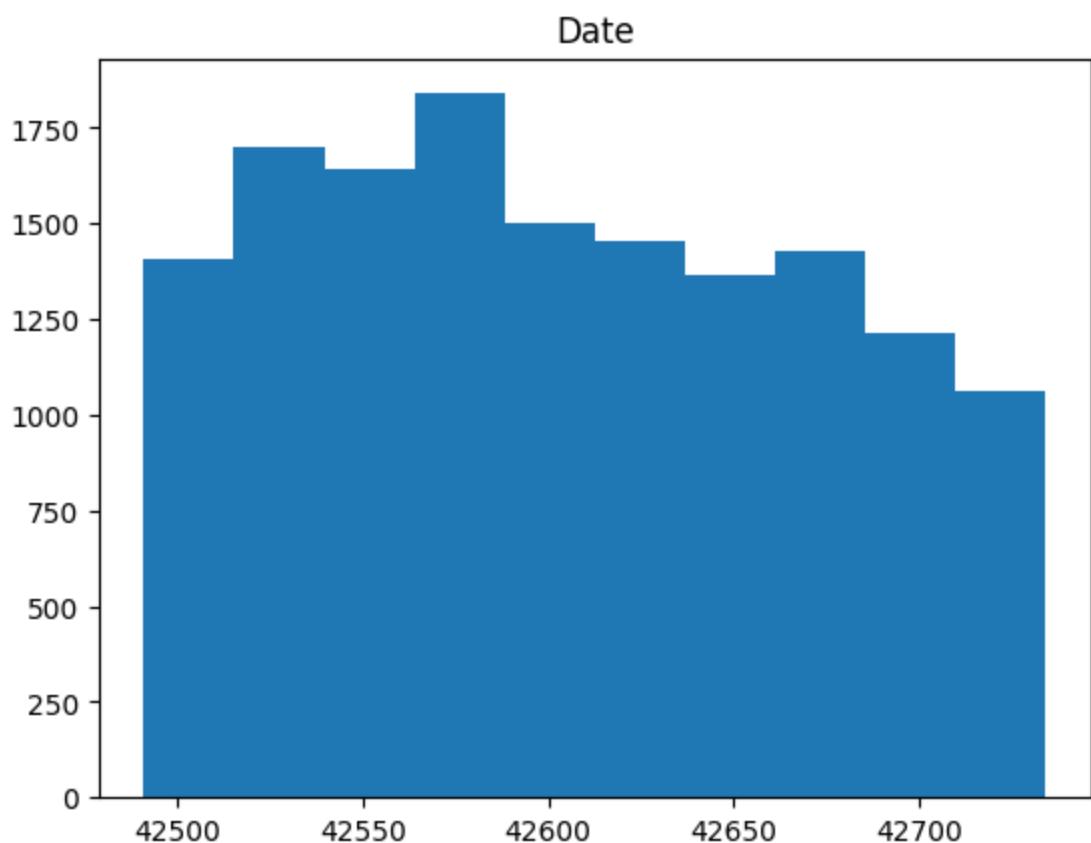
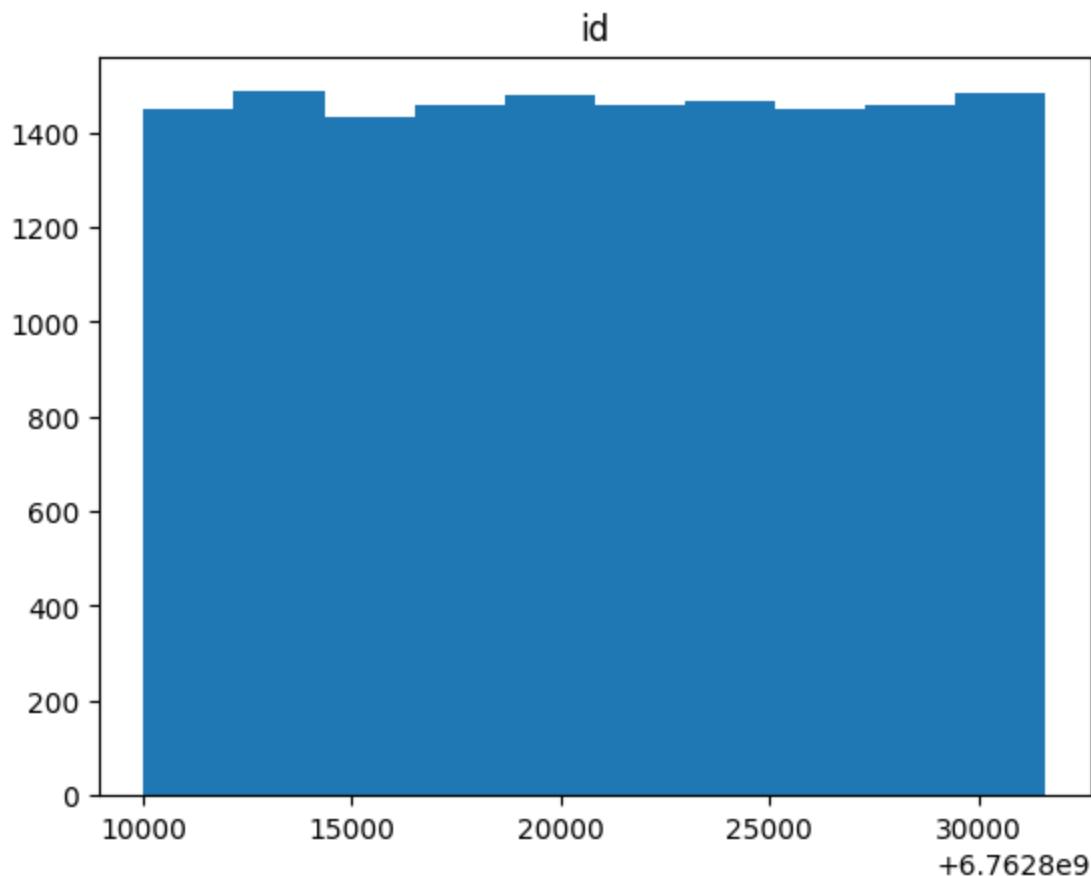
In []: `df.dtypes`

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

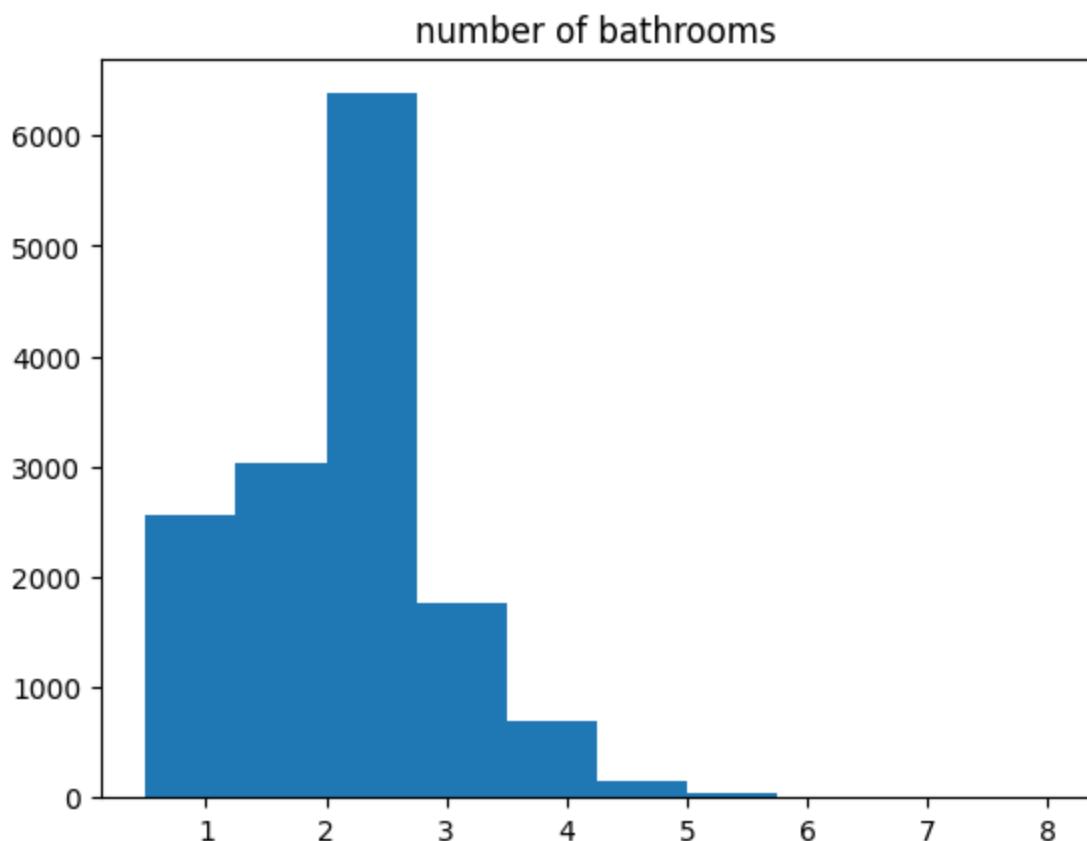
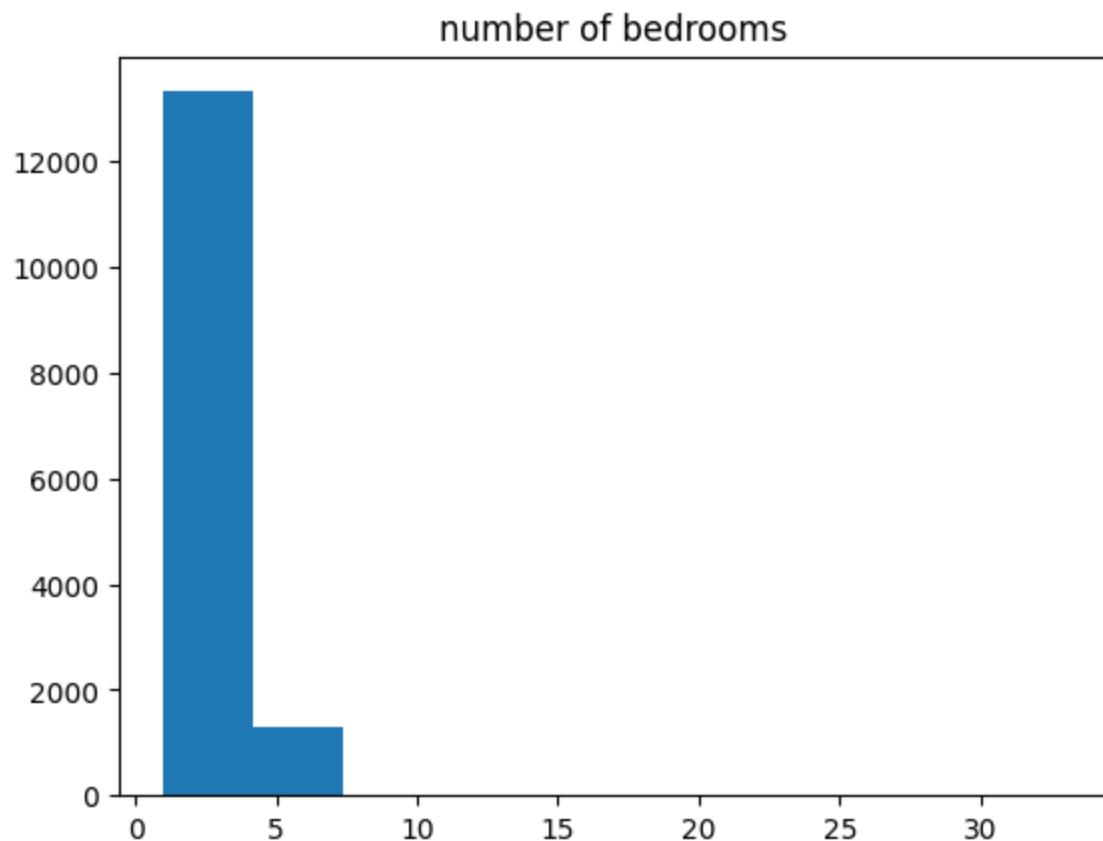
```
Out[ ]: id                      int64
        Date                     int64
        number of bedrooms       int64
        number of bathrooms      float64
        living area                int64
        lot area                     int64
        number of floors          float64
        waterfront 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
        Latitude                    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
```

```
In [ ]: # Univariate Analysis

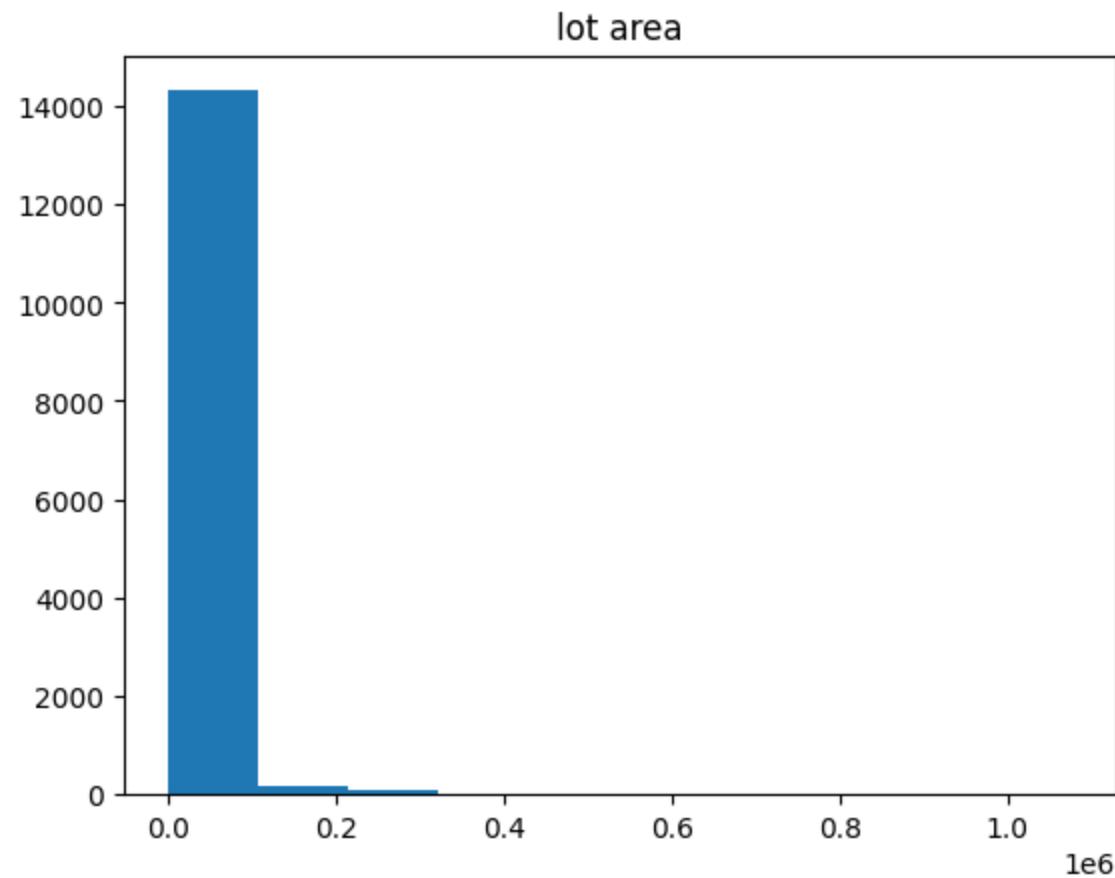
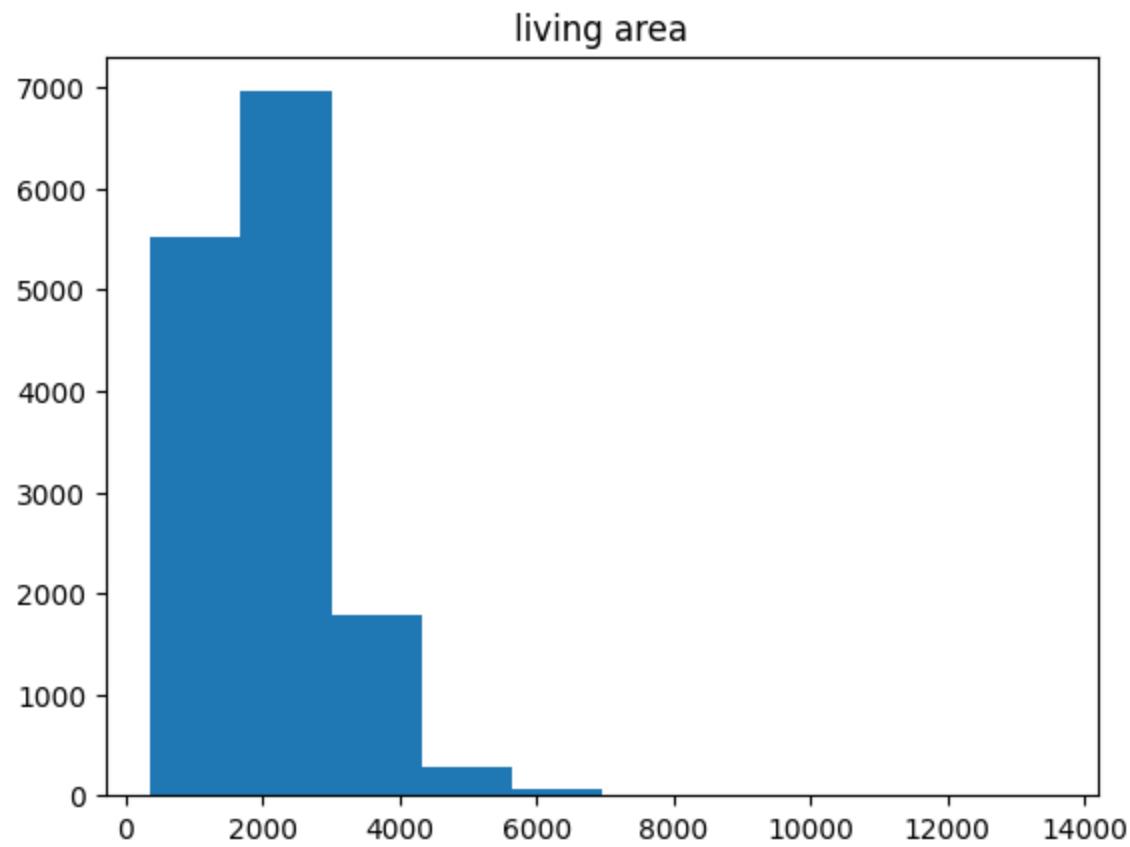
for i in df.columns:
    plt.hist(df[i])
    plt.title(i)
    plt.show()
```



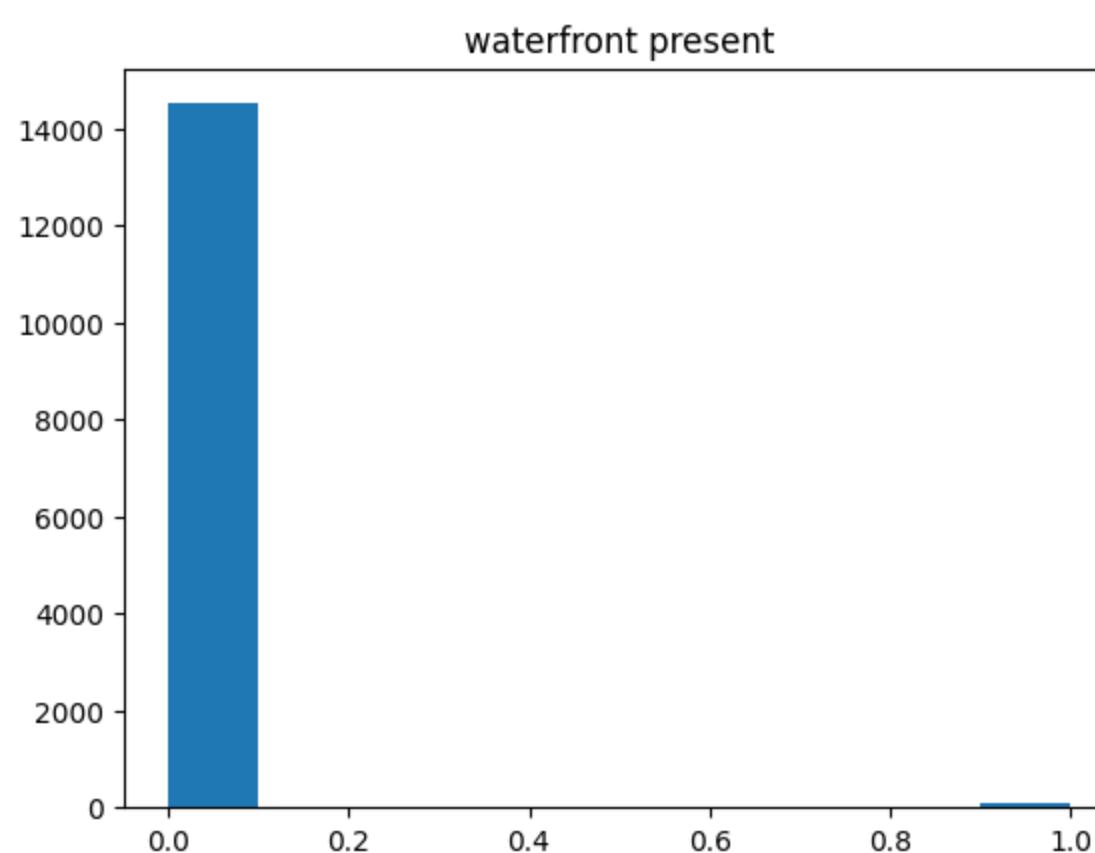
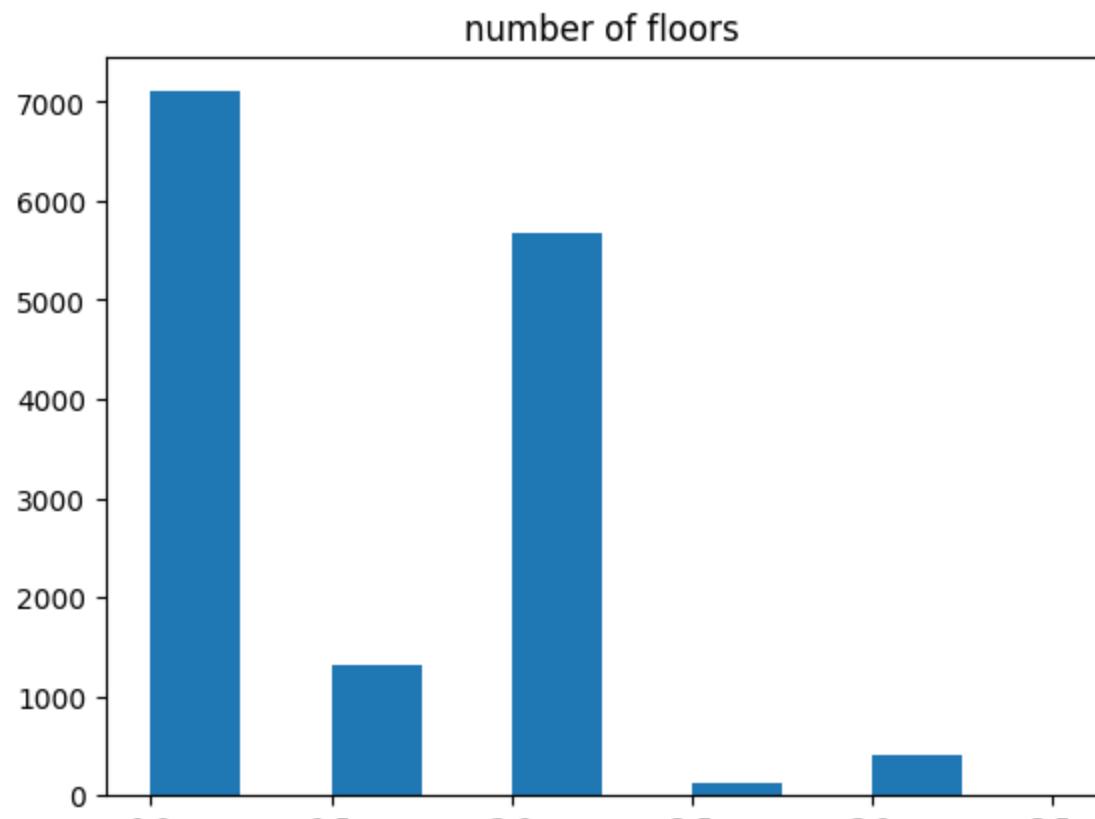
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



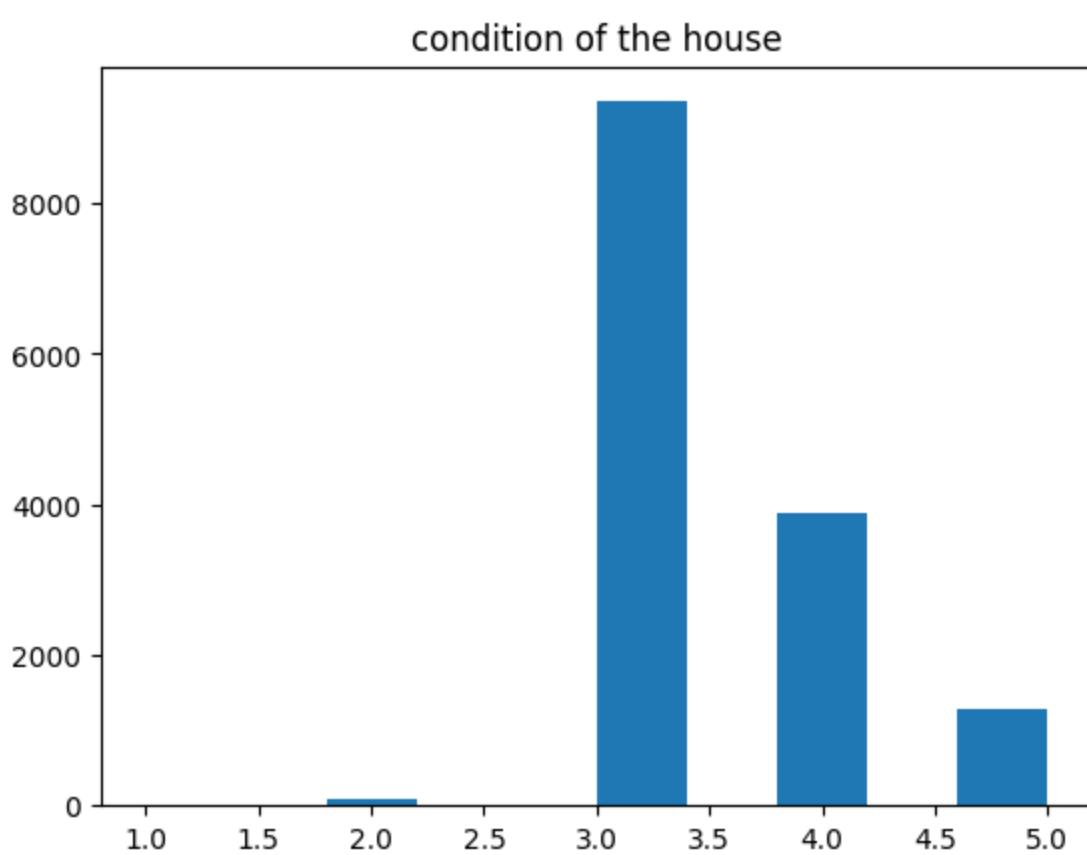
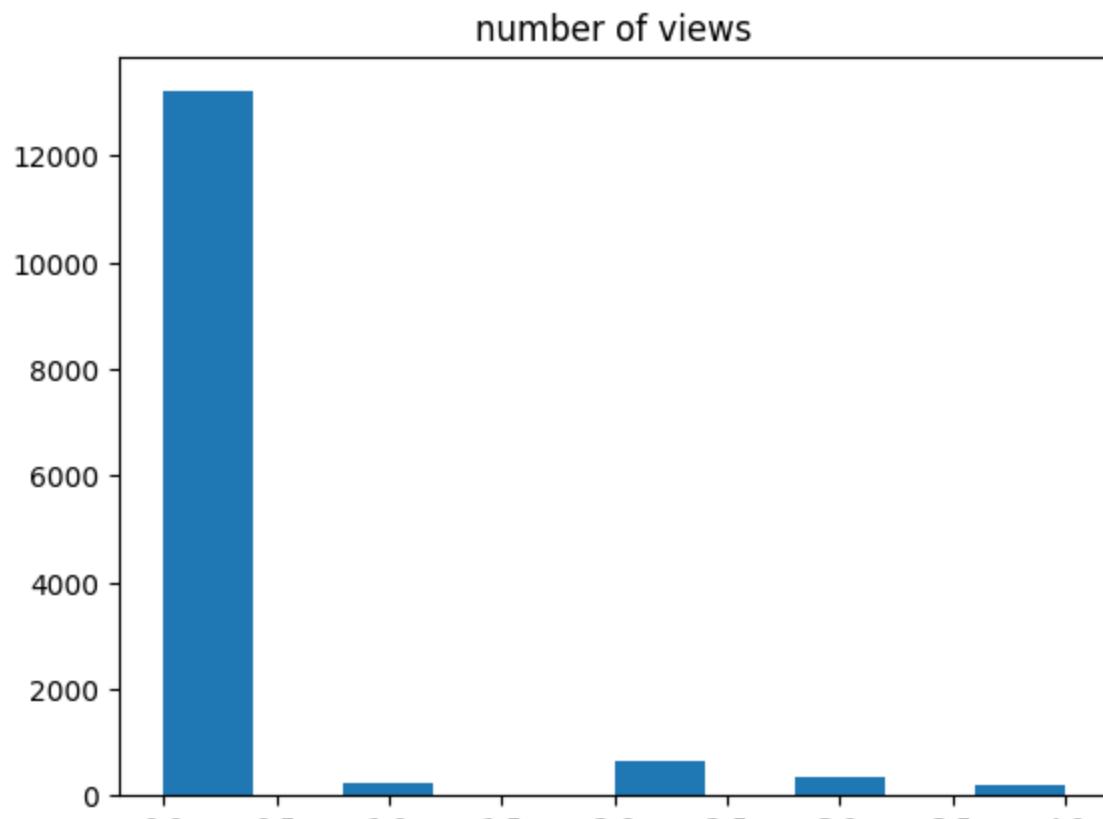
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



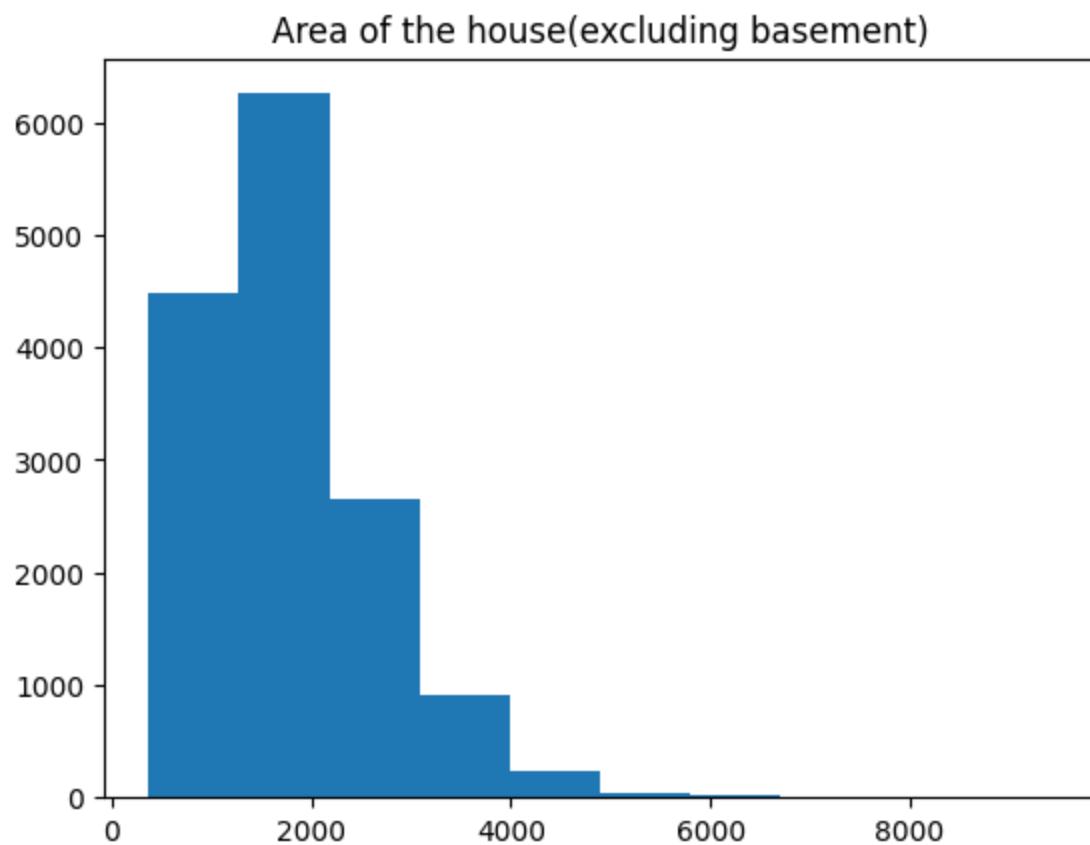
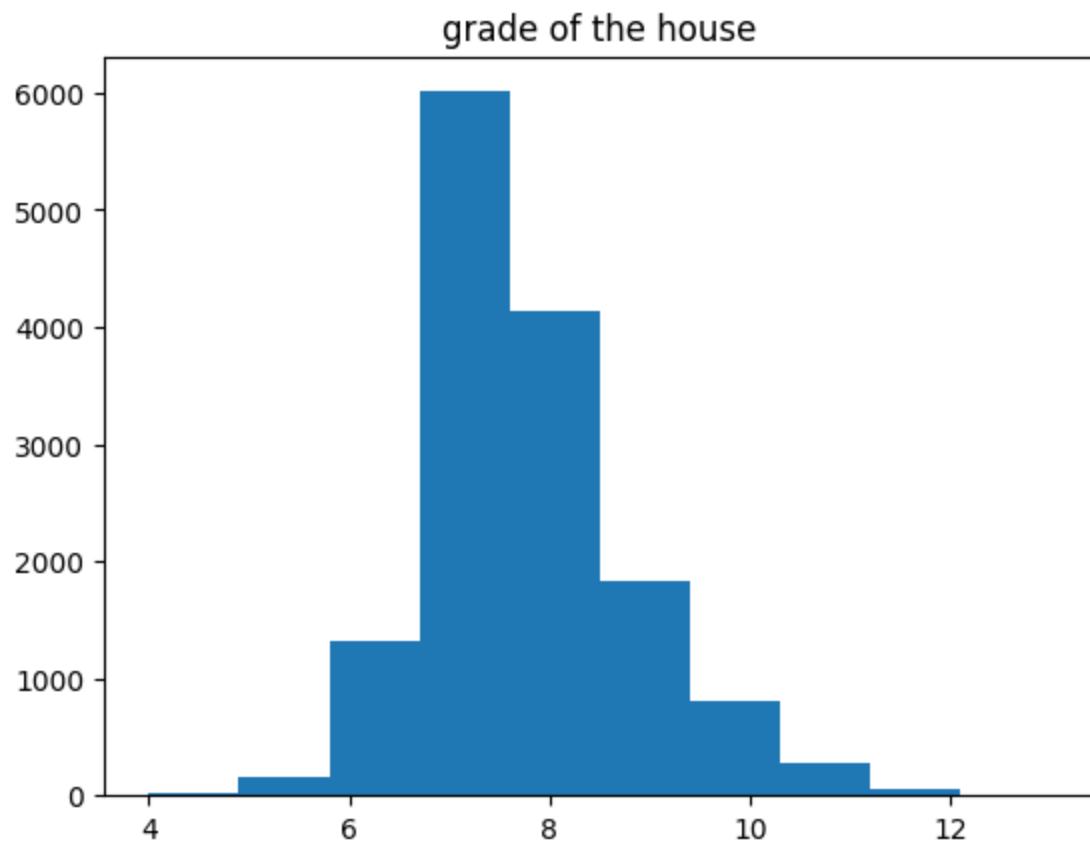
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

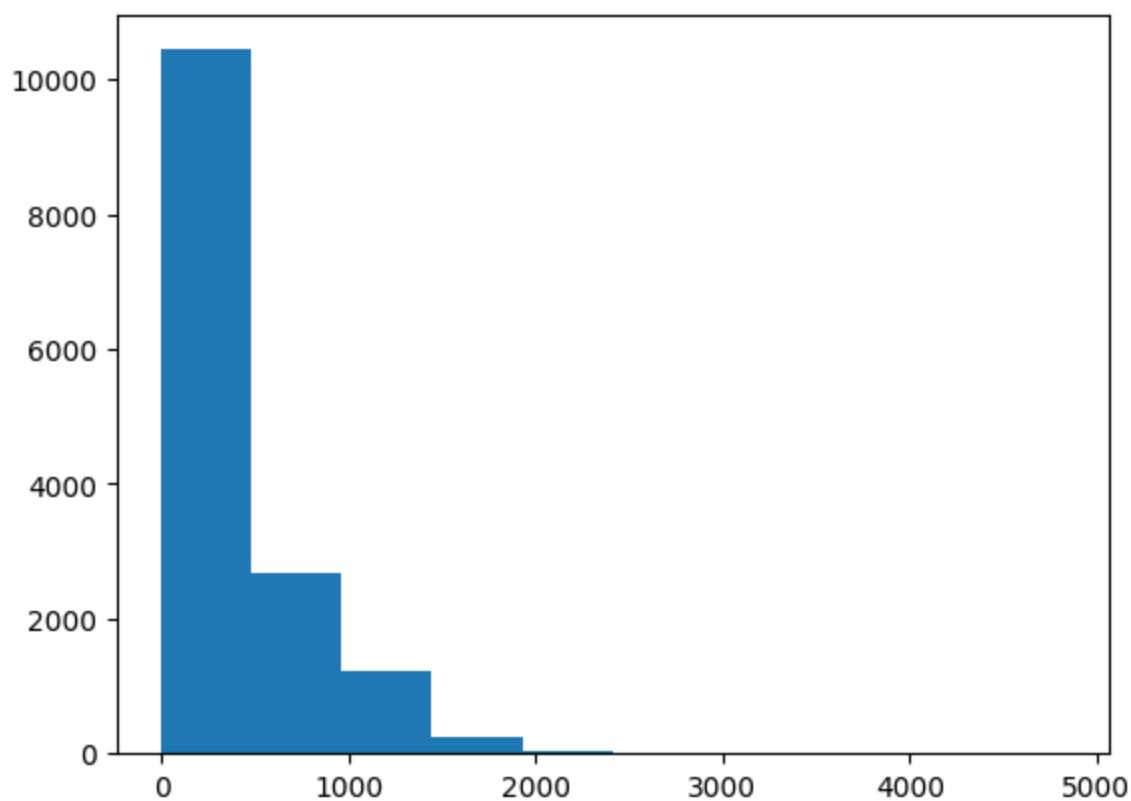


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

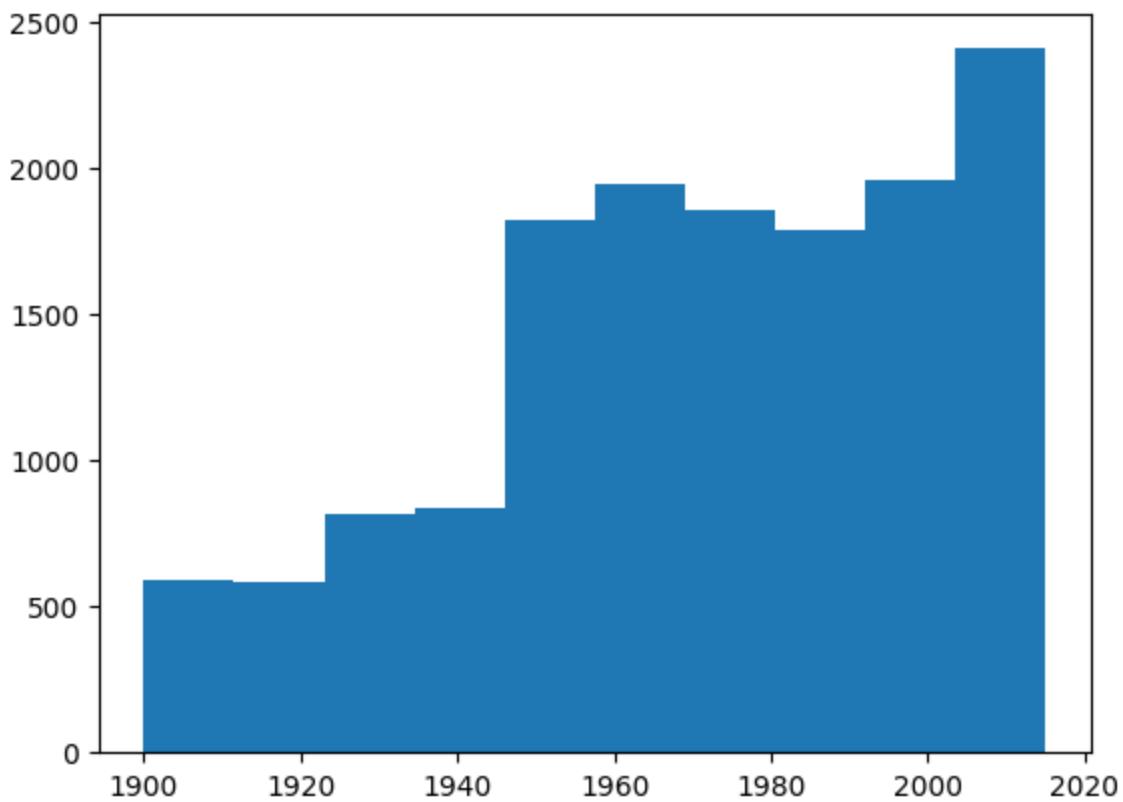


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

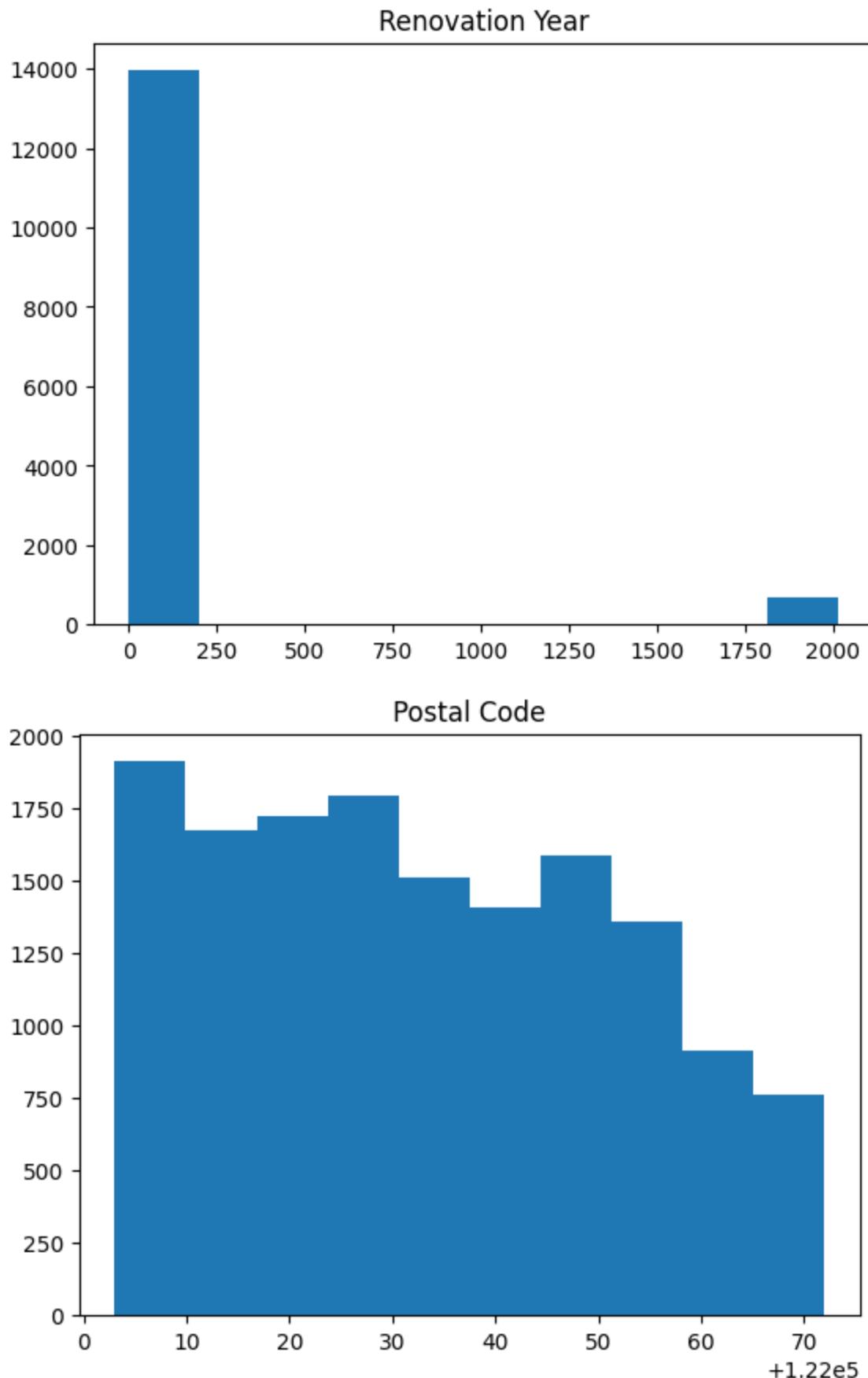
Area of the basement



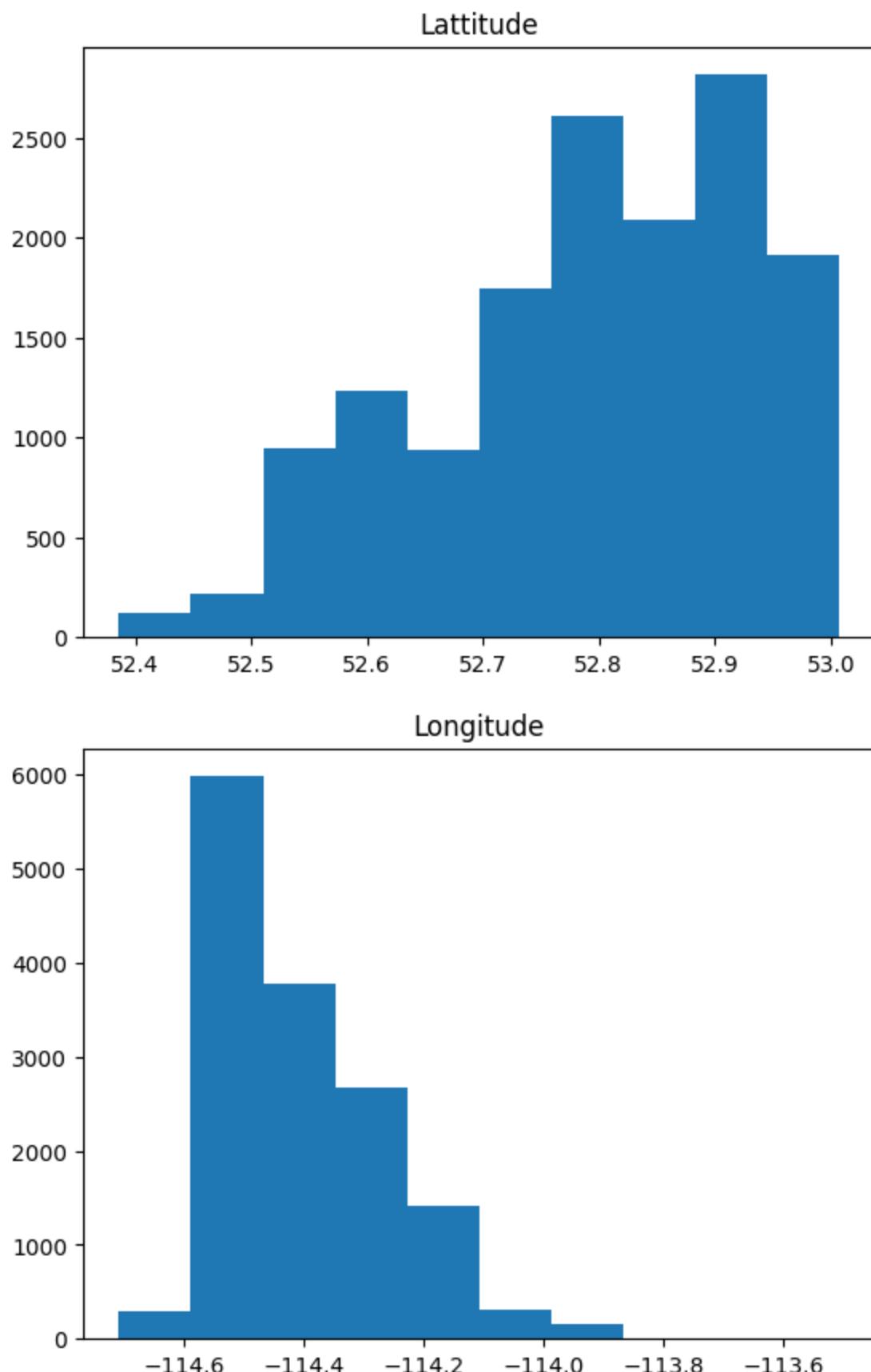
Built Year



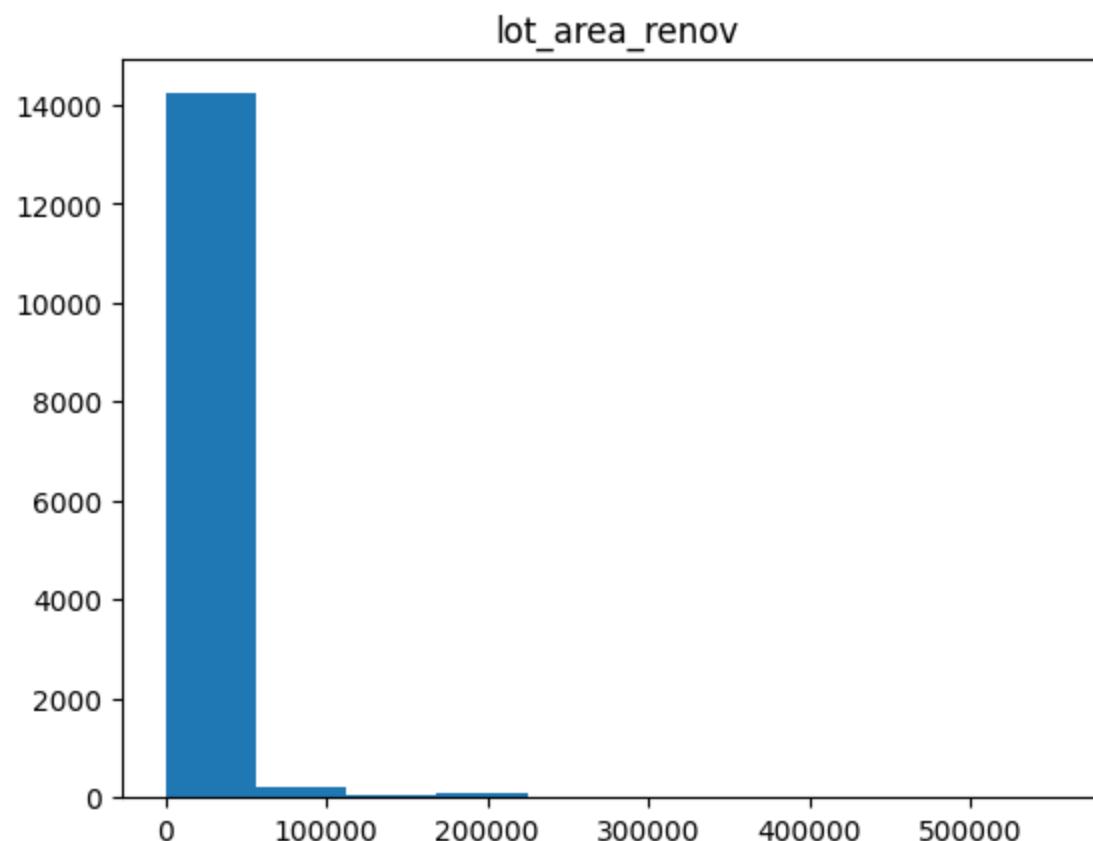
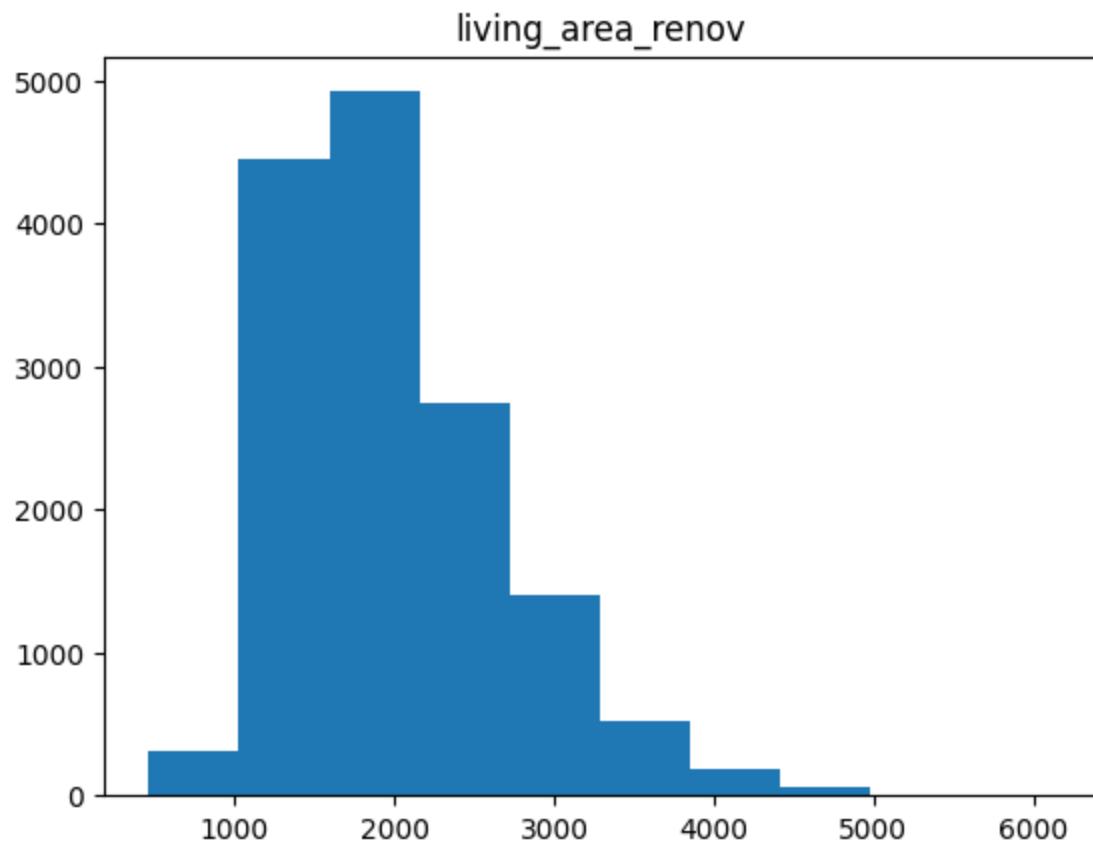
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

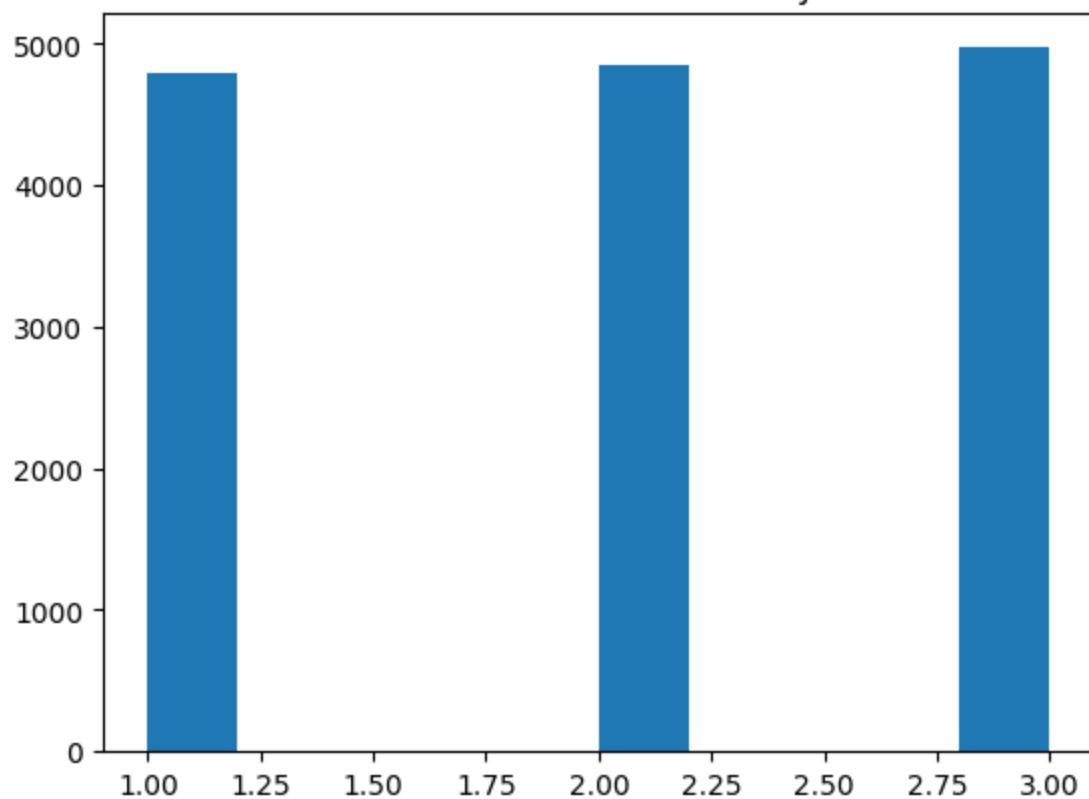


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

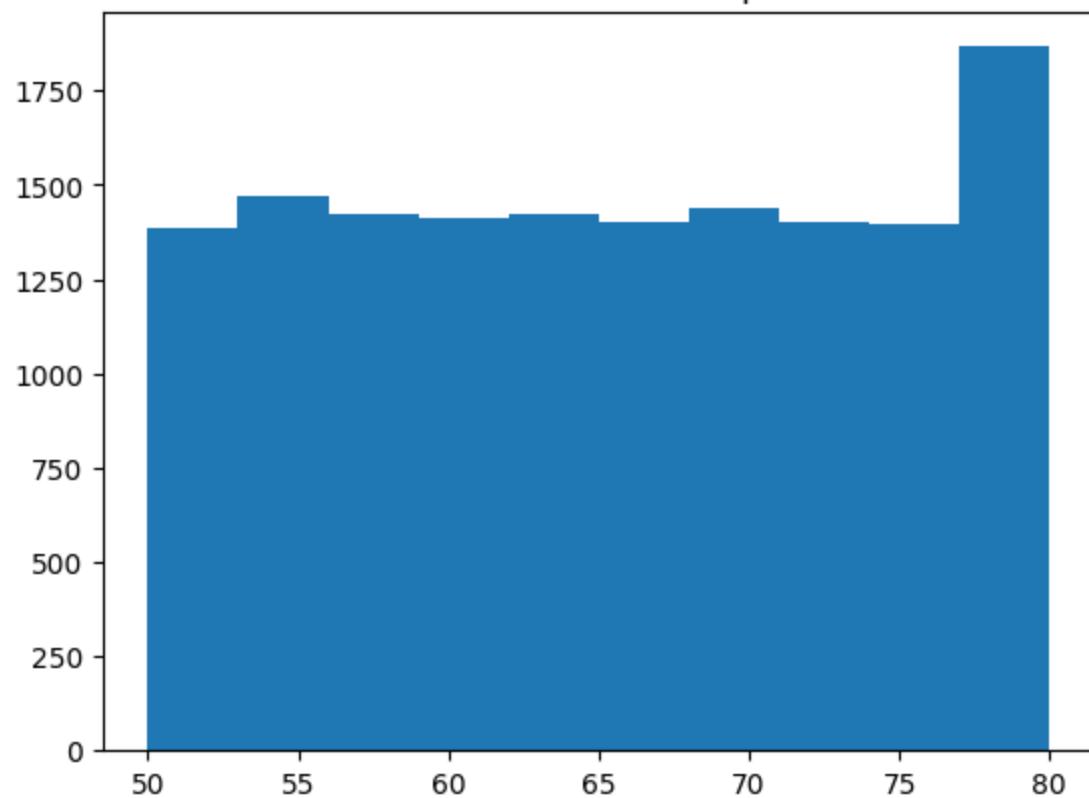


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

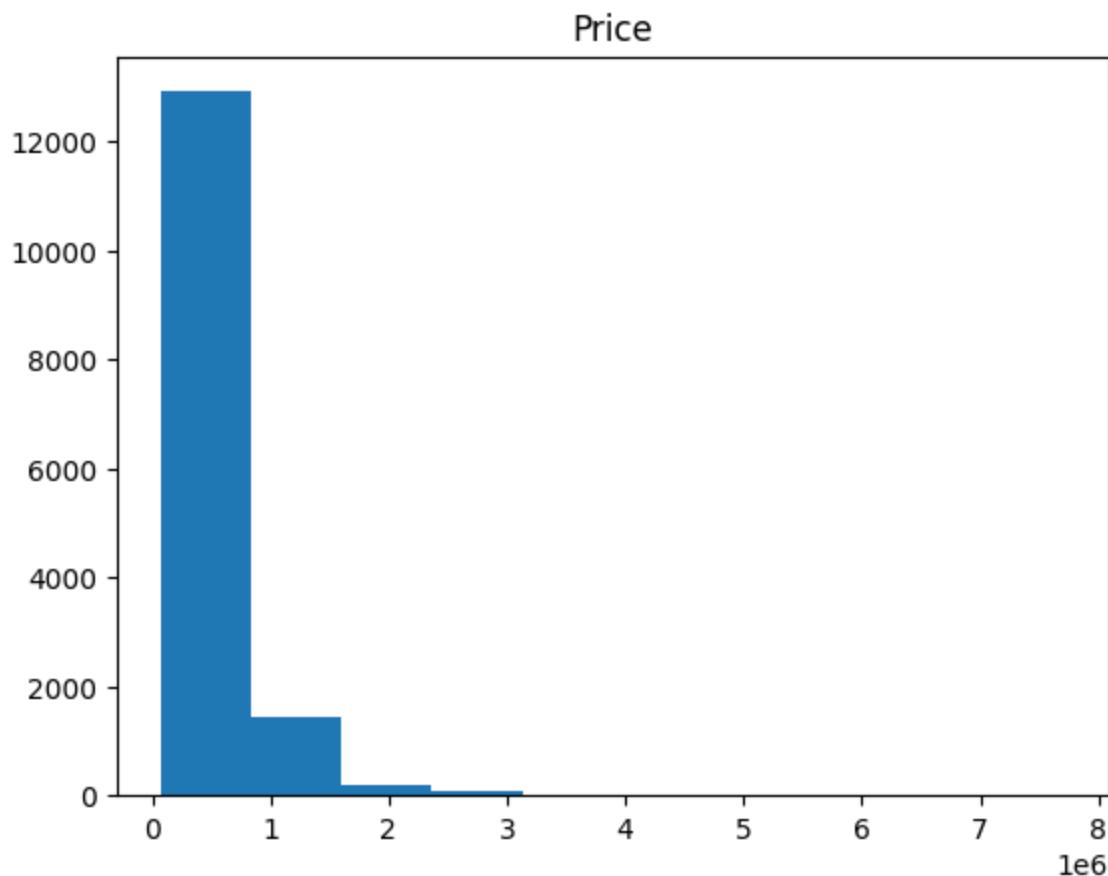
Number of schools nearby



Distance from the airport

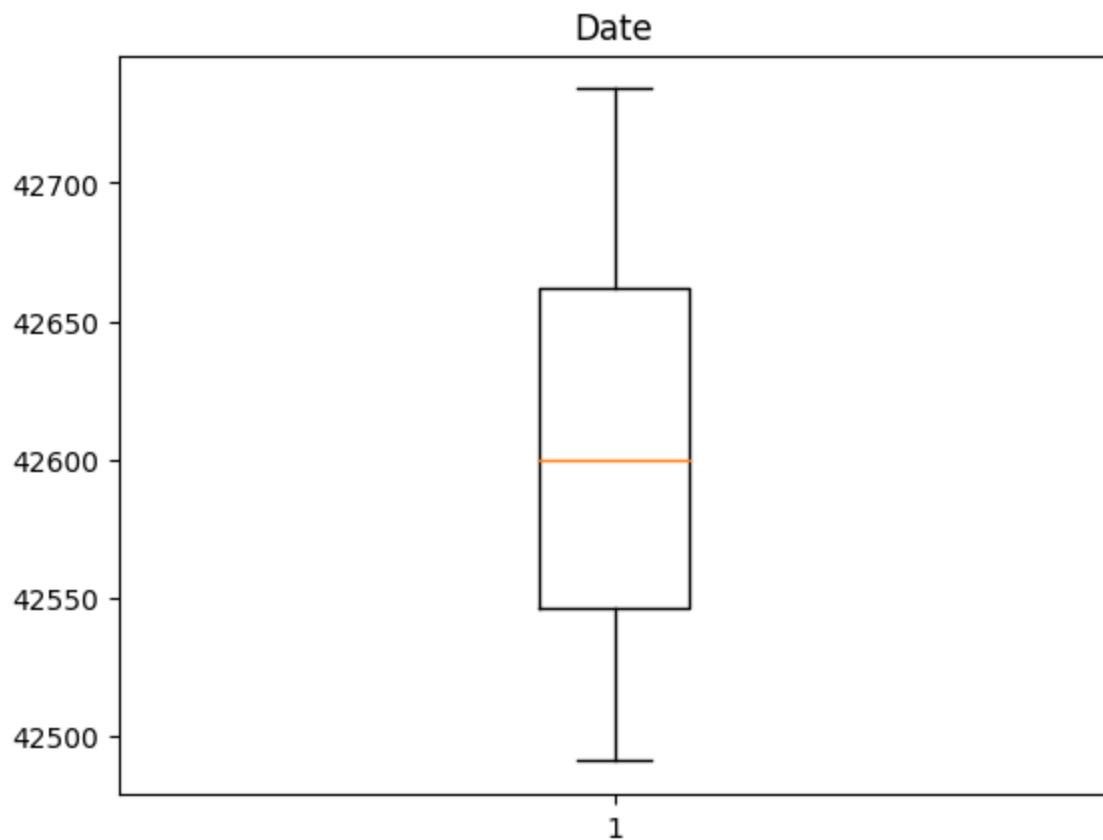
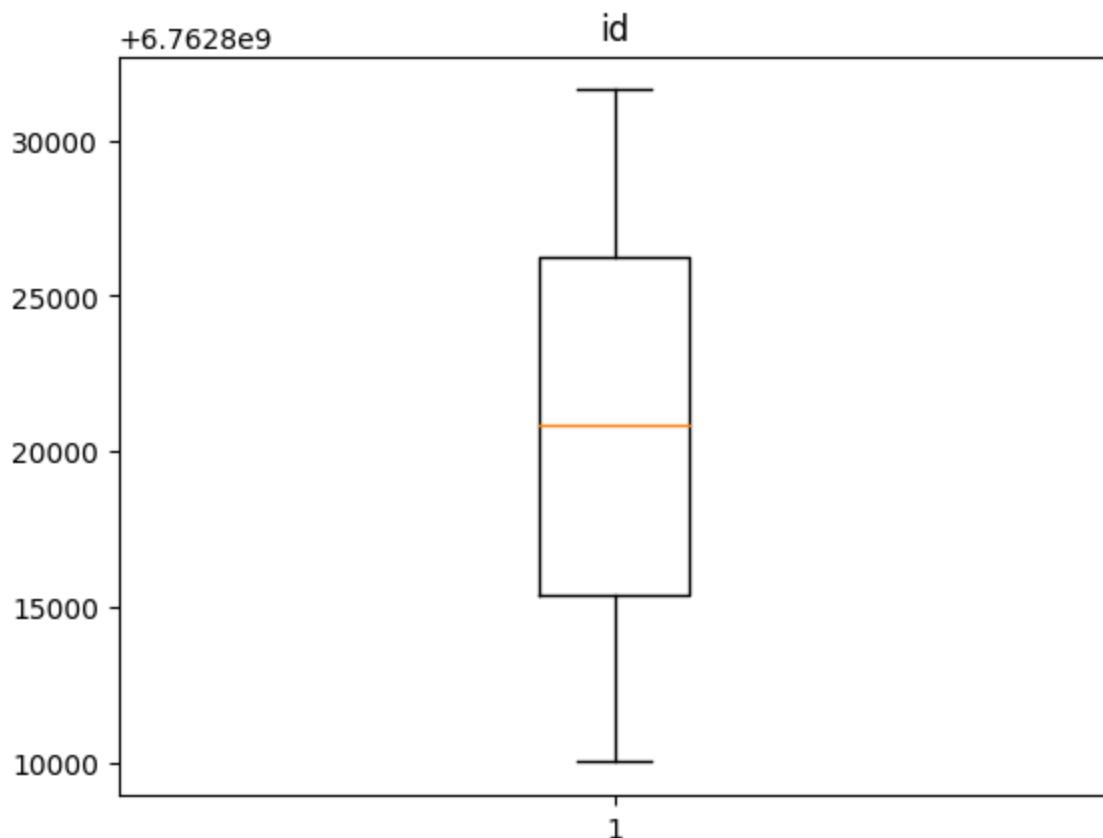


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

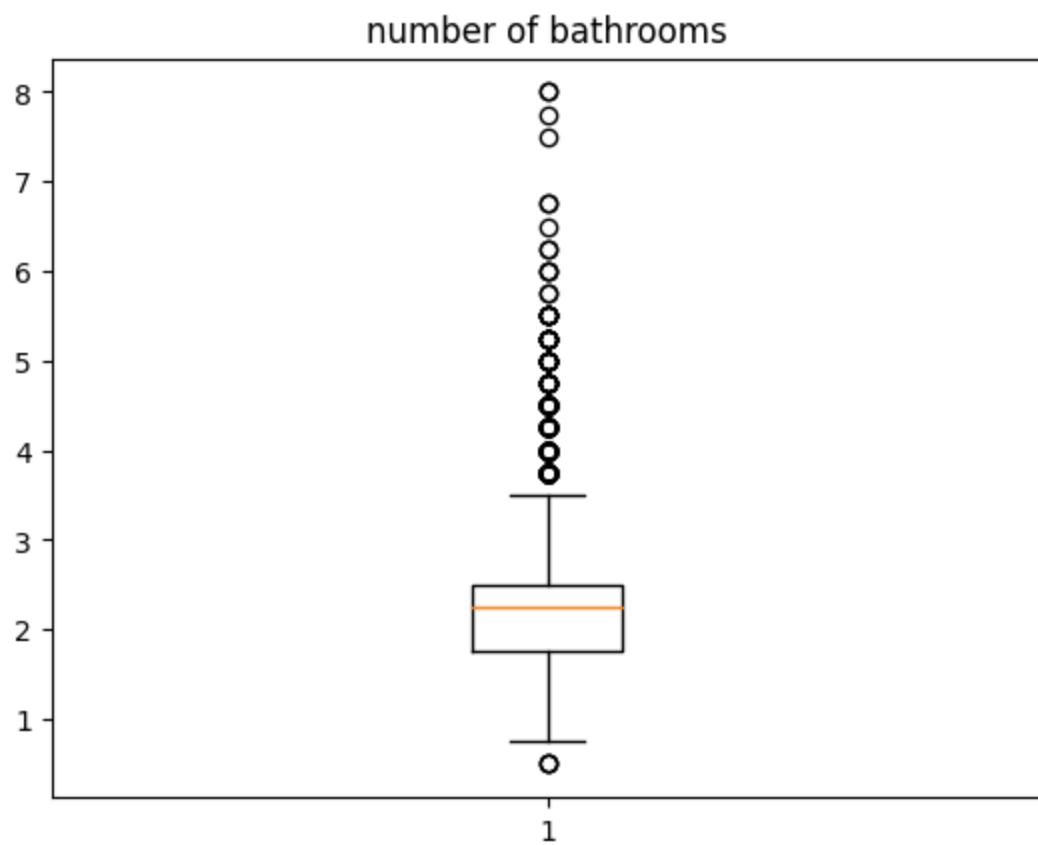
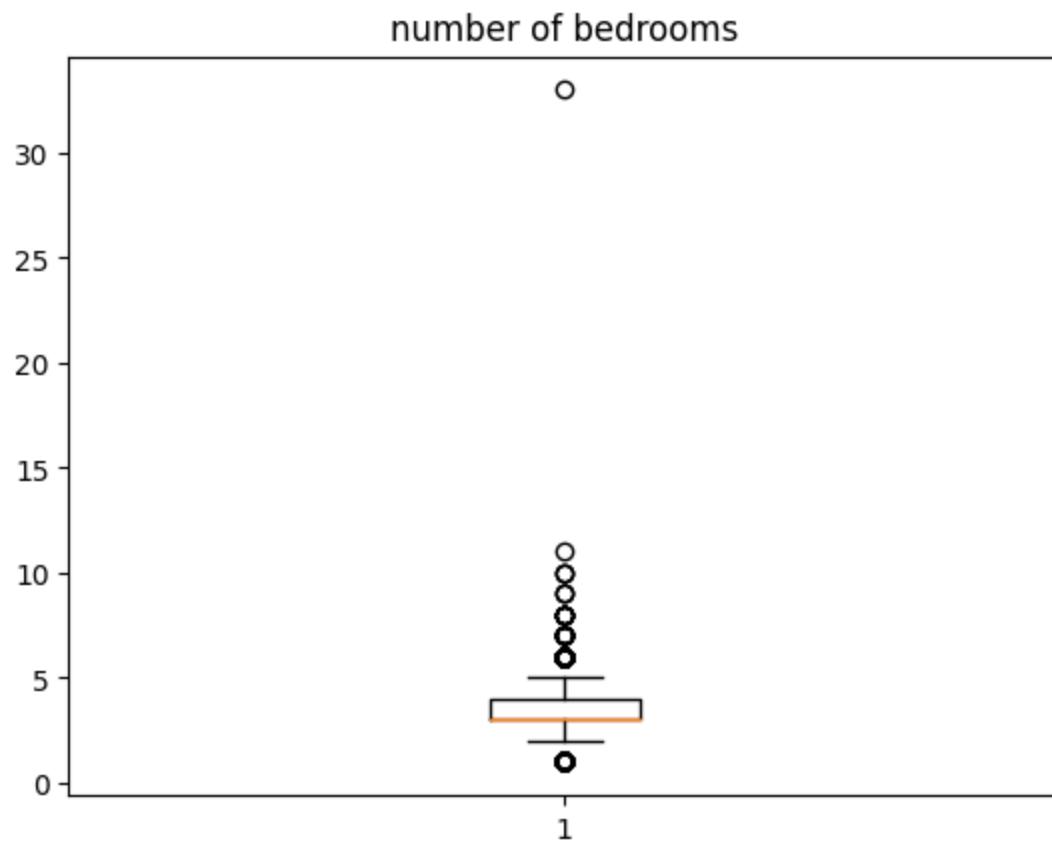


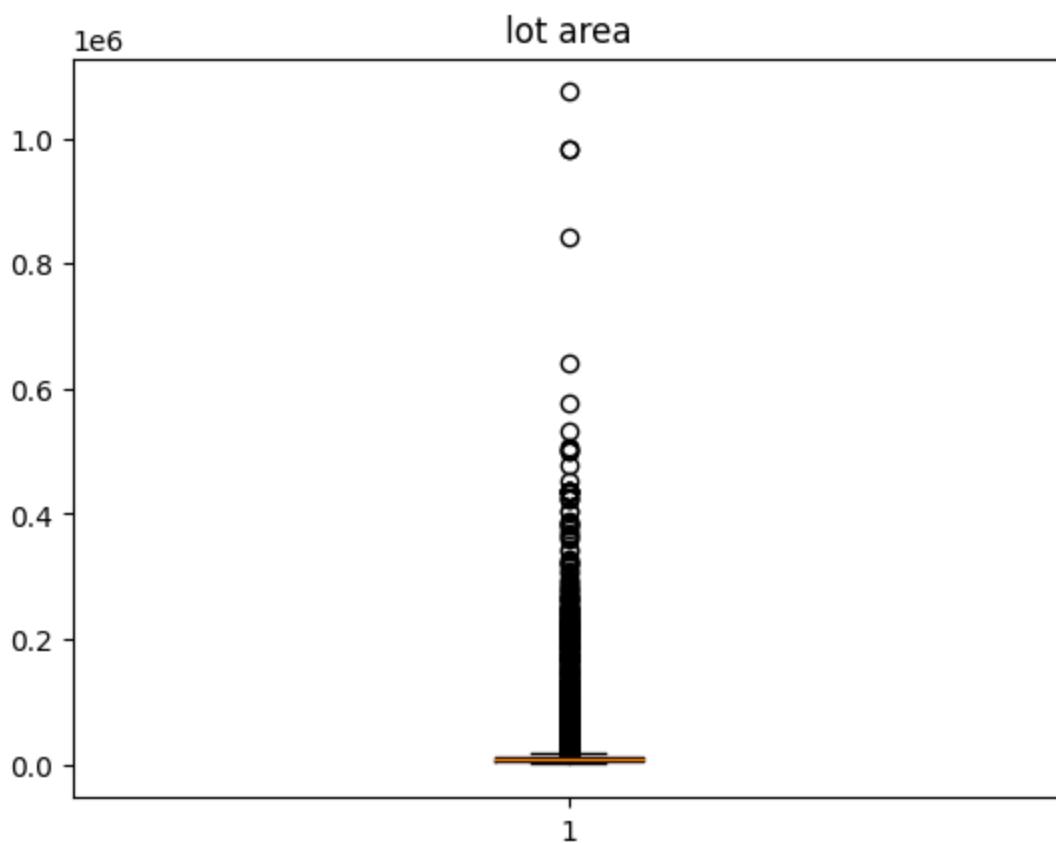
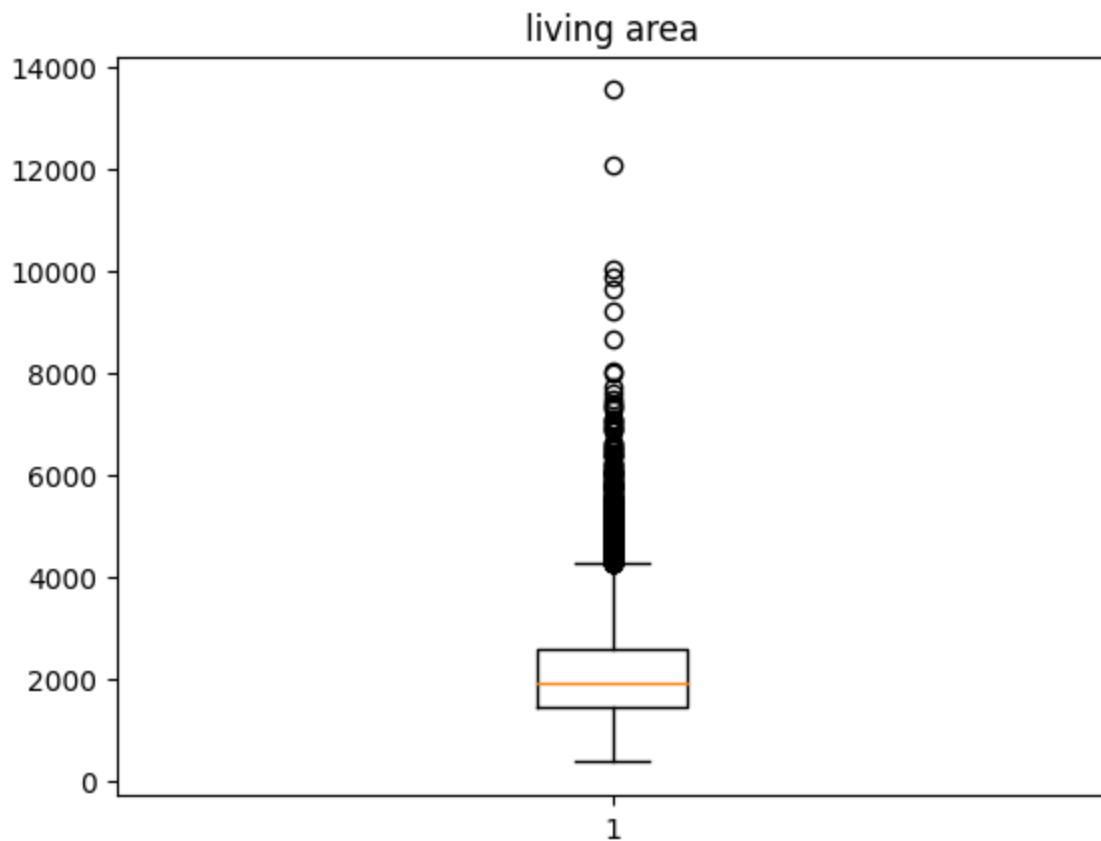
```
In [ ]: for i in df.columns:  
    plt.boxplot(df[i])  
    plt.title(i)  
    plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

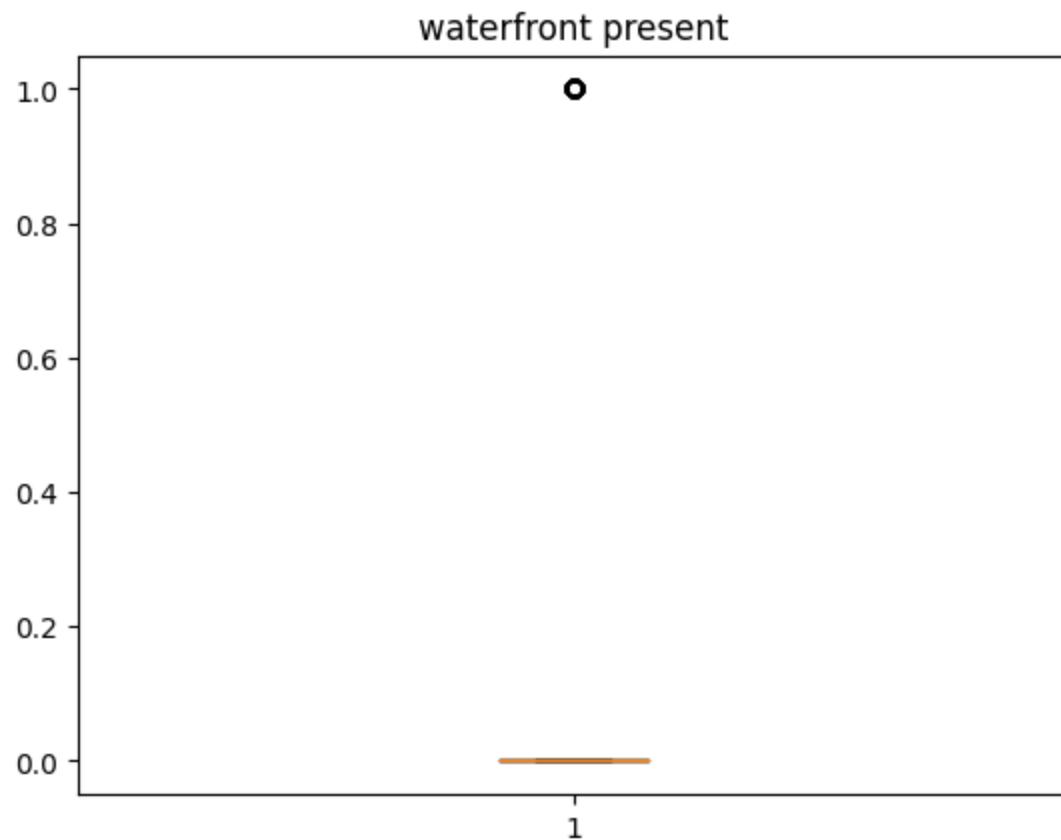
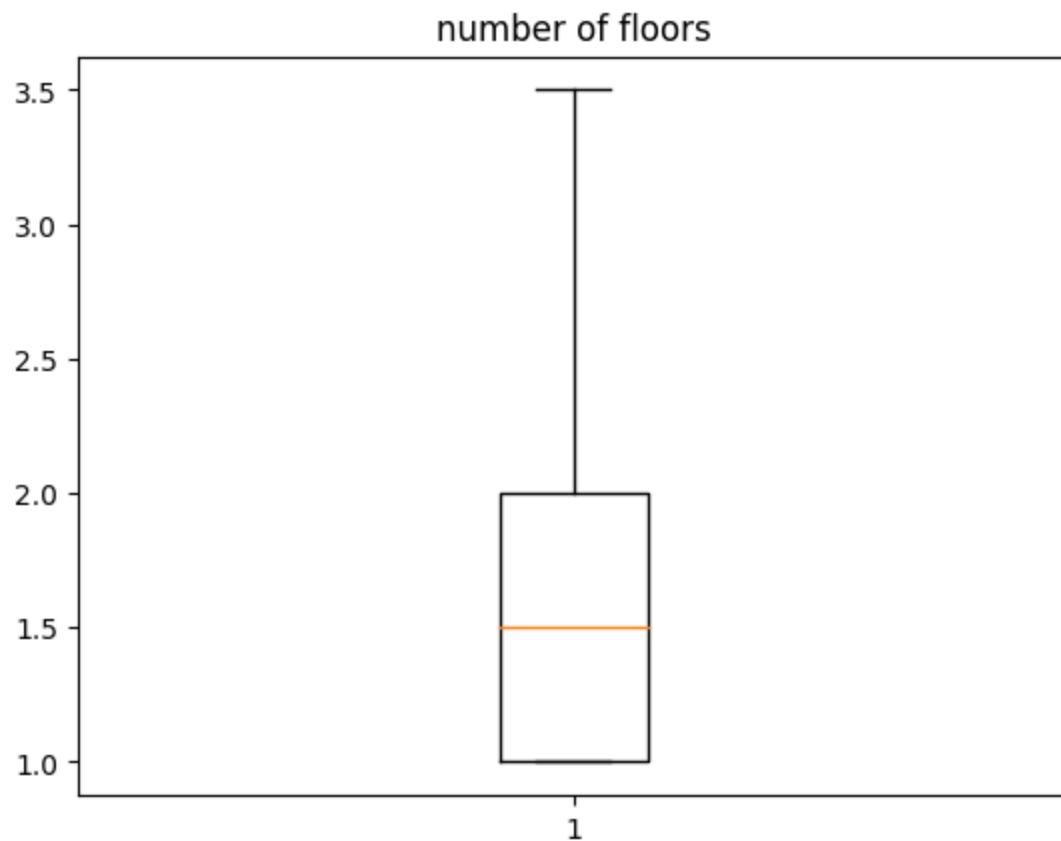


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

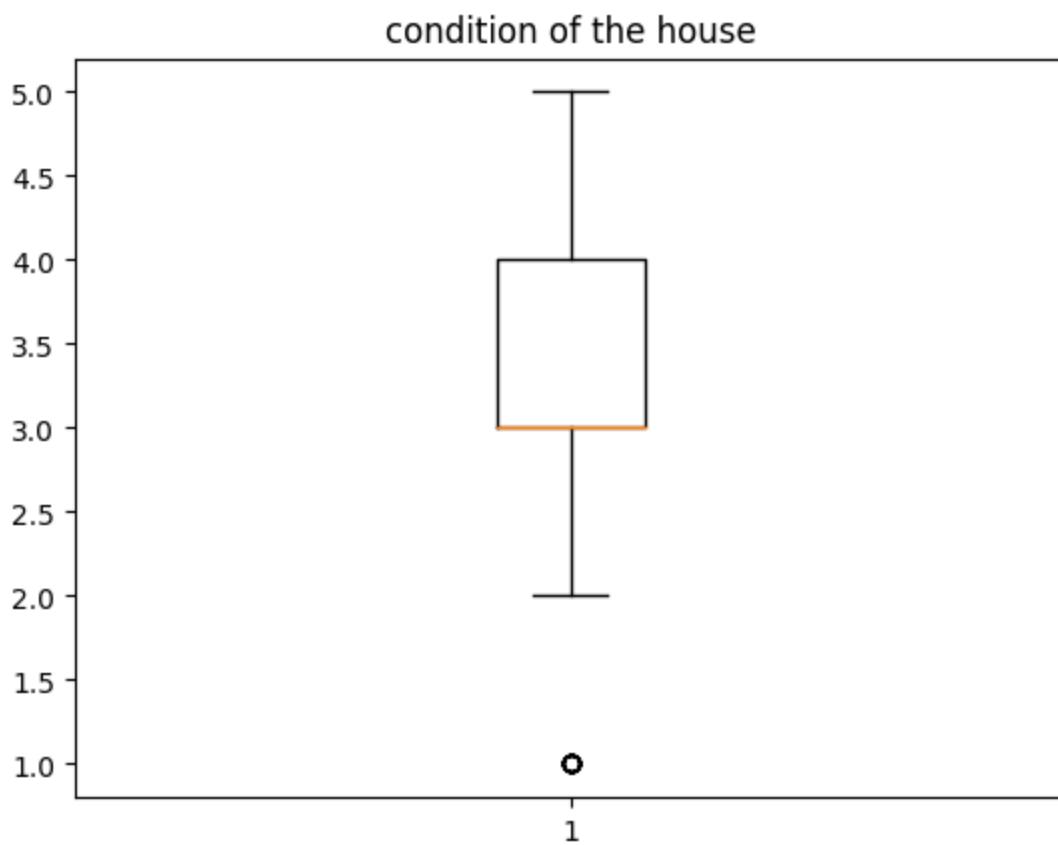
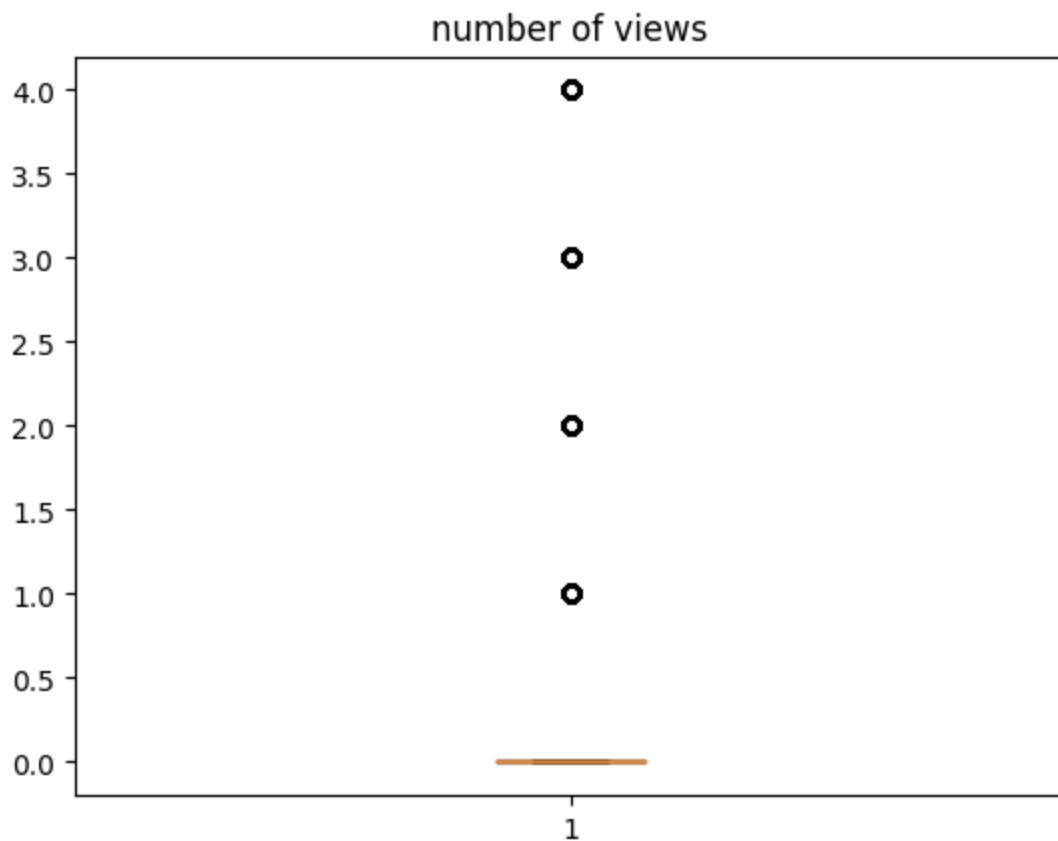




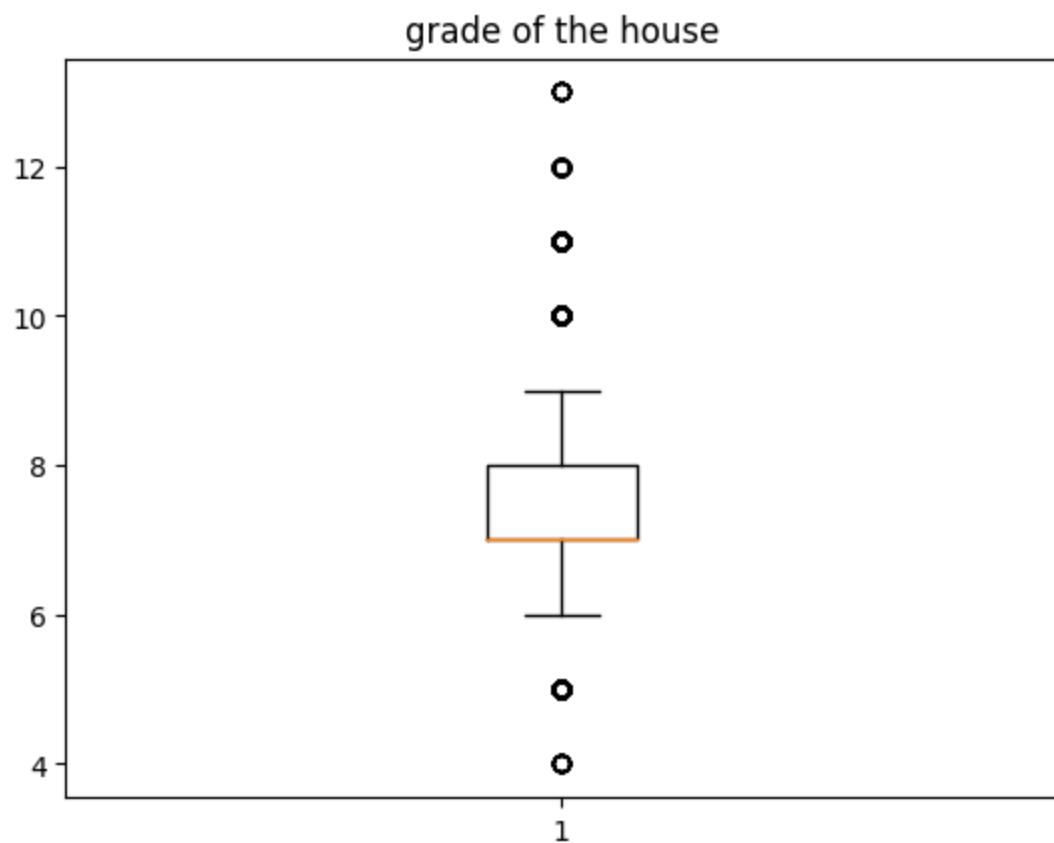
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



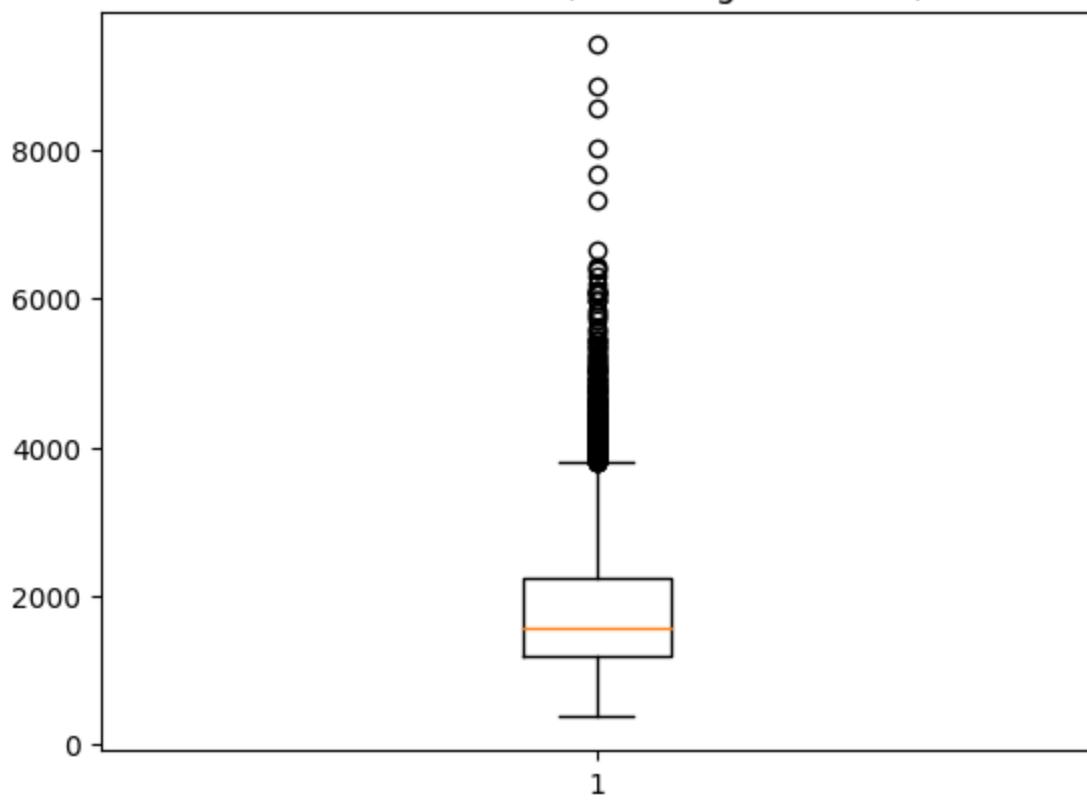
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

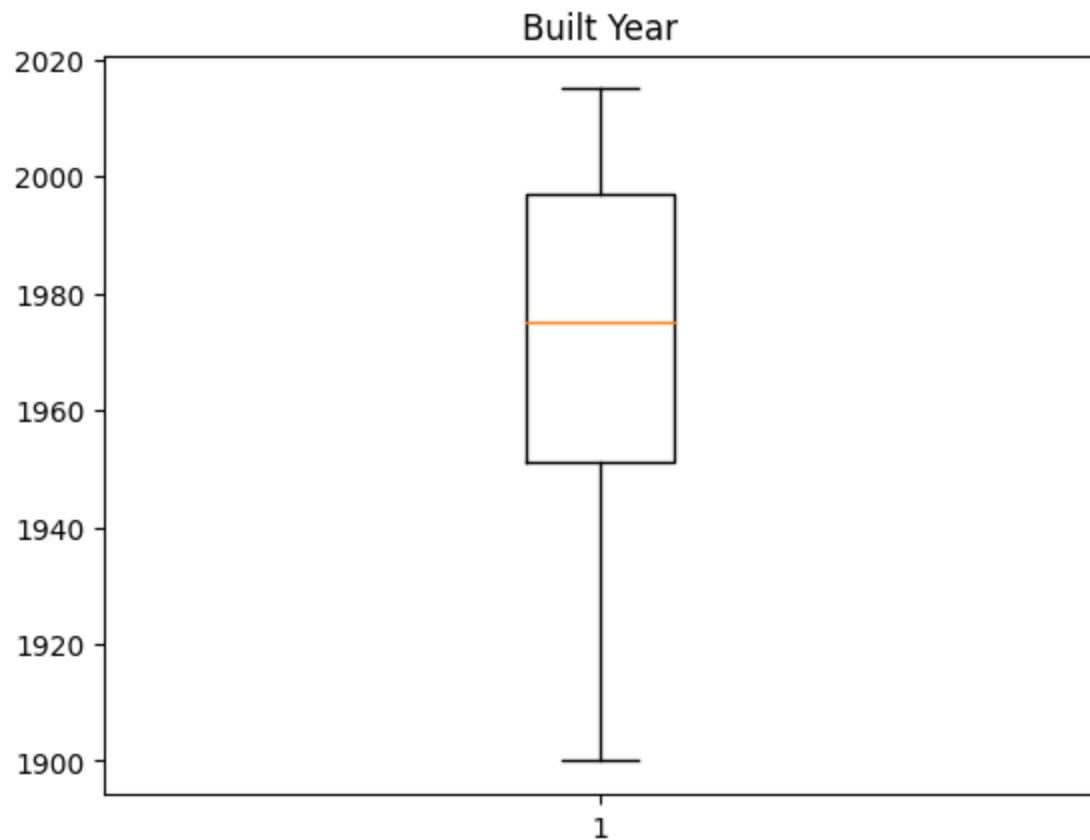
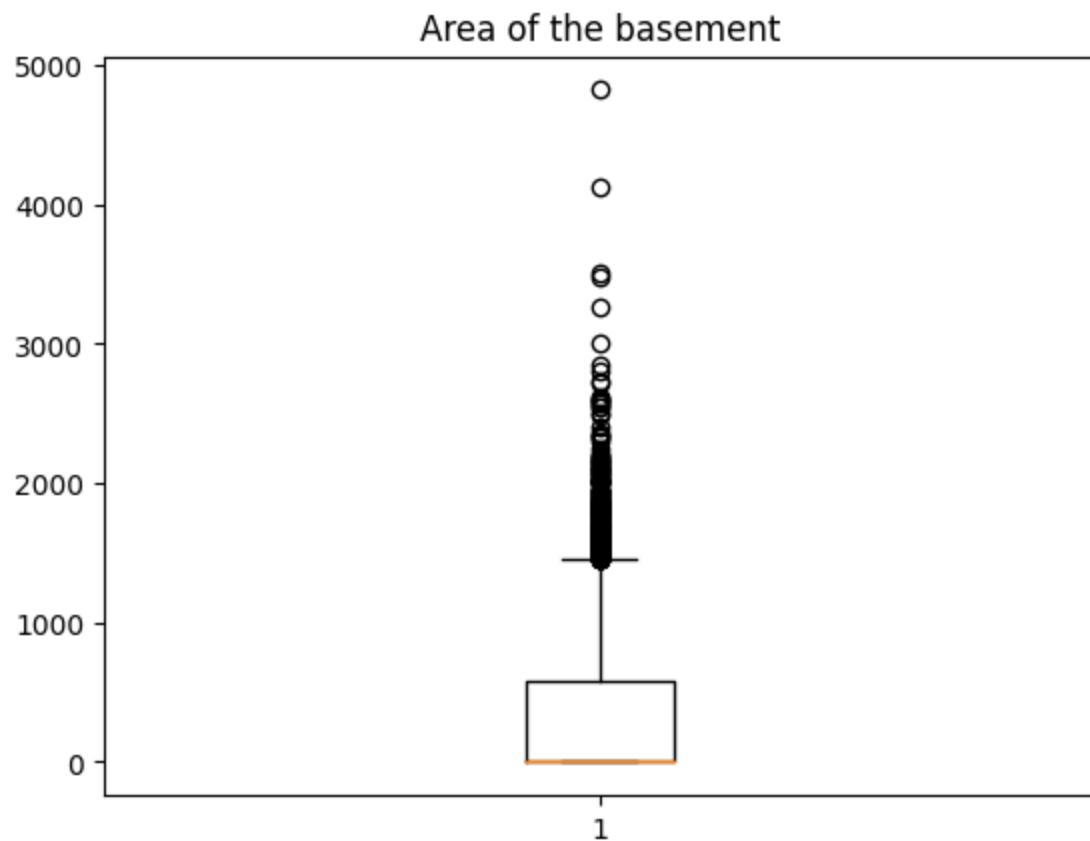


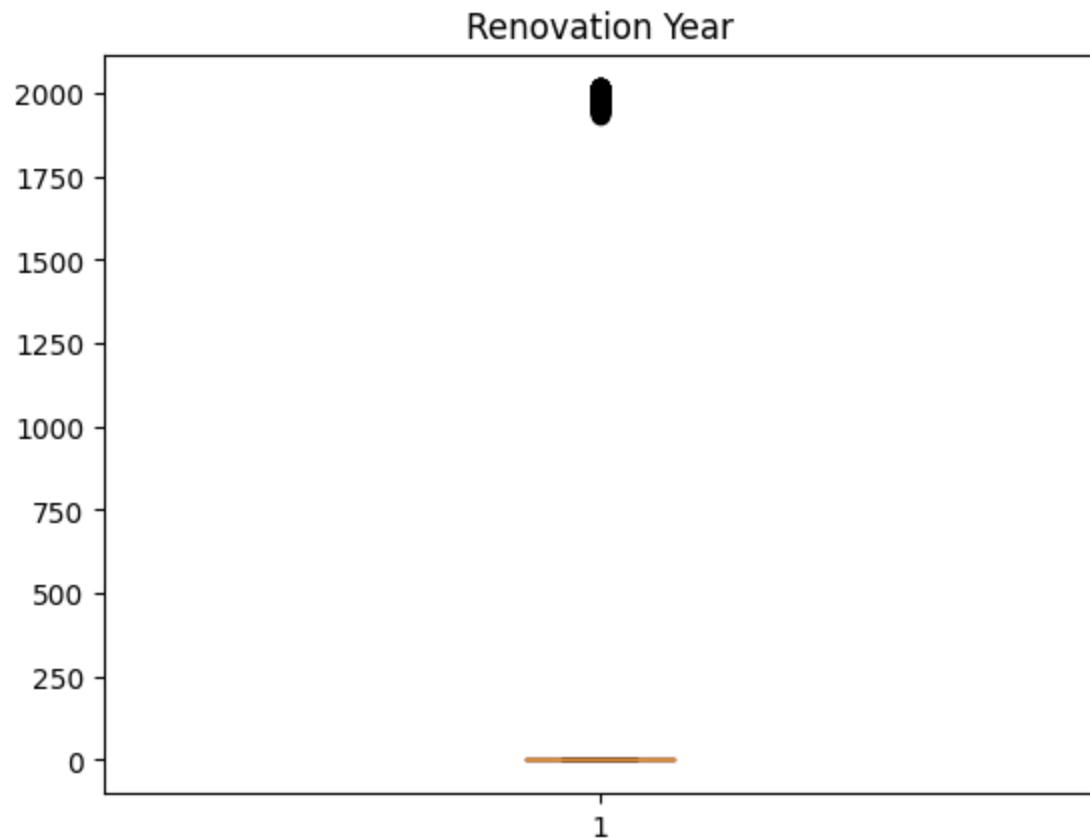
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



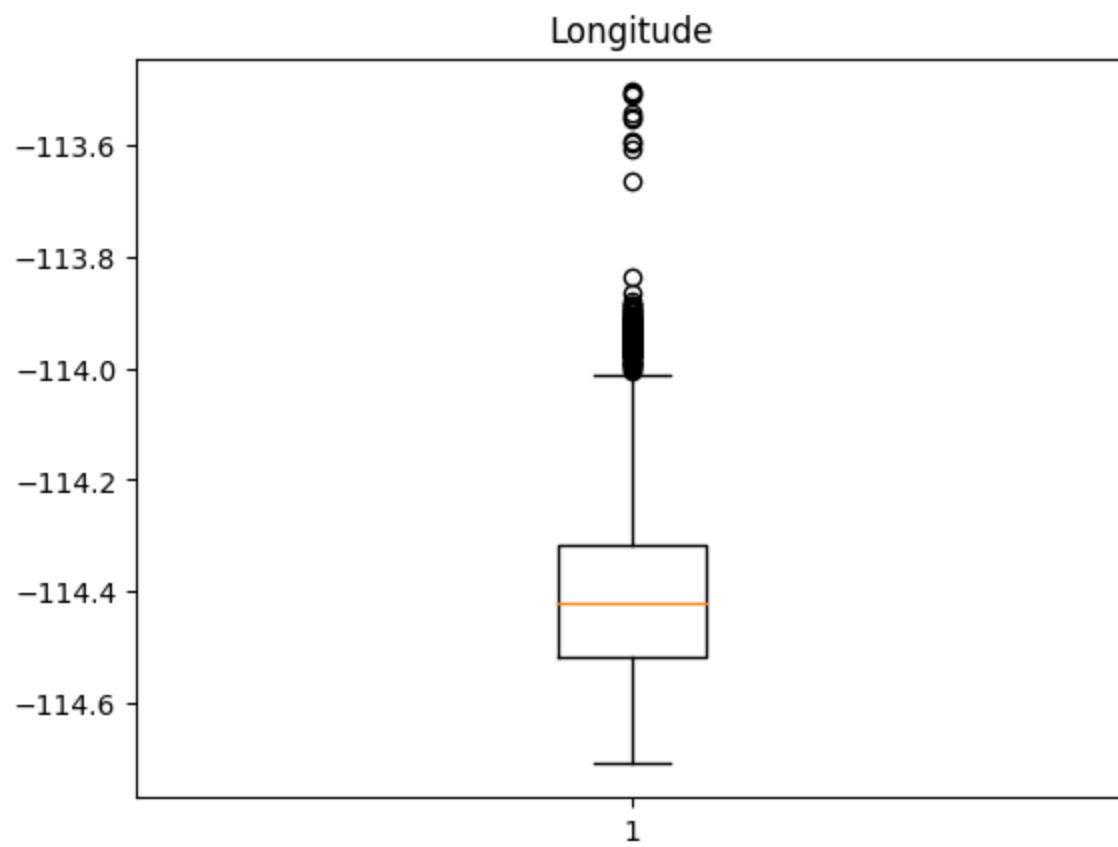
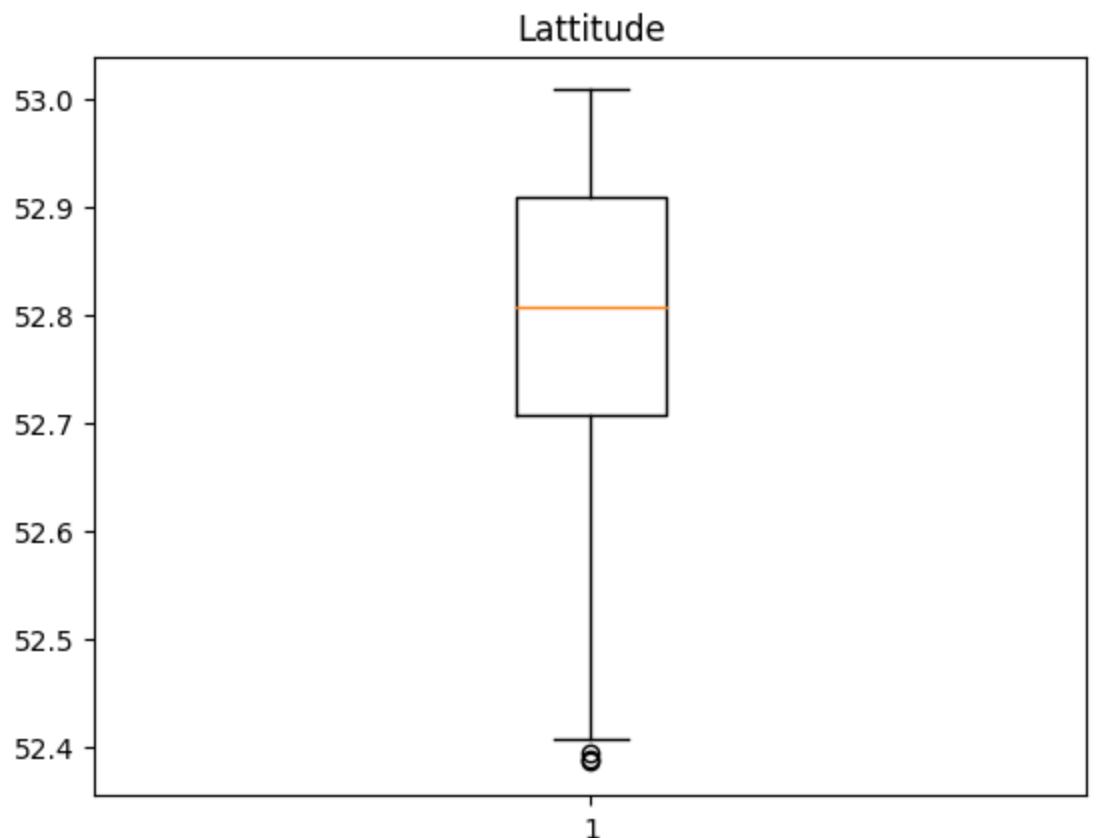
Area of the house(excluding basement)



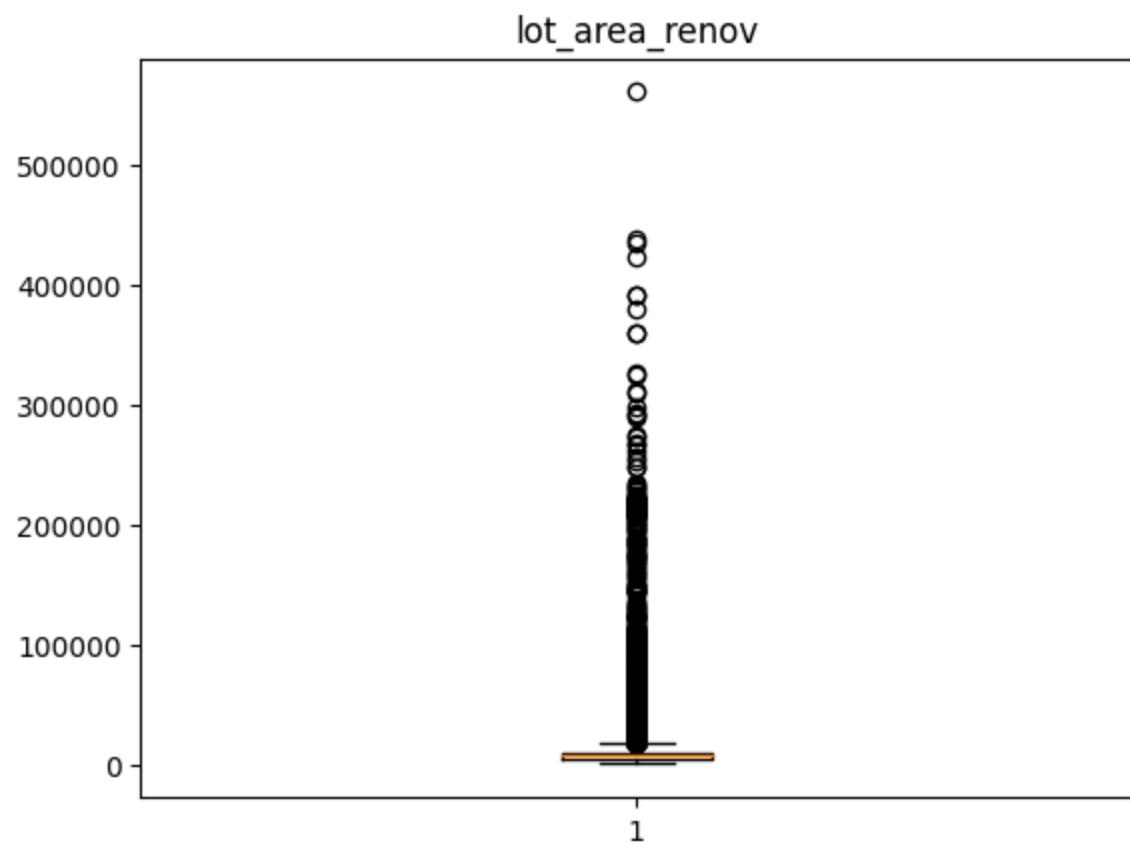
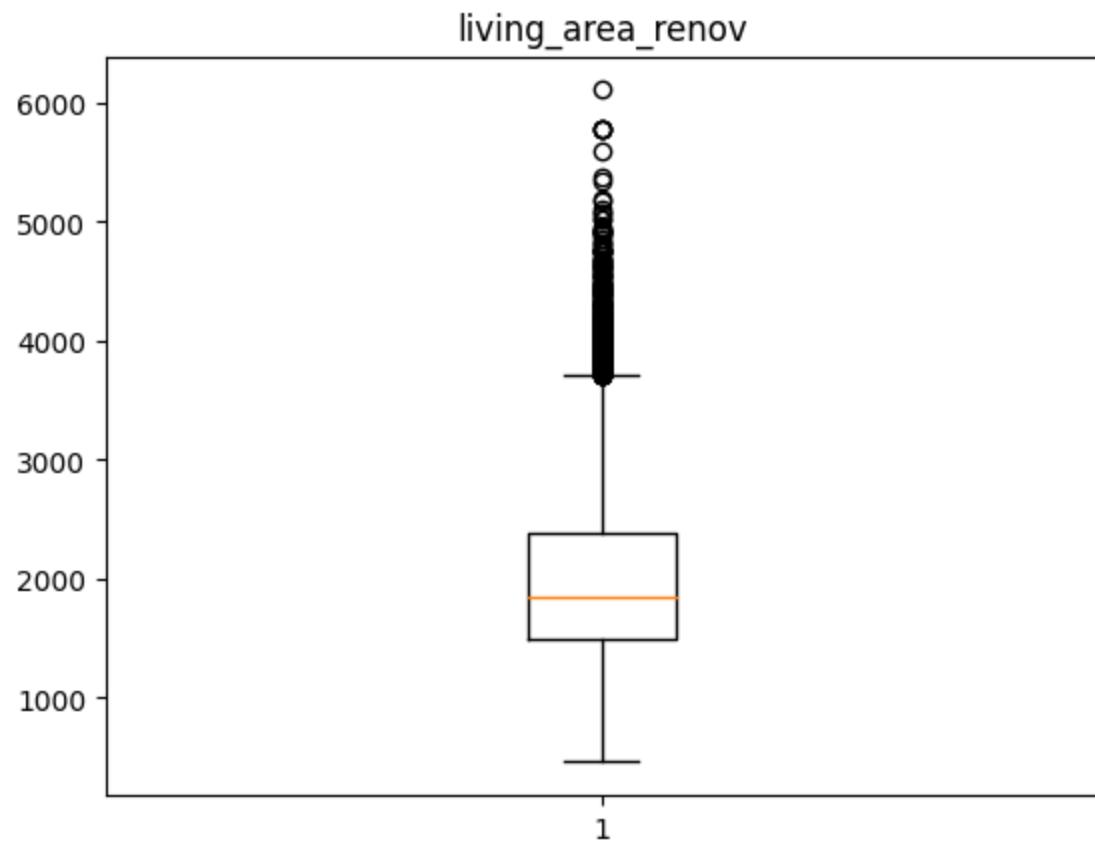




Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

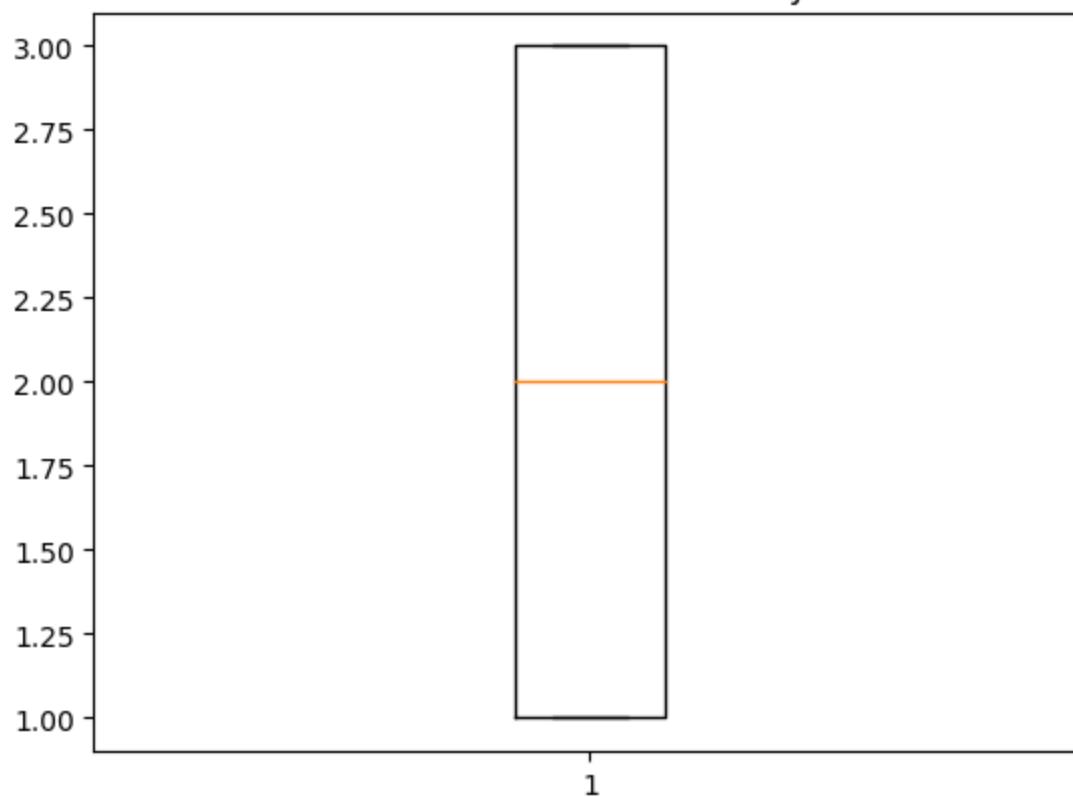


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

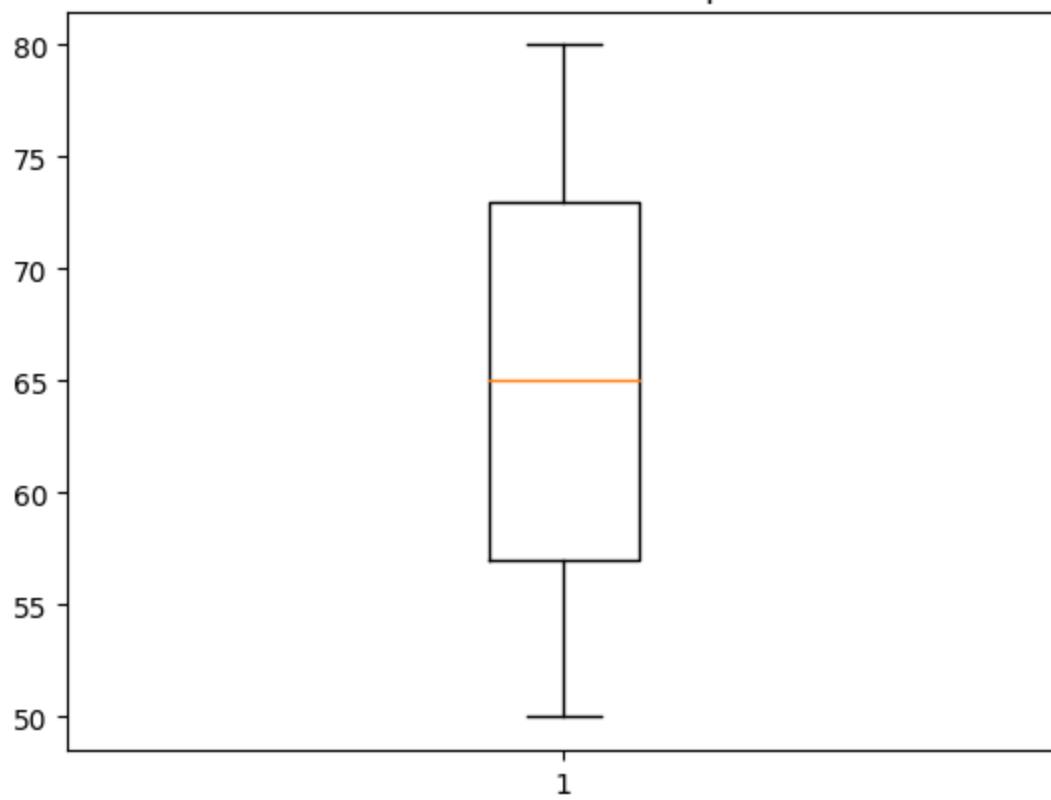


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

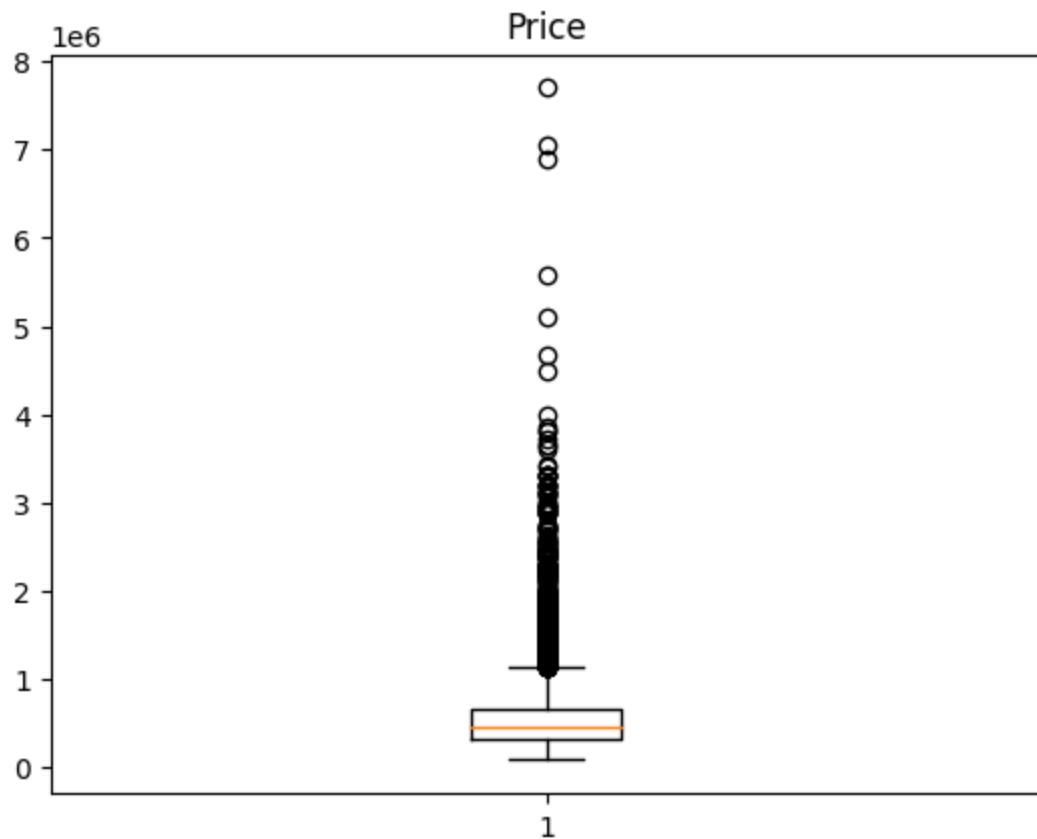
Number of schools nearby



Distance from the airport



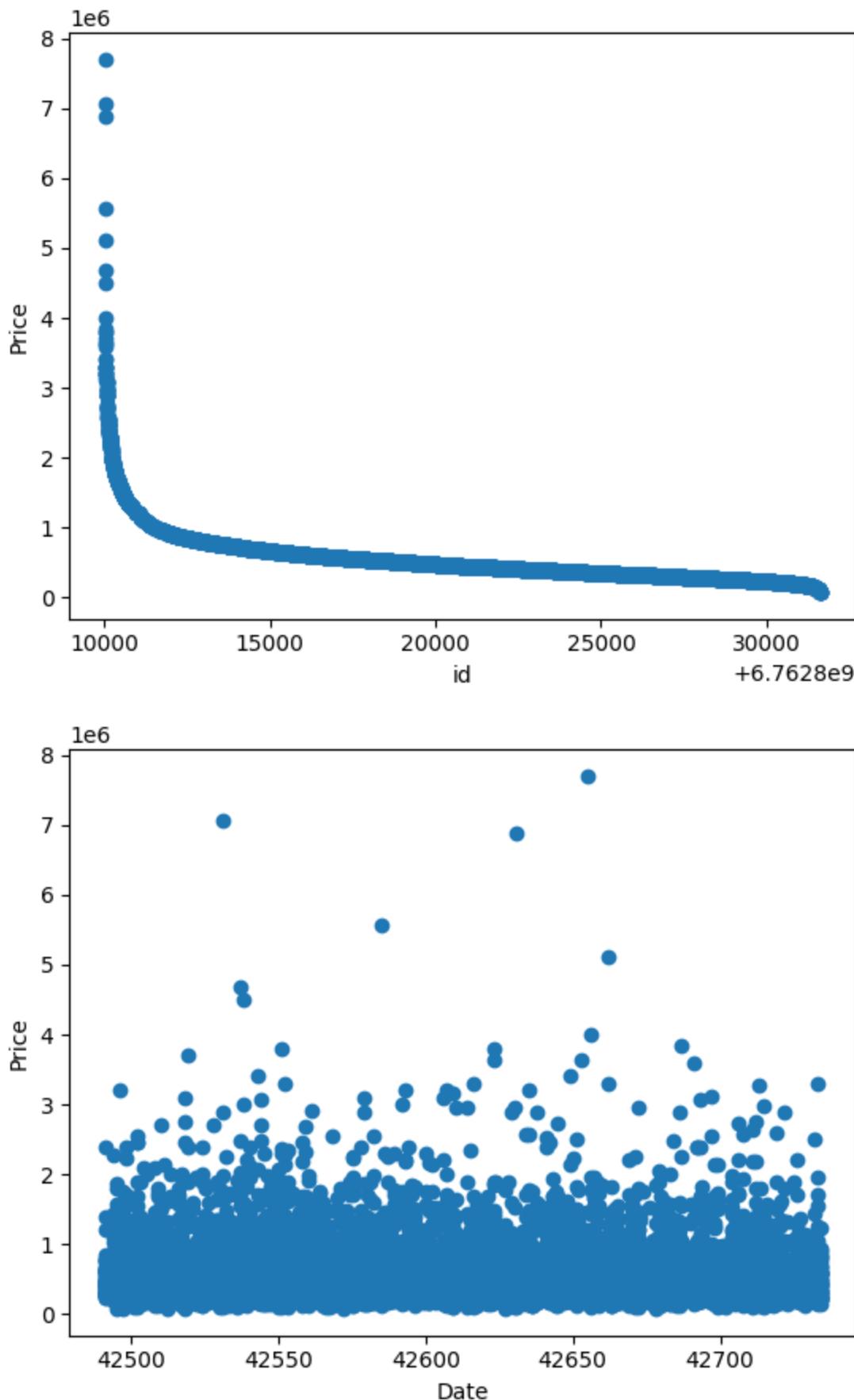
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



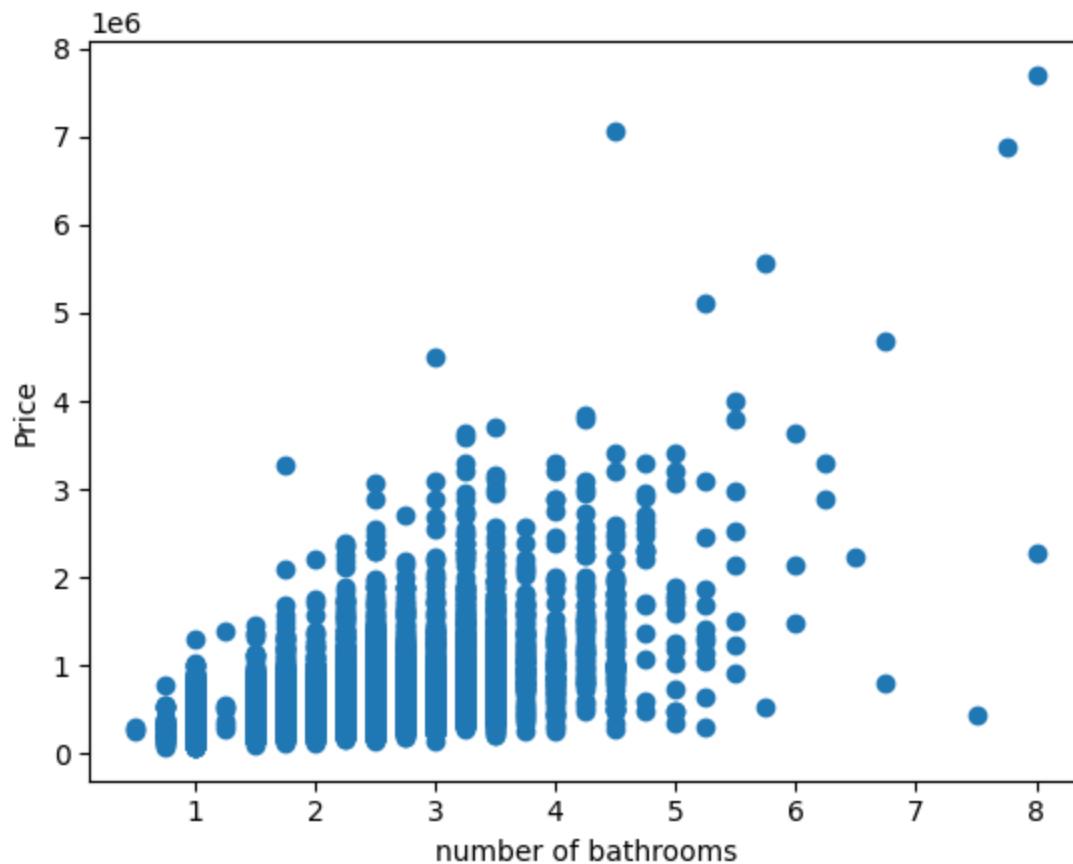
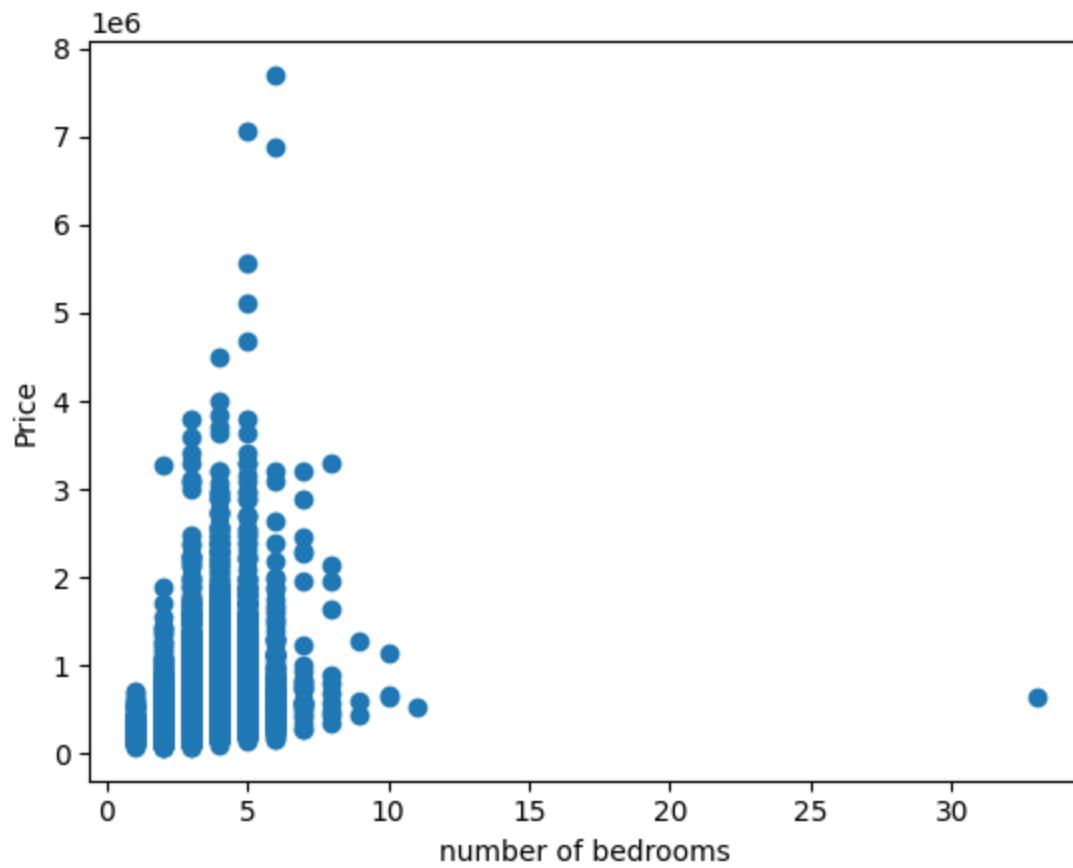
```
In [ ]: # Bi Variate Analysis
```

```
for i in df.columns:  
    plt.scatter(df[i],df['Price'])  
    plt.xlabel(i)  
    plt.ylabel("Price")  
    plt.show()
```

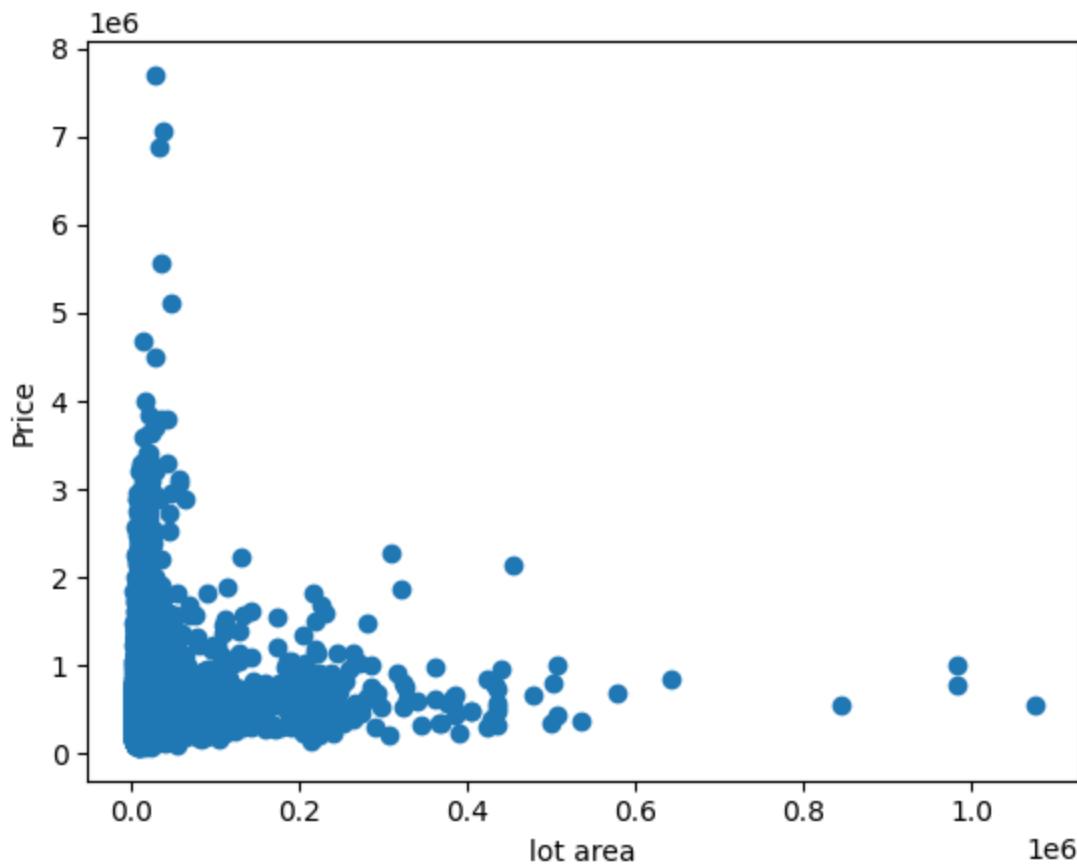
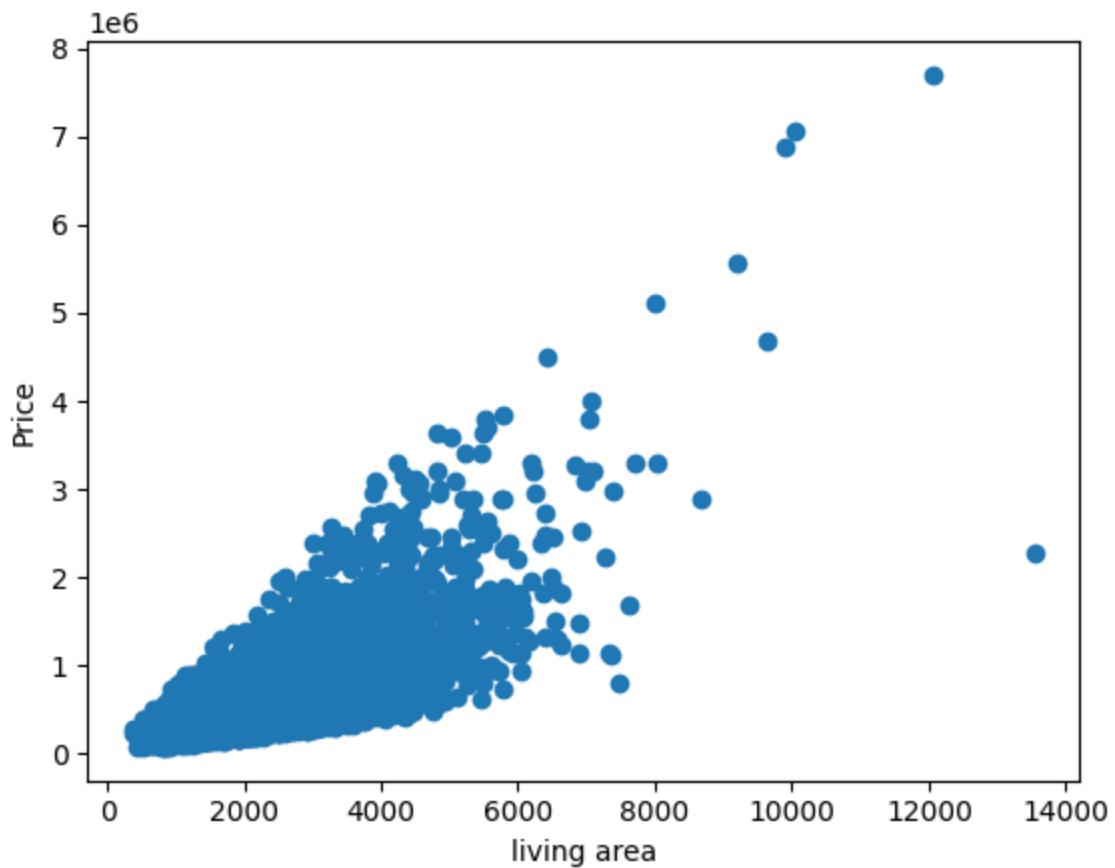
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



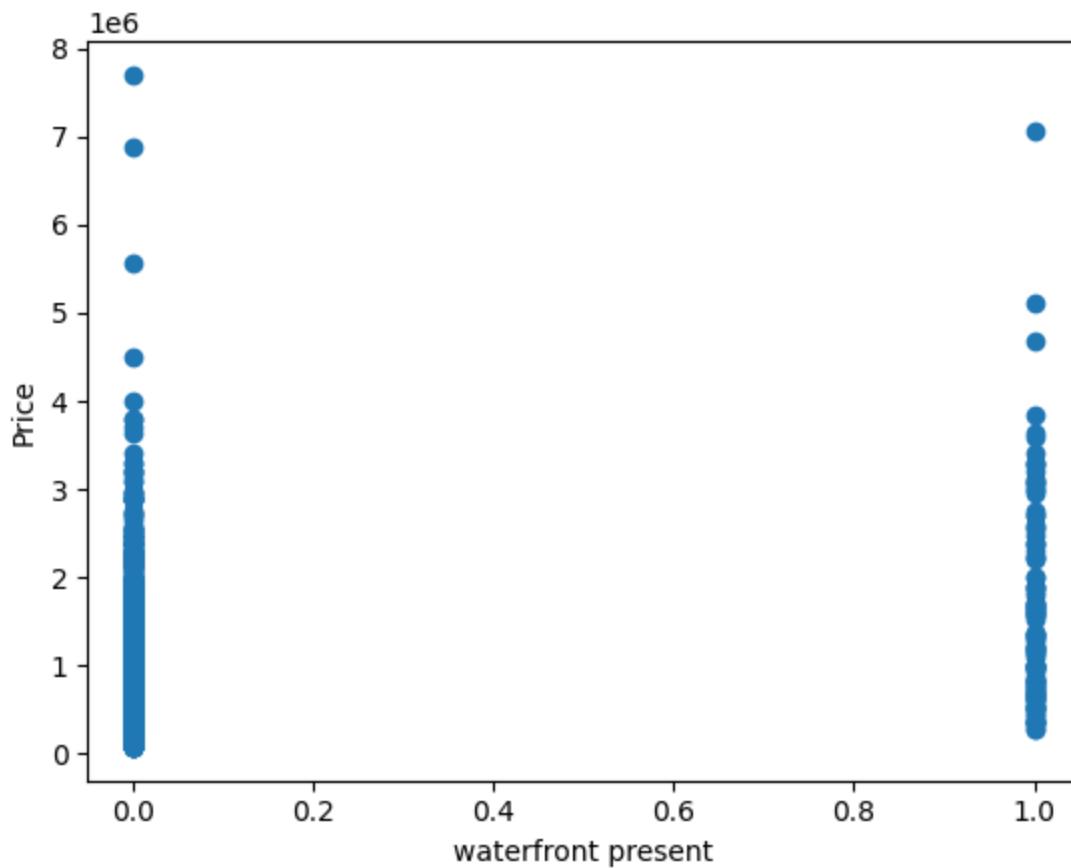
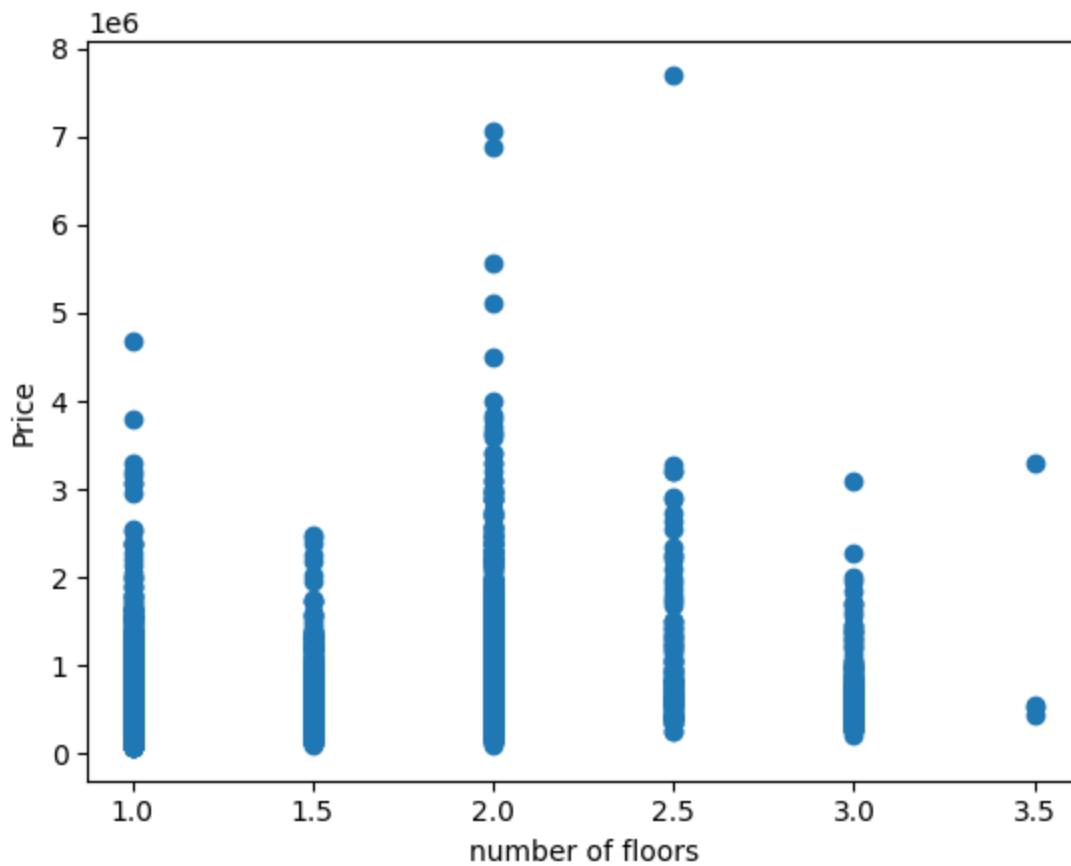
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

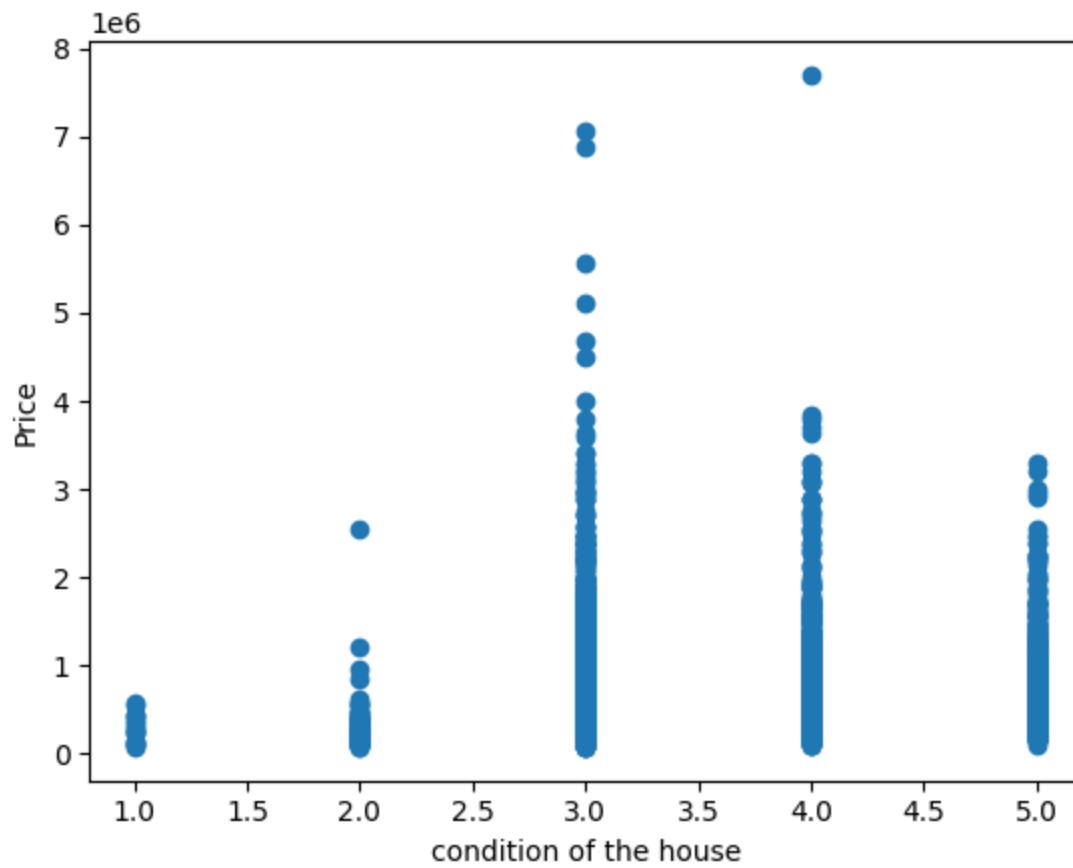
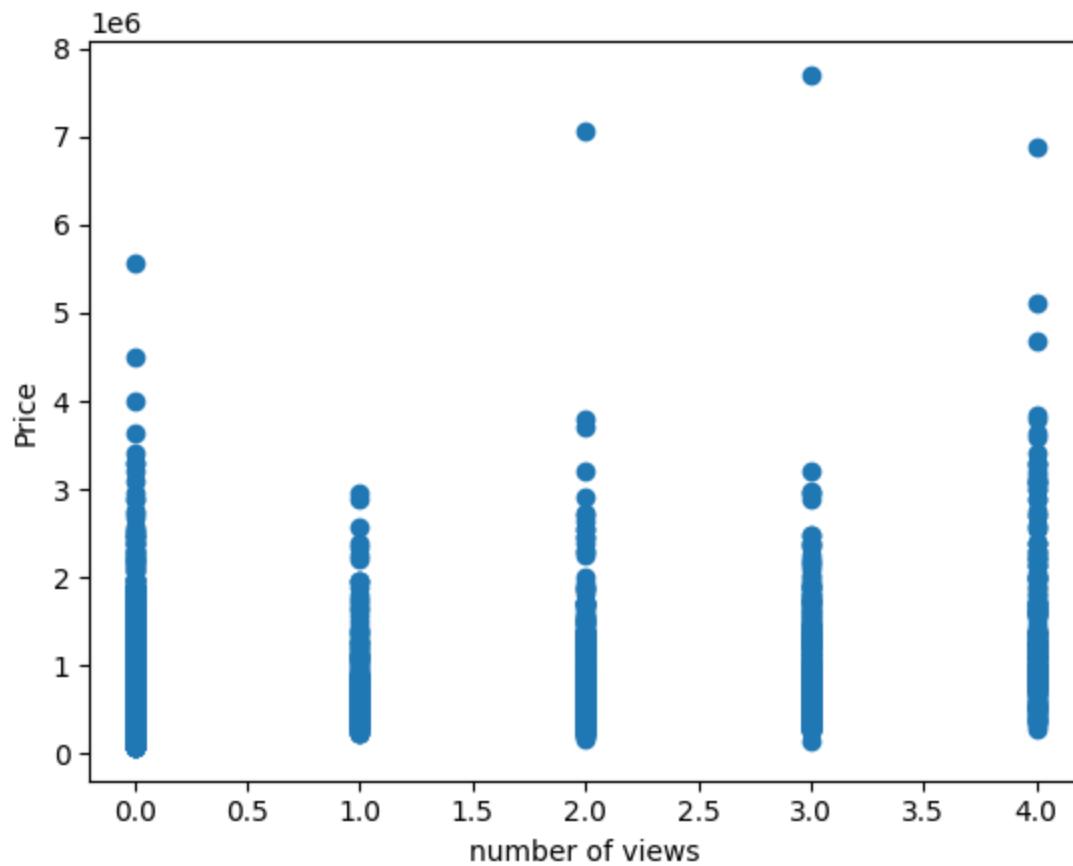


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

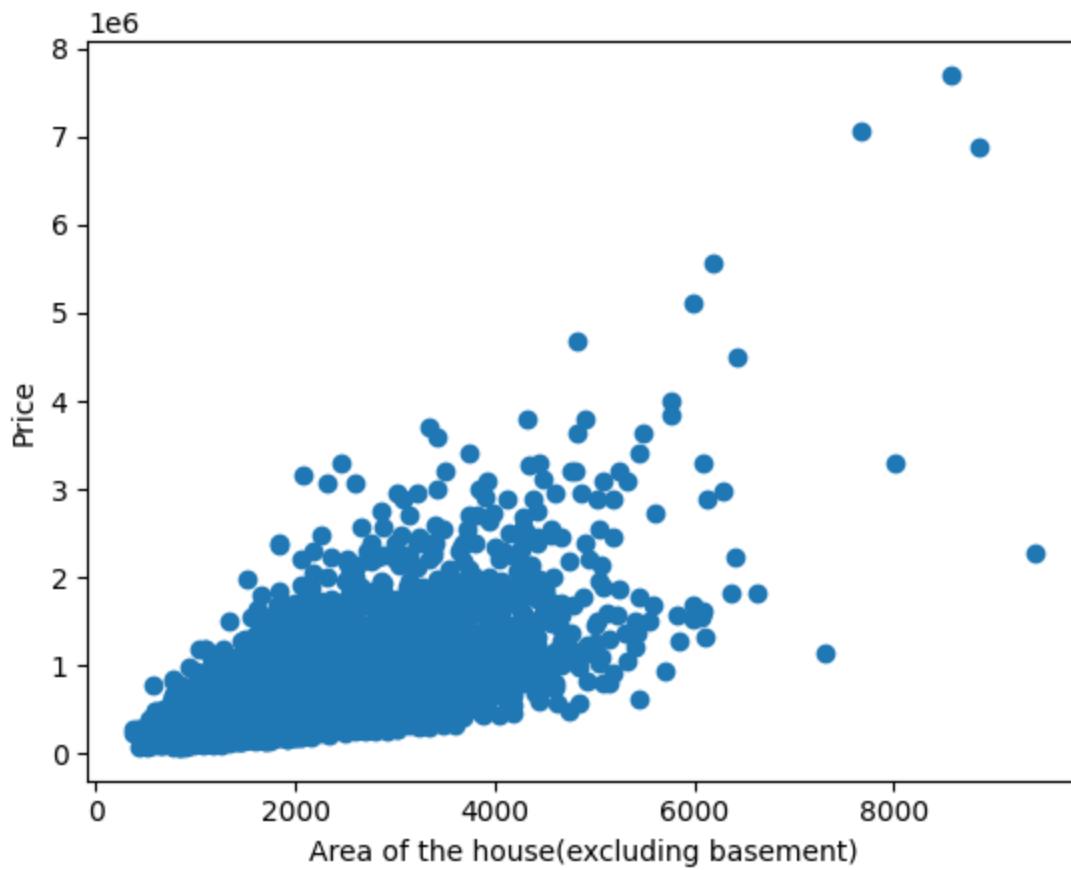
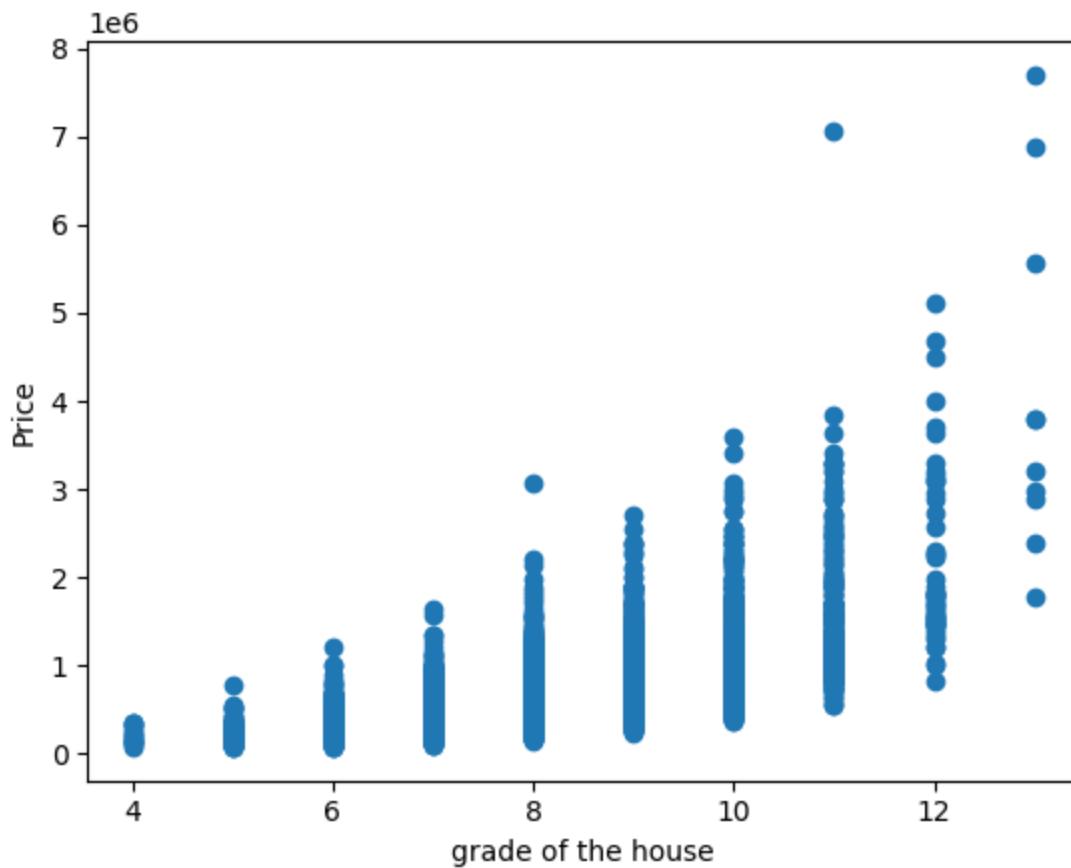


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

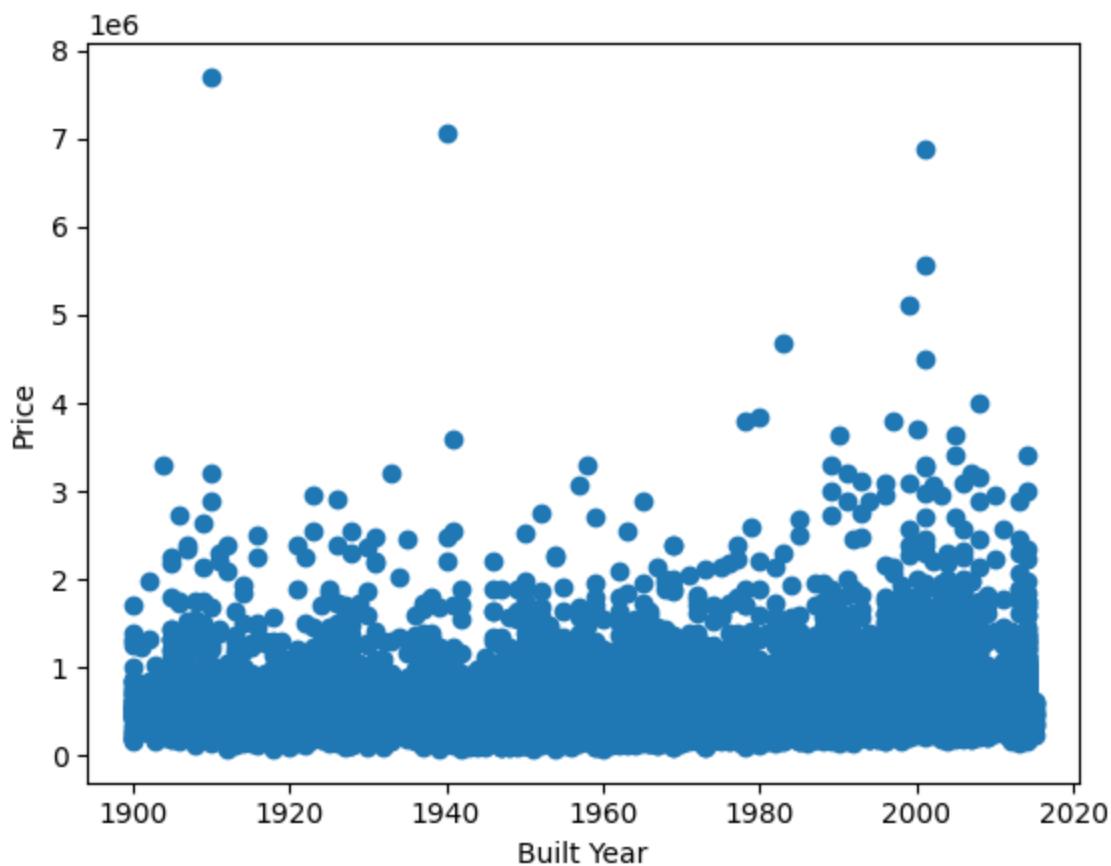
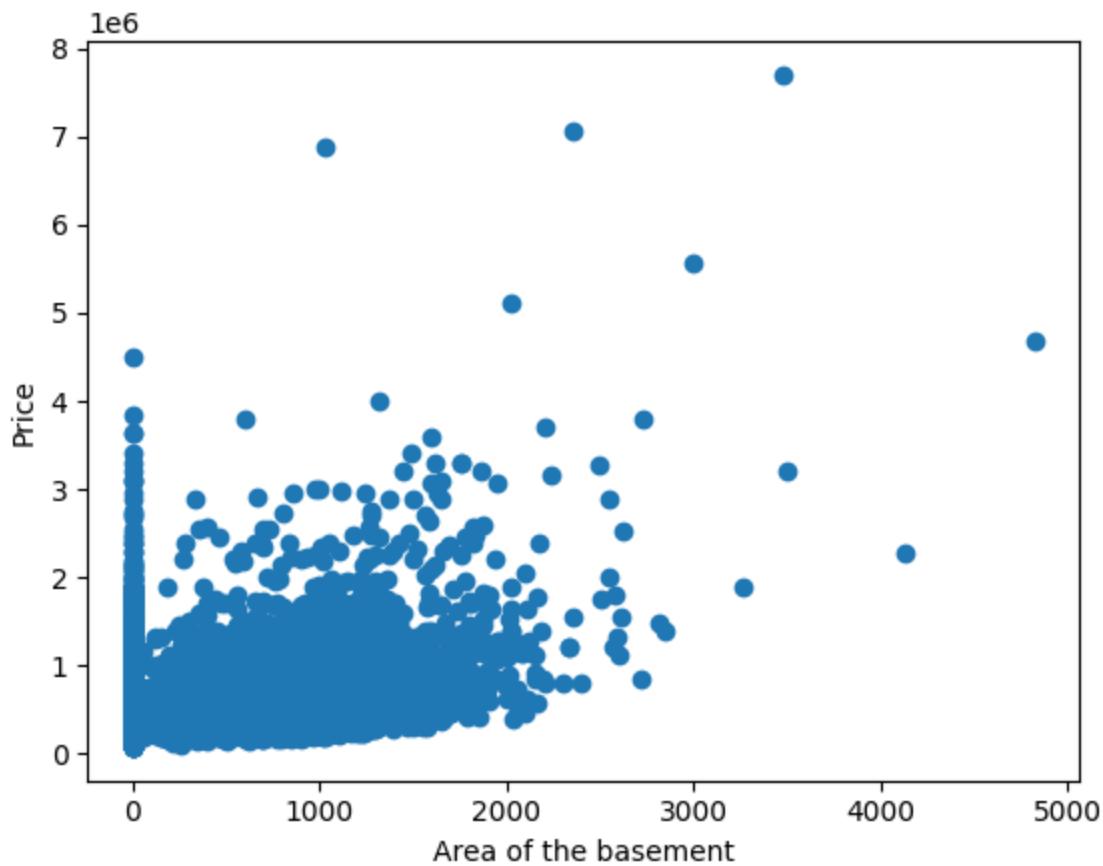




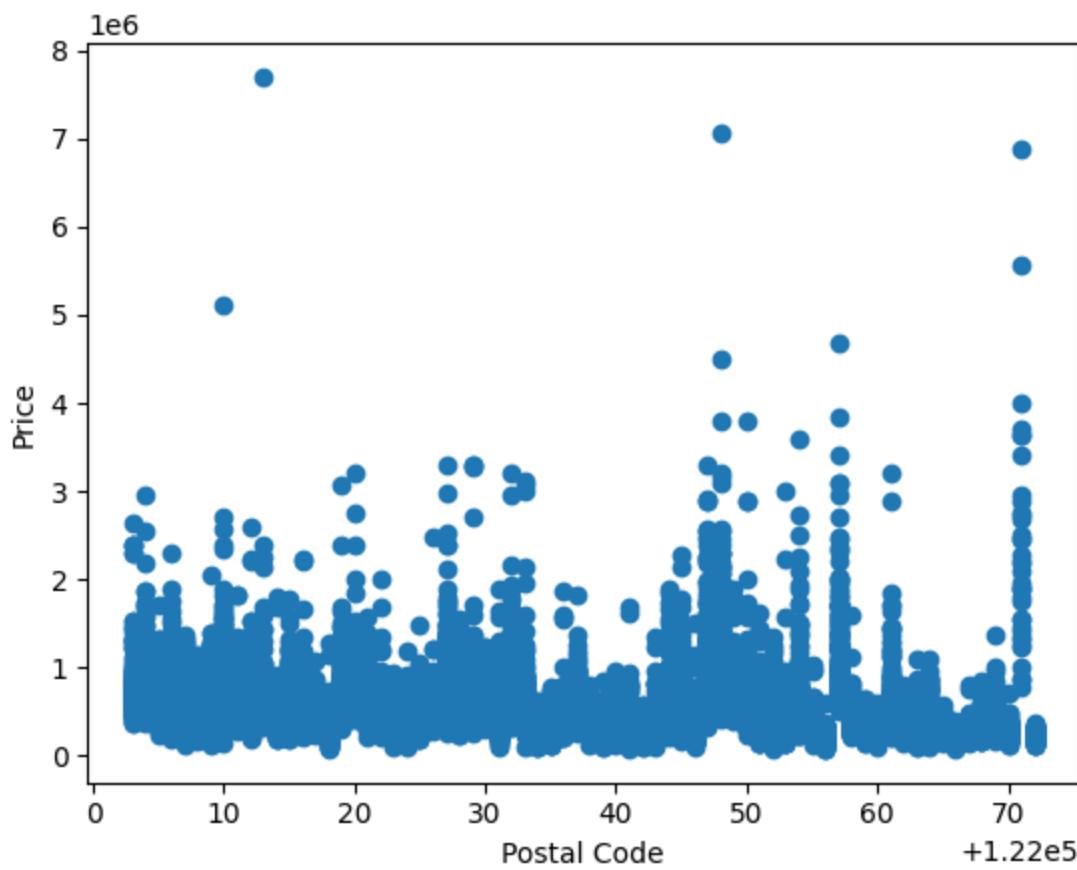
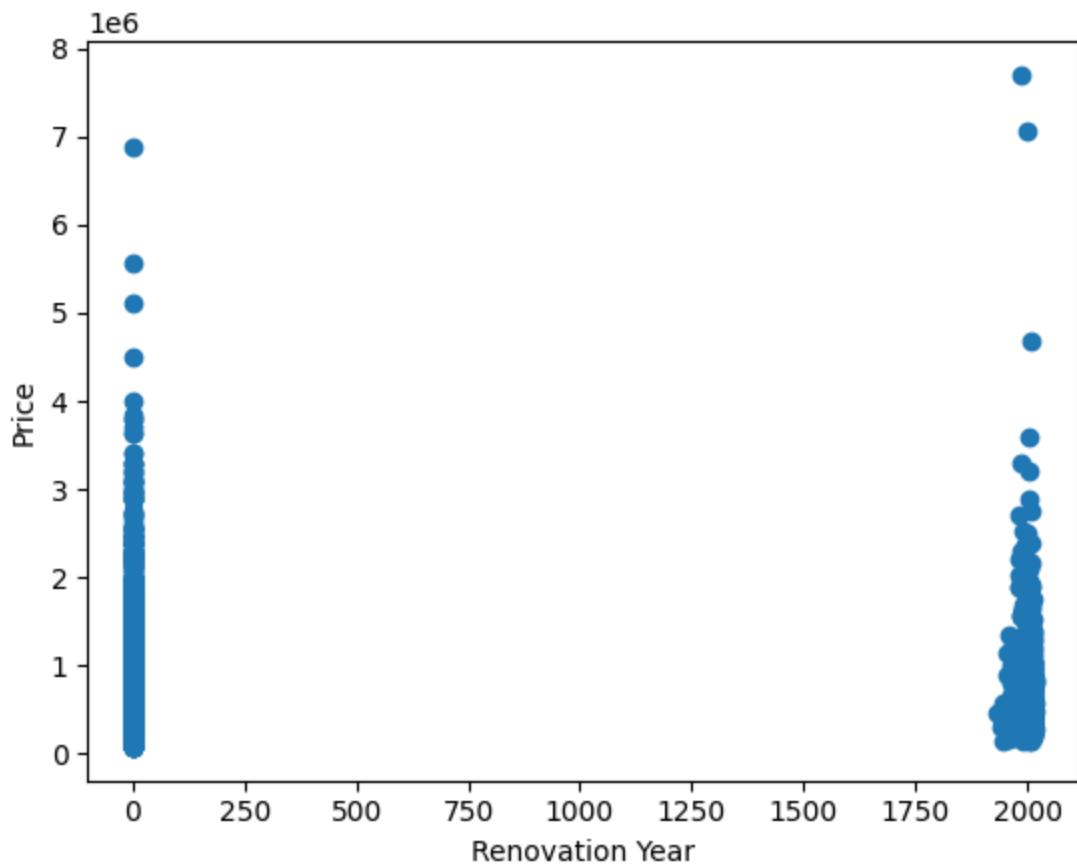
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



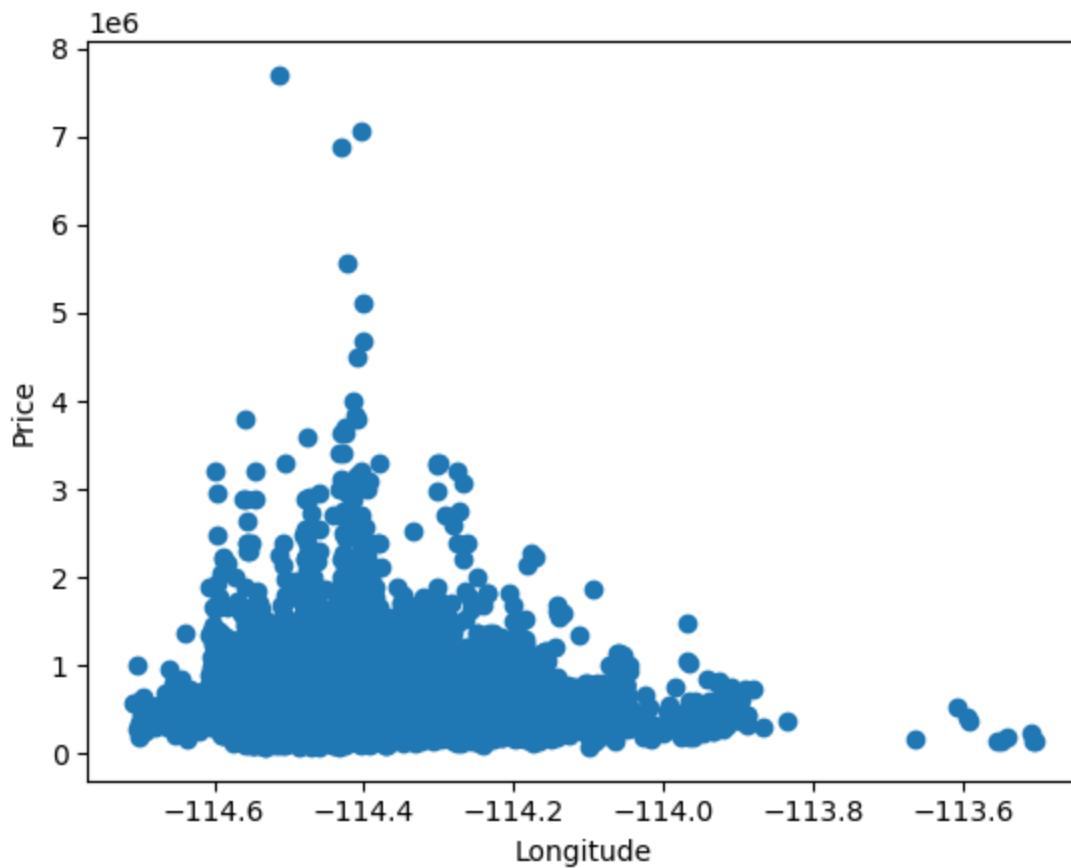
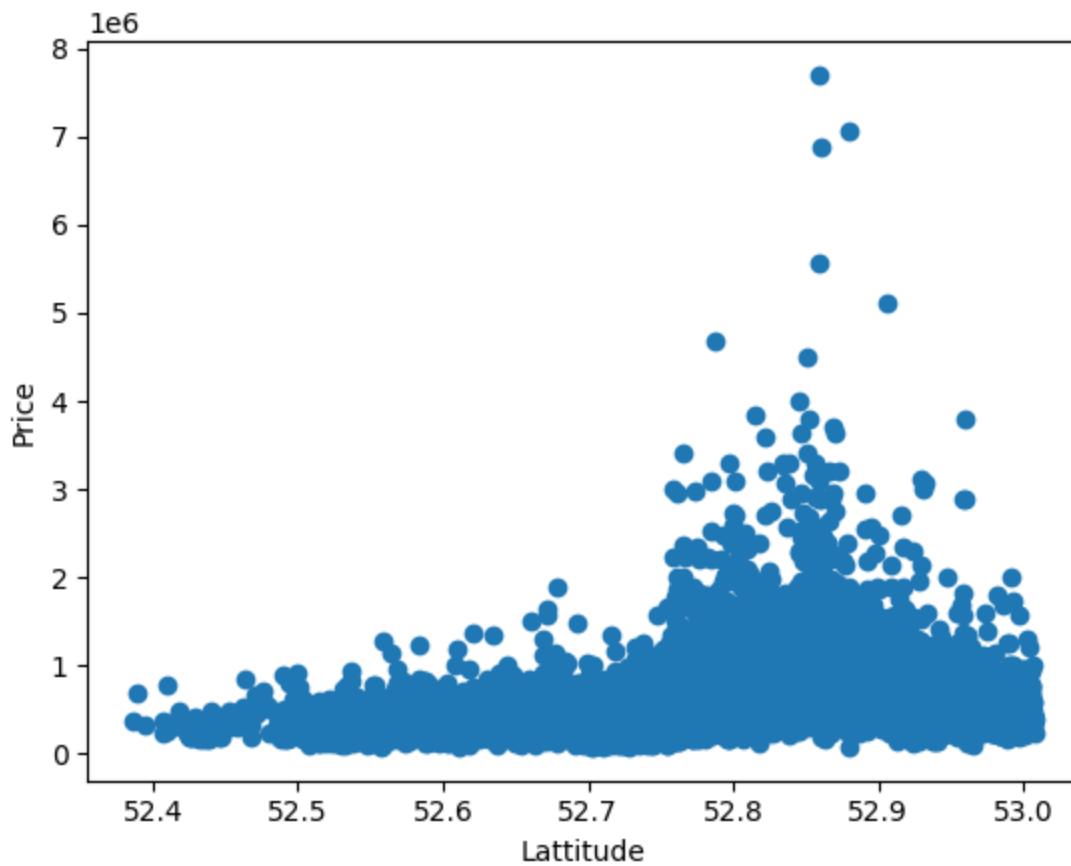
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



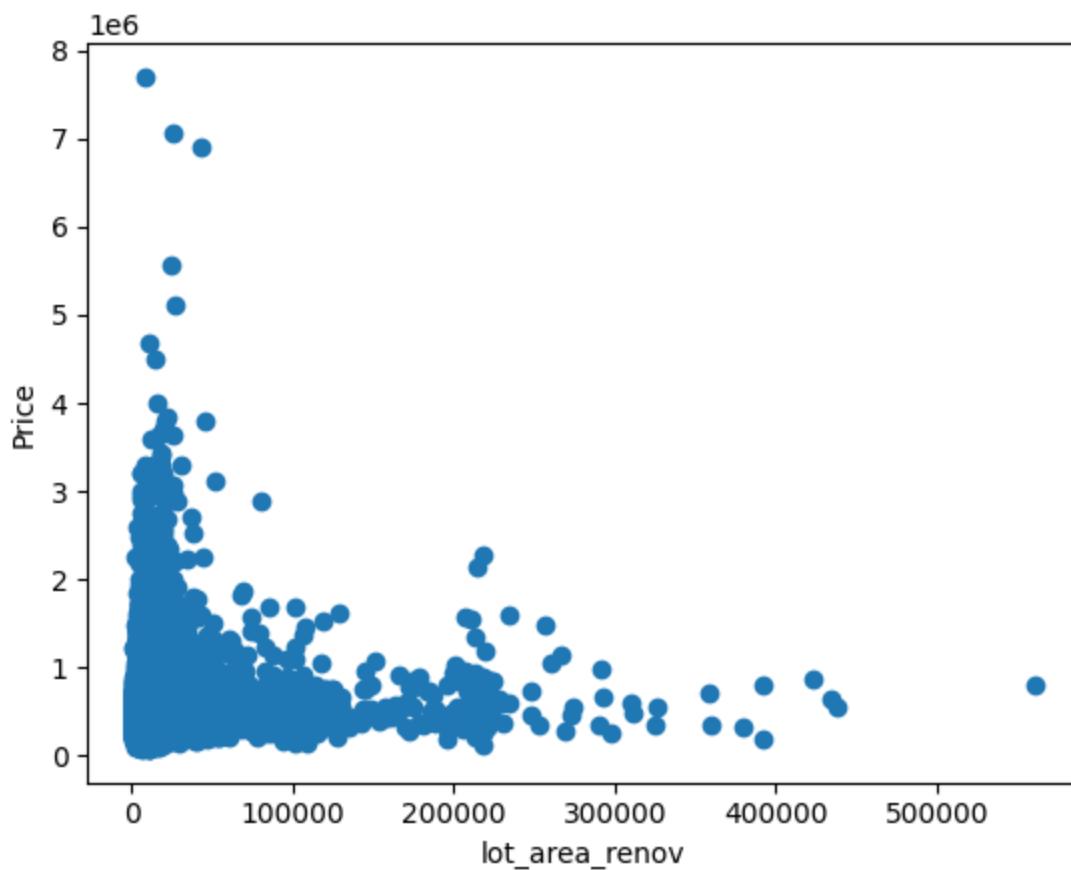
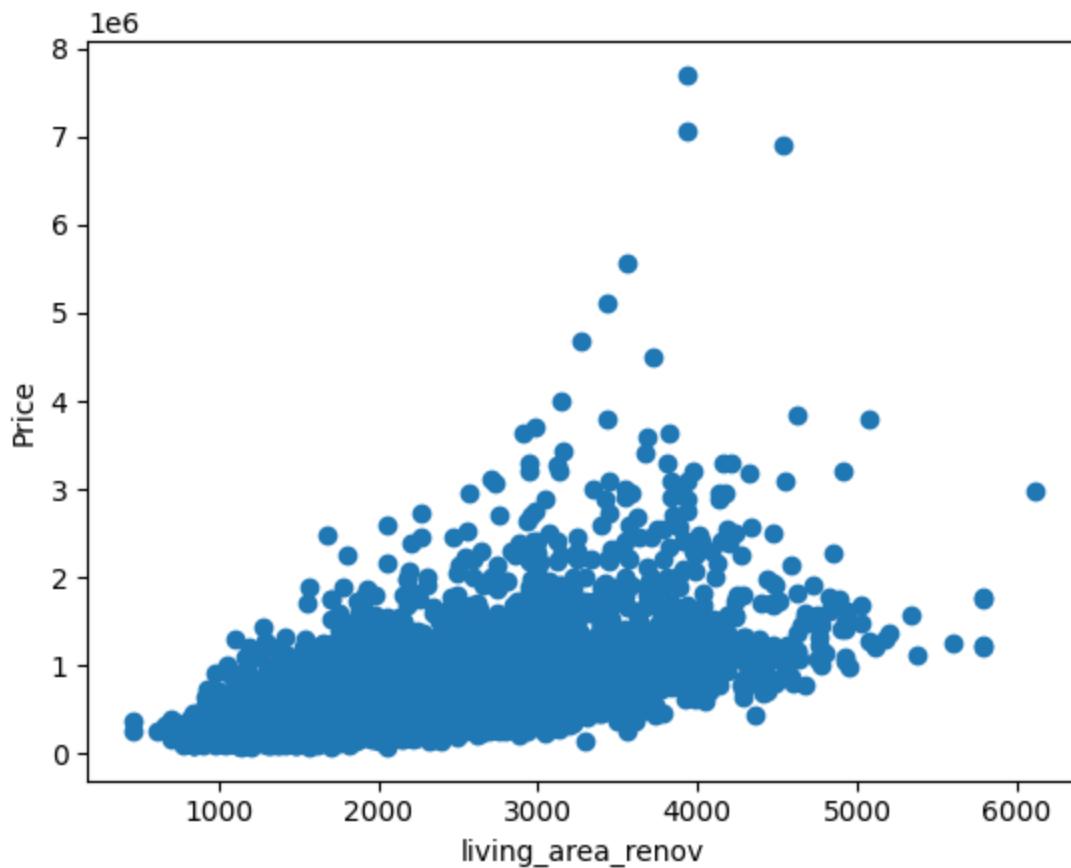
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



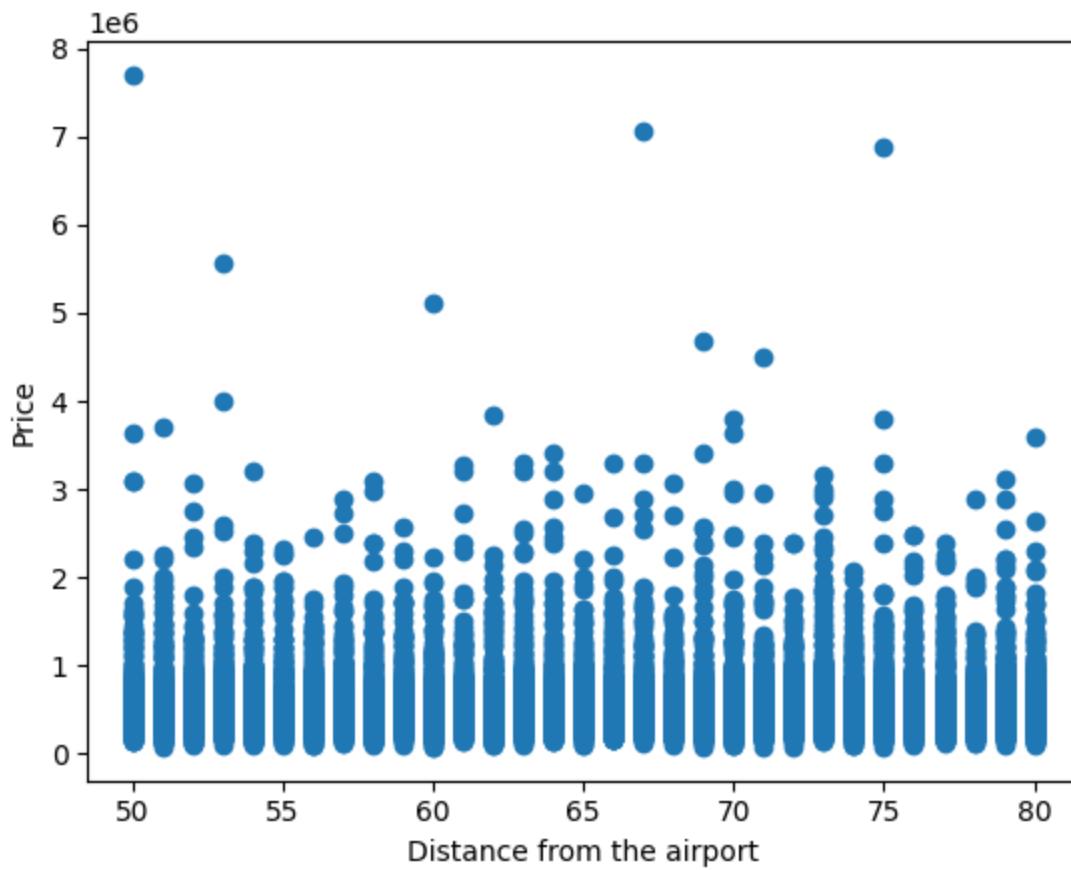
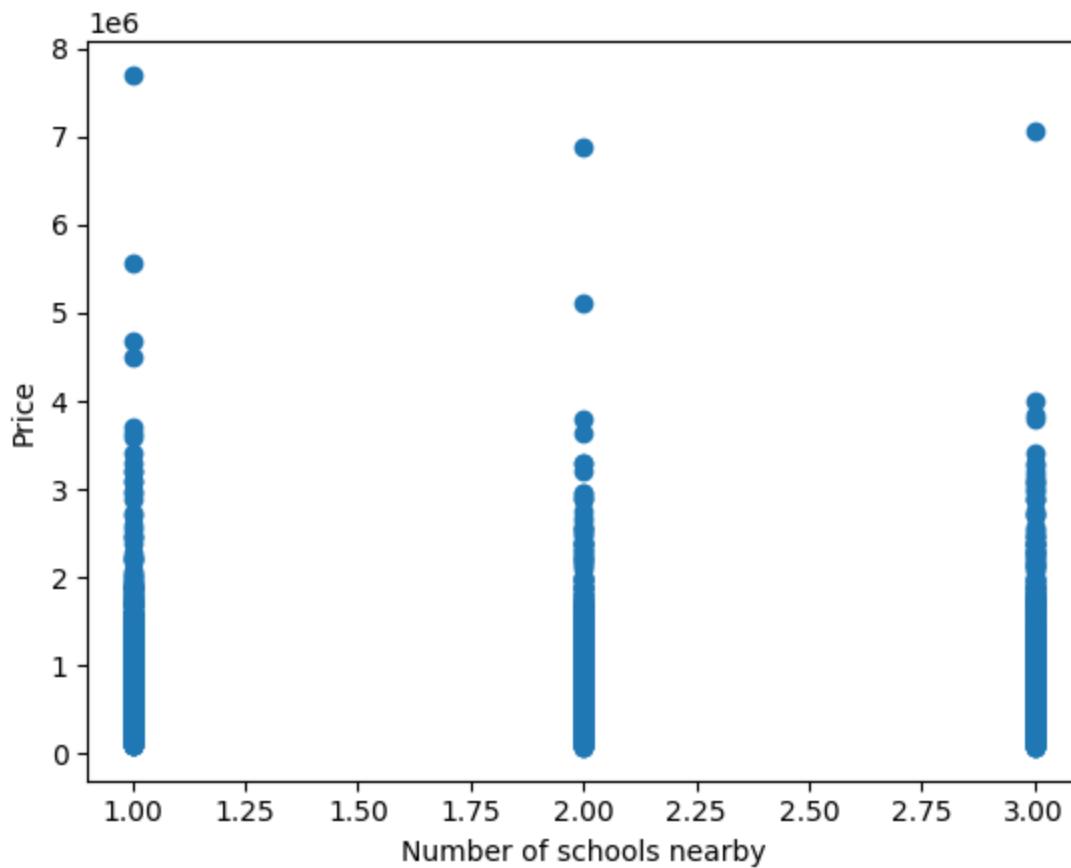
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



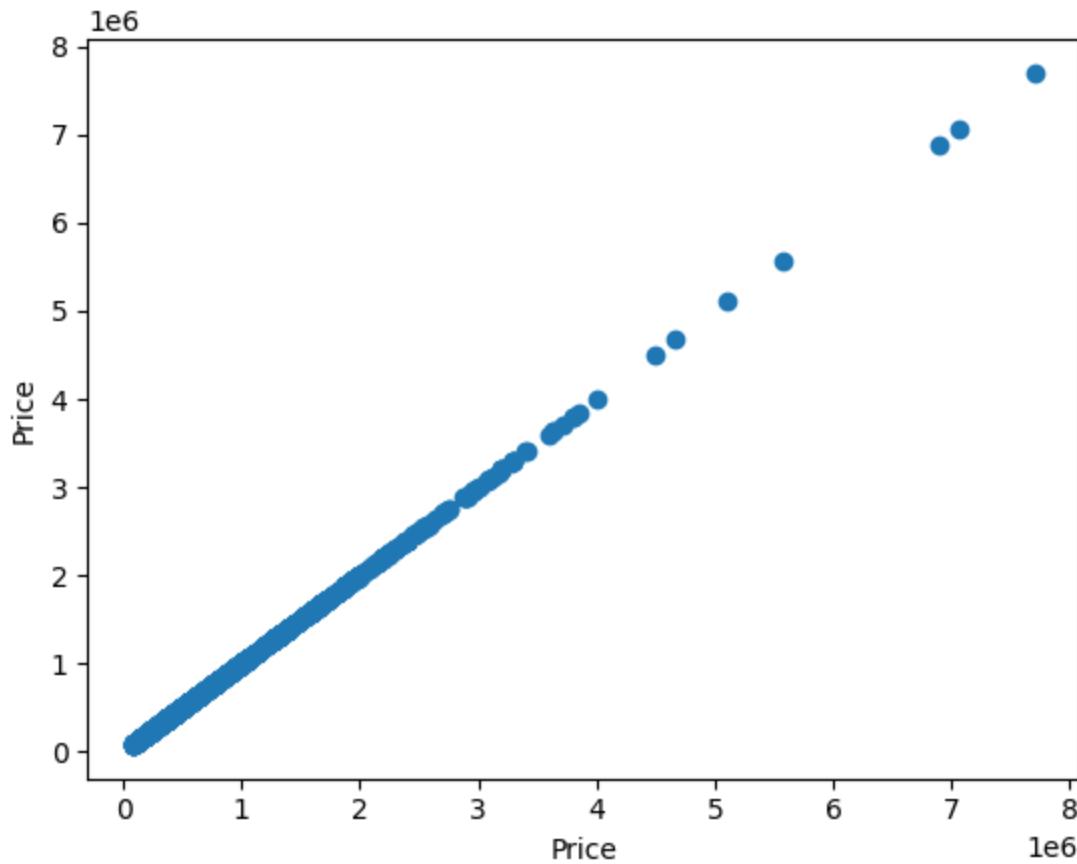
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

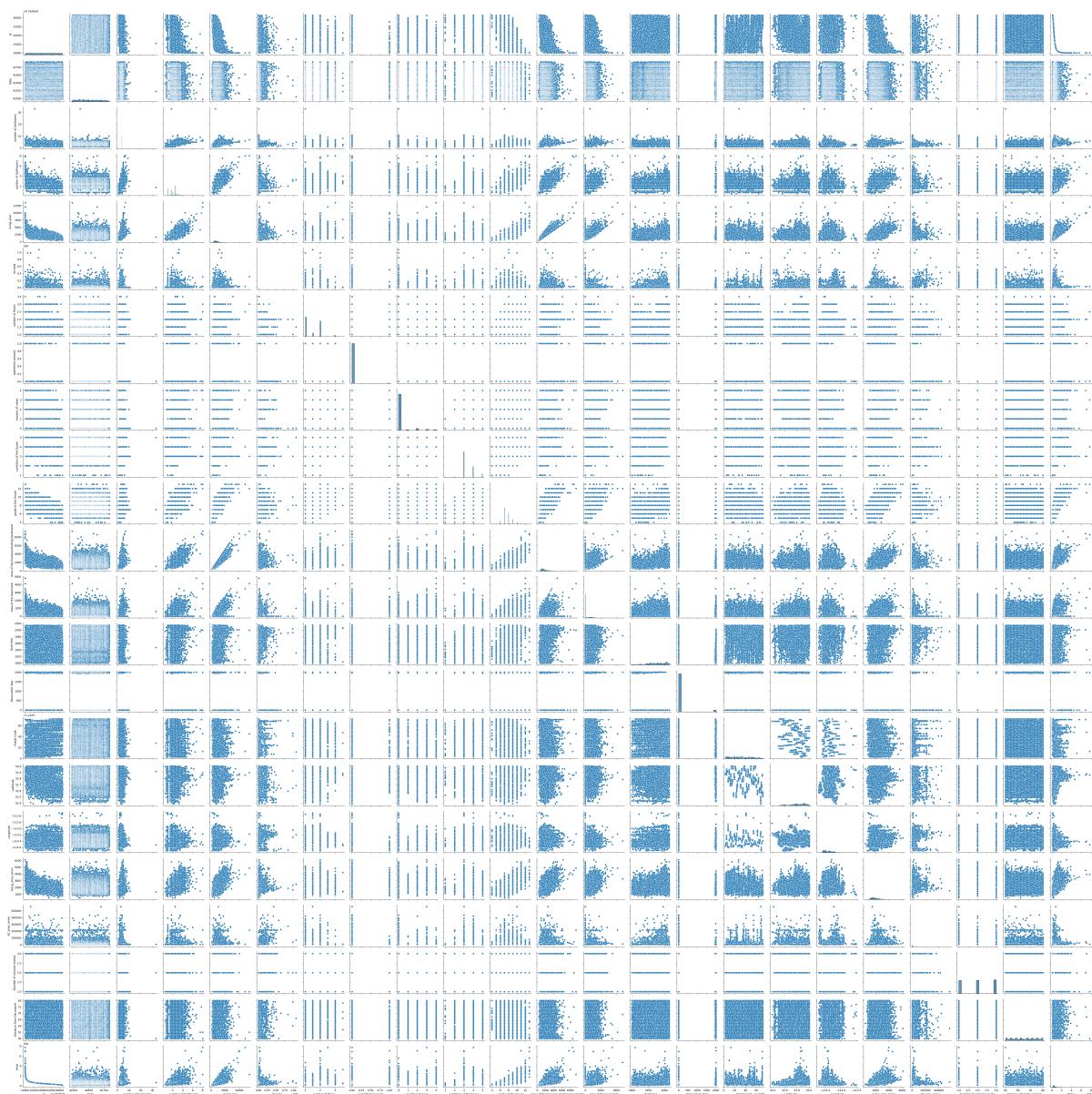


```
In [ ]: # Multivariate Variate Analysis
```

```
      sns.pairplot(df)
```

```
Out[ ]: <seaborn.axisgrid.PairGrid at 0x27b81bd0640>
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js