In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler, StandardScaler, LabelEncoder, Or
```

In [2]:
```
df = pd.read_csv("C:\\Users\\Gouri\\Downloads\\House_Pricing.csv")
df
```

Out[2]:

	ID	Date House was Sold	Sale Price	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors
0	7129300520	14 October 2017	221900.0	3	1.00	1180.0	5650.0	1.0
1	6414100192	14 December 2017	538000.0	3	2.25	2570.0	7242.0	2.0
2	5631500400	15 February 2016	180000.0	2	1.00	770.0	10000.0	1.0
3	2487200875	14 December 2017	604000.0	4	3.00	1960.0	5000.0	1.0
4	1954400510	15 February 2016	510000.0	3	2.00	1680.0	8080.0	1.0
...	...	...	...	...	...	...	...	...
21608	263000018	14 May 2017	360000.0	3	2.50	1530.0	1131.0	3.0
21609	6600060120	15 February 2016	400000.0	4	2.50	2310.0	5813.0	2.0
21610	1523300141	14 June 2017	402101.0	2	0.75	1020.0	1350.0	2.0
21611	291310100	15 January 2016	400000.0	3	2.50	1600.0	2388.0	2.0
21612	1523300157	14 October 2017	325000.0	2	0.75	1020.0	1076.0	2.0

21613 rows × 21 columns

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 21 columns):
 #   Column                                    Non-Null Count  Dtype  
---  ------                                    --------------  -----  
 0   ID                                        21613 non-null  int64  
 1   Date House was Sold                       21613 non-null  object 
 2   Sale Price                                21609 non-null  float64
 3   No of Bedrooms                            21613 non-null  int64  
 4   No of Bathrooms                           21609 non-null  float64
 5   Flat Area (in Sqft)                       21604 non-null  float64
 6   Lot Area (in Sqft)                        21604 non-null  float64
 7   No of Floors                              21613 non-null  float64
 8   Waterfront View                           21613 non-null  object 
 9   No of Times Visited                       2124 non-null   object 
 10  Condition of the House                    21613 non-null  object 
 11  Overall Grade                             21613 non-null  int64  
 12  Area of the House from Basement (in Sqft) 21610 non-null  float64
 13  Basement Area (in Sqft)                   21613 non-null  int64  
 14  Age of House (in Years)                   21613 non-null  int64  
 15  Renovated Year                            21613 non-null  int64  
 16  Zipcode                                   21612 non-null  float64
 17  Latitude                                  21612 non-null  float64
 18  Longitude                                 21612 non-null  float64
 19  Living Area after Renovation (in Sqft)    21612 non-null  float64
 20  Lot Area after Renovation (in Sqft)       21613 non-null  int64  
dtypes: float64(10), int64(7), object(4)
memory usage: 3.5+ MB
```

In [4]:
```
df.describe()
```

Out[4]:

	ID	Sale Price	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Are
count	2.161300e+04	2.160900e+04	21613.000000	21609.000000	21604.000000	2.160400
mean	4.580302e+09	5.401984e+05	3.370842	2.114732	2079.931772	1.510776
std	2.876566e+09	3.673890e+05	0.930062	0.770138	918.487597	4.142827
min	1.000102e+06	7.500000e+04	0.000000	0.000000	290.000000	5.200000
25%	2.123049e+09	3.219500e+05	3.000000	1.750000	1429.250000	5.040000
50%	3.904930e+09	4.500000e+05	3.000000	2.250000	1910.000000	7.617500
75%	7.308900e+09	6.450000e+05	4.000000	2.500000	2550.000000	1.068825
max	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000	1.651359

Checking for duplicate rows and columns

In [6]:
```
df.duplicated().sum() #no duplicated rows
```

Out[6]:
```
0
```

In [7]:
```
dup = df.columns.duplicated()
dup
# no duplicated columns
```

Out[7]:
```
array([False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False])
```

Checking for missing values and handling of missing values

In [9]:
```
df.isna().sum()
#There are missing values in the columns, sale price, no of bathrooms, flat area
#zipcode, latitude, longitude, living area after renovation.
```

Out[9]:
```
ID                                            0
Date House was Sold                           0
Sale Price                                    4
No of Bedrooms                                0
No of Bathrooms                               4
Flat Area (in Sqft)                           9
Lot Area (in Sqft)                            9
No of Floors                                  0
Waterfront View                               0
No of Times Visited                       19489
Condition of the House                        0
Overall Grade                                 0
Area of the House from Basement (in Sqft)     3
Basement Area (in Sqft)                       0
Age of House (in Years)                       0
Renovated Year                                0
Zipcode                                       1
Latitude                                      1
Longitude                                     1
Living Area after Renovation (in Sqft)        1
Lot Area after Renovation (in Sqft)           0
dtype: int64
```

In [10]:
```
df.drop('No of Times Visited', axis=1, inplace=True)
df
```

Out[10]:

	ID	Date House was Sold	Sale Price	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors
0	7129300520	14 October 2017	221900.0	3	1.00	1180.0	5650.0	1.0
1	6414100192	14 December 2017	538000.0	3	2.25	2570.0	7242.0	2.0
2	5631500400	15 February 2016	180000.0	2	1.00	770.0	10000.0	1.0
3	2487200875	14 December 2017	604000.0	4	3.00	1960.0	5000.0	1.0
4	1954400510	15 February 2016	510000.0	3	2.00	1680.0	8080.0	1.0
...	...	...	...	...	...	...	...	...
21608	263000018	14 May 2017	360000.0	3	2.50	1530.0	1131.0	3.0
21609	6600060120	15 February 2016	400000.0	4	2.50	2310.0	5813.0	2.0
21610	1523300141	14 June 2017	402101.0	2	0.75	1020.0	1350.0	2.0
21611	291310100	15 January 2016	400000.0	3	2.50	1600.0	2388.0	2.0
21612	1523300157	14 October 2017	325000.0	2	0.75	1020.0	1076.0	2.0

21613 rows × 20 columns

In [11]:
```
numerical = df.select_dtypes(include=["number"])
numerical
```

Out[11]:

	ID	Sale Price	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors	Overall Grade
0	7129300520	221900.0	3	1.00	1180.0	5650.0	1.0	7
1	6414100192	538000.0	3	2.25	2570.0	7242.0	2.0	7
2	5631500400	180000.0	2	1.00	770.0	10000.0	1.0	6
3	2487200875	604000.0	4	3.00	1960.0	5000.0	1.0	7
4	1954400510	510000.0	3	2.00	1680.0	8080.0	1.0	8
...	...	...	...	...	...	...	...	...
21608	263000018	360000.0	3	2.50	1530.0	1131.0	3.0	8
21609	6600060120	400000.0	4	2.50	2310.0	5813.0	2.0	8
21610	1523300141	402101.0	2	0.75	1020.0	1350.0	2.0	7
21611	291310100	400000.0	3	2.50	1600.0	2388.0	2.0	8
21612	1523300157	325000.0	2	0.75	1020.0	1076.0	2.0	7

21613 rows × 17 columns

In [12]:
```
numerical.columns.tolist()
```

Out[12]:
```
['ID',
 'Sale Price',
 'No of Bedrooms',
 'No of Bathrooms',
 'Flat Area (in Sqft)',
 'Lot Area (in Sqft)',
 'No of Floors',
 'Overall Grade',
 'Area of the House from Basement (in Sqft)',
 'Basement Area (in Sqft)',
 'Age of House (in Years)',
 'Renovated Year',
 'Zipcode',
 'Latitude',
 'Longitude',
 'Living Area after Renovation (in Sqft)',
 'Lot Area after Renovation (in Sqft)']
```

In [13]:
```
numerical.isna().sum()
```

Out[13]:
```
ID                                             0
Sale Price                                     4
No of Bedrooms                                 0
No of Bathrooms                                4
Flat Area (in Sqft)                            9
Lot Area (in Sqft)                             9
No of Floors                                   0
Overall Grade                                  0
Area of the House from Basement (in Sqft)      3
Basement Area (in Sqft)                        0
Age of House (in Years)                        0
Renovated Year                                 0
Zipcode                                        1
Latitude                                       1
Longitude                                      1
Living Area after Renovation (in Sqft)         1
Lot Area after Renovation (in Sqft)            0
dtype: int64
```

In [14]:
```
for col in numerical:
  numerical[col] = numerical[col].fillna(numerical[col].median())

numerical
```

Out[14]:

	ID	Sale Price	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors	Overall Grade
0	7129300520	221900.0	3	1.00	1180.0	5650.0	1.0	7
1	6414100192	538000.0	3	2.25	2570.0	7242.0	2.0	7
2	5631500400	180000.0	2	1.00	770.0	10000.0	1.0	6
3	2487200875	604000.0	4	3.00	1960.0	5000.0	1.0	7
4	1954400510	510000.0	3	2.00	1680.0	8080.0	1.0	8
...	...	...	...	...	...	...	...	...
21608	263000018	360000.0	3	2.50	1530.0	1131.0	3.0	8
21609	6600060120	400000.0	4	2.50	2310.0	5813.0	2.0	8
21610	1523300141	402101.0	2	0.75	1020.0	1350.0	2.0	7
21611	291310100	400000.0	3	2.50	1600.0	2388.0	2.0	8
21612	1523300157	325000.0	2	0.75	1020.0	1076.0	2.0	7

21613 rows × 17 columns

In [15]:
```
numerical.isna().sum()
#filled all missing values in numerical column
```

Out[15]:
```
ID                                          0
Sale Price                                  0
No of Bedrooms                              0
No of Bathrooms                             0
Flat Area (in Sqft)                         0
Lot Area (in Sqft)                          0
No of Floors                                0
Overall Grade                               0
Area of the House from Basement (in Sqft)   0
Basement Area (in Sqft)                     0
Age of House (in Years)                     0
Renovated Year                              0
Zipcode                                     0
Latitude                                    0
Longitude                                   0
Living Area after Renovation (in Sqft)      0
Lot Area after Renovation (in Sqft)         0
dtype: int64
```

In [16]:
```
object = df.select_dtypes(include=["object_"])
object
```

Out[16]:

	Date House was Sold	Waterfront View	Condition of the House
0	14 October 2017	No	Fair
1	14 December 2017	No	Fair
2	15 February 2016	No	Fair
3	14 December 2017	No	Excellent
4	15 February 2016	No	Fair
...	...	...	...
21608	14 May 2017	No	Fair
21609	15 February 2016	No	Fair
21610	14 June 2017	No	Fair
21611	15 January 2016	No	Fair
21612	14 October 2017	No	Fair

21613 rows × 3 columns

In [17]:
```
object.isna().sum()
```

Out[17]:
```
Date House was Sold       0
Waterfront View           0
Condition of the House    0
dtype: int64
```

In [18]:
```
filled = pd.concat([numerical,object], axis=1)
filled
#creating new dataframe with all missing values filled.
```

Out[18]:

	ID	Sale Price	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors	Overall Grade
0	7129300520	221900.0	3	1.00	1180.0	5650.0	1.0	7
1	6414100192	538000.0	3	2.25	2570.0	7242.0	2.0	7
2	5631500400	180000.0	2	1.00	770.0	10000.0	1.0	6
3	2487200875	604000.0	4	3.00	1960.0	5000.0	1.0	7
4	1954400510	510000.0	3	2.00	1680.0	8080.0	1.0	8
...	...	...	...	...	...	...	...	...
21608	263000018	360000.0	3	2.50	1530.0	1131.0	3.0	8
21609	6600060120	400000.0	4	2.50	2310.0	5813.0	2.0	8
21610	1523300141	402101.0	2	0.75	1020.0	1350.0	2.0	7
21611	291310100	400000.0	3	2.50	1600.0	2388.0	2.0	8
21612	1523300157	325000.0	2	0.75	1020.0	1076.0	2.0	7

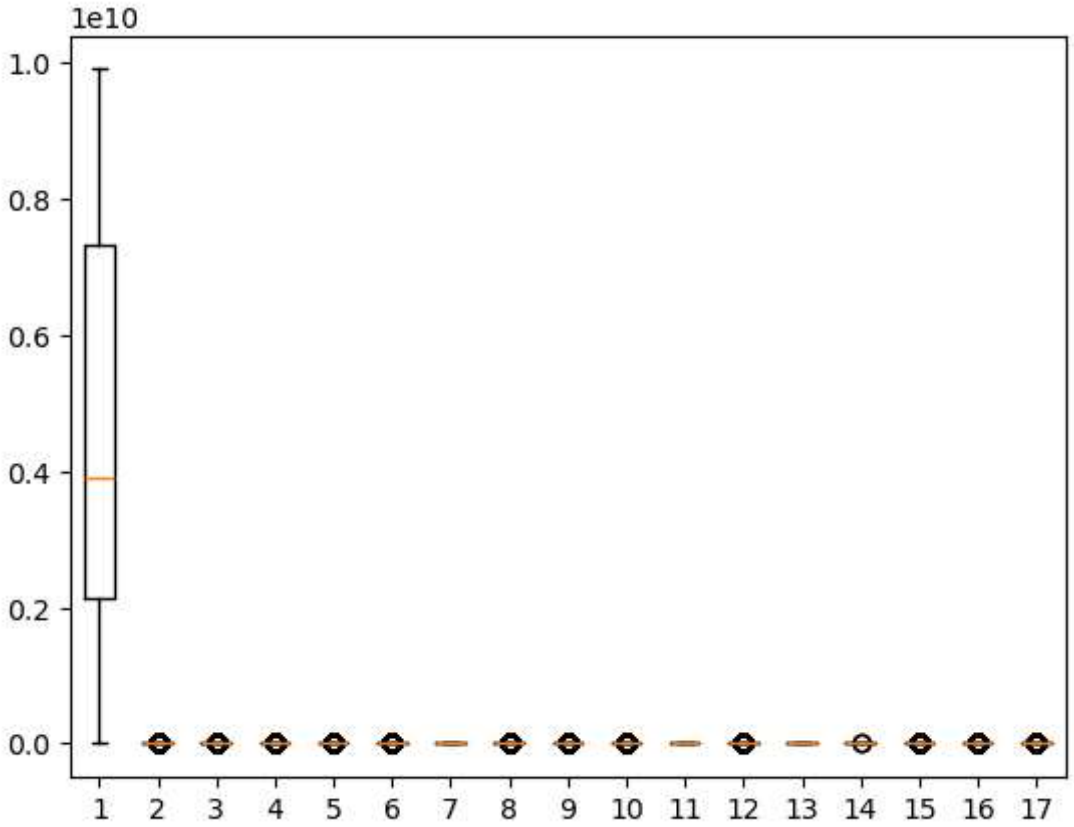
21613 rows × 20 columns

In [19]:
```
filled.isna().sum()
#checking if new data has any missing values.
```

Out[19]:
```
ID                                              0
Sale Price                                      0
No of Bedrooms                                  0
No of Bathrooms                                 0
Flat Area (in Sqft)                             0
Lot Area (in Sqft)                              0
No of Floors                                    0
Overall Grade                                   0
Area of the House from Basement (in Sqft)       0
Basement Area (in Sqft)                         0
Age of House (in Years)                         0
Renovated Year                                  0
Zipcode                                         0
Latitude                                        0
Longitude                                       0
Living Area after Renovation (in Sqft)          0
Lot Area after Renovation (in Sqft)             0
Date House was Sold                             0
Waterfront View                                 0
Condition of the House                          0
dtype: int64
```

Finding outliers and removing them

In [20]:
```
plt.boxplot(numerical)
plt.show()
```



In [40]:
```
def remove_outliers(numerical, column_name):
    q1 = numerical[column_name].quantile(0.25)
    q3 = numerical[column_name].quantile(0.75)
    iqr = q3 - q1
    upper_bound = q3 + 1.5 * iqr
```

```
    lower_bound = q1 - 1.5 * iqr
    numerical[column_name] = numerical[column_name].clip(upper=upper_bound)
    numerical[column_name] = numerical[column_name].clip(lower=lower_bound)
    return numerical[column_name]
```

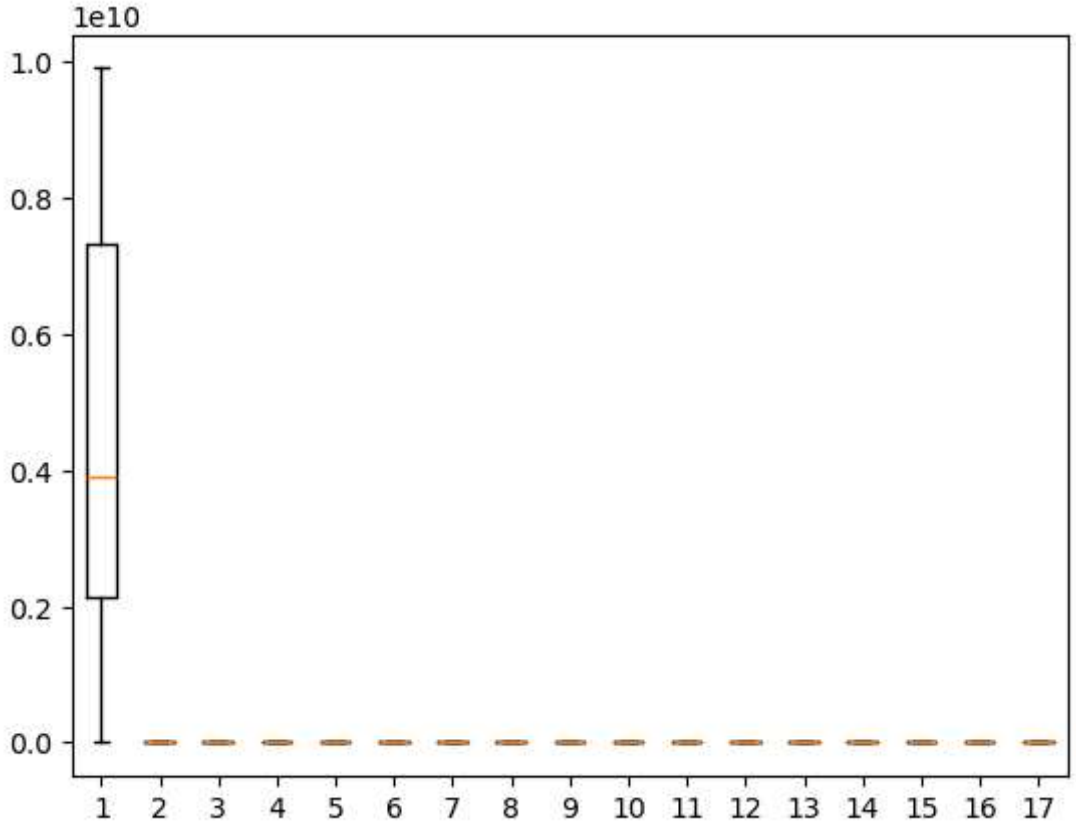
In [46]:
```
for col in numerical:
  numerical[col] = remove_outliers(numerical, col)

numerical
```

Out[46]:

	ID	Sale Price	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors	Overall Grade
0	7129300520	221900.0	3.0	1.00	1180.0	5650.0	1.0	7.0
1	6414100192	538000.0	3.0	2.25	2570.0	7242.0	2.0	7.0
2	5631500400	180000.0	2.0	1.00	770.0	10000.0	1.0	6.0
3	2487200875	604000.0	4.0	3.00	1960.0	5000.0	1.0	7.0
4	1954400510	510000.0	3.0	2.00	1680.0	8080.0	1.0	8.0
...	...	...	...	...	...	...	...	...
21608	263000018	360000.0	3.0	2.50	1530.0	1131.0	3.0	8.0
21609	6600060120	400000.0	4.0	2.50	2310.0	5813.0	2.0	8.0
21610	1523300141	402101.0	2.0	0.75	1020.0	1350.0	2.0	7.0
21611	291310100	400000.0	3.0	2.50	1600.0	2388.0	2.0	8.0
21612	1523300157	325000.0	2.0	0.75	1020.0	1076.0	2.0	7.0

21613 rows × 17 columns

In [48]:
```
plt.boxplot(numerical)
plt.show()
```



Scaling

In [109…

```
min_max_scaler = MinMaxScaler(feature_range=(1,2))
df['flat_area_scaled'] = min_max_scaler.fit_transform(df[['Flat Area (in Sqft)']
df
```

Out[109…

	ID	Date House was Sold	Sale Price	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors
0	7129300520	14 October 2017	221900.0	3	1.00	1180.0	5650.0	1.0
1	6414100192	14 December 2017	538000.0	3	2.25	2570.0	7242.0	2.0
2	5631500400	15 February 2016	180000.0	2	1.00	770.0	10000.0	1.0
3	2487200875	14 December 2017	604000.0	4	3.00	1960.0	5000.0	1.0
4	1954400510	15 February 2016	510000.0	3	2.00	1680.0	8080.0	1.0
...	...	...	...	...	...	...	...	...
21608	263000018	14 May 2017	360000.0	3	2.50	1530.0	1131.0	3.0
21609	6600060120	15 February 2016	400000.0	4	2.50	2310.0	5813.0	2.0
21610	1523300141	14 June 2017	402101.0	2	0.75	1020.0	1350.0	2.0
21611	291310100	15 January 2016	400000.0	3	2.50	1600.0	2388.0	2.0
21612	1523300157	14 October 2017	325000.0	2	0.75	1020.0	1076.0	2.0

21613 rows × 21 columns

In [111…

```
std_scaler = StandardScaler()
df['lot_area_scaled']= std_scaler.fit_transform(df[['Lot Area (in Sqft)']])
df
```

Out[111…

	ID	Date House was Sold	Sale Price	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors
0	7129300520	14 October 2017	221900.0	3	1.00	1180.0	5650.0	1.0
1	6414100192	14 December 2017	538000.0	3	2.25	2570.0	7242.0	2.0
2	5631500400	15 February 2016	180000.0	2	1.00	770.0	10000.0	1.0
3	2487200875	14 December 2017	604000.0	4	3.00	1960.0	5000.0	1.0
4	1954400510	15 February 2016	510000.0	3	2.00	1680.0	8080.0	1.0
...	...	...	...	...	...	...	...	...
21608	263000018	14 May 2017	360000.0	3	2.50	1530.0	1131.0	3.0
21609	6600060120	15 February 2016	400000.0	4	2.50	2310.0	5813.0	2.0
21610	1523300141	14 June 2017	402101.0	2	0.75	1020.0	1350.0	2.0
21611	291310100	15 January 2016	400000.0	3	2.50	1600.0	2388.0	2.0
21612	1523300157	14 October 2017	325000.0	2	0.75	1020.0	1076.0	2.0

21613 rows × 22 columns

In [113…

```
std_scaler = StandardScaler()
df['living_area_scaled']= std_scaler.fit_transform(df[['Living Area after Renova
df
```

Out[113…

	ID	Date House was Sold	Sale Price	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors
0	7129300520	14 October 2017	221900.0	3	1.00	1180.0	5650.0	1.0
1	6414100192	14 December 2017	538000.0	3	2.25	2570.0	7242.0	2.0
2	5631500400	15 February 2016	180000.0	2	1.00	770.0	10000.0	1.0
3	2487200875	14 December 2017	604000.0	4	3.00	1960.0	5000.0	1.0
4	1954400510	15 February 2016	510000.0	3	2.00	1680.0	8080.0	1.0
...	...	...	...	...	...	...	...	...
21608	263000018	14 May 2017	360000.0	3	2.50	1530.0	1131.0	3.0
21609	6600060120	15 February 2016	400000.0	4	2.50	2310.0	5813.0	2.0
21610	1523300141	14 June 2017	402101.0	2	0.75	1020.0	1350.0	2.0
21611	291310100	15 January 2016	400000.0	3	2.50	1600.0	2388.0	2.0
21612	1523300157	14 October 2017	325000.0	2	0.75	1020.0	1076.0	2.0

21613 rows × 23 columns

In [115…

```
num = df.select_dtypes(include=['number'])
num
```

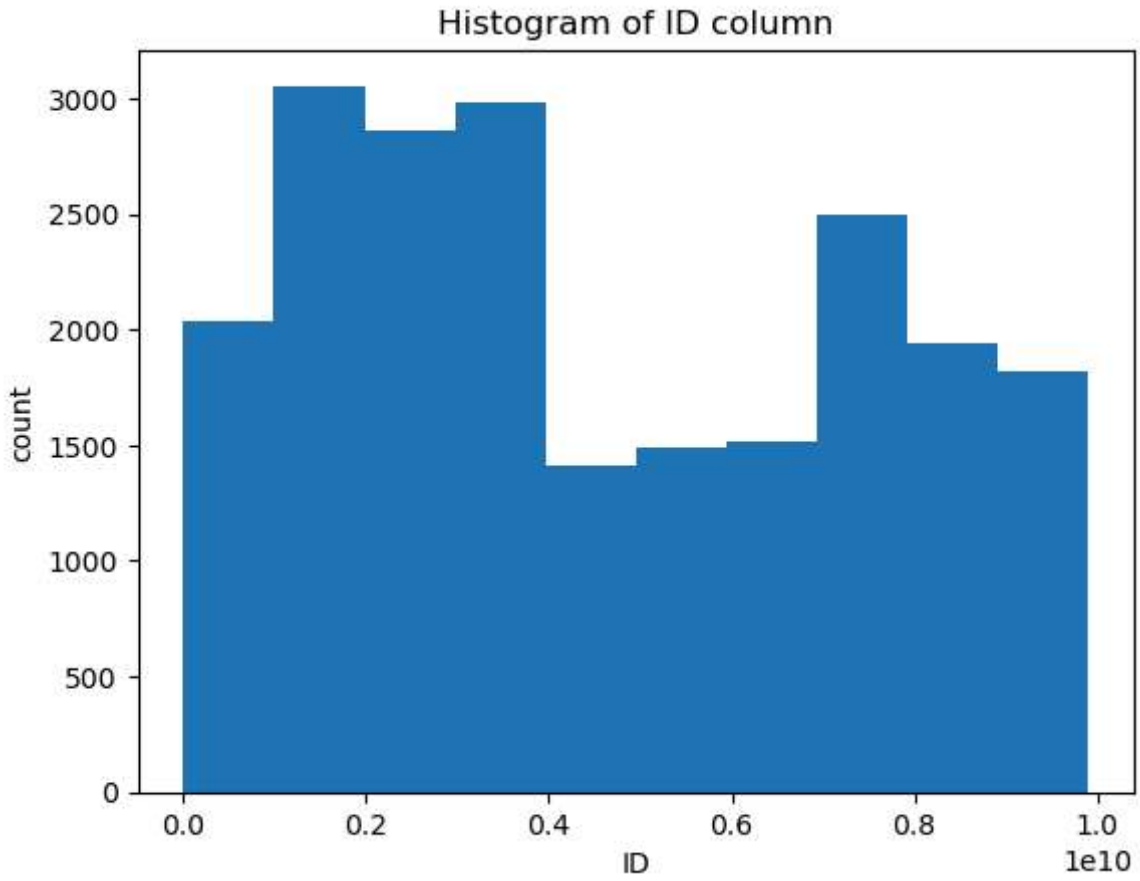
Out[115…

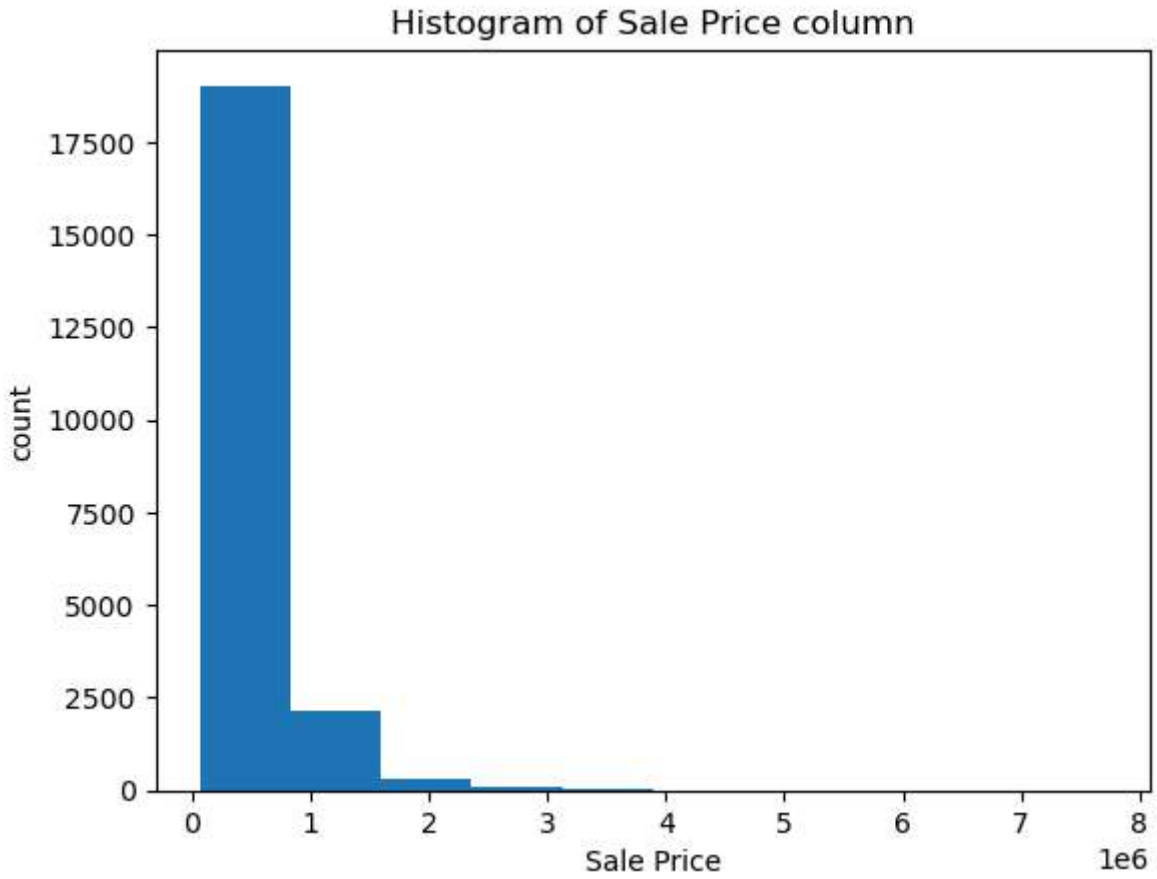
	ID	Sale Price	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors	Overall Grade
0	7129300520	221900.0	3	1.00	1180.0	5650.0	1.0	7
1	6414100192	538000.0	3	2.25	2570.0	7242.0	2.0	7
2	5631500400	180000.0	2	1.00	770.0	10000.0	1.0	6
3	2487200875	604000.0	4	3.00	1960.0	5000.0	1.0	7
4	1954400510	510000.0	3	2.00	1680.0	8080.0	1.0	8
…	…	…	…	…	…	…	…	…
21608	263000018	360000.0	3	2.50	1530.0	1131.0	3.0	8
21609	6600060120	400000.0	4	2.50	2310.0	5813.0	2.0	8
21610	1523300141	402101.0	2	0.75	1020.0	1350.0	2.0	7
21611	291310100	400000.0	3	2.50	1600.0	2388.0	2.0	8
21612	1523300157	325000.0	2	0.75	1020.0	1076.0	2.0	7

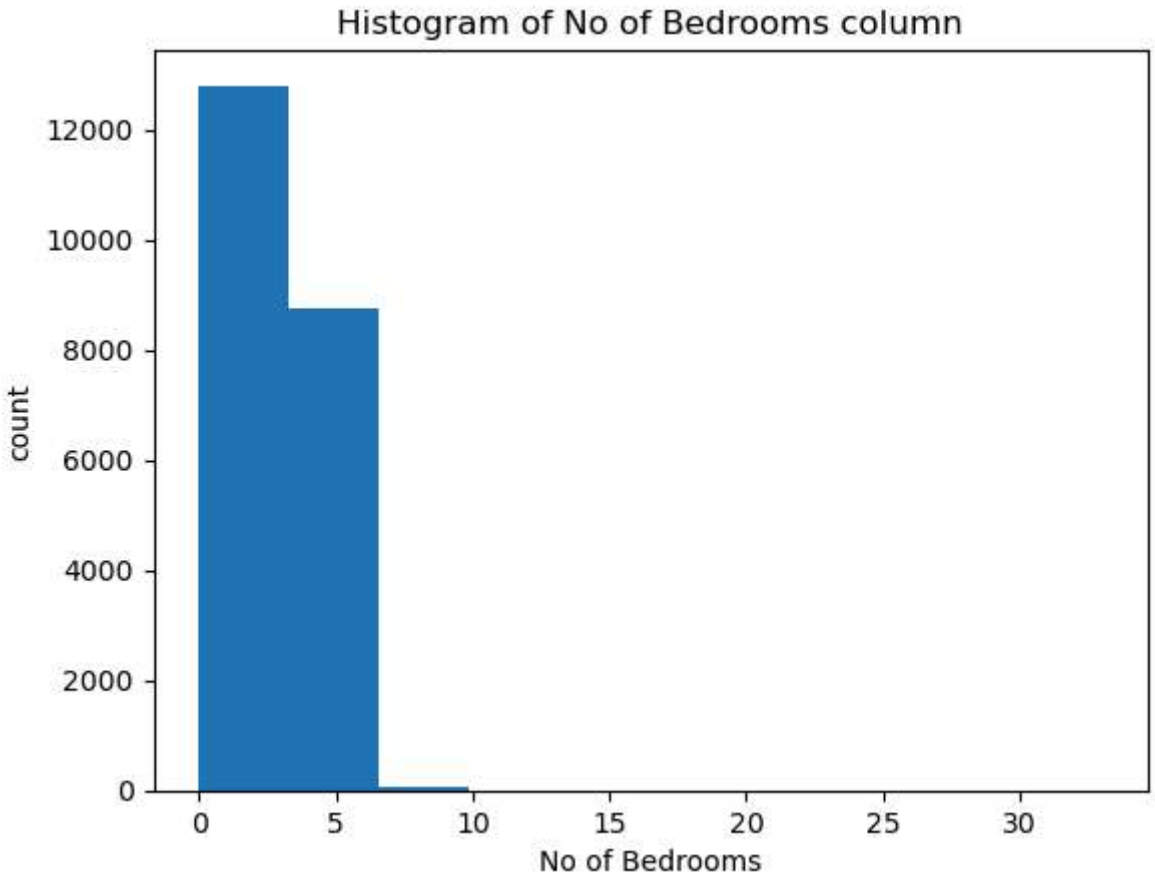
21613 rows × 20 columns

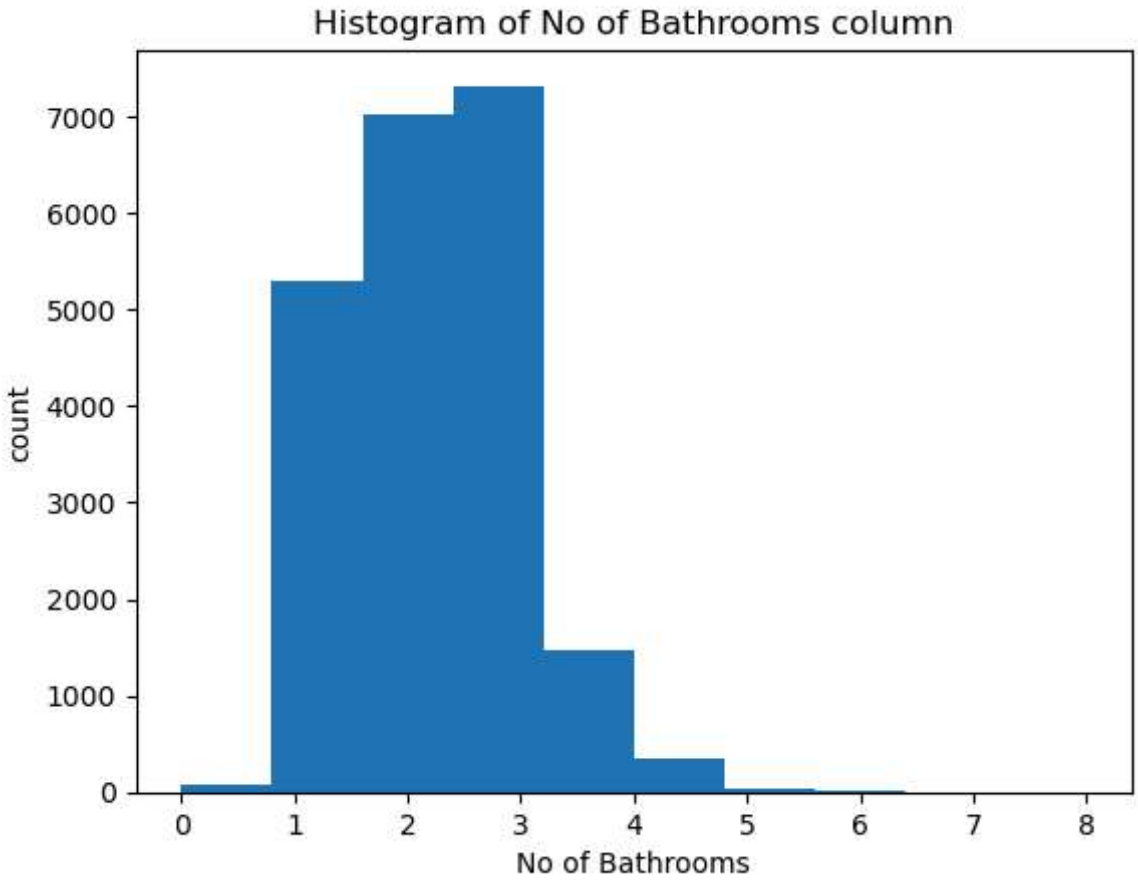
In [117…

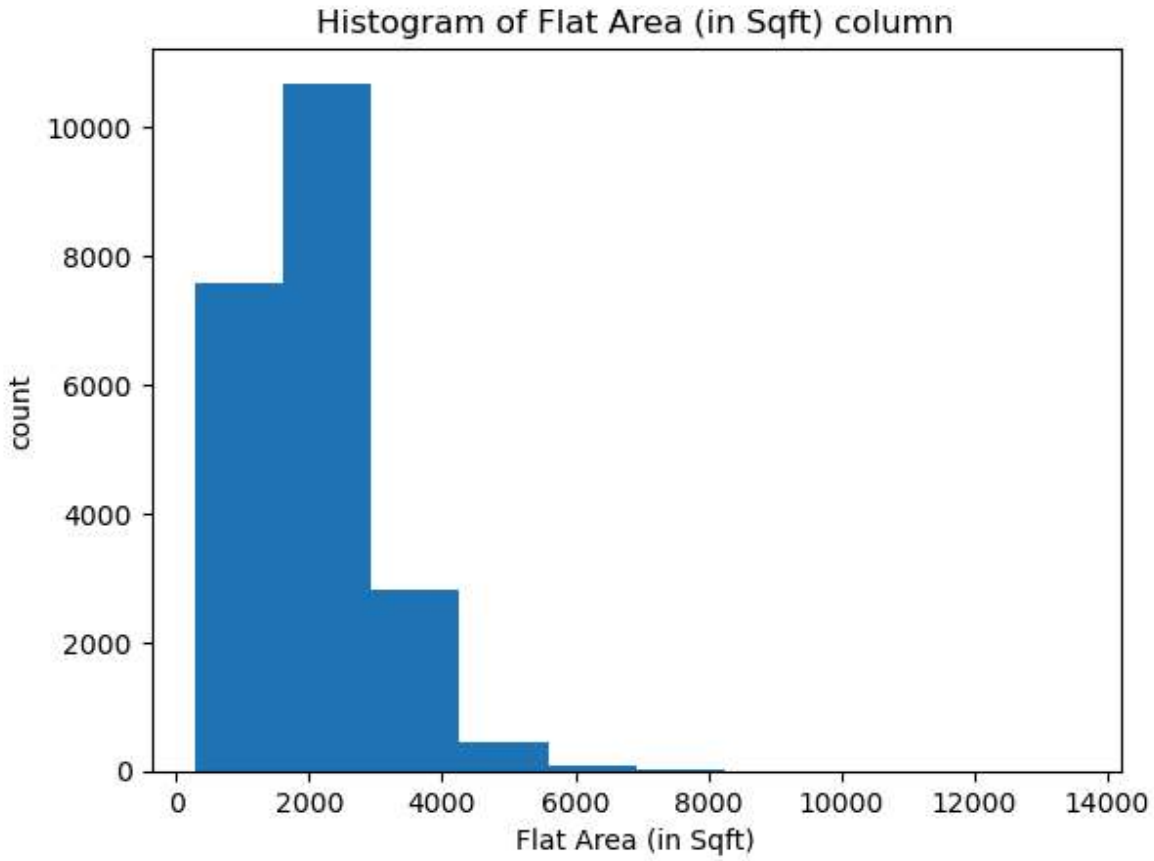
```
for c in num:
  plt.hist(num[c])
  plt.title("Histogram of {} column".format(c))
  plt.xlabel(c)
  plt.ylabel("count")
  plt.show()
```

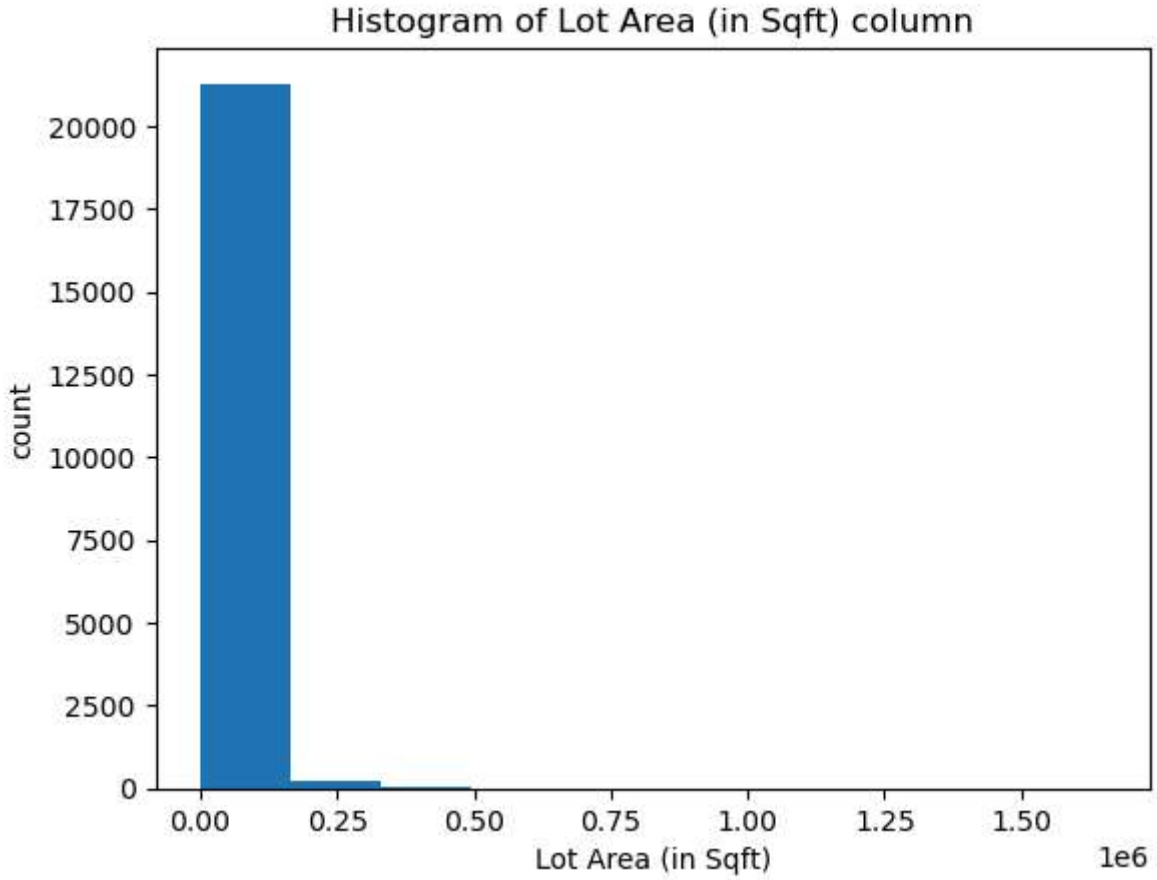

Histogram of ID column
count
ID
1e10
0
500
1000
1500
2000
2500
3000
0.0
0.2
0.4
0.6
0.8
1.0

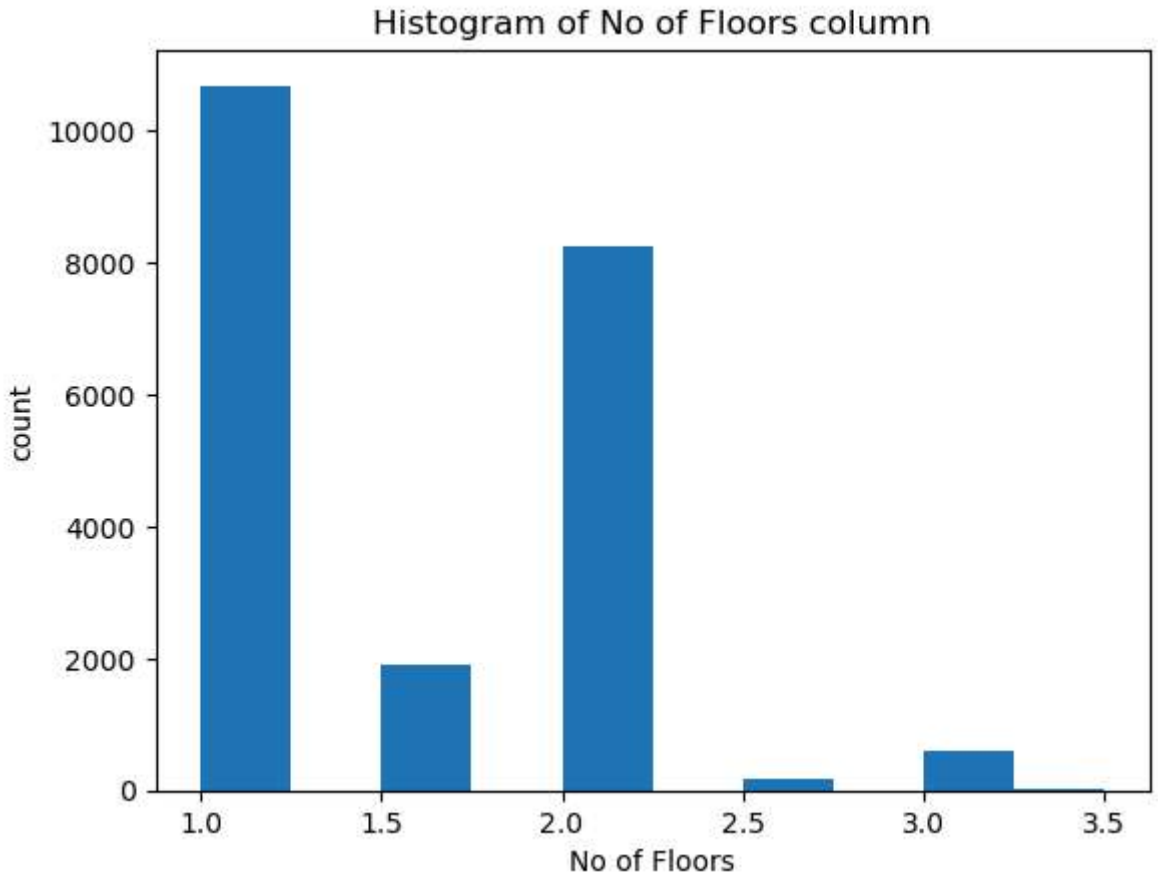

Histogram of Sale Price column
count
Sale Price
1e6
0
2500
5000
7500
10000
12500
15000
17500
0
1
2
3
4
5
6
7
8

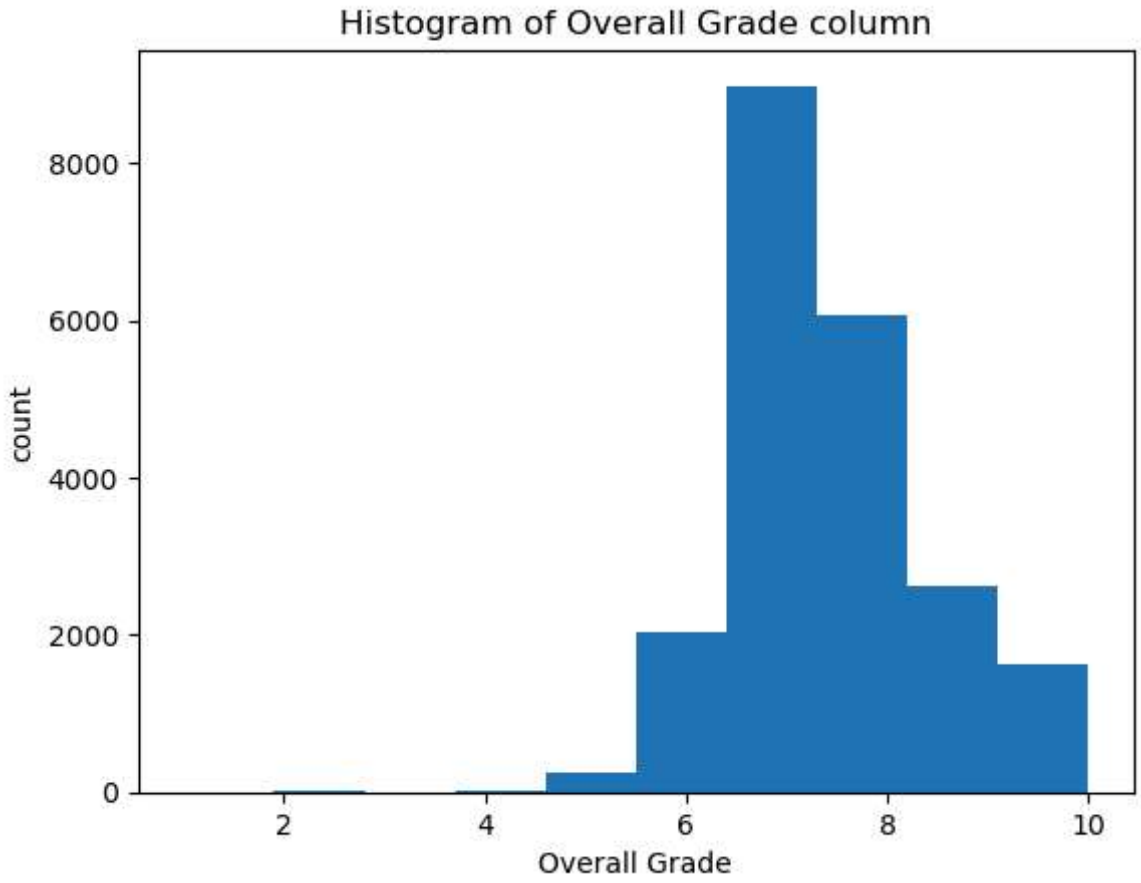

Histogram of No of Bedrooms column
count
No of Bedrooms
0
2000
4000
6000
8000
10000
12000
0
5
10
15
20
25
30

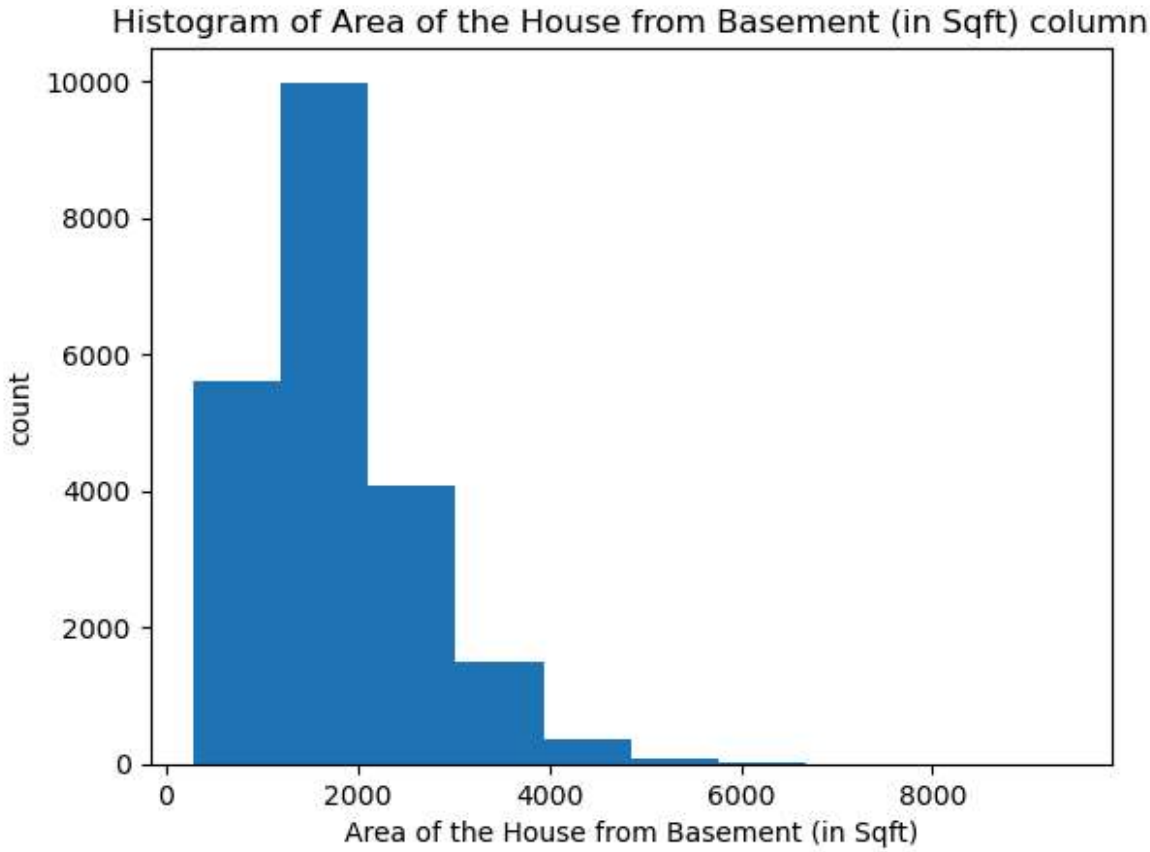

Histogram of No of Bathrooms column
count
No of Bathrooms
0
1000
2000
3000
4000
5000
6000
7000
0
1
2
3
4
5
6
7
8

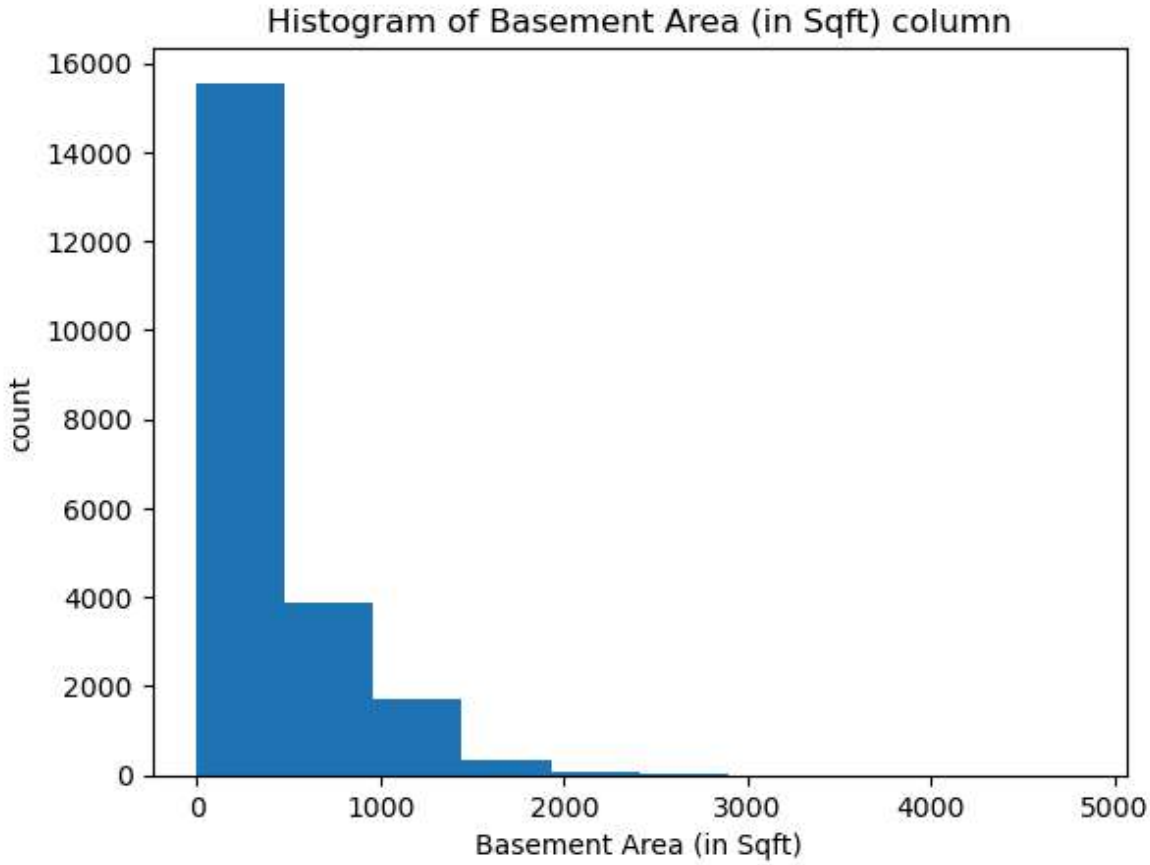

Histogram of Flat Area (in Sqft) column
count
Flat Area (in Sqft)
0
2000
4000
6000
8000
10000
12000
14000

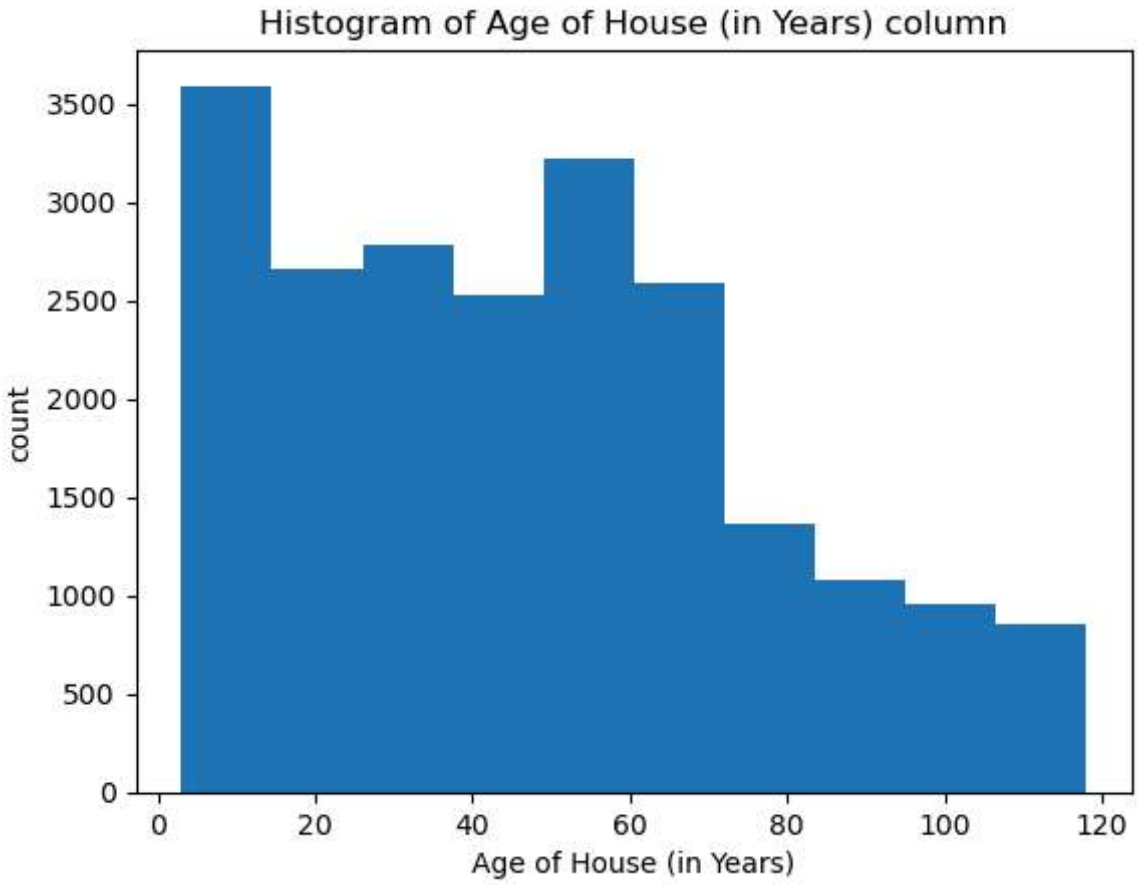

Histogram of Lot Area (in Sqft) column
count
Lot Area (in Sqft)
1e6
0.00
0.25
0.50
0.75
1.00
1.25
1.50

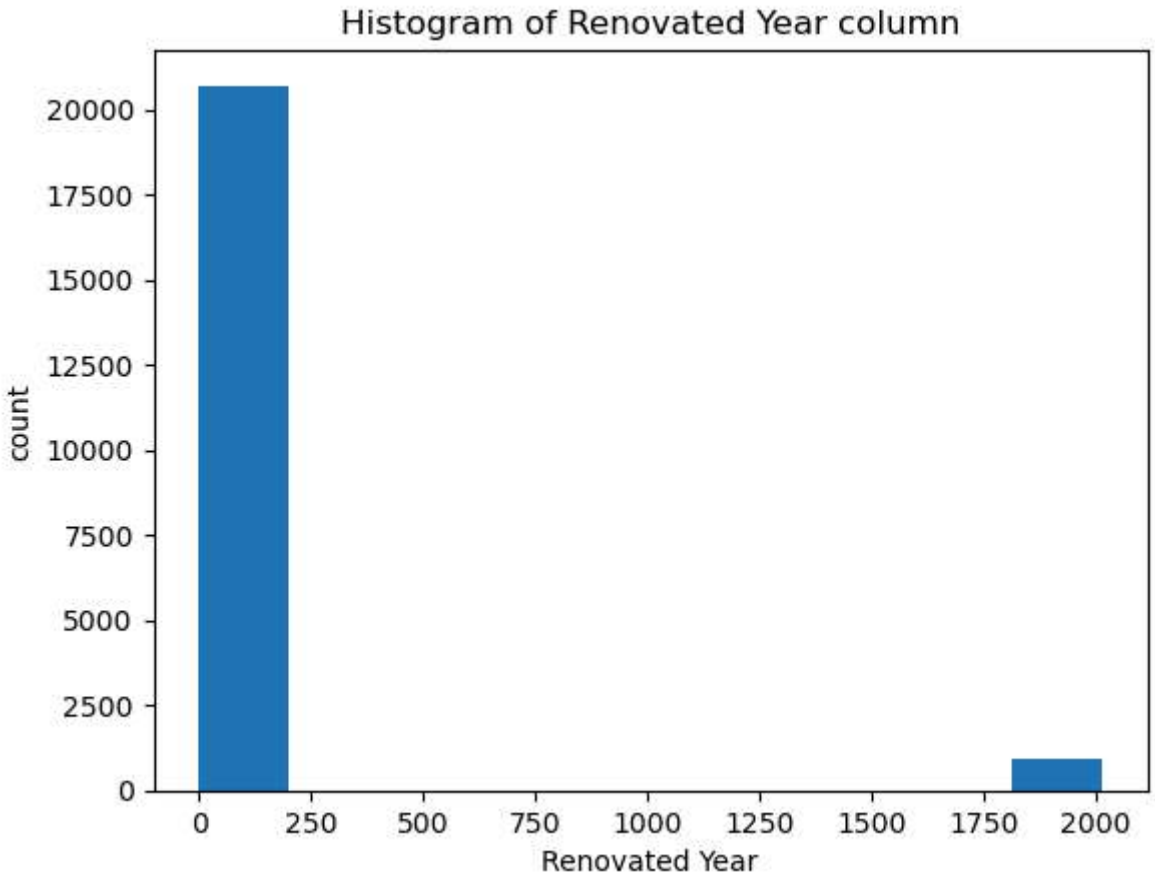

Histogram of No of Floors column
count
10000
8000
6000
4000
2000
0
1.0
1.5
2.0
2.5
3.0
3.5
No of Floors

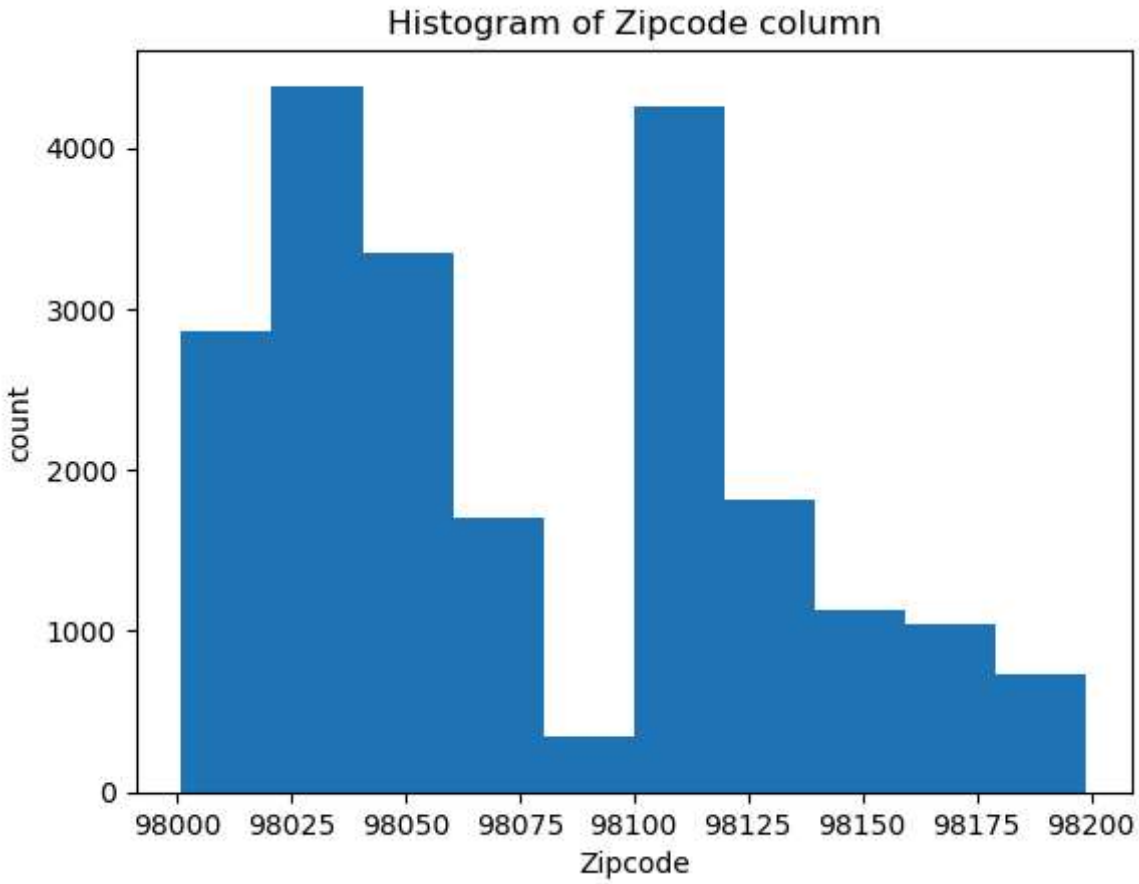

Histogram of Overall Grade column
count
8000
6000
4000
2000
0
2
4
6
8
10
Overall Grade

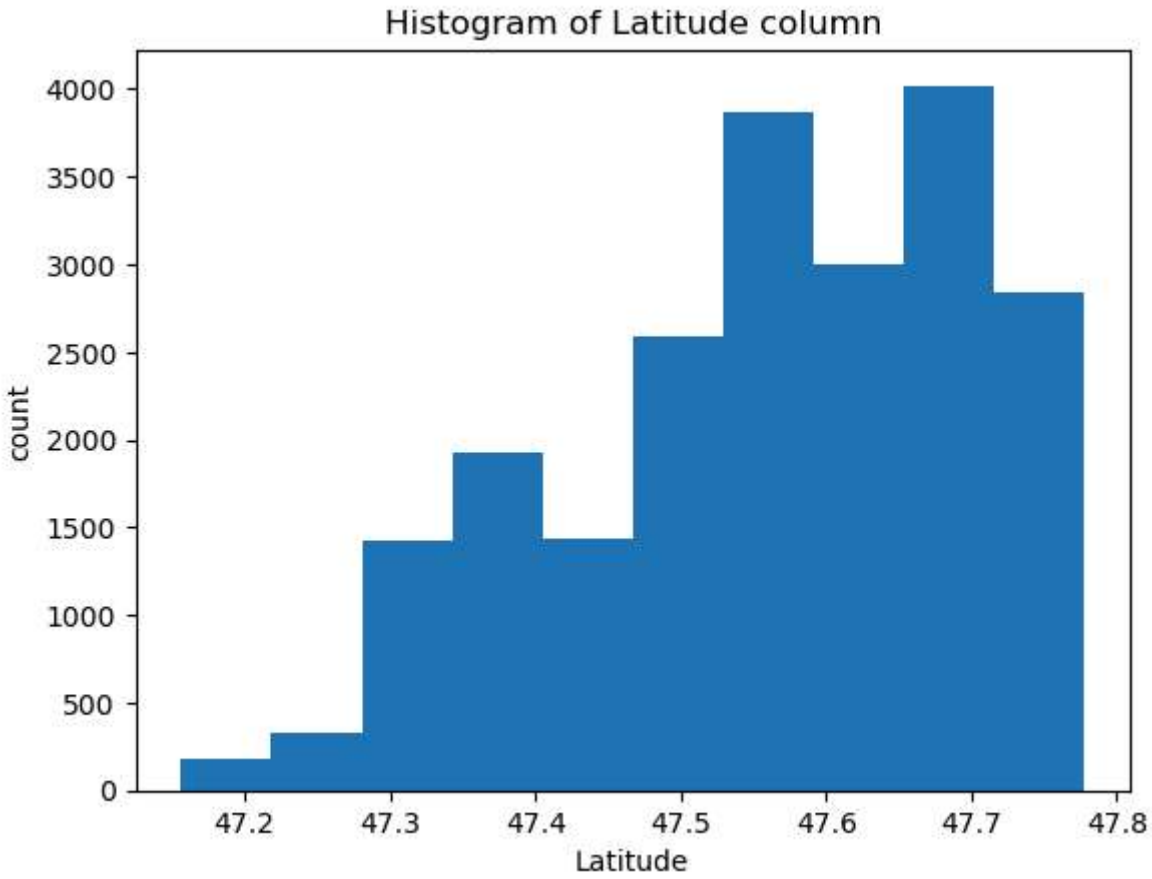

Histogram of Area of the House from Basement (in Sqft) column
count
Area of the House from Basement (in Sqft)
0
2000
4000
6000
8000
10000

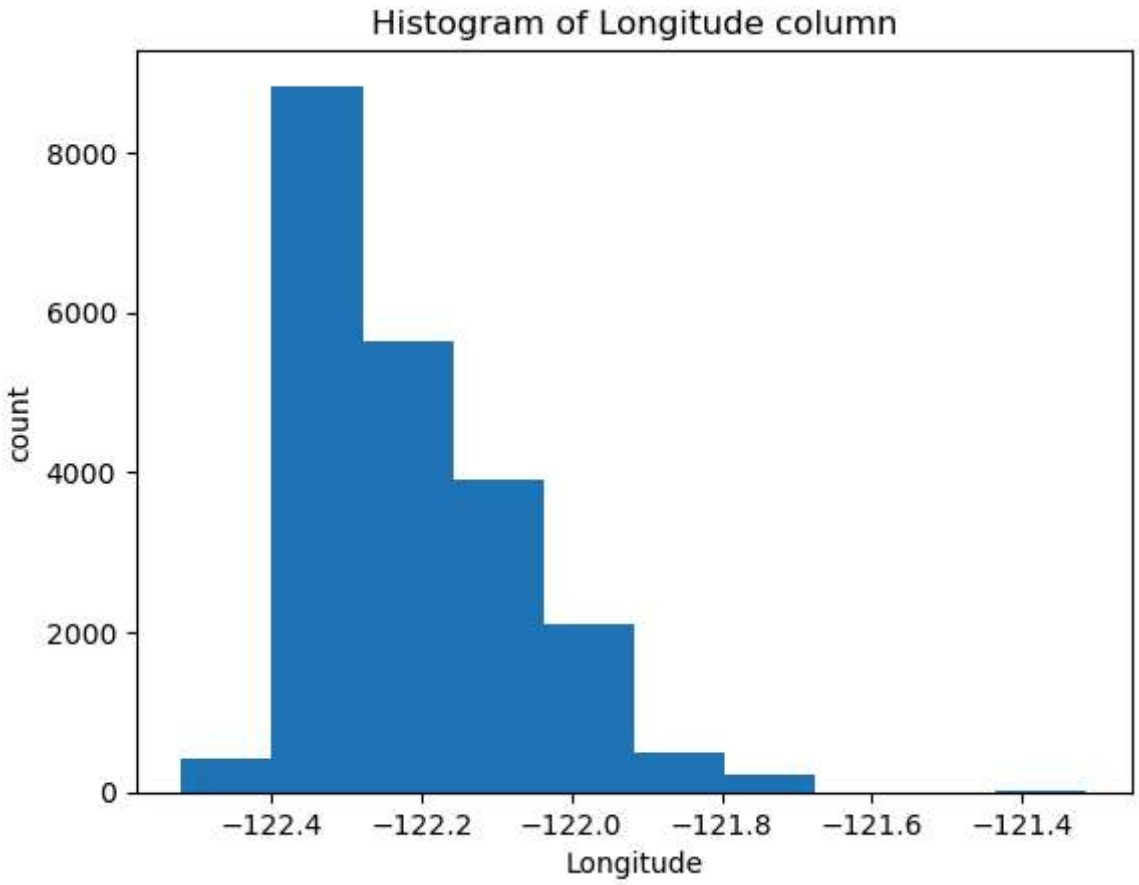

Histogram of Basement Area (in Sqft) column
count
Basement Area (in Sqft)
0
1000
2000
3000
4000
5000
6000
8000
10000
12000
14000
16000

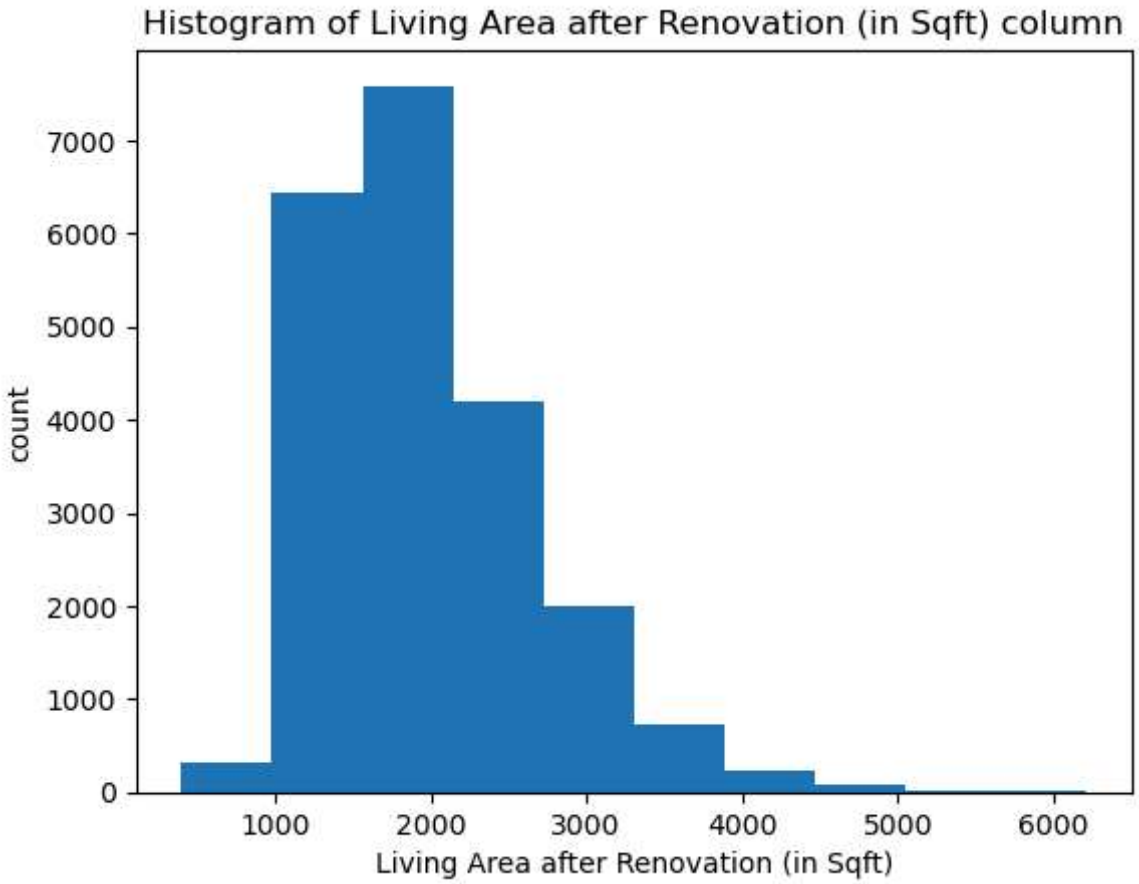

Histogram of Age of House (in Years) column
count
Age of House (in Years)
0
500
1000
1500
2000
2500
3000
3500
0
20
40
60
80
100
120

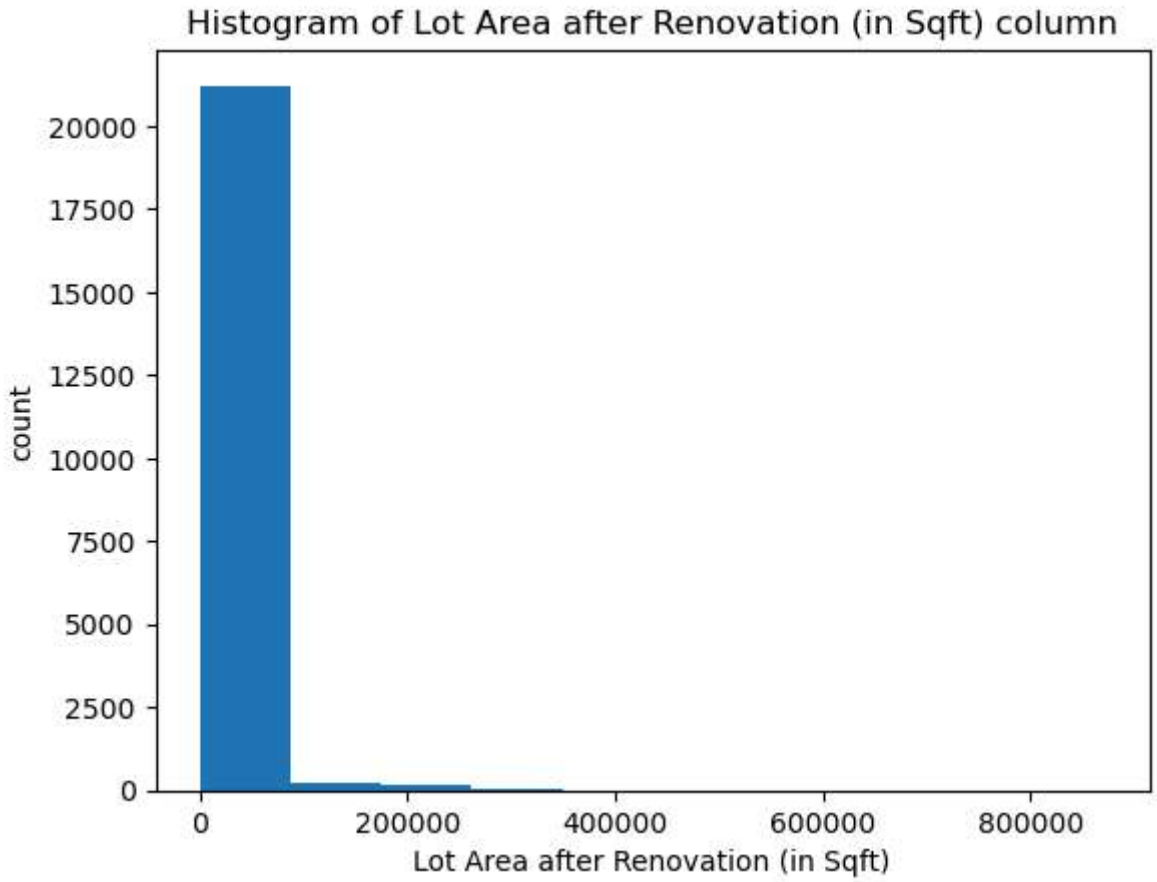

Histogram of Renovated Year column
count
Renovated Year
0
2500
5000
7500
10000
12500
15000
17500
20000
0
250
500
750
1000
1250
1500
1750
2000

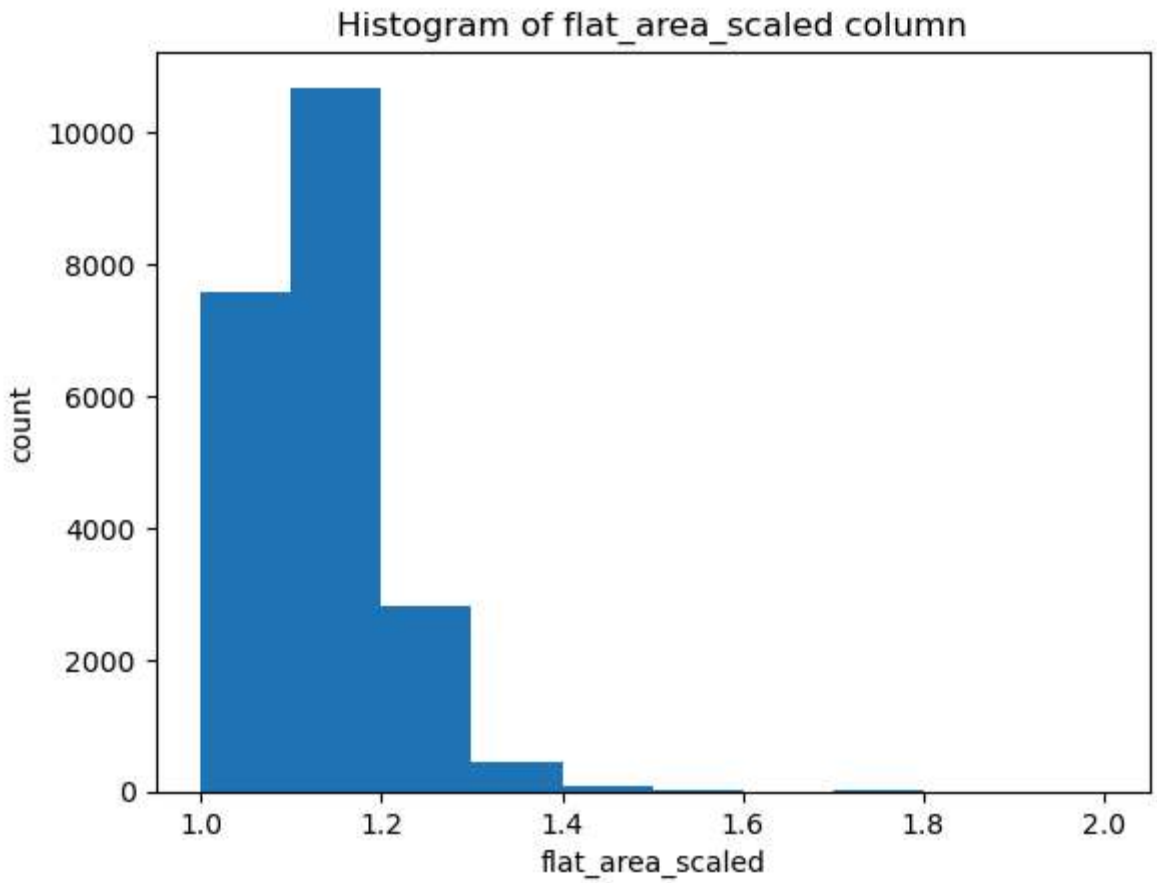

Histogram of Zipcode column
count
0
1000
2000
3000
4000
98000
98025
98050
98075
98100
98125
98150
98175
98200
Zipcode

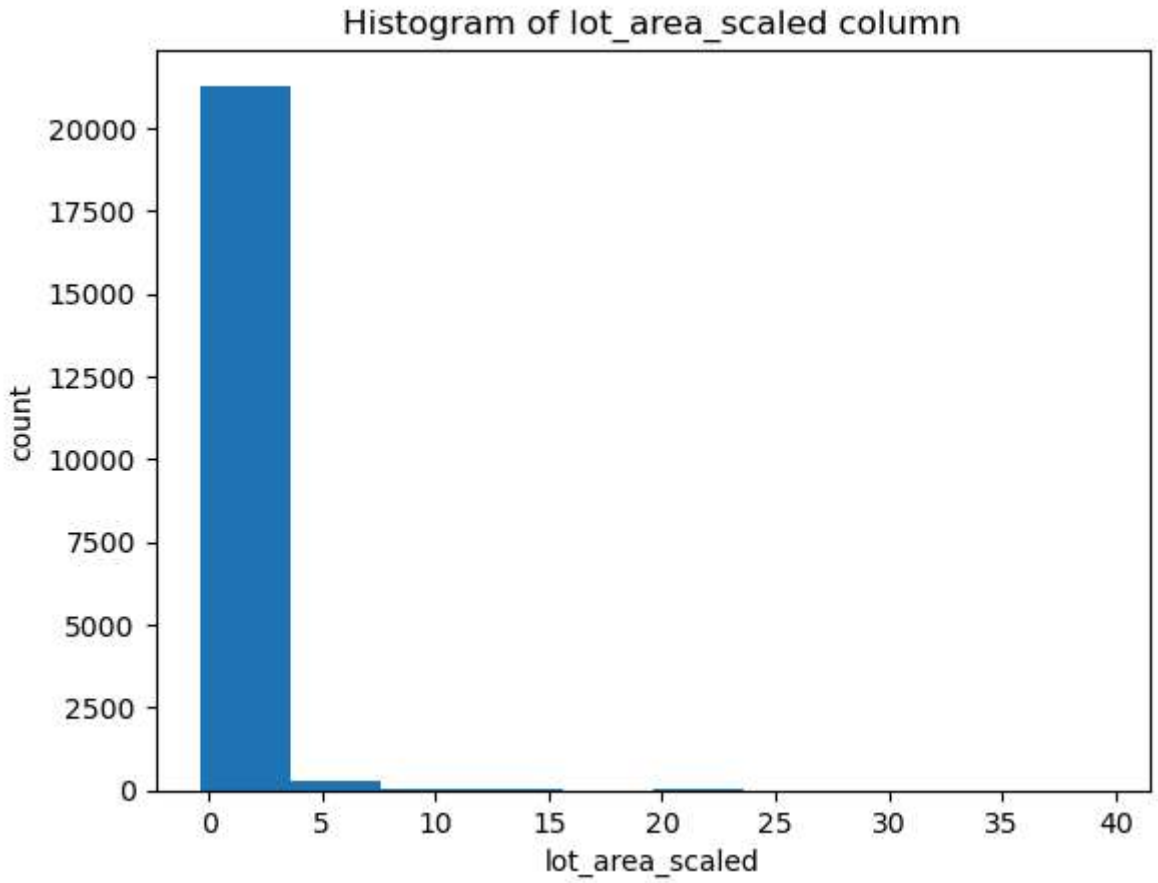

Histogram of Latitude column
count
0
500
1000
1500
2000
2500
3000
3500
4000
47.2
47.3
47.4
47.5
47.6
47.7
47.8
Latitude

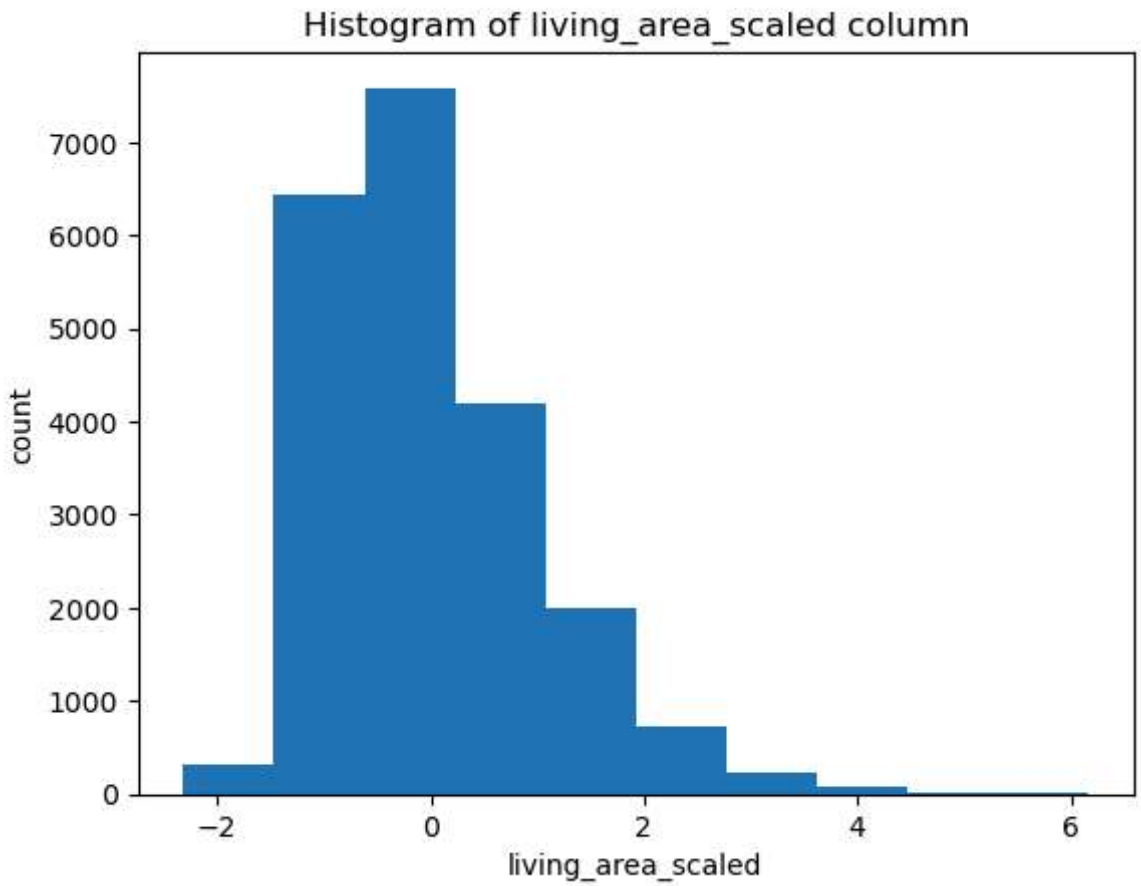

Histogram of Longitude column
count
8000
6000
4000
2000
0
−122.4
−122.2
−122.0
−121.8
−121.6
−121.4
Longitude


Histogram of Living Area after Renovation (in Sqft) column
count
7000
6000
5000
4000
3000
2000
1000
0
1000
2000
3000
4000
5000
6000
Living Area after Renovation (in Sqft)


Histogram of Lot Area after Renovation (in Sqft) column
count
20000
17500
15000
12500
10000
7500
5000
2500
0
0
200000
400000
600000
800000
Lot Area after Renovation (in Sqft)


Histogram of flat_area_scaled column
count
10000
8000
6000
4000
2000
0
1.0
1.2
1.4
1.6
1.8
2.0
flat_area_scaled





In [124…

```
cat = df.select_dtypes(include=["category"])
cat
```

Out[124…

0
1
2
3
4
...
21608
21609
21610
21611
21612

21613 rows × 0 columns

In [136…

```
label_enc = LabelEncoder()
df['View'] = label_enc.fit_transform(df['Waterfront View'])
df
```

Out[136…

	ID	Date House was Sold	Sale Price	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors
0	7129300520	14 October 2017	221900.0	3	1.00	1180.0	5650.0	1.0
1	6414100192	14 December 2017	538000.0	3	2.25	2570.0	7242.0	2.0
2	5631500400	15 February 2016	180000.0	2	1.00	770.0	10000.0	1.0
3	2487200875	14 December 2017	604000.0	4	3.00	1960.0	5000.0	1.0
4	1954400510	15 February 2016	510000.0	3	2.00	1680.0	8080.0	1.0
…	…	…	…	…	…	…	…	…
21608	263000018	14 May 2017	360000.0	3	2.50	1530.0	1131.0	3.0
21609	6600060120	15 February 2016	400000.0	4	2.50	2310.0	5813.0	2.0
21610	1523300141	14 June 2017	402101.0	2	0.75	1020.0	1350.0	2.0
21611	291310100	15 January 2016	400000.0	3	2.50	1600.0	2388.0	2.0
21612	1523300157	14 October 2017	325000.0	2	0.75	1020.0	1076.0	2.0

21613 rows × 24 columns

In [140…

```
label_enc = LabelEncoder()
df['condition'] = label_enc.fit_transform(df['Condition of the House'])
df

#there are no columns for which one hot encoding eeds to be done. so only label
```

Out[140…

	ID	Date House was Sold	Sale Price	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors
0	7129300520	14 October 2017	221900.0	3	1.00	1180.0	5650.0	1.0
1	6414100192	14 December 2017	538000.0	3	2.25	2570.0	7242.0	2.0
2	5631500400	15 February 2016	180000.0	2	1.00	770.0	10000.0	1.0
3	2487200875	14 December 2017	604000.0	4	3.00	1960.0	5000.0	1.0
4	1954400510	15 February 2016	510000.0	3	2.00	1680.0	8080.0	1.0
…	…	…	…	…	…	…	…	…
21608	263000018	14 May 2017	360000.0	3	2.50	1530.0	1131.0	3.0
21609	6600060120	15 February 2016	400000.0	4	2.50	2310.0	5813.0	2.0
21610	1523300141	14 June 2017	402101.0	2	0.75	1020.0	1350.0	2.0
21611	291310100	15 January 2016	400000.0	3	2.50	1600.0	2388.0	2.0
21612	1523300157	14 October 2017	325000.0	2	0.75	1020.0	1076.0	2.0

21613 rows × 25 columns

In [167…

```
waterfront_onehot = pd.get_dummies(df, columns=['Waterfront View'], prefix='wf')
waterfront_onehot
```

Out[167…

	ID	Date House was Sold	Sale Price	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors
0	7129300520	14 October 2017	221900.0	3	1.00	1180.0	5650.0	1.0
1	6414100192	14 December 2017	538000.0	3	2.25	2570.0	7242.0	2.0
2	5631500400	15 February 2016	180000.0	2	1.00	770.0	10000.0	1.0
3	2487200875	14 December 2017	604000.0	4	3.00	1960.0	5000.0	1.0
4	1954400510	15 February 2016	510000.0	3	2.00	1680.0	8080.0	1.0
…	…	…	…	…	…	…	…	…
21608	263000018	14 May 2017	360000.0	3	2.50	1530.0	1131.0	3.0
21609	6600060120	15 February 2016	400000.0	4	2.50	2310.0	5813.0	2.0
21610	1523300141	14 June 2017	402101.0	2	0.75	1020.0	1350.0	2.0
21611	291310100	15 January 2016	400000.0	3	2.50	1600.0	2388.0	2.0
21612	1523300157	14 October 2017	325000.0	2	0.75	1020.0	1076.0	2.0

21613 rows × 26 columns

In [173…

```
y=waterfront_onehot[["Sale Price"]]
y
```

Out[173…

	Sale Price
0	221900.0
1	538000.0
2	180000.0
3	604000.0
4	510000.0
…	…
21608	360000.0
21609	400000.0
21610	402101.0
21611	400000.0
21612	325000.0

21613 rows × 1 columns

In [175…

```
x=waterfront_onehot.drop(['Sale Price'],axis=1)
x
```

Out[175…

	ID	Date House was Sold	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors	Conditio of th Hous
0	7129300520	14 October 2017	3	1.00	1180.0	5650.0	1.0	Fa
1	6414100192	14 December 2017	3	2.25	2570.0	7242.0	2.0	Fa
2	5631500400	15 February 2016	2	1.00	770.0	10000.0	1.0	Fa
3	2487200875	14 December 2017	4	3.00	1960.0	5000.0	1.0	Excelle
4	1954400510	15 February 2016	3	2.00	1680.0	8080.0	1.0	Fa
...	...	...	...	...	...	...	...	
21608	263000018	14 May 2017	3	2.50	1530.0	1131.0	3.0	Fa
21609	6600060120	15 February 2016	4	2.50	2310.0	5813.0	2.0	Fa
21610	1523300141	14 June 2017	2	0.75	1020.0	1350.0	2.0	Fa
21611	291310100	15 January 2016	3	2.50	1600.0	2388.0	2.0	Fa
21612	1523300157	14 October 2017	2	0.75	1020.0	1076.0	2.0	Fa

21613 rows × 25 columns

In [177…

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_
x_train
```

Out[177…

	ID	Date House was Sold	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors	Conditi of t Hou
15166	3304700130	15 January 2016	4	4.00	3860.0	67953.0	2.0	Go
13416	1115100119	14 December 2017	4	2.00	1920.0	7803.0	1.0	F
206	9526600140	14 September 2017	3	2.50	2440.0	4587.0	2.0	F
4286	9264921140	15 March 2016	3	2.75	1910.0	15508.0	1.0	F
18065	3826500730	14 September 2017	4	2.50	2130.0	9100.0	1.0	F
…	…	…	…	…	…	…	…	
1099	7227500830	14 May 2017	2	1.00	720.0	4222.0	1.0	Go
18898	984220330	14 August 2017	4	2.50	1820.0	9161.0	1.0	Go
11798	3629920990	14 June 2017	4	3.25	3440.0	7661.0	2.0	F
6637	3126049439	15 January 2016	2	1.50	870.0	747.0	2.0	F
2575	3629970240	14 November 2017	3	2.50	2820.0	5001.0	2.0	F

17290 rows × 25 columns

In [179…

```
y_train
```

Out[179…

	Sale Price
15166	1760000.0
13416	360000.0
206	677900.0
4286	300000.0
18065	220000.0
…	…
1099	151000.0
18898	325000.0
11798	905000.0
6637	313000.0
2575	690000.0

17290 rows × 1 columns

In [181…

```
x_test
```

Out[181…

	ID	Date House was Sold	No of Bedrooms	No of Bathrooms	Flat Area (in Sqft)	Lot Area (in Sqft)	No of Floors	Conditic of th Hou
6638	3491300052	15 April 2016	4	2.25	2410.0	4250.0	1.5	Excelle
7366	8917100020	14 June 2017	3	1.50	2170.0	16600.0	1.0	Fa
3158	5100401429	14 October 2017	2	1.00	1450.0	6380.0	1.0	Fa
9117	1454600156	14 June 2017	5	3.25	4500.0	9648.0	2.0	Goc
3392	1917300025	15 January 2016	2	1.00	860.0	6000.0	1.0	Fa
...	...	...	...	...	...	...	...	
8494	7305300090	14 November 2017	4	1.75	1530.0	8152.0	1.0	Excelle
5359	2310030500	14 July 2017	3	1.75	1580.0	9187.0	1.0	Fa
5242	1515910290	14 August 2017	4	2.50	2650.0	9451.0	2.0	Goc
13777	7625703065	14 June 2017	2	1.00	820.0	6250.0	1.0	Goc
1080	7888400560	14 October 2017	4	2.75	1810.0	8677.0	1.5	Fa

4323 rows × 25 columns

In [183…

```
y_test
```

Out[183…

	Sale Price
6638	735000.0
7366	1150000.0
3158	350500.0
9117	860000.0
3392	122000.0
…	…
8494	338000.0
5359	263000.0
5242	397450.0
13777	375000.0
1080	208000.0

4323 rows × 1 columns

In [ ]: