

MOUNIKA G

Perform the Below Tasks to complete the assignment:- Tasks:-

- 1. Download the dataset
- 2. Load the dataset.
- 3. Perform the Below Visualizations. • Univariate Analysis • Bi - Variate Analysis • Multi-Variate Analysis
- 4. Perform descriptive statistics on the dataset.
- 5. Handle the Missing values.

```
from google.colab import files
uploaded = files.upload()
```

Choose Files

House Price India.csv

- House Price India.csv(text/csv) - 1524561 bytes, last modified: 3/27/2023 - 100% done

Saving House Price India.csv to House Price India.csv

```
import pandas as pd
import seaborn as sns
import numpy as np
import io
df = pd.read_csv('House Price India.csv')
df.head()
```

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

```
df.tail()
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condition of the house	...	Built Year	Renovation Year	Postal Code
14615	6762830250	42734	2	1.5	1556	20000	1.0	0	0	4	...	1957	0	122066
14616	6762830339	42734	3	2.0	1680	7000	1.5	0	0	4	...	1968	0	122072
14617	6762830618	42734	2	1.0	1070	6120	1.0	0	0	3	...	1962	0	122056
14618	6762830709	42734	4	1.0	1030	6621	1.0	0	0	4	...	1955	0	122042
14619	6762831463	42734	3	1.0	900	4770	1.0	0	0	3	...	1969	2009	122018

```
df
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condition of the house	...	Built Year	Renovation Year	Postal Code
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5	...	1921	0	122003
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5	...	1909	0	122004
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3	...	1939	0	122004
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3	...	2001	0	122005

df.columns

```
Index(['id', 'Date', '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', 'Latitude',  
      'Longitude', 'living_area_renov', 'lot_area_renov',  
      'Number of schools nearby', 'Distance from the airport', 'Price'],  
      dtype='object')
```

df.dtypes

```
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
```

df.info()

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 14620 entries, 0 to 14619  
Data columns (total 23 columns):  
#   Column                                Non-Null Count  Dtype  
---  ---                                -  
0   id                                    14620 non-null  int64  
1   Date                                  14620 non-null  int64  
2   number of bedrooms                  14620 non-null  int64  
3   number of bathrooms                 14620 non-null  float64  
4   living area                         14620 non-null  int64  
5   lot area                            14620 non-null  int64  
6   number of floors                    14620 non-null  float64  
7   waterfront present                  14620 non-null  int64  
8   number of views                     14620 non-null  int64  
9   condition of the house              14620 non-null  int64  
10  grade of the house                  14620 non-null  int64  
11  Area of the house(excluding basement) 14620 non-null  int64  
12  Area of the basement                14620 non-null  int64  
13  Built Year                          14620 non-null  int64  
14  Renovation Year                     14620 non-null  int64  
15  Postal Code                         14620 non-null  int64  
16  Latitude                            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
```

print(df.describe())

mean	6.762821e+03	42074.000000	3.575000	2.125000
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	Latitude	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]

```
import matplotlib.pyplot as plt
plt.hist(df['Price'])
```

```
(array([1.2916e+04, 1.4260e+03, 1.9100e+02, 6.0000e+01, 1.9000e+01,  
       2.0000e+00, 2.0000e+00, 1.0000e+00, 1.0000e+00, 2.0000e+00]),  
 array([ 78000., 840200., 1602400., 2364600., 3126800., 3889000.,  
        4651700  5413400  6175600  6937800  7700000 1])
```

```
sns.countplot(df['number of floors'])
```

<Axes: ylabel='count'>



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

<Axes: ylabel='count'>

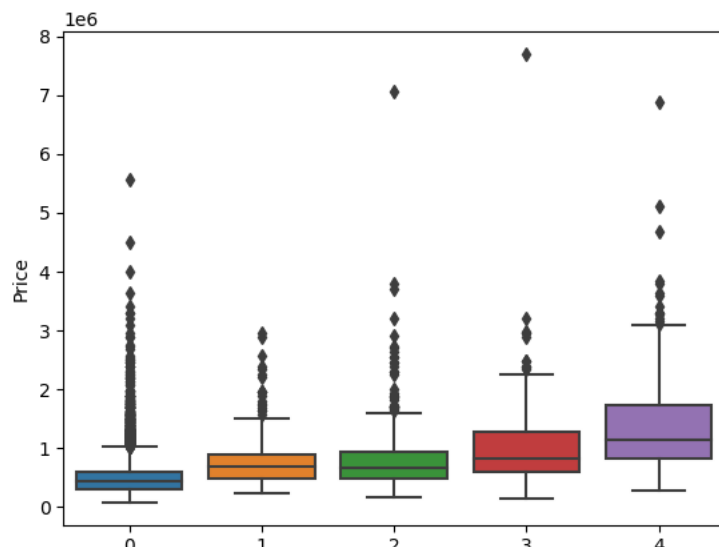


```
sns.boxplot(data=df, x="Price")
```

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

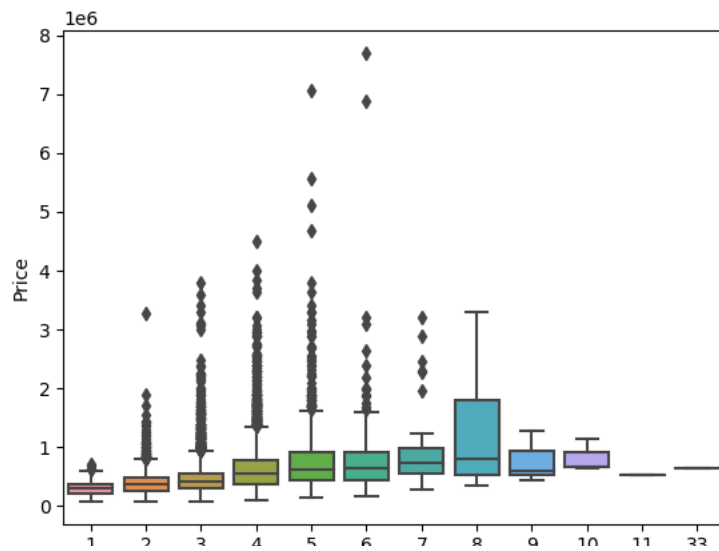
```
sns.boxplot(x=df['number of views'],y=df['Price'])
```

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



```
sns.boxplot(x=df['number of bedrooms'],y=df['Price'])
```

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

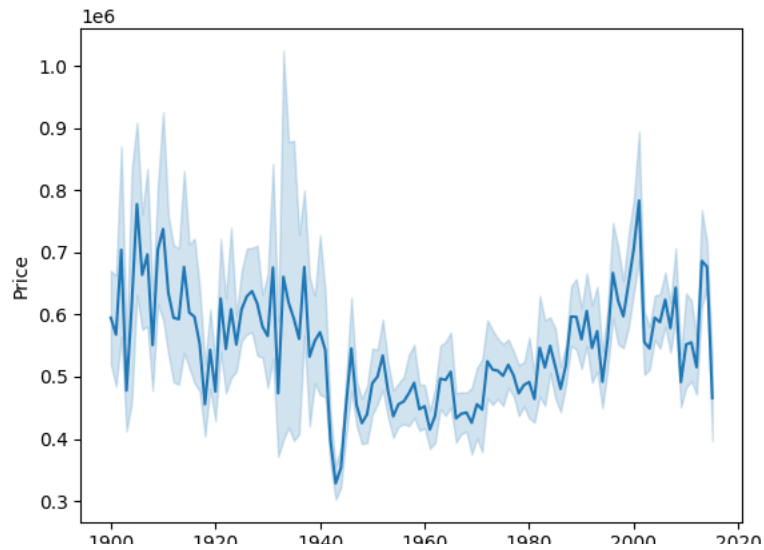


```
sns.lineplot(x=df.groupby('Built Year').mean().index, y=df.groupby('Built Year').mean()['Price'])
plt.show()
```



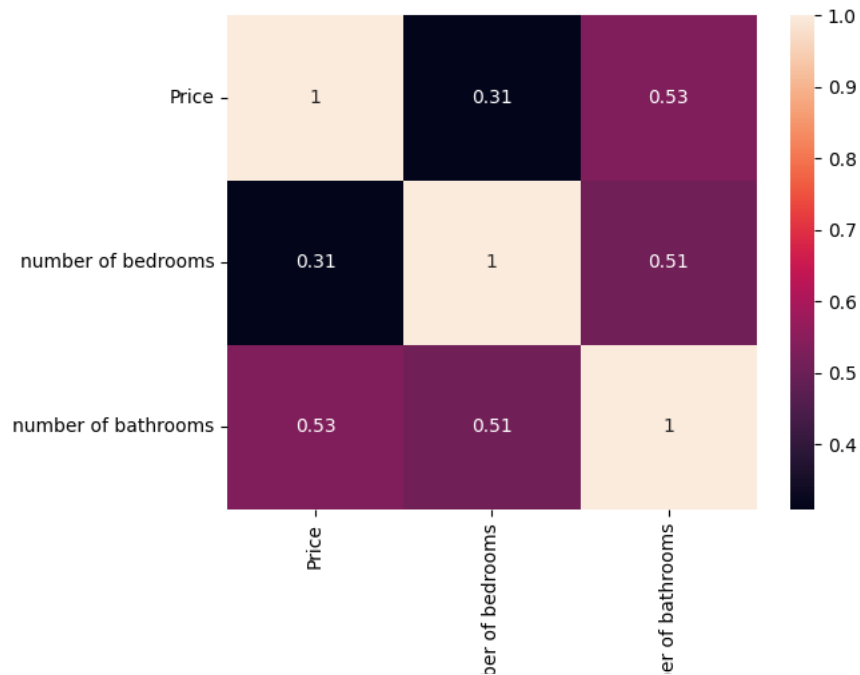
```
sns.lineplot(x=df['Built Year'],y=df['Price'])
```

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



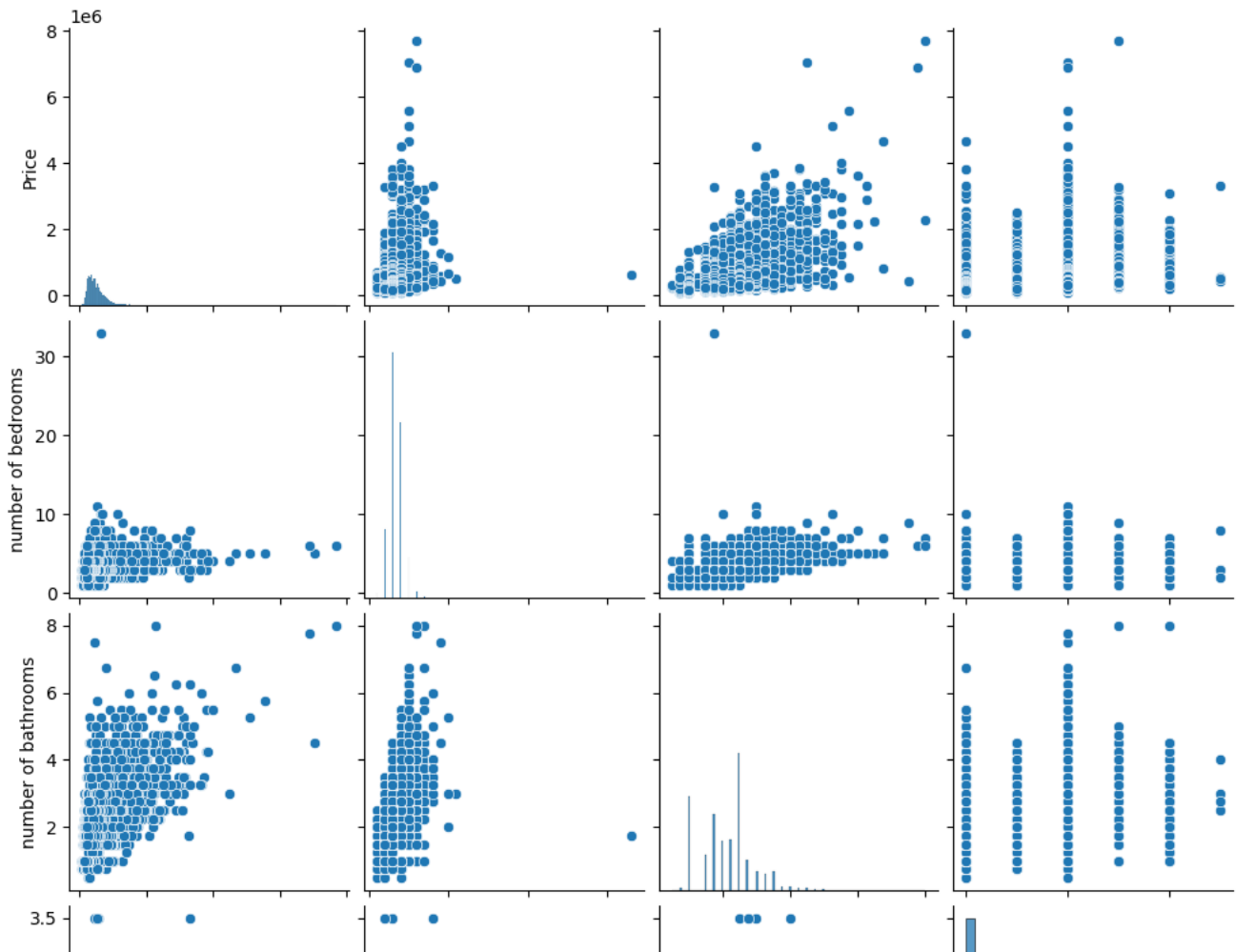
```
sns.heatmap(df[['Price', 'number of bedrooms', 'number of bathrooms']].corr(), annot=True)
```

<Axes: >



```
sns.pairplot(df[['Price', 'number of bedrooms', 'number of bathrooms', 'number of floors']])
```

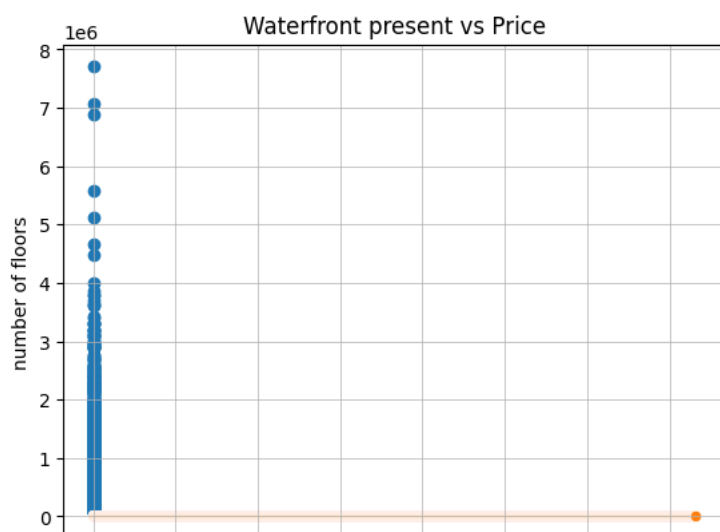
```
<seaborn.axisgrid.PairGrid at 0x7fcd63f242b0>
```



```
df.duplicated().sum()
```

```
0
```

```
plt.scatter(df['waterfront present'], df['Price'])
plt.title("Waterfront present vs Price")
plt.grid(linestyle='-', linewidth=0.7)
sns.scatterplot(df['number of floors'])
plt.grid(linestyle='-', linewidth=0.5)
```



```
plt.subplots(figsize=(15, 15))
sns.heatmap(df.drop(['id'], axis=1).corr(), linewidth=0.3, annot=True)
plt.show()
```

Date	1	-0.016	0.026	-0.022	0.0044	-0.01	0.012	0.0048	0.027	0.033	-0.016	-0.016	0.0059	0.012	0.018	-0.023	0.018	-0.032	-5e-05	0.004
number of bedrooms	-0.016	1	0.51	0.57	0.034	0.18	-0.0068	0.079	0.027	0.35	0.47	0.3	0.15	0.016	-0.044	-0.013	0.14	0.39	0.029	0.0034
number of bathrooms	-0.026	0.51	1	0.75	0.081	0.5	0.06	0.18	-0.13	0.66	0.68	0.29	0.5	0.05	-0.11	0.031	0.22	0.57	0.079	0.0022
living area	-0.022	0.57	0.75	1	0.17	0.35	0.11	0.29	-0.063	0.76	0.88	0.44	0.31	0.059	-0.08	0.055	0.24	0.76	0.18	0.0024
lot area	-0.0044	0.034	0.081	0.17	1	0.004	0.026	0.078	0.0085	0.11	0.18	0.02	0.052	0.0068	0.07	-0.091	0.22	0.15	0.71	-0.013
number of floors	-0.01	0.18	0.5	0.35	-0.004	1	0.016	0.02	-0.27	0.46	0.53	-0.24	0.48	0.0067	-0.13	0.051	0.13	0.29	-0.01	-0.007
waterfront present	0.012	0.0068	0.06	0.11	0.026	0.016	1	0.4	0.019	0.08	0.072	0.085	-0.024	0.086	0.038	-0.022	0.048	0.086	0.032	0.0016
number of views	-0.0048	0.079	0.18	0.29	0.078	0.02	0.4	1	0.053	0.25	0.16	0.29	-0.055	0.1	0.039	0.0046	-0.08	0.28	0.072	0.008
condition of the house	-0.027	0.027	-0.13	-0.063	0.0085	-0.27	0.019	0.053	1	-0.15	-0.17	0.18	-0.38	-0.062	0.045	-0.003	-0.12	-0.1	-0.004	-0.006
grade of the house	-0.033	0.35	0.66	0.76	0.11	0.46	0.08	0.25	-0.15	1	0.76	0.17	0.44	0.015	-0.15	0.12	0.2	0.72	0.12	0.0009
Area of the house(excluding basement)	-0.016	0.47	0.68	0.88	0.18	0.53	0.072	0.16	-0.17	0.76	1	-0.046	0.42	0.026	-0.084	8.8e-05	0.35	0.74	0.19	-0.0025
Area of the basement	-0.016	0.3	0.29	0.44	0.02	-0.24	0.085	0.29	0.18	0.17	-0.046	1	-0.14	0.075	-0.011	0.11	-0.15	0.2	0.011	0.01
Built Year	-0.0059	0.15	0.5	0.31	0.052	0.48	-0.024	-0.055	-0.38	0.44	0.42	-0.14	1	-0.23	-0.062	-0.14	0.41	0.33	0.073	0.0016
Renovation Year	-0.012	0.016	0.05	0.059	0.0068	0.0067	0.086	0.1	-0.062	0.015	0.026	0.075	-0.23	1	0.018	0.029	-0.08	0.0026	0.0058	0.0008
Postal Code	0.018	-0.044	-0.11	-0.08	0.07	-0.13	0.038	0.039	0.045	-0.15	-0.084	-0.011	0.062	0.018	1	-0.31	-0.099	-0.11	0.077	0.011
Latitude	-0.023	-0.013	0.031	0.055	-0.091	0.051	-0.022	0.0046	-0.003	0.12	-8.8e-05	0.11	-0.14	0.029	-0.31	1	-0.13	0.046	-0.092	0.015
Longitude	-0.018	0.14	0.22	0.24	0.22	0.13	-0.048	-0.08	-0.12	0.2	0.35	-0.15	0.41	-0.08	-0.099	-0.13	1	0.34	0.26	-0.01
living_area_renov	-0.032	0.39	0.57	0.76	0.15	0.29	0.086	0.28	-0.1	0.72	0.74	0.2	0.33	-0.0026	-0.11	0.046	0.34	1	0.19	-0.0015
lot_area_renov	-5e-05	0.029	0.079	0.18	0.71	-0.01	0.032	0.072	0.0047	0.12	0.19	0.011	0.073	0.0059	0.077	-0.092	0.26	0.19	1	-0.025

```
print(df.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
Latitude             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(df.corr())
```

```

price          -0.115906  0.297490  0.024414
living_area_renov  lot_area_renov  \
id          -0.599900  -0.089604
Date        -0.032495  -0.000050
number of bedrooms  0.389855  0.029400
number of bathrooms  0.570530  0.078627
living area    0.757571  0.180312
lot area       0.149744  0.706812
number of floors  0.285093  -0.010120
waterfront present  0.085743  0.032055
number of views   0.281452  0.072300
condition of the house -0.099743  -0.004748
grade of the house   0.720019  0.116725
Area of the house(excluding basement) 0.737744  0.194670
Area of the basement  0.196403  0.011283
Built Year           0.328625  0.072874
Renovation Year      -0.002601  0.005869
Postal Code          -0.108454  0.077483
Latitude             0.046148  -0.091622
Longitude            0.341221  0.258066
living_area_renov    1.000000  0.189225
lot_area_renov       0.189225  1.000000
Number of schools nearby -0.001203  -0.025014
Distance from the airport -0.005673  -0.014587
Price                0.584924  0.075535

Number of schools nearby  \
id          -0.004821
Date        -0.004071
number of bedrooms  0.003397
number of bathrooms  0.002180
living area    0.002370
lot area       -0.012671
number of floors  -0.007579
waterfront present  0.001563
number of views   0.008004
condition of the house -0.006939
grade of the house   0.000986
Area of the house(excluding basement) -0.002894
Area of the basement  0.010284
Built Year          -0.001631
Renovation Year      -0.000826
Postal Code         0.010605
Latitude            0.014949
Longitude           -0.010163
living_area_renov   -0.001203
lot_area_renov      -0.025014
Number of schools nearby 1.000000
Distance from the airport 0.004035
Price                0.009890

Distance from the airport  Price
id          -0.004542 -0.773114
Date        0.011457 -0.027919
number of bedrooms -0.006157  0.308460
number of bathrooms  0.009206  0.531735
living area    0.002511  0.712169
lot area       0.003291  0.081992

```

```
print(df.describe())
```

```

mean    6.762021e+03  72007.550040  0.937543  2.125505
std      6.237575e+03  67.347991  0.938719  0.769934

```

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	Latitude	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

```
-- -- - -
```

```
print(df['Number of schools nearby'].value_counts())
```

```
3    4973
2    4853
1    4794
Name: Number of schools nearby, dtype: int64
```

```
print('Mean:', df['Distance from the airport'].mean())
print('Median:', df['Area of the basement'].median())
print('Mode:', df['grade of the house'].mode())
```

```
Mean: 64.95095759233926
Median: 0.0
Mode: 0    7
Name: grade of the house, dtype: int64
```

```
df.dropna(inplace=True)
df.fillna(0, inplace=True)
df.interpolate(inplace=True)
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
x=df.drop(['Price','Date'],axis=1)
x.set_index(['id'],inplace=True)
y=df[['id','Price']]
x.head()
```

	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	Renov
id													
6762810145	5	2.50	3650	9050	2.0	0	4	5	10	3370	280	1921	
6762810635	4	2.50	2920	4000	1.5	0	0	5	8	1910	1010	1909	

```
print(df.isnull().sum())
```

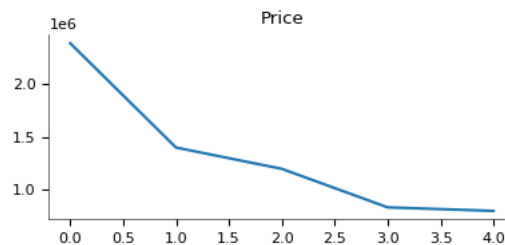
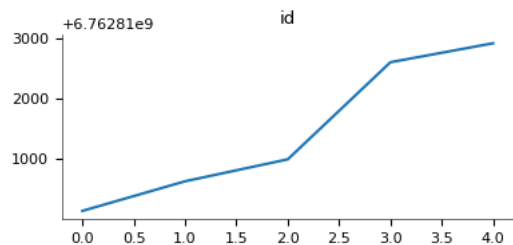
id	0
Date	0
number of bedrooms	0
number of bathrooms	0
living area	0
lot area	0
number of floors	0
waterfront present	0
number of views	0
condition of the house	0
grade of the house	0
Area of the house(excluding basement)	0
Area of the basement	0
Built Year	0
Renovation Year	0
Postal Code	0
Latitude	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	

```
sc=StandardScaler()
sc=MinMaxScaler()
x=pd.DataFrame(sc.fit_transform(x),columns=x.columns.values)
x.head()
```

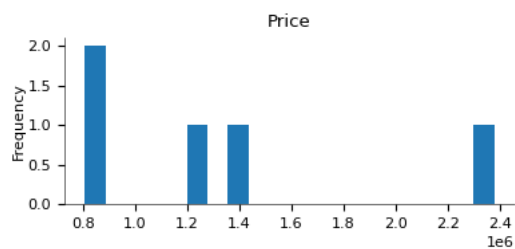
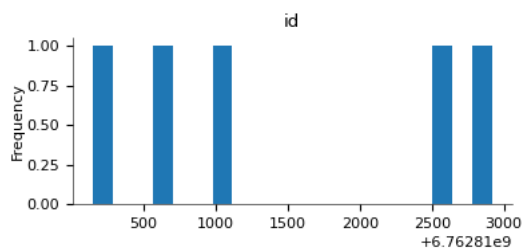
	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condition of the house	grade of the house	Area of the house(excluding basement)	Area of the basement	Built Year	Renov
0	0.12500	0.266667	0.249051	0.007945	0.4	0.0	1.0	1.00	0.666667	0.331858	0.058091	0.182609	
1	0.09375	0.266667	0.193622	0.003241	0.2	0.0	0.0	1.00	0.444444	0.170354	0.209544	0.078261	
2	0.12500	0.300000	0.192863	0.008345	0.2	0.0	0.0	0.50	0.444444	0.280973	0.000000	0.339130	
3	0.09375	0.266667	0.223235	0.039562	0.4	0.0	0.0	0.50	0.555556	0.325221	0.000000	0.878261	
4	0.06250	0.200000	0.177677	0.003707	0.2	0.0	0.0	0.75	0.444444	0.167035	0.172100	0.252174	

```
y.head()
```

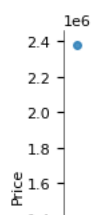
## Values



## Distributions



## 2-d distributions



```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import r2_score
x_train,x_test,y_train,y_test = train_test_split(x , y['Price'],test_size =0.1,random_state=2)
model = GradientBoostingRegressor(n_estimators= 400,max_depth=5,min_samples_split=2,learning_rate=0.1)
model.fit(x_train,y_train)
```

```
y_pred = model.predict(x_test)
model.score(x_test,y_test)
```

```
0.9111918507210961
```

```
r2_score(y_pred,y_test)
```

```
0.900323276786278
```

```
y_pred
```

```
array([[497766.12740438, 244495.3776842 , 293819.40063242, ...],
       [698495.60350629, 297006.00386358, 245881.76921871]])
```

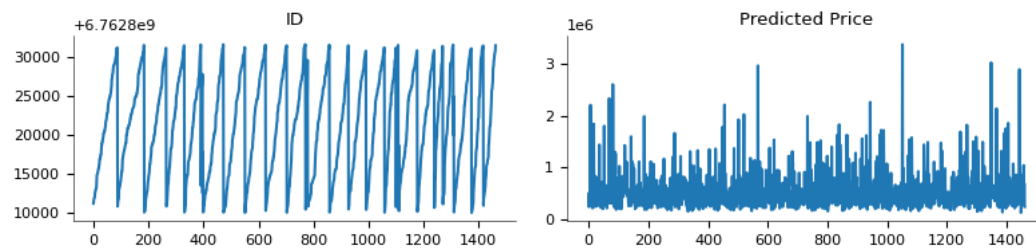
```
y_pred_list = y['id'][-len(y_pred):].tolist()
y_pred_df=pd.DataFrame(y_pred_list,columns=['ID'])
y_pred_df['Predicted Price'] = y_pred.round(2)
```

```
y_pred_df
```

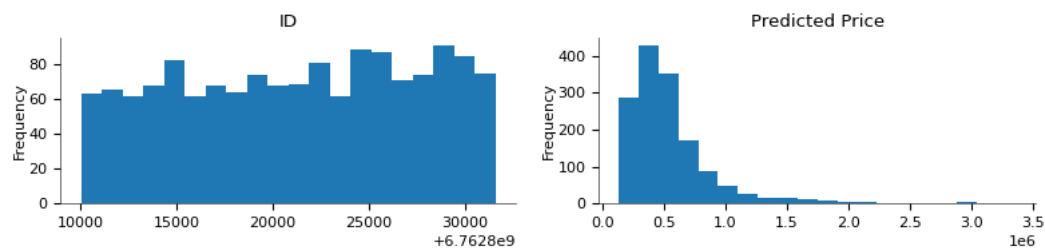
```
...      ...      ...  
1457 6762830250      1041014.57  
1458 6762830339      317512.59  
1459 6762830618      698495.60  
1460 6762830709      297006.00  
1461 6762831463      245881.77
```

1462 rows × 2 columns

### Values



### Distributions



### 2-d distributions

