```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
from google.colab import files
import pandas as pd

uploaded = files.upload()

for fn in uploaded.keys():
  print('User uploaded file "{name}" with length {length}
bytes'.format(name=fn, length=len(uploaded[fn])))

data = pd.read_csv(next(iter(uploaded)))
```

```
<IPython.core.display.HTML object>
```

```
Saving House Price India.csv to House Price India.csv
User uploaded file "House Price India.csv" with length 1524561 bytes
```

perform the Univariate Analysis

```
data.dtypes
```

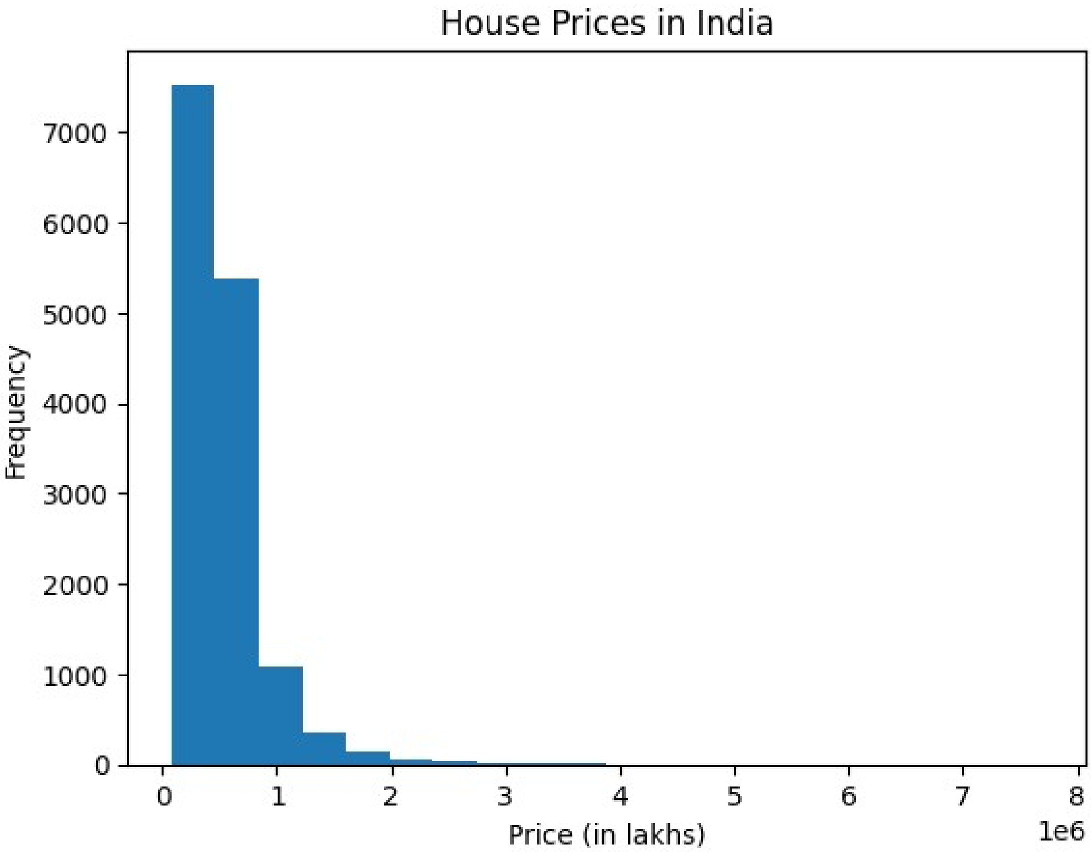
```
id                                         int64
Date                                       int64
number of bedrooms                         int64
number of bathrooms                      float64
living area                                int64
lot area                                   int64
number 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
Lattitude                                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
```

```
data.describe()
```

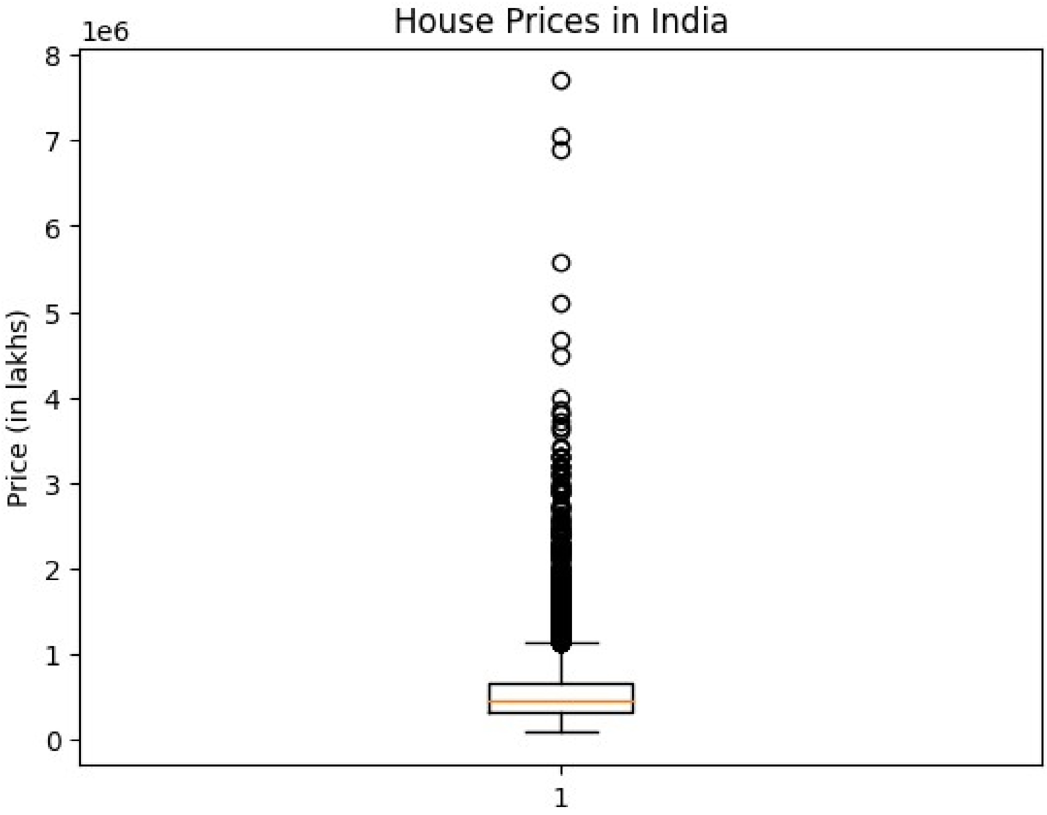
```
                 id         Date  number of bedrooms  number of
bathrooms  \
count  1.462000e+04  14620.000000        14620.000000
14620.000000
mean   6.762821e+09  42604.538646            3.379343
2.129583
std    6.237575e+03     67.347991            0.938719
0.769934
min    6.762810e+09  42491.000000            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]
```

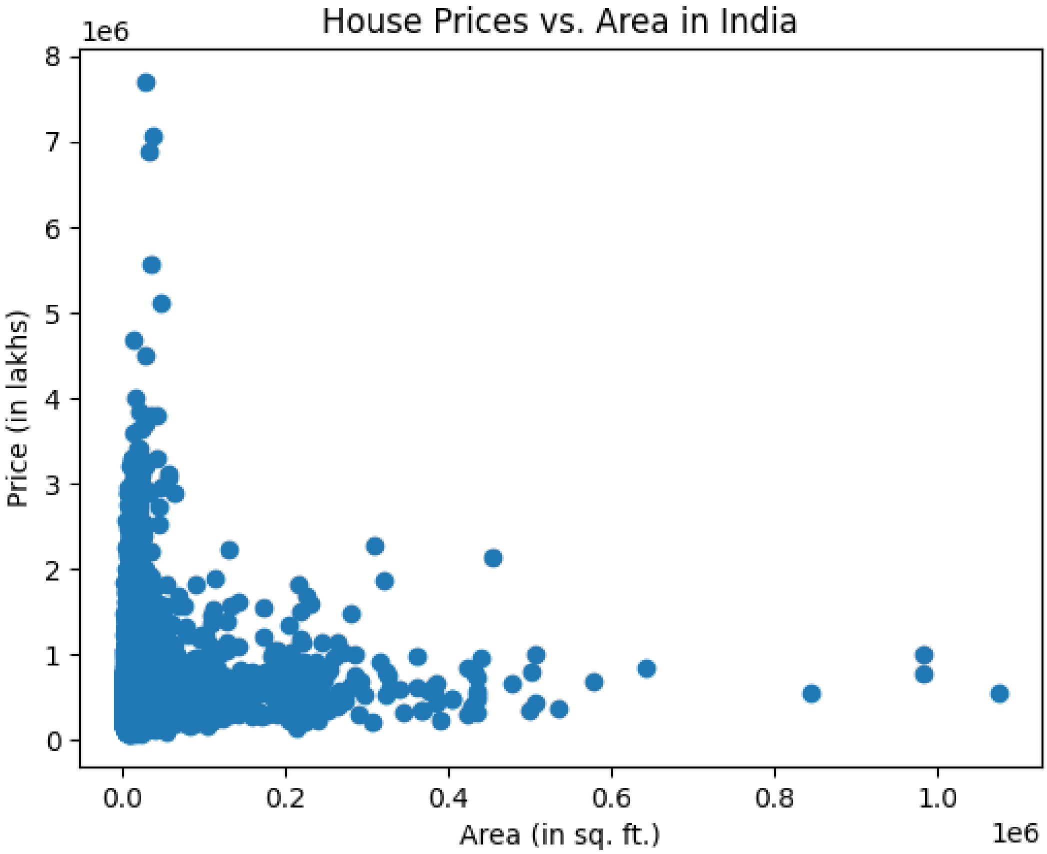
```
plt.hist(data['Price'], bins=20)
plt.title('House Prices in India')
plt.xlabel('Price (in lakhs)')
plt.ylabel('Frequency')
plt.show()
```



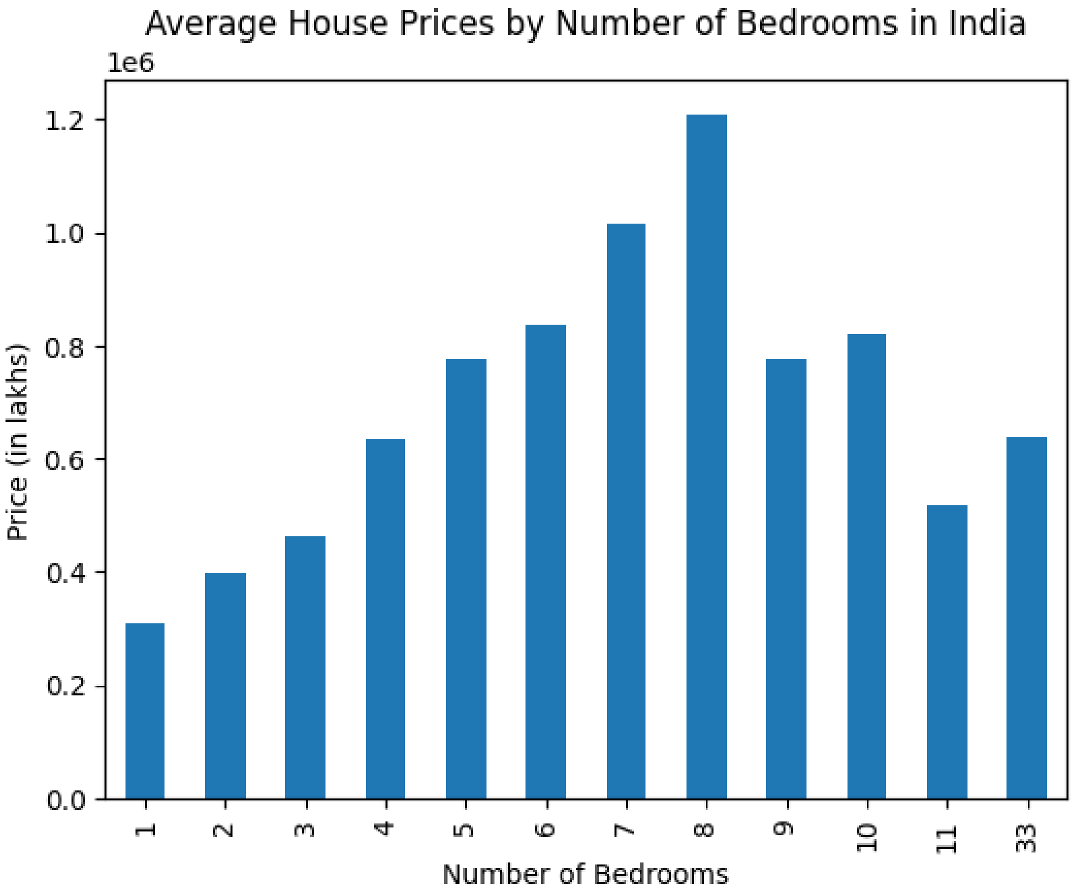
```
plt.boxplot(data['Price'])
plt.title('House Prices in India')
plt.ylabel('Price (in lakhs)')
plt.show()
```



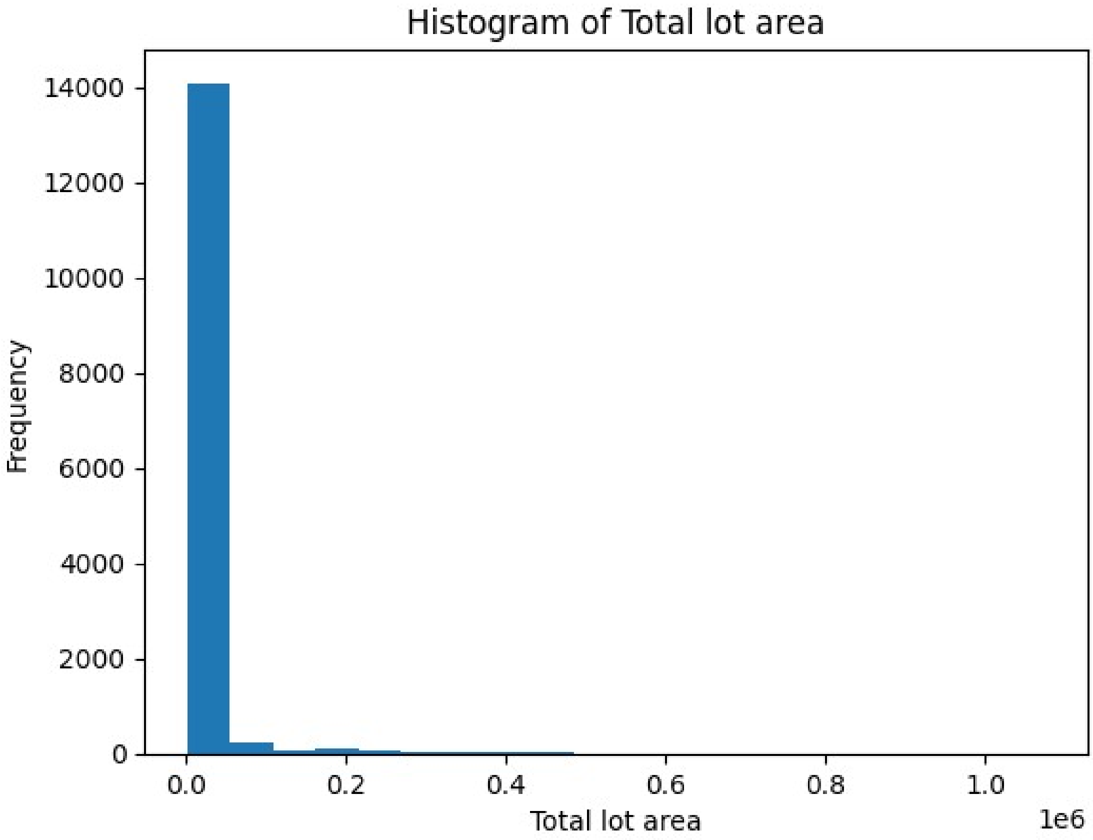
```
plt.scatter(data['lot area'], data['Price'])
plt.title('House Prices vs. Area in India')
plt.xlabel('Area (in sq. ft.)')
plt.ylabel('Price (in lakhs)')
plt.show()
```



```
data.groupby('number of bedrooms')['Price'].mean().plot(kind='bar')
plt.title('Average House Prices by Number of Bedrooms in India')
plt.xlabel('Number of Bedrooms')
plt.ylabel('Price (in lakhs)')
plt.show()
```

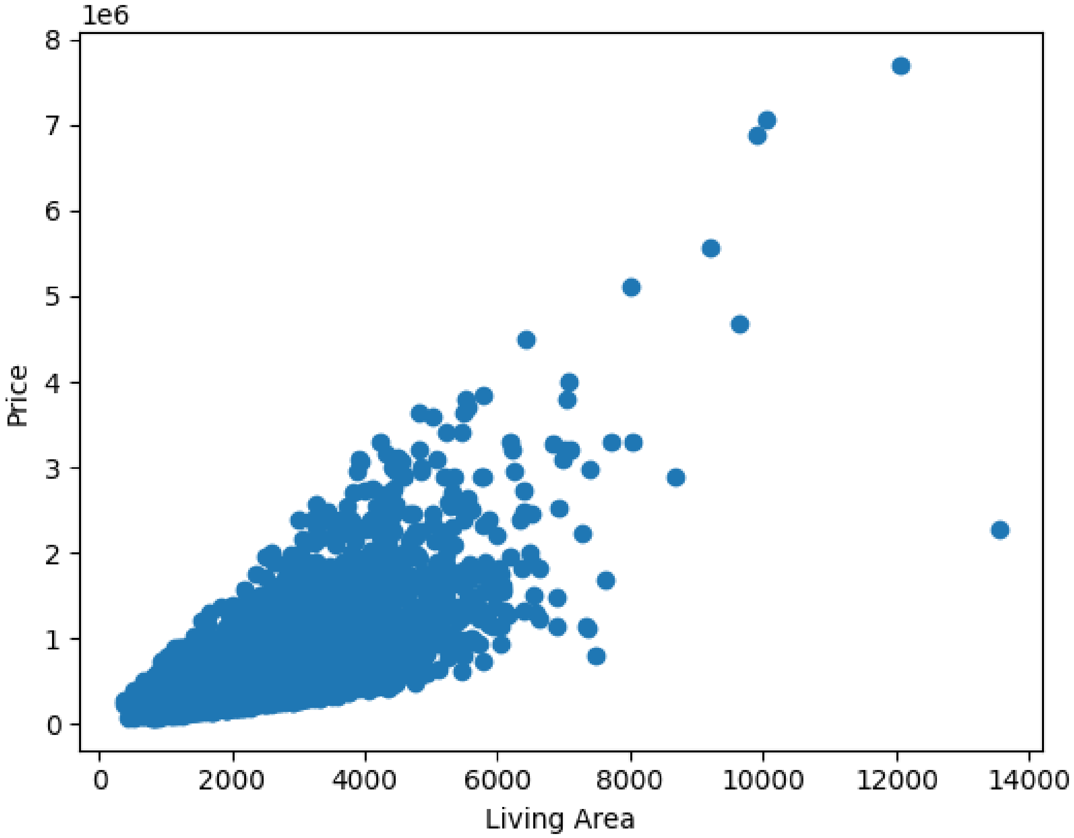


```
plt.hist(data['lot area'], bins=20)
plt.xlabel('Total lot area')
plt.ylabel('Frequency')
plt.title('Histogram of Total lot area')
plt.show()
```

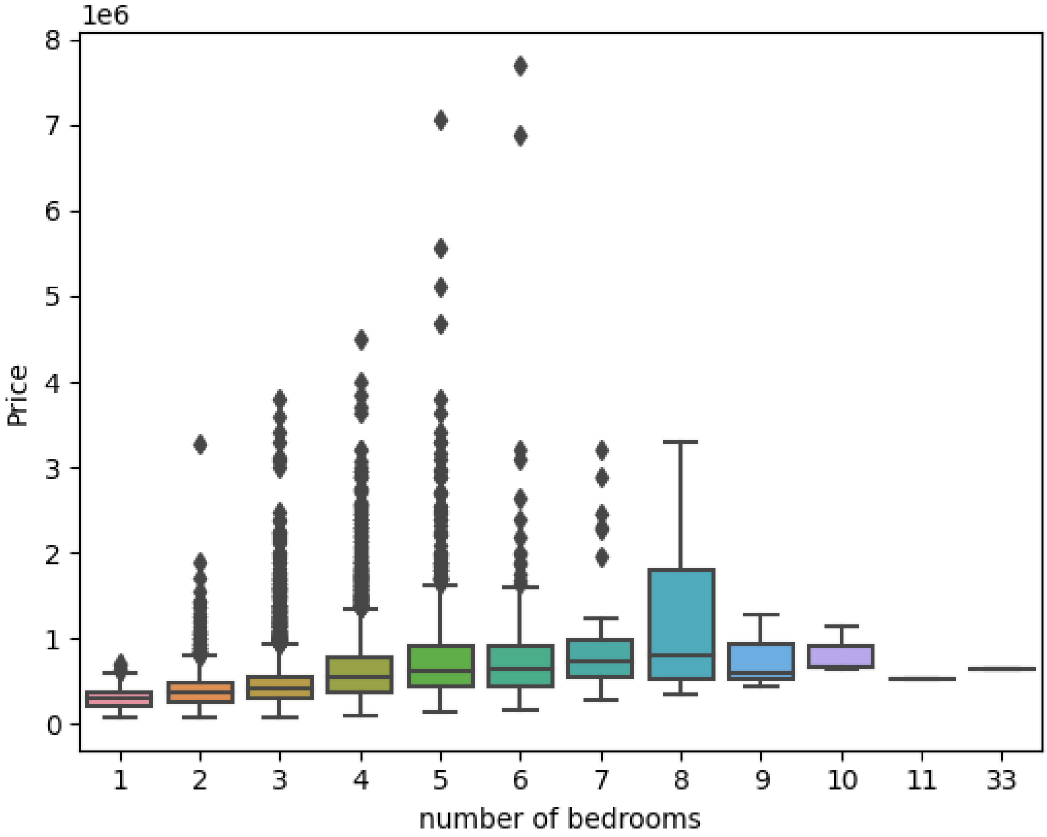


Bi-Variate Analysis

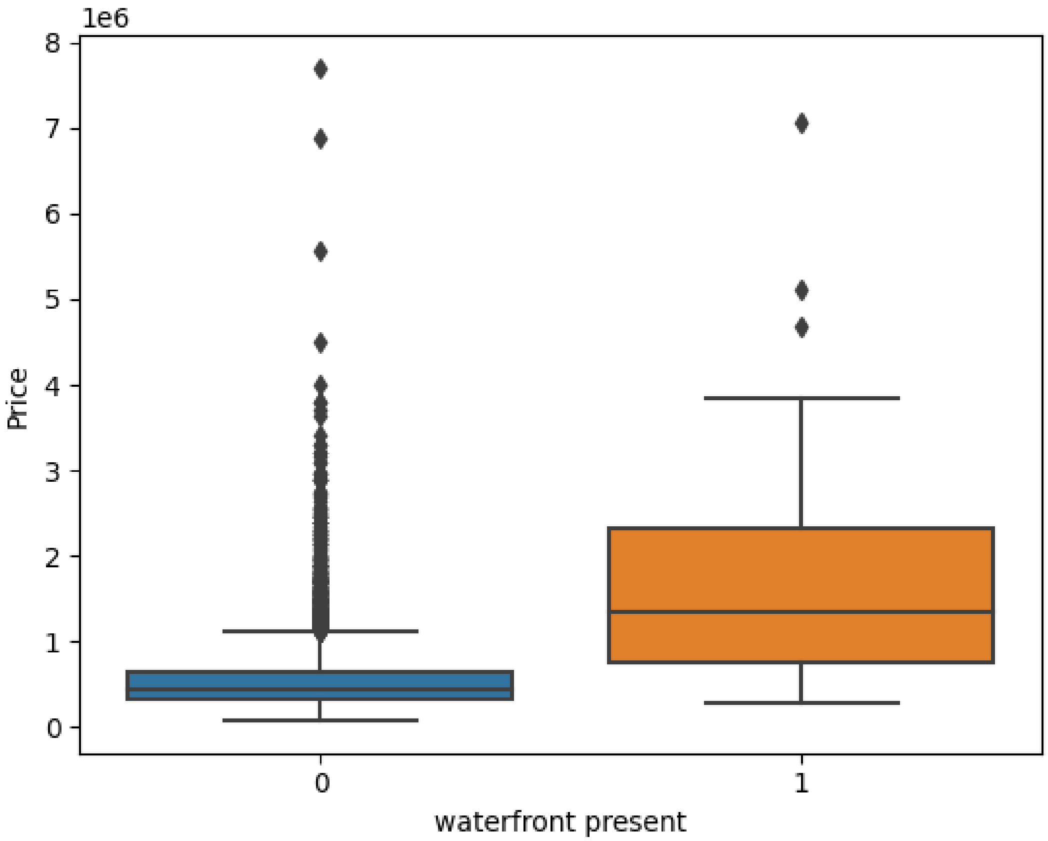
```
plt.scatter(data['living area'], data['Price'])
plt.xlabel('Living Area')
plt.ylabel('Price')
plt.show()
```



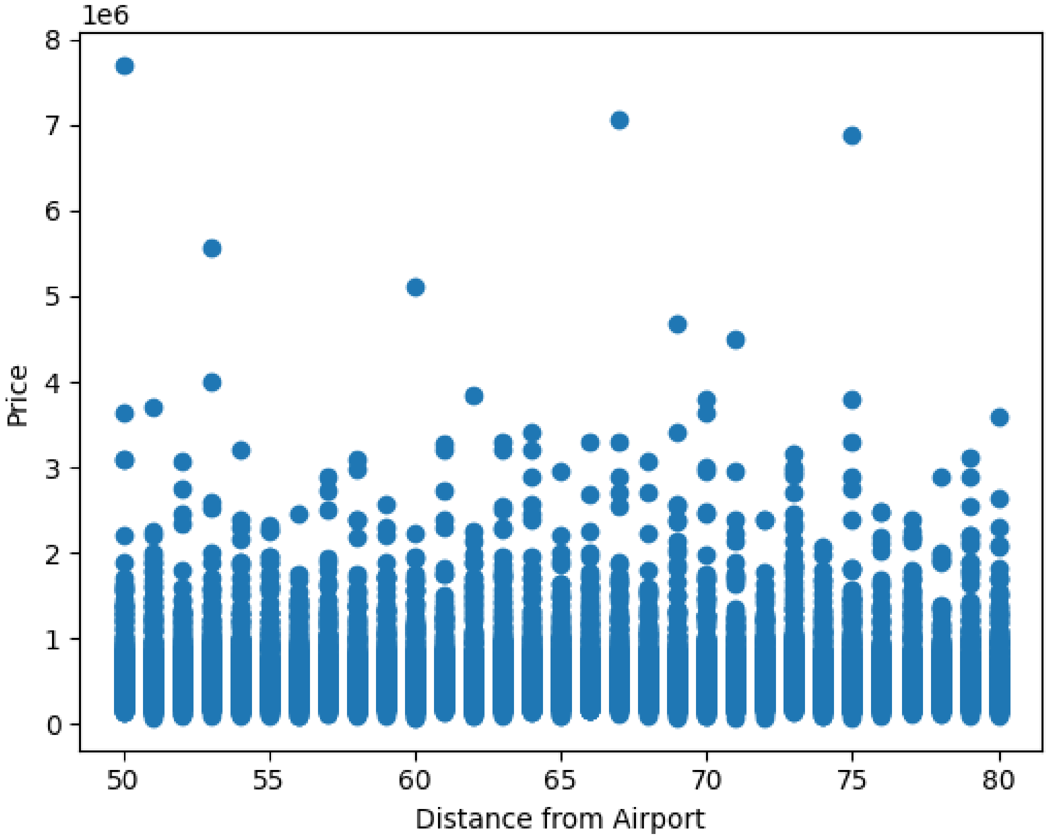
```
import seaborn as sns

sns.boxplot(x='number of bedrooms', y='Price', data=data)
plt.show()
```



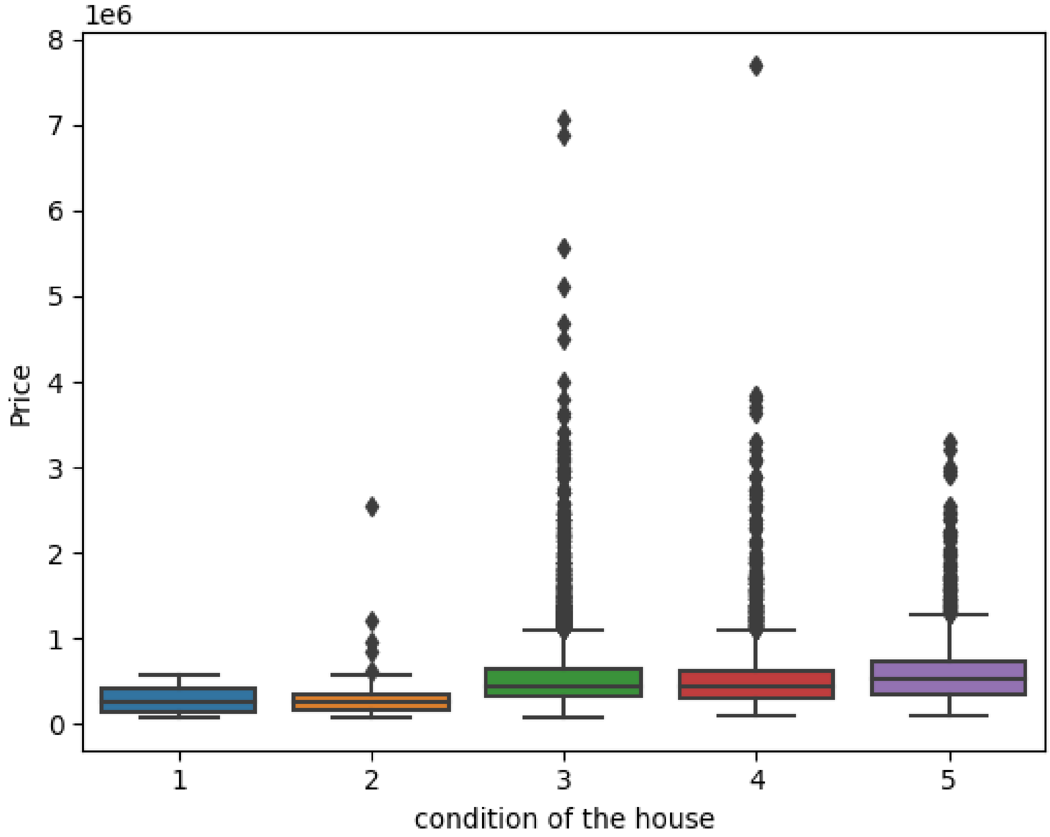
```
sns.boxplot(x='waterfront present', y='Price', data=data)
plt.show()
```



```
plt.scatter(data['Distance from the airport'], data['Price'])
plt.xlabel('Distance from Airport')
plt.ylabel('Price')
plt.show()
```



```
sns.boxplot(x='condition of the house', y='Price', data=data)
plt.show()
```

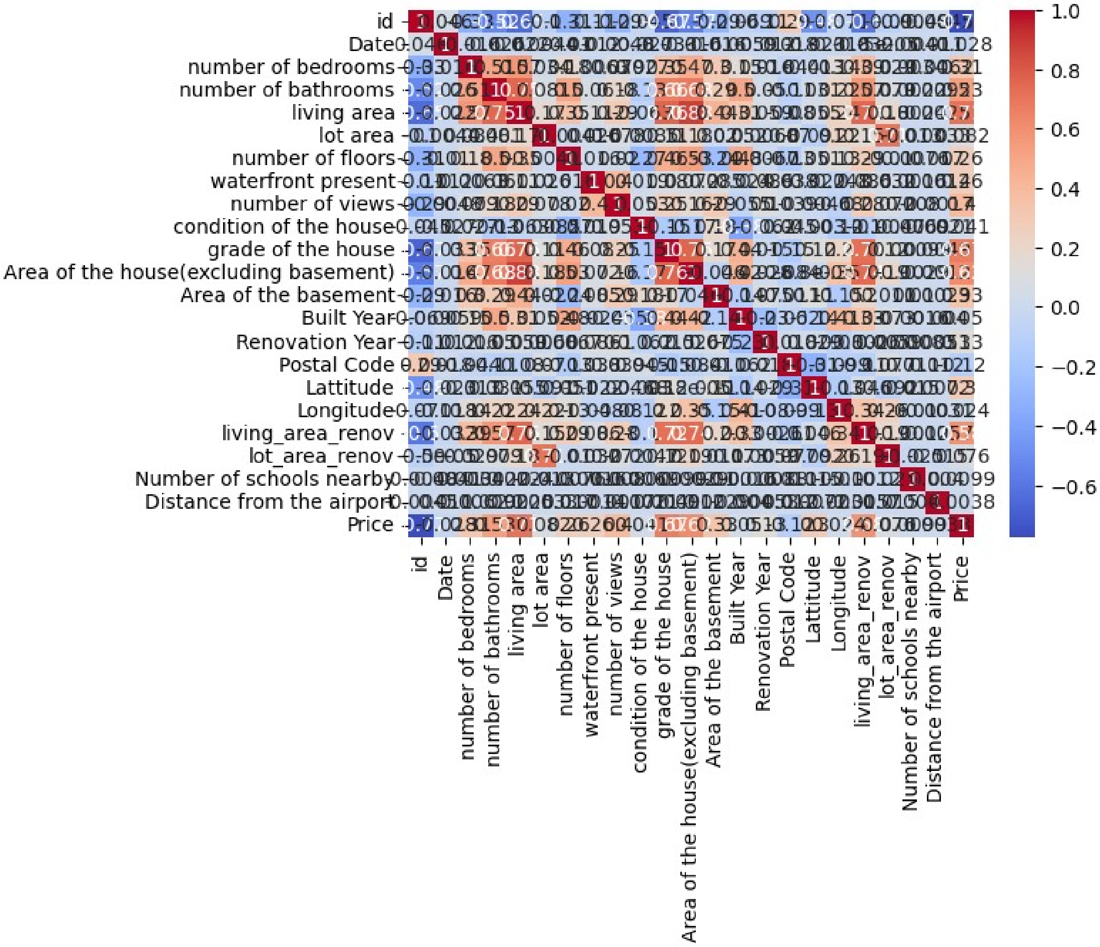


Multi-Variate Ananlsis

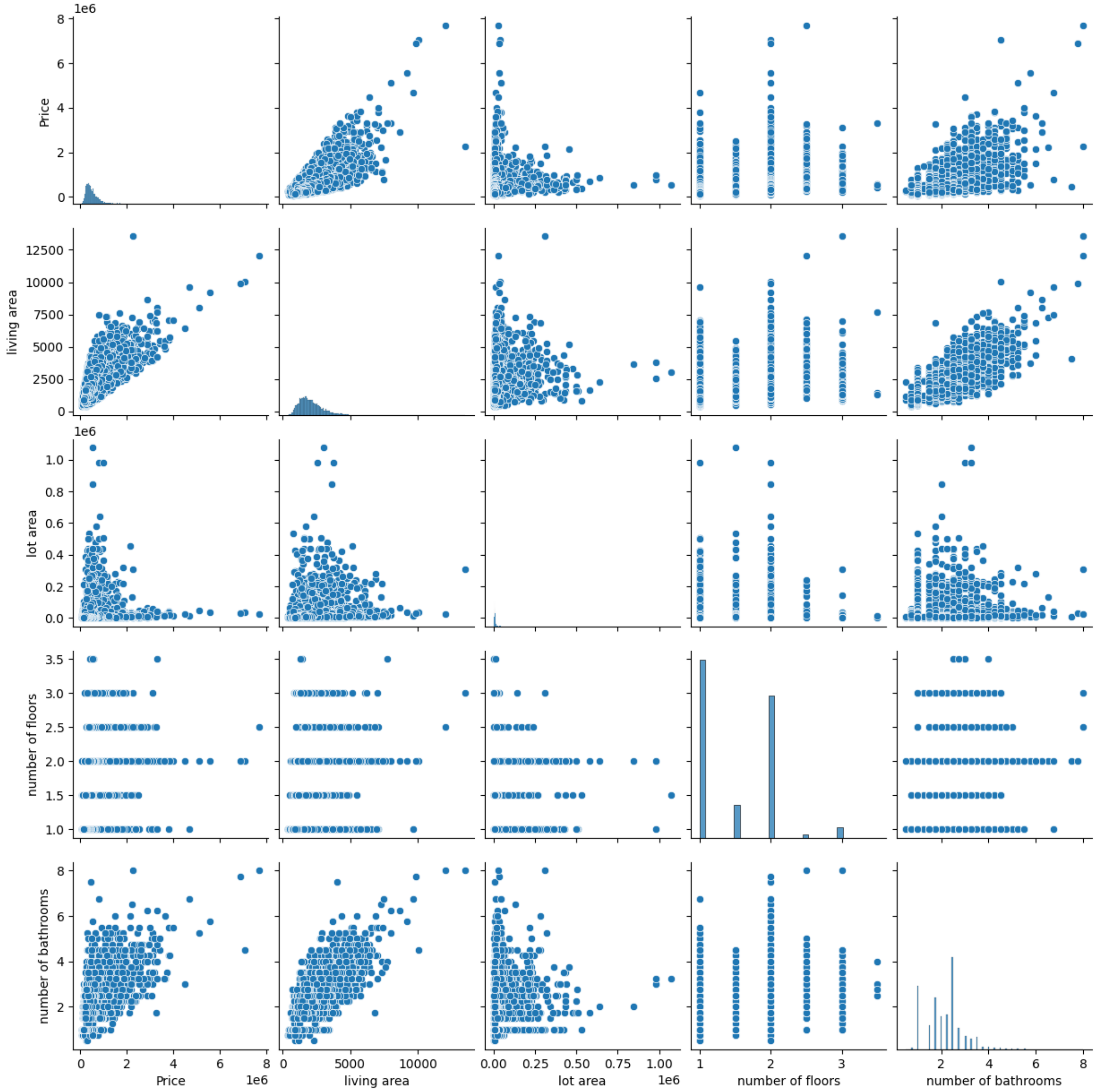
```
print(data.isnull().sum())
```

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

```
Number of schools nearby                 0
Distance from the airport                0
Price                                    0
dtype: int64

# Check the correlation matrix
corr_matrix = data.corr()
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm")
plt.show()
```



```
# Create a pairplot to visualize the relationship between each pair of variables
sns.pairplot(data, vars=['Price', 'living area', 'lot area', 'number of floors', 'number of bathrooms'])
plt.show()
```



```
df = pd.get_dummies(data, columns=['waterfront present'],
drop_first=True)

# Fit a multiple linear regression model to predict price based on all
other variables
from sklearn.linear_model import LinearRegression
model = LinearRegression()
X = data.drop(['id', 'Date', 'Price'], axis=1)
y = data['Price']
model.fit(X, y)
print('Intercept:', model.intercept_)
print('Coefficients:', model.coef_)
```

```
Intercept: -70412827.25487897
Coefficients: [-3.38012415e+04  4.09965286e+04  1.13030194e+02
2.75590250e-03
  1.39276481e+03  5.80094627e+05  4.90632154e+04  3.25732896e+04
  9.81475925e+04  7.60466665e+01  3.69835276e+01 -2.45282471e+03
  2.40176230e+01  2.77734783e+02  5.53258540e+05 -9.98005641e+04
  1.63455328e+01 -3.72467553e-01  1.74973054e+03 -1.23159493e+02]
```

```
# Evaluate the model
from sklearn.metrics import r2_score
y_pred = model.predict(X)
print('R2 score:', r2_score(y, y_pred))
```

```
R2 score: 0.7026950180179408
```

Perform descriptive statistics on the dataset

```
# Select the relevant columns
columns = ['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 basement', 'Built Year', 'Renovation Year',
'Postal Code', 'Lattitude',
           'Longitude', 'living_area_renov', 'lot_area_renov', 'Number
of schools nearby',
           'Distance from the airport', 'Price']
df = data[columns]

# Print descriptive statistics
print(df.describe())
```

```
       number of bedrooms  number of bathrooms   living area      lot
area  \
count        14620.000000         14620.000000  14620.000000
1.462000e+04
mean             3.379343             2.129583   2098.262996
1.509328e+04
std              0.938719             0.769934    928.275721
3.791962e+04
min              1.000000             0.500000    370.000000
5.200000e+02
25%              3.000000             1.750000   1440.000000
5.010750e+03
50%              3.000000             2.250000   1930.000000
7.620000e+03
75%              4.000000             2.500000   2570.000000
1.080000e+04
max             33.000000             8.000000  13540.000000
1.074218e+06

       number of floors  waterfront present  number of views  \
count      14620.000000        14620.000000     14620.000000
mean           1.502360            0.007661         0.233105
std            0.540239            0.087193         0.766259
```

```
min           1.000000          0.000000        0.000000
25%           1.000000          0.000000        0.000000
50%           1.500000          0.000000        0.000000
75%           2.000000          0.000000        0.000000
max           3.500000          1.000000        4.000000

       condition of the house  grade of the house  Area of the
basement  \
count            14620.000000        14620.000000
14620.000000
mean                 3.430506            7.682421
296.479070
std                  0.664151            1.175033
448.551409
min                  1.000000            4.000000
0.000000
25%                  3.000000            7.000000
0.000000
50%                  3.000000            7.000000
0.000000
75%                  4.000000            8.000000
580.000000
max                  5.000000           13.000000
4820.000000

         Built Year  Renovation Year    Postal Code     Lattitude  \
count  14620.000000     14620.000000   14620.000000  14620.000000
mean    1970.926402        90.924008  122033.062244     52.792848
std       29.493625       416.216661      19.082418      0.137522
min     1900.000000         0.000000  122003.000000     52.385900
25%     1951.000000         0.000000  122017.000000     52.707600
50%     1975.000000         0.000000  122032.000000     52.806400
75%     1997.000000         0.000000  122048.000000     52.908900
max     2015.000000      2015.000000  122072.000000     53.007600

          Longitude  living_area_renov  lot_area_renov  \
count  14620.000000       14620.000000    14620.000000
mean    -114.404007        1996.702257    12753.500068
std        0.141326         691.093366    26058.414467
min     -114.709000         460.000000      651.000000
25%     -114.519000        1490.000000     5097.750000
50%     -114.421000        1850.000000     7620.000000
75%     -114.315000        2380.000000    10125.000000
max     -113.505000        6110.000000   560617.000000

       Number of schools nearby  Distance from the airport
Price
count              14620.000000               14620.000000
1.462000e+04
mean                   2.012244                  64.950958
```

```
5.389322e+05
std                    0.817284                   8.936008
3.675324e+05
min                    1.000000                  50.000000
7.800000e+04
25%                    1.000000                  57.000000
3.200000e+05
50%                    2.000000                  65.000000
4.500000e+05
75%                    3.000000                  73.000000
6.450000e+05
max                    3.000000                  80.000000
7.700000e+06
```

Handle the Missing Values

```
# Check for missing values
print(df.isnull().sum())
```

```
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 basement         0
Built Year                   0
Renovation Year              0
Postal Code                  0
Lattitude                    0
Longitude                    0
living_area_renov            0
lot_area_renov               0
Number of schools nearby     0
Distance from the airport    0
Price                        0
dtype: int64
```

```
# Drop rows with missing values
df = df.dropna()
```

```
# Fill missing values with mean or median
data['Area of the basement'] = data['Area of the
basement'].fillna(data['Area of the basement'].median())
data['Renovation Year'] = data['Renovation
Year'].fillna(data['Renovation Year'].mean())
```

```
# Replace missing values with a constant
data['waterfront present'] = data['waterfront
present'].fillna('Unknown')

# Check for missing values after handling
print(df.isnull().sum())
```

```
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 basement         0
Built Year                   0
Renovation Year              0
Postal Code                  0
Lattitude                    0
Longitude                    0
living_area_renov            0
lot_area_renov               0
Number of schools nearby     0
Distance from the airport    0
Price                        0
dtype: int64
```