assignment-2-Sai Bhaskar k

September 6, 2023

0.0.1 Task-1: Downloading the dataset

0.0.2 Task-2: Loading the dataset

[ ]:

```
import pandas as pd

dataset = pd.read_csv('/content/House Price India.csv')

print("House Price India Dataset is loaded successfully.")
```

```
House Price India Dataset is loaded successfully.
```

0.0.3 Task-3: Perform the Below Visualizations for that dataset

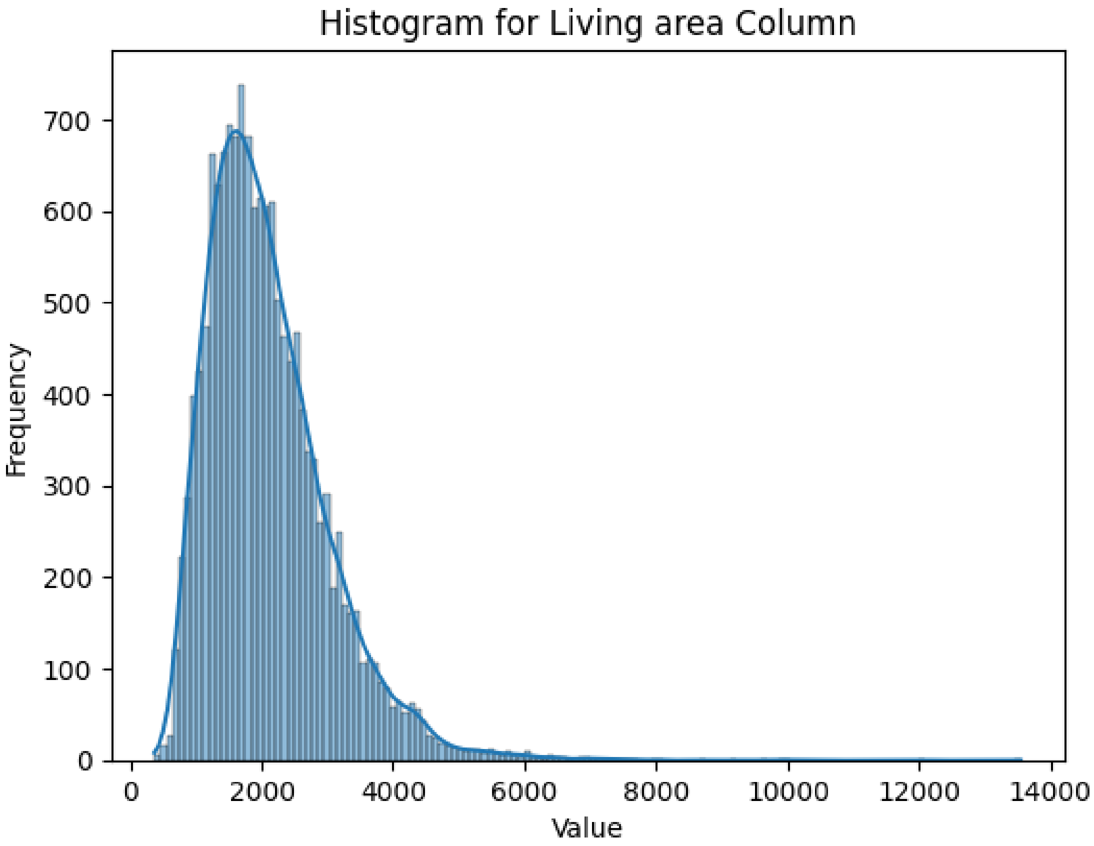
1. Univariate Analysis
2. Bi - Variate Analysis
3. Multivariate Analysis

1. **Univariate Analysis:**

[ ]:

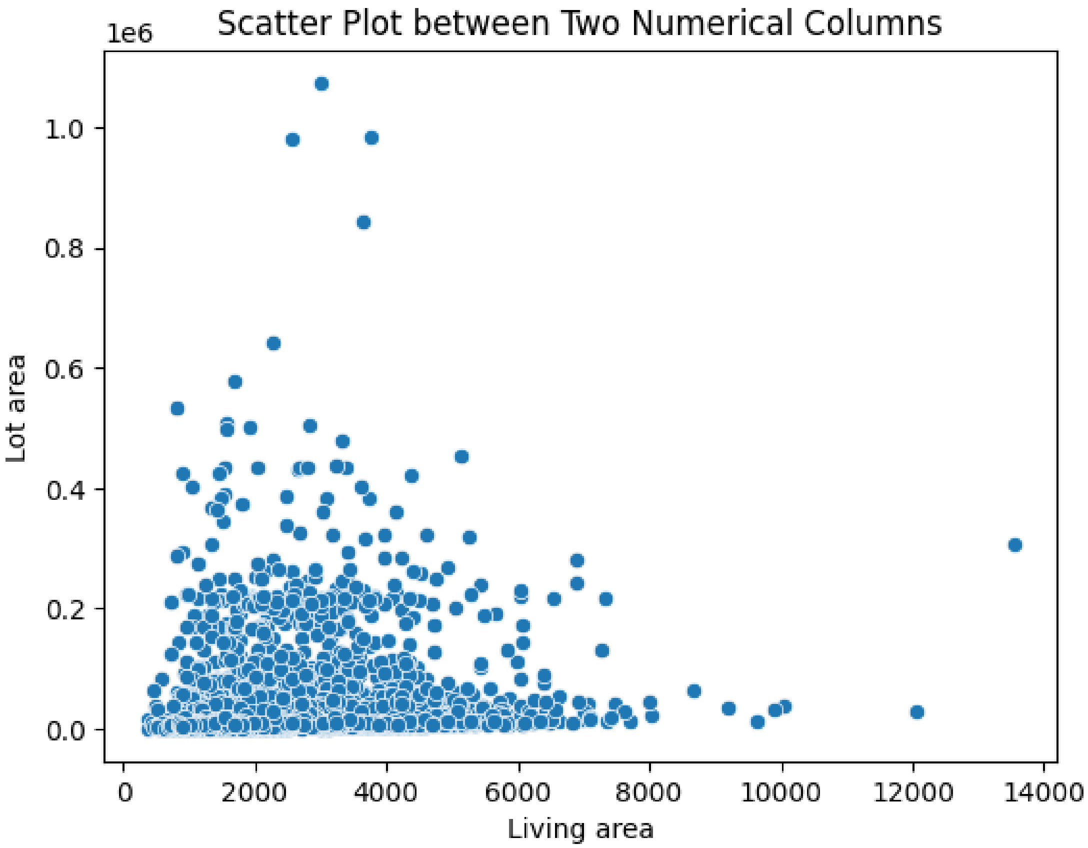
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.histplot(dataset['living area'], kde=True)
plt.title('Histogram for Living area Column')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.show()
```



2. Bi-Variate Analysis:

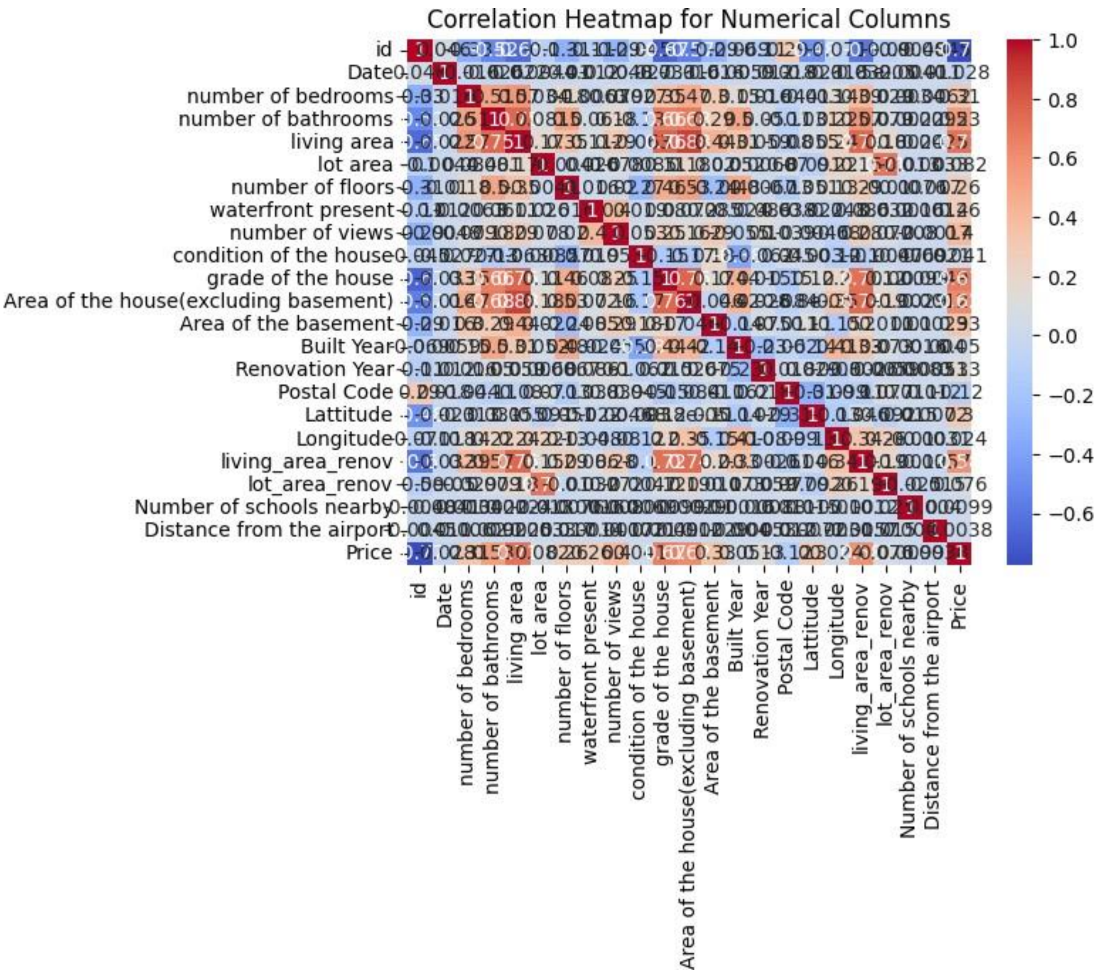
[ ]:
```
sns.scatterplot(data=dataset, x='living area', y='lot area')
plt.title('Scatter Plot between Two Numerical Columns')
plt.xlabel('Living area')
plt.ylabel('Lot area')
plt.show()
```



3. Multi-Variate Analysis

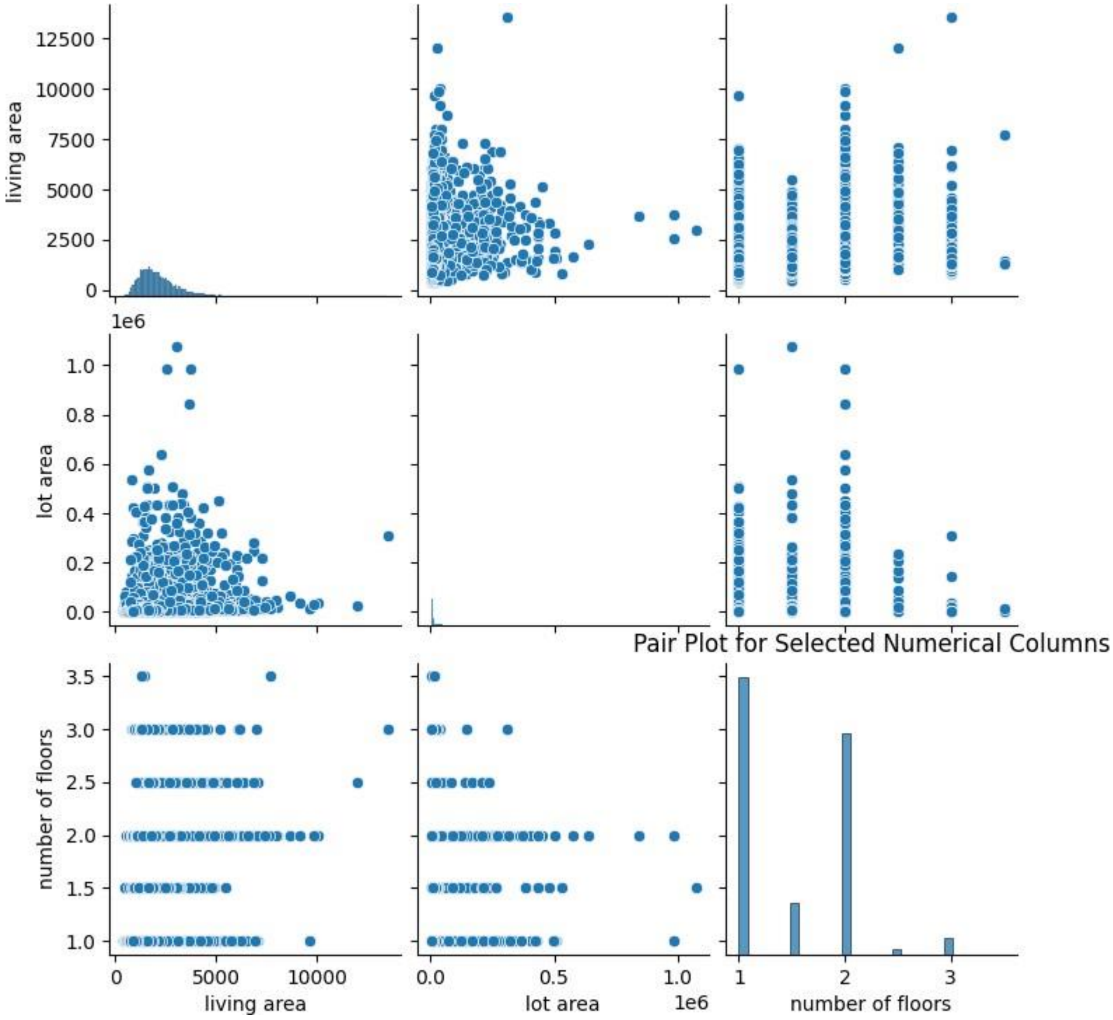
[ ]:

```
correlation_matrix = dataset.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap for Numerical Columns')
plt.show()
```



Multi-variate Analysis using Pairplot:

```
[ ]: sns.pairplot(dataset[['living area', 'lot area', 'number of floors']])
     plt.title('Pair Plot for Selected Numerical Columns')
     plt.show()
```



Pair Plot for Selected Numerical Columns

0.0.4 Task-4: Perform descriptive statistics on the dataset

```
[ ]: numerical_stats = dataset.describe()

     print("Descriptive Statistics for Numerical Columns:")
     print(numerical_stats)
```

```
Descriptive Statistics for Numerical Columns:
                 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]
```

0.0.5 Task-5: Handling the missing values:

[ ]:
```python
missing_values = dataset.isna().sum()
print("Missing Values per Column:")
print(missing_values)

# Remove rows with missing values
dataset_cleaned = dataset.dropna()

# Impute missing values for numerical columns with mean
numerical_columns = dataset.select_dtypes(include='number')
for column in numerical_columns:
    dataset[column].fillna(dataset[column].mean(), inplace=True)

# Impute missing values for categorical columns with mode
categorical_columns = dataset.select_dtypes(include='object')
for column in categorical_columns:
    dataset[column].fillna(dataset[column].mode()[0], inplace=True)

# Mark missing values in a categorical column with 'Missing'
dataset['living area'].fillna('Missing', inplace=True)

# Drop columns with too many missing values
threshold = len(dataset) * 0.2
dataset.dropna(axis=1, thresh=threshold, inplace=True)

# Interpolate missing values for a numerical column
dataset['living area'].interpolate(method='linear', inplace=True)

# Display the cleaned dataset
print("Cleaned Dataset:")
print(dataset.head())
```

```
Missing Values per Column:
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
Cleaned Dataset:
           id   Date  number of bedrooms  number of bathrooms  living area  \
0  6762810145  42491                   5                 2.50         3650   
1  6762810635  42491                   4                 2.50         2920   
2  6762810998  42491                   5                 2.75         2910   
3  6762812605  42491                   4                 2.50         3310   
4  6762812919  42491                   3                 2.00         2710   

   lot area  number of floors  waterfront present  number of views  \
0      9050               2.0                   0                4   
1      4000               1.5                   0                0   
2      9480               1.5                   0                0   
3     42998               2.0                   0                0   
4      4500               1.5                   0                0   

   condition of the house  ...  Built Year  Renovation Year  Postal Code  \
0                       5  ...        1921                0       122003   
1                       5  ...        1909                0       122004   
2                       3  ...        1939                0       122004   
3                       3  ...        2001                0       122005   
4                       4  ...        1929                0       122006   

   Lattitude  Longitude  living_area_renov  lot_area_renov  \
0    52.8645   -114.557               2880            5400   
1    52.8878   -114.470               2470            4000   
2    52.8852   -114.468               2940            6600   
3    52.9532   -114.321               3350           42847   
4    52.9047   -114.485               2060            4500   

   Number of schools nearby  Distance from the airport    Price  
0                         2                         58  2380000  
```

```
1                          2                          51  1400000
2                          1                          53  1200000
3                          3                          76   838000
4                          1                          51   805000

[5 rows x 23 columns]
```