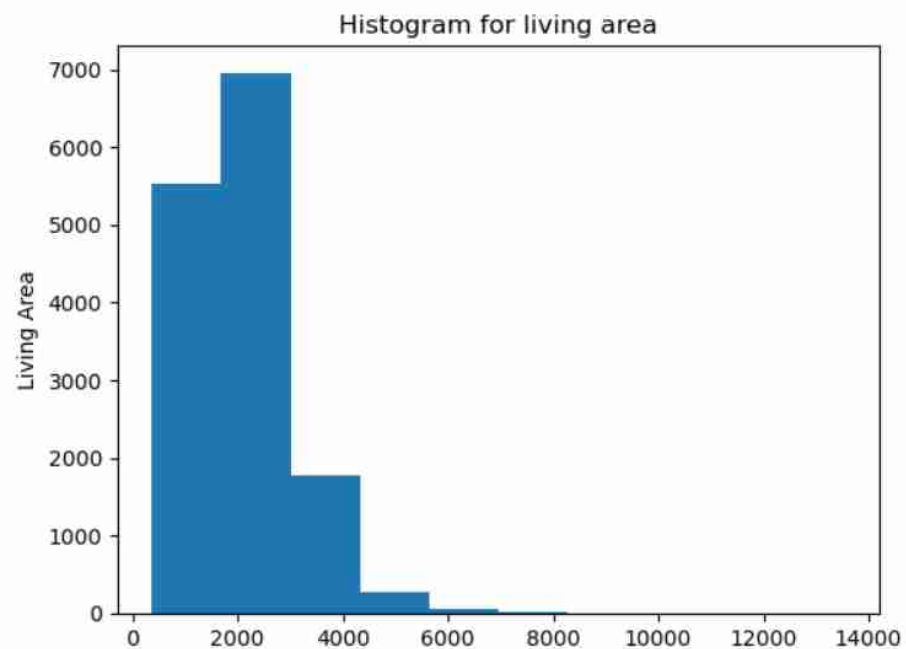




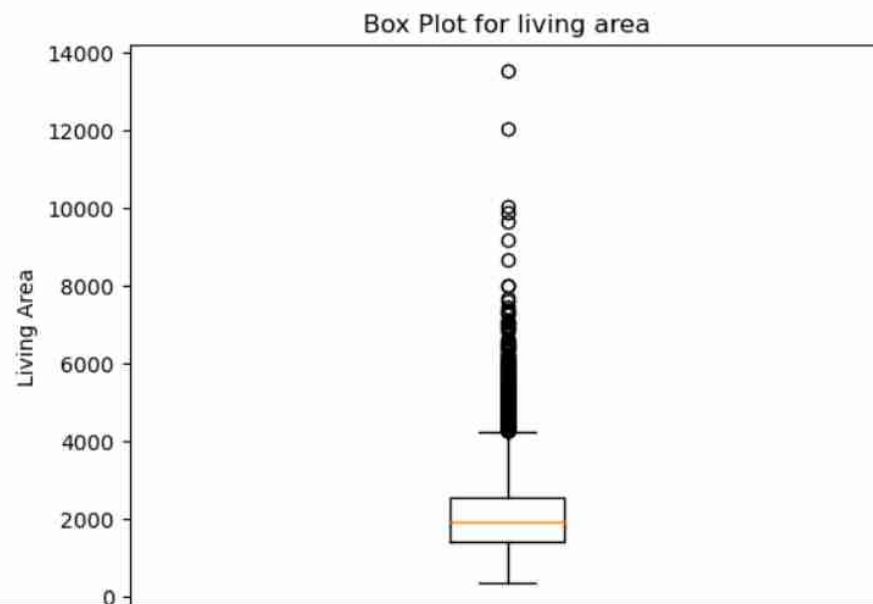
```
In [27]: plt.title('Histogram for living area')
plt.xlabel('Index')
plt.ylabel('Living Area')
plt.hist(df['living area'])
```

```
Out[27]: (array([5.530e+03, 6.958e+03, 1.779e+03, 2.770e+02, 5.500e+01, 1.400e+01,
2.000e+00, 3.000e+00, 1.000e+00, 1.000e+00]),
array([ 370., 1687., 3004., 4321., 5638., 6955., 8272., 9589.,
10906., 12223., 13540.]),
<BarContainer object of 10 artists>)
```



```
In [32]: plt.title('Box Plot for living area')
plt.xlabel('Index')
plt.ylabel('Living Area')
plt.boxplot(df['living area'])
```

```
Out[32]: {'whiskers': [<matplotlib.lines.Line2D at 0x270a6216350>,
<matplotlib.lines.Line2D at 0x270a6216590>],
'caps': [<matplotlib.lines.Line2D at 0x270a6215c00>,
<matplotlib.lines.Line2D at 0x270a6214b80>],
'boxes': [<matplotlib.lines.Line2D at 0x270a6215810>],
'medians': [<matplotlib.lines.Line2D at 0x270a6215180>],
'fliers': [<matplotlib.lines.Line2D at 0x270a6214d90>],
'means': []}
```





```
In [ ]: # 1. Download the dataset
        # 2. Load the dataset
```

```
In [3]: import pandas as pd
        import numpy as np
```

```
In [5]: df=pd.read_csv('House Price India.csv')
```

Out[5]:

	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	Latitude	Longitude	liv
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5	...	1921	0	122003	52.8645	-114.557	
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5	...	1909	0	122004	52.8878	-114.470	
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3	...	1939	0	122004	52.8852	-114.468	
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3	...	2001	0	122005	52.9532	-114.321	
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	4	...	1929	0	122006	52.9047	-114.485	
...
14615	6762830250	42734	2	1.50	1556	20000	1.0	0	0	4	...	1957	0	122066	52.6191	-114.472	
14616	6762830339	42734	3	2.00	1680	7000	1.5	0	0	4	...	1968	0	122072	52.5075	-114.393	
14617	6762830618	42734	2	1.00	1070	6120	1.0	0	0	3	...	1962	0	122056	52.7289	-114.507	
14618	6762830709	42734	4	1.00	1030	6621	1.0	0	0	4	...	1955	0	122042	52.7157	-114.411	
14619	6762831463	42734	3	1.00	900	4770	1.0	0	0	3	...	1969	2009	122018	52.5338	-114.552	

14620 rows × 23 columns





In []: *# 4. perform the descriptive statistics for the dataset*

In [2]:

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

In [3]: df.describe()

Out[3]:

	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
	462000e+04	14620.000000	14620.000000	14620.000000	14620.000000	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.000000	...	14620.000000
	762821e+09	42604.538646	3.379343	2.129583	2098.262996	1.509328e+04	1.502360	0.007661	0.233105	3.430506	...	1970.926402
	237575e+03	67.347991	0.938719	0.769934	928.275721	3.791962e+04	0.540239	0.087193	0.766259	0.664151	...	29.493625
	762810e+09	42491.000000	1.000000	0.500000	370.000000	5.200000e+02	1.000000	0.000000	0.000000	1.000000	...	1900.000000
	762815e+09	42546.000000	3.000000	1.750000	1440.000000	5.010750e+03	1.000000	0.000000	0.000000	3.000000	...	1951.000000
	762821e+09	42600.000000	3.000000	2.250000	1930.000000	7.620000e+03	1.500000	0.000000	0.000000	3.000000	...	1975.000000
	762826e+09	42662.000000	4.000000	2.500000	2570.000000	1.080000e+04	2.000000	0.000000	0.000000	4.000000	...	1997.000000
	762832e+09	42734.000000	33.000000	8.000000	13540.000000	1.074218e+06	3.500000	1.000000	4.000000	5.000000	...	2015.000000

23 columns



In [4]: *# 5.Handle the missing values*

In [5]:

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

```
id          0
Date        0
number of bedrooms
```

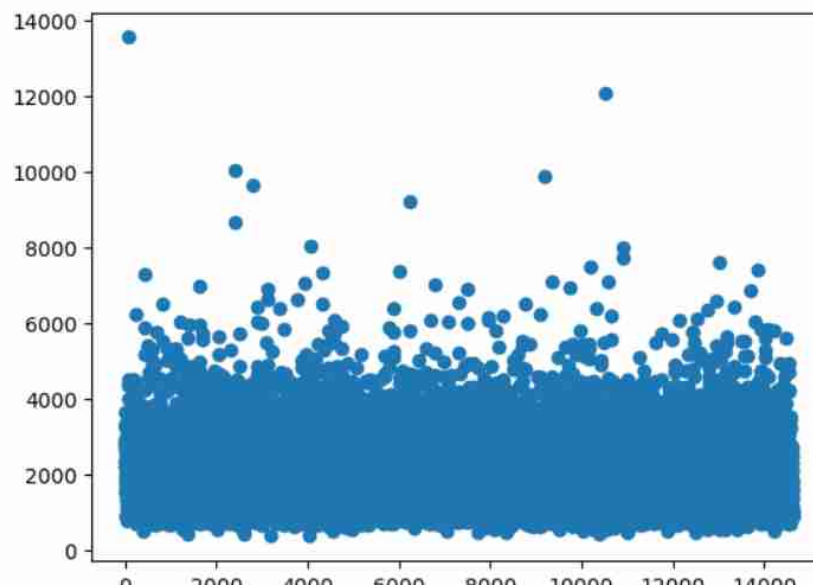


```
In [ ]: # 3. Perform the univariate, Bivariate and Multivariate Analysis visualizations
```

```
In [7]: import matplotlib.pyplot as plt  
import seaborn as sns
```

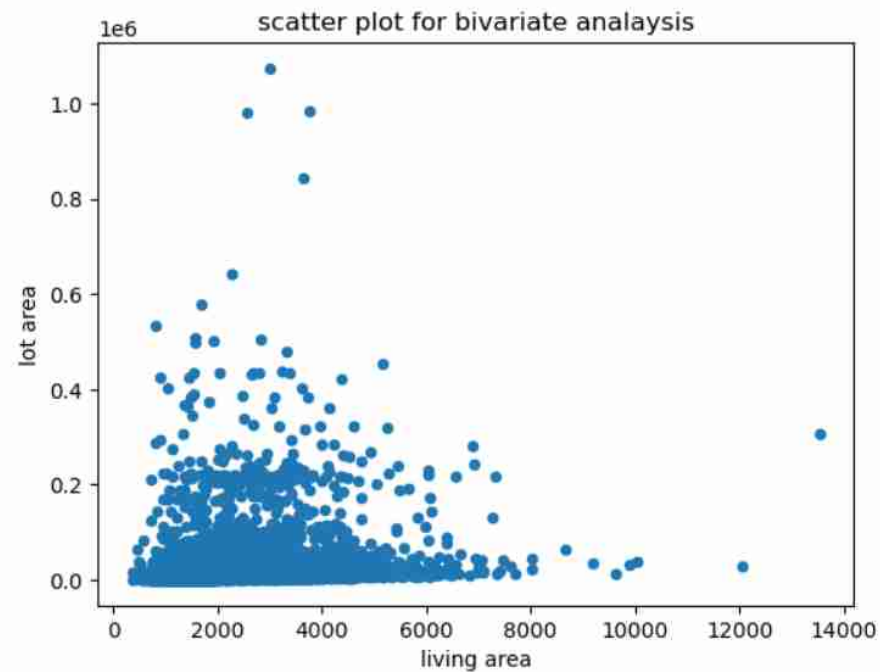
```
In [ ]: # VISUALIZATION OF UNIVARIATE ANALYSIS FOR LIVING AREA IN THE HOUSE PRICE PREDICTION DATASET
```

```
In [13]: plt.scatter(df.index, df['living area'])  
plt.show()
```

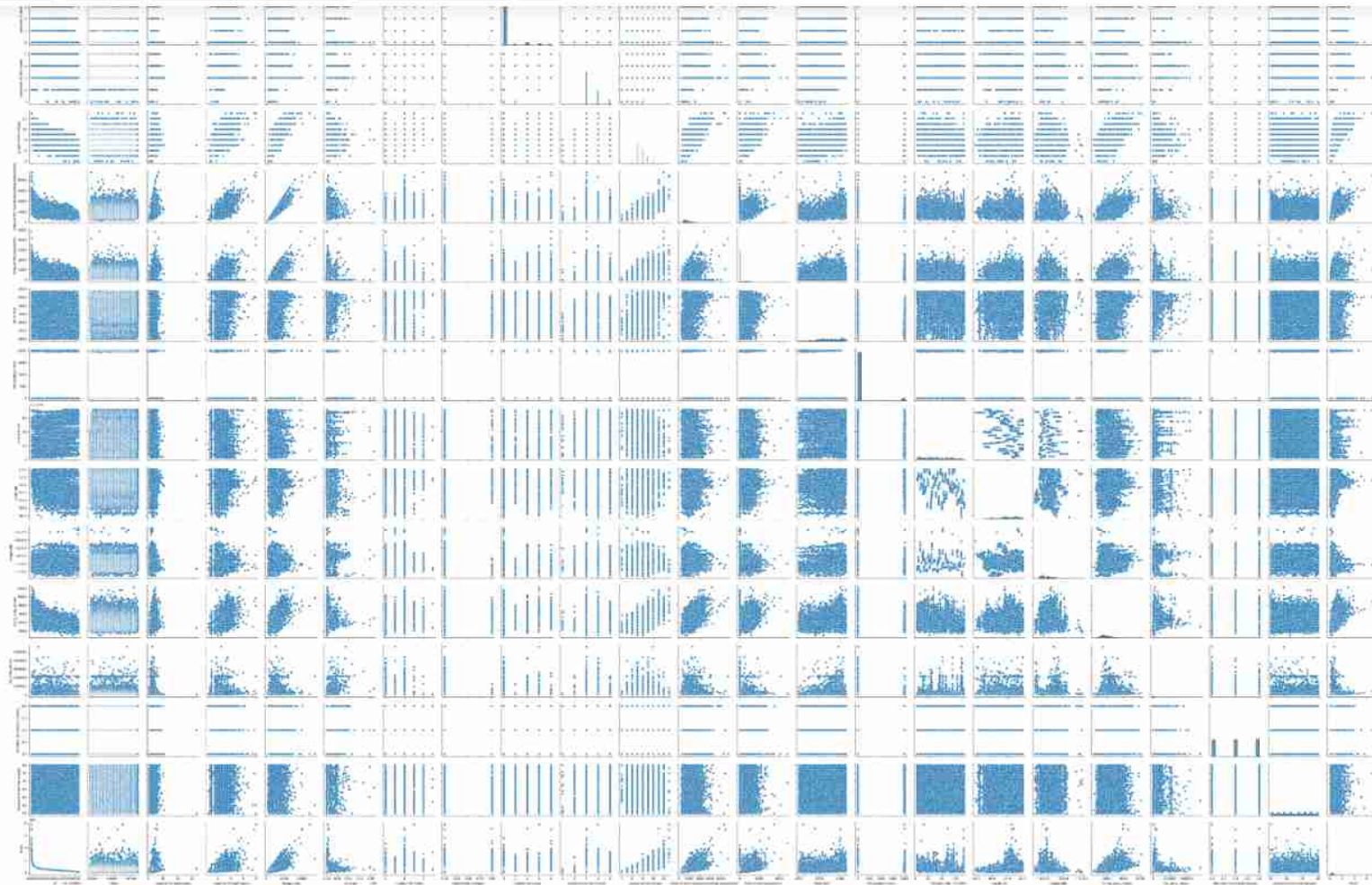


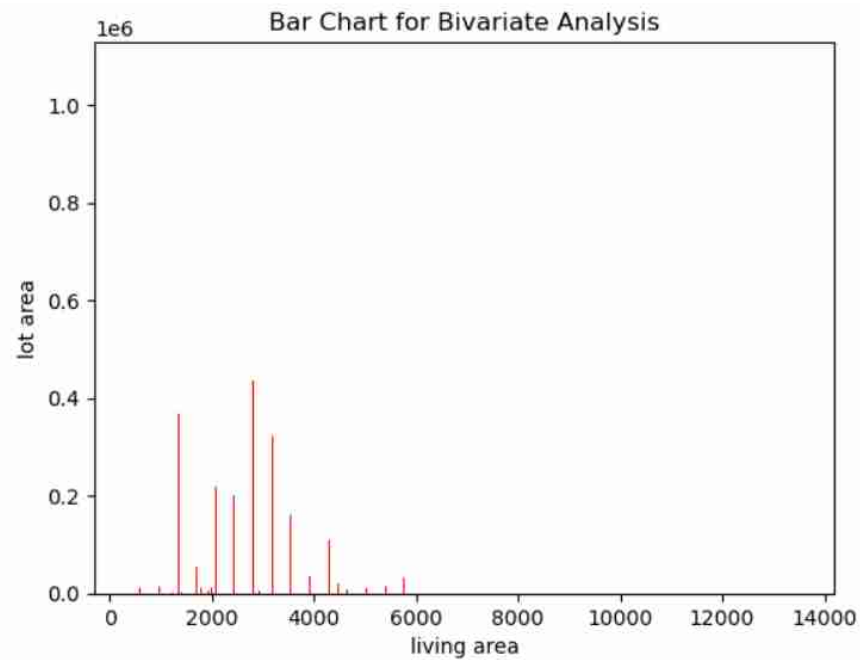
```
In [ ]: #VISUALIZATION OF BIVARIATE ANALYSIS FOR LIVING AREA IN THE HOUSE PRICE PREDICTION DATASET
```

```
In [45]: df.plot(title='scatter plot for bivariate analysis',x='living area',y='lot area',kind='scatter')
plt.show()
```



```
In [47]: plt.title('Bar Chart for Bivariate Analysis')
plt.xlabel('living area')
```



```
In [ ]: #VISUALIZATION OF MULTI-VARIATE ANALYSIS FOR LIVING AREA IN THE HOUSE PRICE PREDICTION DATASET
```

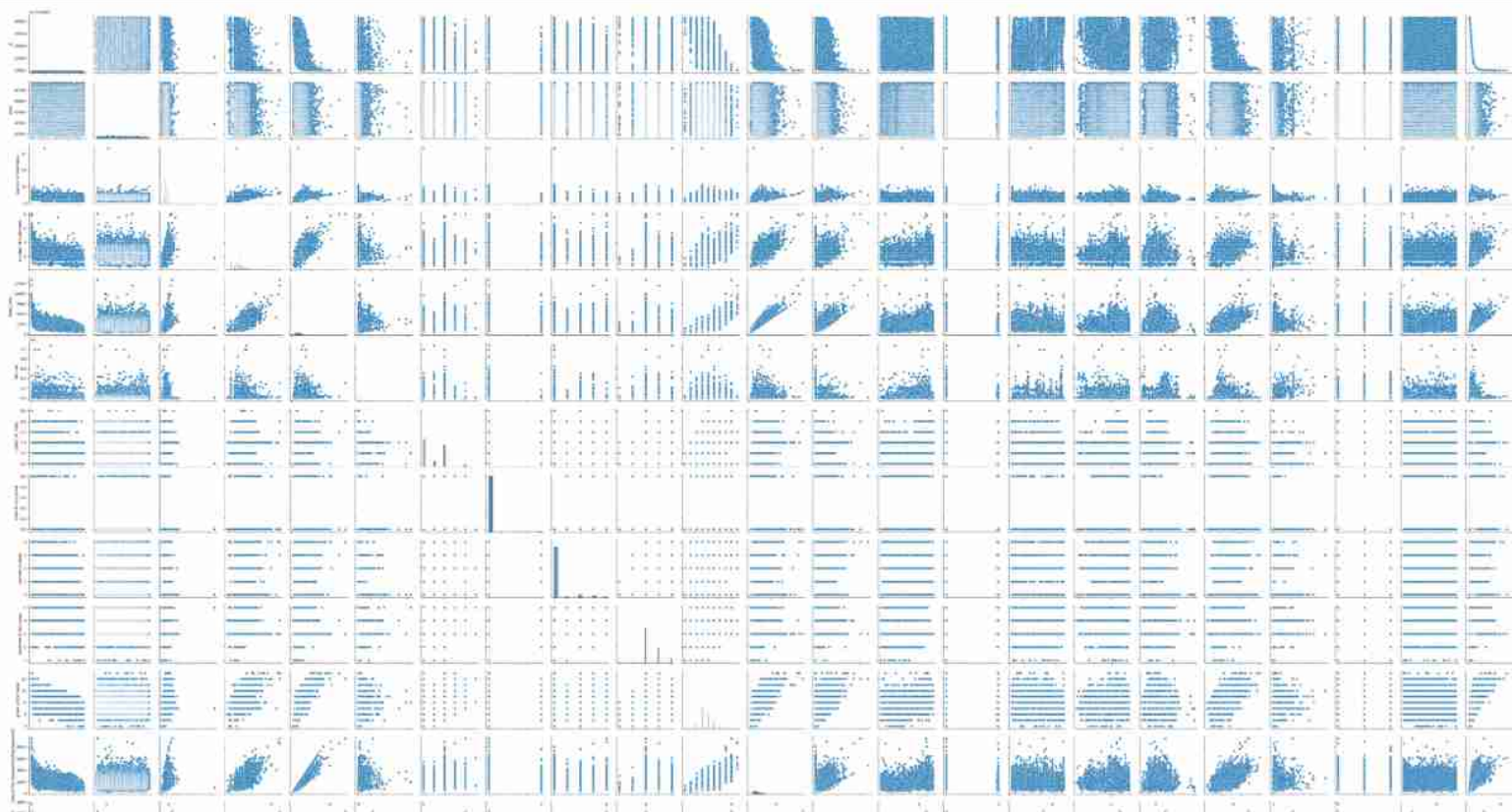
```
In [51]: sns.heatmap(df.corr(),annot=True)
```

```
Out[51]: <Axes: >
```

id -1 0.46389266-0.31112903075729069029-0.87160908047 - 1.0



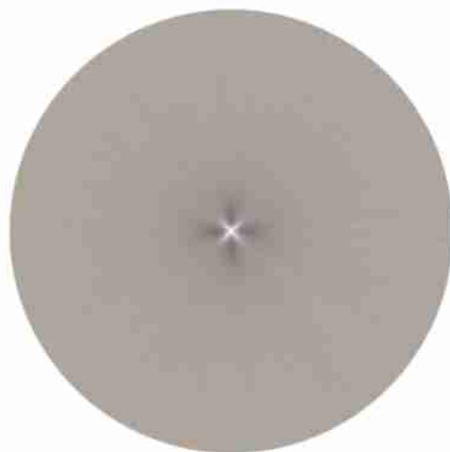
```
In [10]: import seaborn as sns
import matplotlib.pyplot as plt
sns.pairplot(df)
plt.show()
```





```
In [40]: plt.title('Pie Chart for living area')
plt.pie(df['living area'])
plt.show()
```

Pie Chart for living area

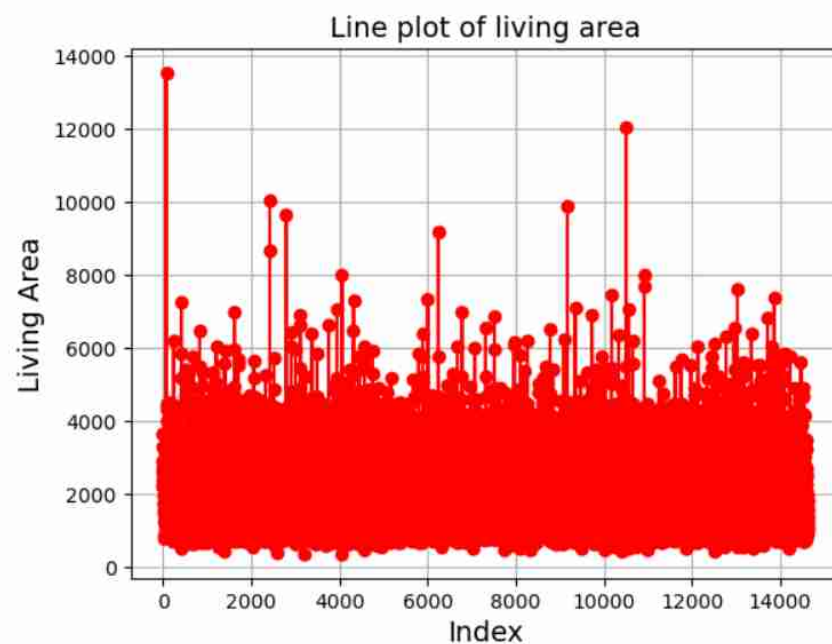


```
In [ ]:
```

```
In [ ]: #VISUALIZATION OF BIVARIATE ANALYSIS FOR LIVING AREA IN THE HOUSE PRICE PREDICTION DATASET
```

```
In [45]:
```

```
In [29]: plt.plot(df.index, df['living area'], color='red',marker='o' )  
plt.title('Line plot of living area', fontsize=14)  
plt.xlabel('Index', fontsize=14)  
plt.ylabel('Living Area', fontsize=14)  
plt.grid(True)  
plt.show()
```



```
In [27]: plt.title('Histogram for living area')
```

```
In [7]: df.dropna(inplace=True)
df.fillna(df.mean(), inplace=True)
df.fillna(0, inplace=True)
df.fillna(df.mode().iloc[0], inplace=True)
```

NameError Traceback (most recent call last)

~\AppData\Local\Temp\ipykernel_11308\3074672682.py in <module>

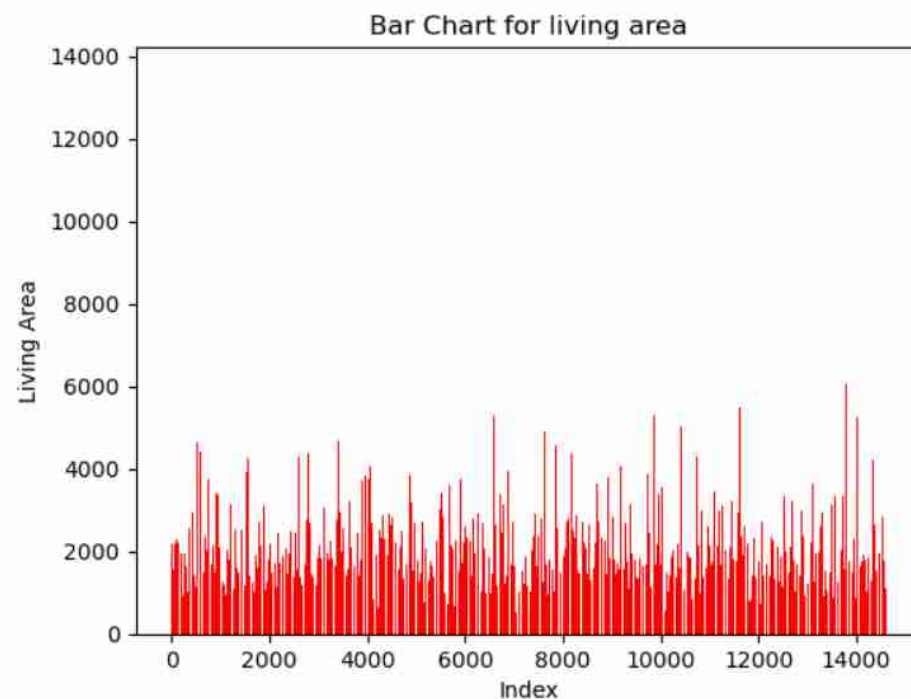
```
----> 1 df.dropna(inplace=True)
      2 df.fillna(df.mean(), inplace=True)
      3 df.fillna(0, inplace=True)
      4 df.fillna(df.mode().iloc[0], inplace=True)
```

NameError: name 'df' is not defined

EXPLANATION: There are no missing values in the dataset and has no null values and already the null paces are filled with 0



```
In [39]: plt.title('Bar Chart for living area')
plt.xlabel('Index')
plt.ylabel('Living Area')
plt.bar(df.index,df['living area'],color='red')
plt.show()
```

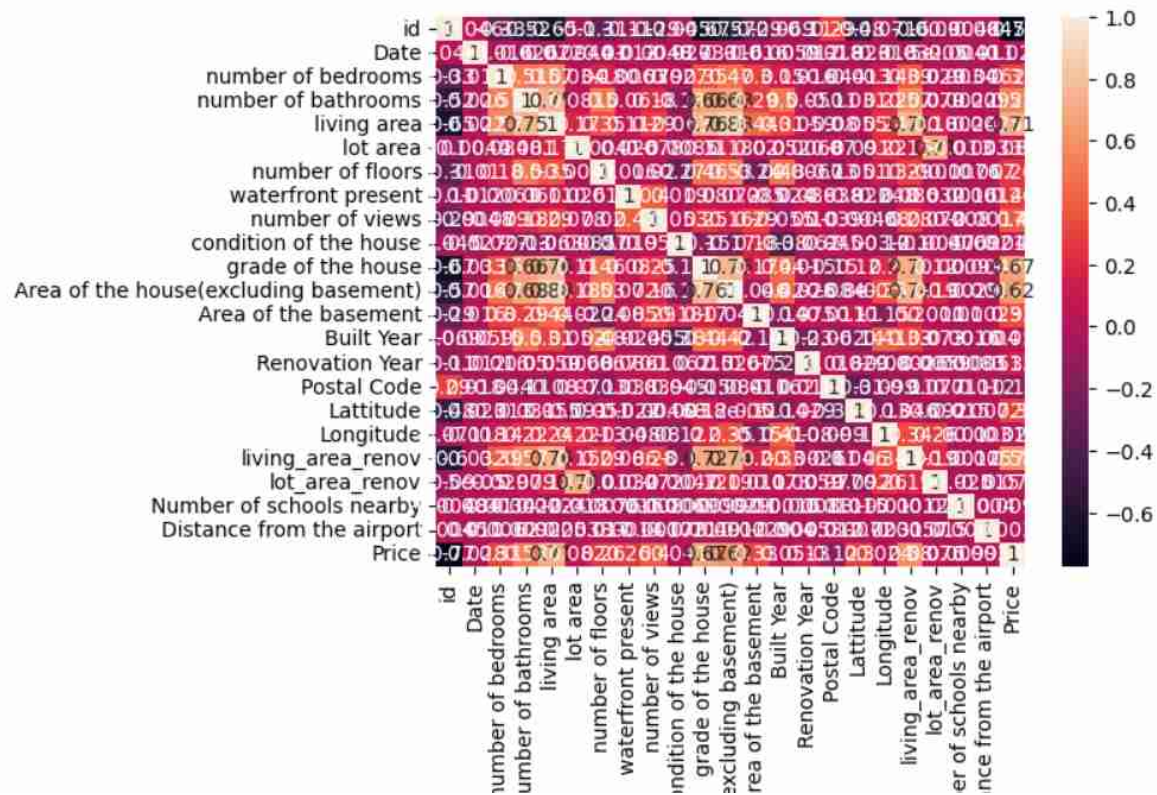


```
In [40]: plt.title('Pie Chart for living area')
plt.pie(df['living area'])
plt.show()
```



```
In [51]: sns.heatmap(df.corr(),annot=True)
```

```
Out[51]: <Axes: >
```





In [4]: # impute the missing values

In [5]: `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
```