ASSIGNMENT-2 (21BCE3220) Anoop Kumar

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import rcParams
import seaborn as sns
```

TASK 1 and TASK 2

In [ ]:
```
df=pd.read_csv("/content/House Price India.csv")
df.head()
```

Out[ ]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condition of the house
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	4

5 rows × 23 columns

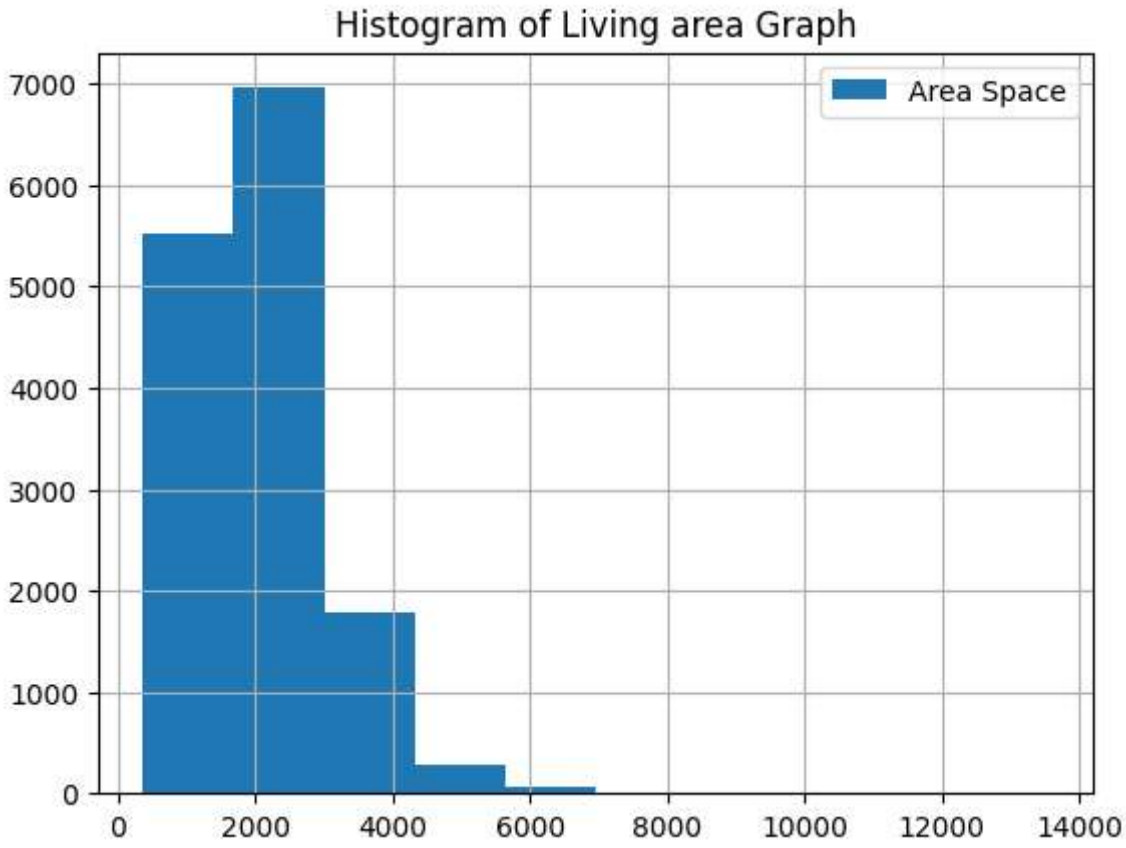
In [ ]:
```
df.shape
```

Out[ ]:
```
(14620, 23)
```

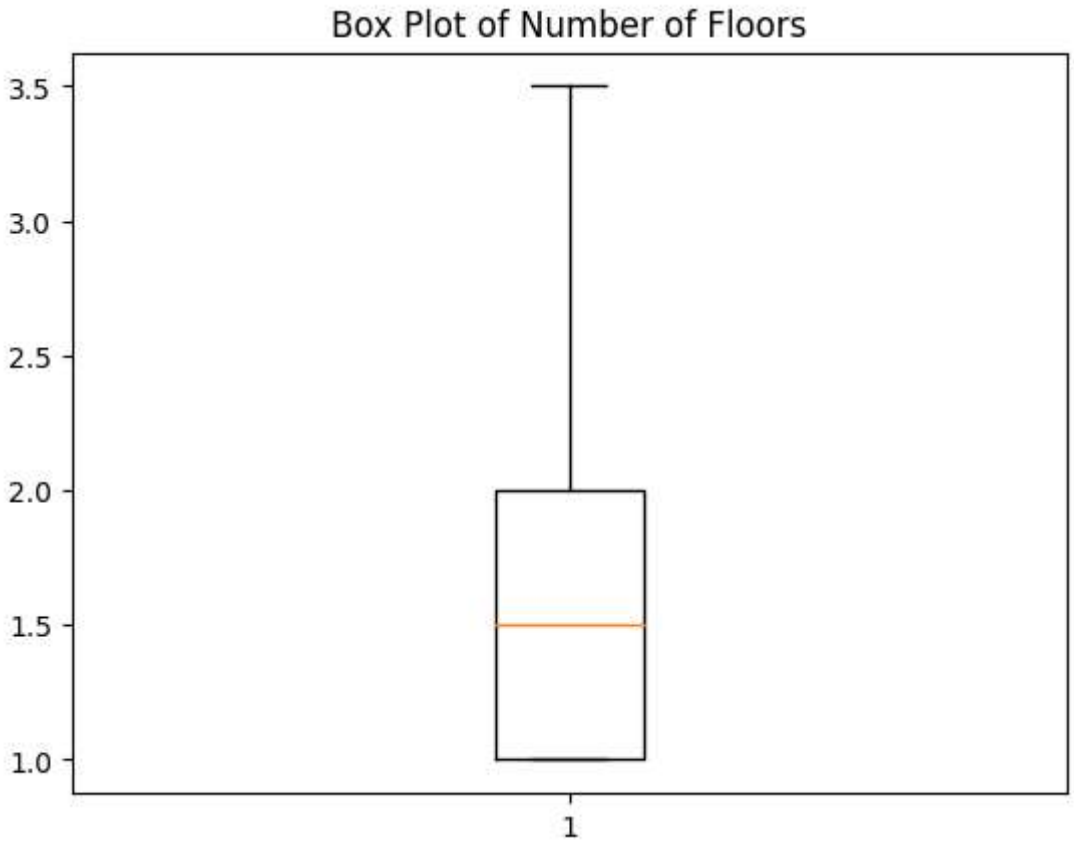
TASK 3

UNIVARIATE ANALYSIS

In [ ]:
```
df["living area"].hist()
plt.legend(["Area Space"])
plt.title("Histogram of Living area Graph")
plt.show()
```

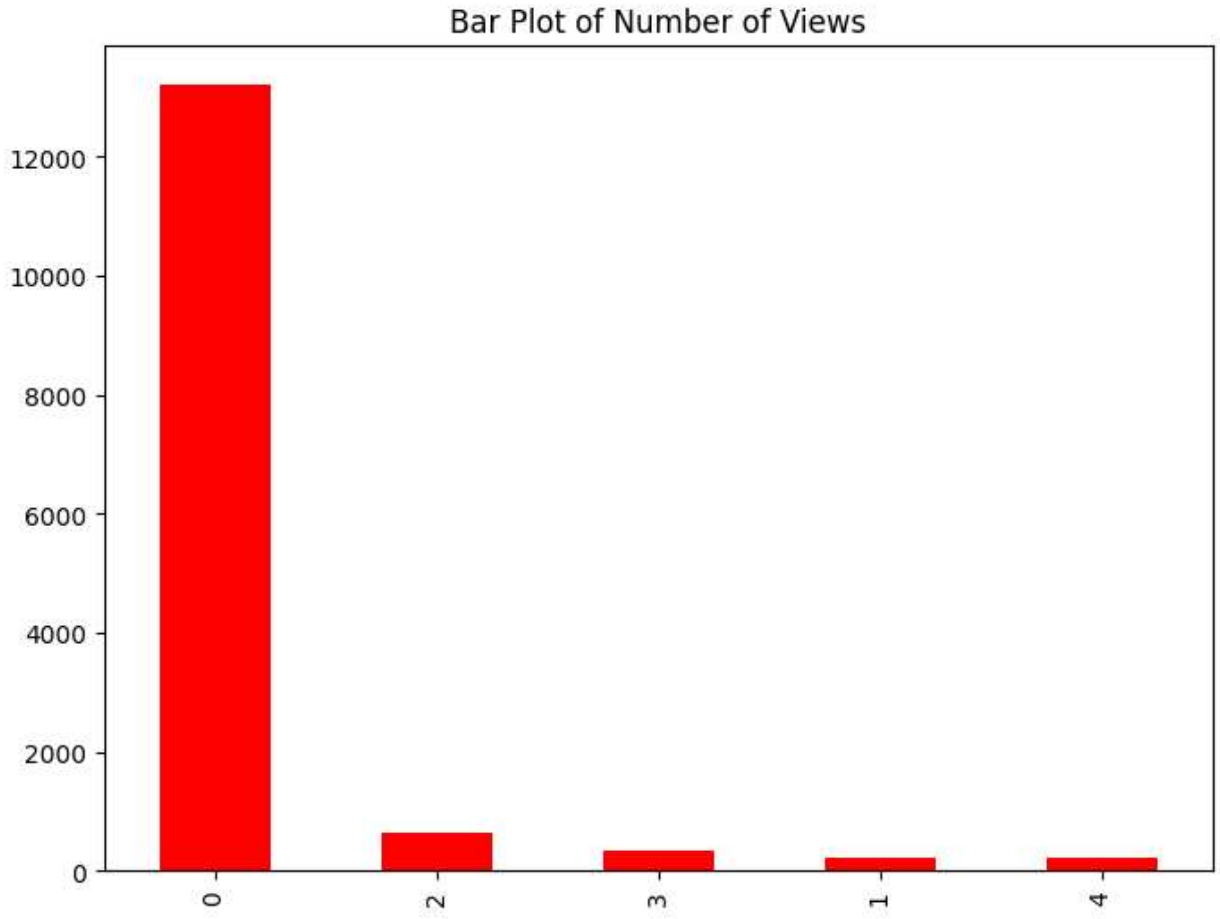


In [ ]:
```
plt.boxplot(df["number of floors"])
plt.title("Box Plot of Number of Floors")
plt.show()
```



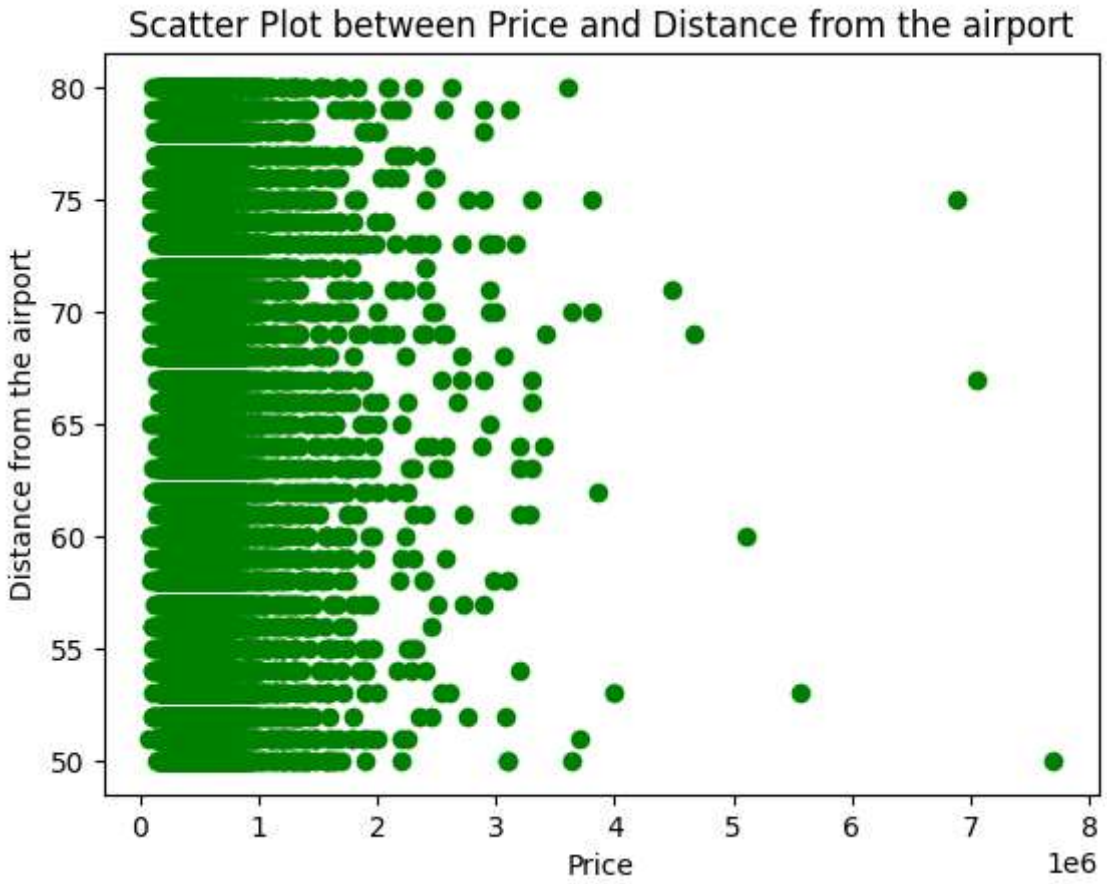
In [ ]:
```
category_counts = df['number of views'].value_counts()

plt.figure(figsize=(8, 6))
category_counts.plot(kind='bar', color='red')
plt.title('Bar Plot of Number of Views')
plt.show()
```



BIVARIATE ANALYSIS

In [ ]:
```
plt.scatter(df['Price'], df['Distance from the airport'],color="green")
plt.xlabel('Price')
plt.ylabel('Distance from the airport')
plt.title('Scatter Plot between Price and Distance from the airport')
plt.show()
```



In [ ]:

MULTIVARIATE ANALYSIS

In [ ]:
```
df.corr()
```

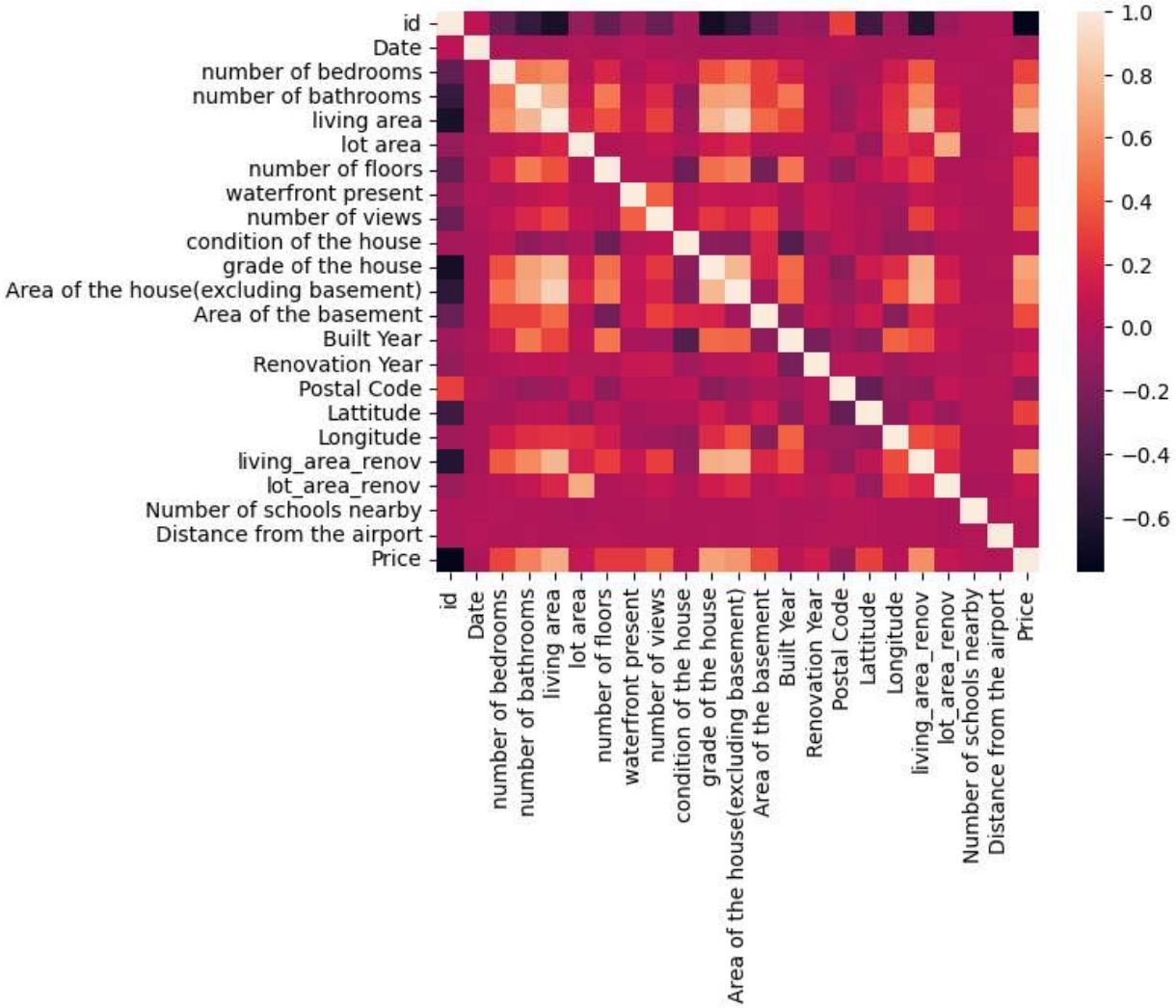
Out[ ]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterf pre
id	1.000000	0.045966	-0.329034	-0.516909	-0.648127	-0.100269	-0.312305	-0.11
Date	0.045966	1.000000	-0.015663	-0.026485	-0.021958	0.004392	-0.010335	0.01
number of bedrooms	-0.329034	-0.015663	1.000000	0.509784	0.570526	0.034416	0.177294	-0.00
number of bathrooms	-0.516909	-0.026485	0.509784	1.000000	0.753517	0.080806	0.502924	0.06
living area	-0.648127	-0.021958	0.570526	0.753517	1.000000	0.174420	0.354743	0.10
lot area	-0.100269	0.004392	0.034416	0.080806	0.174420	1.000000	-0.004138	0.02
number of floors	-0.312305	-0.010335	0.177294	0.502924	0.354743	-0.004138	1.000000	0.01
waterfront present	-0.112937	0.012006	-0.006257	0.060104	0.105837	0.026282	0.016316	1.00
number of views	-0.293004	-0.004782	0.078665	0.183789	0.287728	0.078308	0.020153	0.40
condition of the house	-0.045061	-0.027402	0.026597	-0.128232	-0.063358	-0.008548	-0.269928	0.01
grade of the house	-0.673448	-0.033097	0.352945	0.663054	0.761835	0.110546	0.463082	0.07
Area of the house(excluding basement)	-0.565116	-0.015994	0.473599	0.684391	0.875793	0.183553	0.525643	0.07
Area of the basement	-0.290806	-0.015711	0.300332	0.287190	0.441491	0.019755	-0.242976	0.08
Built Year	-0.068645	-0.005869	0.152954	0.498127	0.309602	0.051615	0.481565	-0.02
Renovation Year	-0.109155	-0.011636	0.016132	0.049669	0.059400	0.006848	0.006705	0.08
Postal Code	0.294709	0.018243	-0.044156	-0.105546	-0.080303	0.070131	-0.129788	0.03
Lattitude	-0.479334	-0.023327	-0.013163	0.031156	0.054518	-0.090983	0.050731	-0.02
Longitude	-0.070841	-0.018231	0.135712	0.223904	0.240208	0.221432	0.127550	-0.04
living_area_renov	-0.599900	-0.032495	0.389855	0.570530	0.757571	0.149744	0.285093	0.08
lot_area_renov	-0.089604	-0.000050	0.029400	0.078627	0.180312	0.706812	-0.010120	0.03
Number of schools nearby	-0.004821	-0.004071	0.003397	0.002180	0.002370	-0.012671	-0.007579	0.00
Distance from the airport	-0.004542	0.011457	-0.006157	0.009206	0.002511	0.003291	0.016567	0.00
Price	-0.773114	-0.027919	0.308460	0.531735	0.712169	0.081992	0.262732	0.26

23 rows × 23 columns

In [ ]:
```
sns.heatmap(df.corr())
```

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



PAIRPLOT

TASK 4

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

Out[ ]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	numbe fl
count	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.000000	1.462000e+04	14620.00
mean	6.762821e+09	42604.538646	3.379343	2.129583	2098.262996	1.509328e+04	1.50
std	6.237575e+03	67.347991	0.938719	0.769934	928.275721	3.791962e+04	0.54
min	6.762810e+09	42491.000000	1.000000	0.500000	370.000000	5.200000e+02	1.00
25%	6.762815e+09	42546.000000	3.000000	1.750000	1440.000000	5.010750e+03	1.00
50%	6.762821e+09	42600.000000	3.000000	2.250000	1930.000000	7.620000e+03	1.50
75%	6.762826e+09	42662.000000	4.000000	2.500000	2570.000000	1.080000e+04	2.00
max	6.762832e+09	42734.000000	33.000000	8.000000	13540.000000	1.074218e+06	3.50

8 rows × 23 columns

TASK 5

In [ ]:
```
df.isnull().any()
```

Out[ ]:
```
id                                    False
Date                                  False
number of bedrooms                    False
number of bathrooms                   False
living area                           False
lot area                              False
number of floors                      False
waterfront present                    False
number of views                       False
condition of the house                False
grade of the house                    False
Area of the house(excluding basement)    False
Area of the basement                  False
Built Year                            False
Renovation Year                       False
Postal Code                           False
Lattitude                             False
Longitude                             False
living_area_renov                     False
lot_area_renov                        False
Number of schools nearby              False
Distance from the airport             False
Price                                 False
dtype: bool
```

In [50]:
```
x = df.iloc[:, :-1].values
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values = np.nan, strategy = 'median')
imputer.fit(x[:, 2:18])
x[:, 2:18] = imputer.transform(x[:, 2:18])
print(x)
```

```
[[6.76281014e+09 4.24910000e+04 5.00000000e+00 ... 5.40000000e+03
  2.00000000e+00 5.80000000e+01]
 [6.76281064e+09 4.24910000e+04 4.00000000e+00 ... 4.00000000e+03
  2.00000000e+00 5.10000000e+01]
 [6.76281100e+09 4.24910000e+04 5.00000000e+00 ... 6.60000000e+03
  1.00000000e+00 5.30000000e+01]
 ...
 [6.76283062e+09 4.27340000e+04 2.00000000e+00 ... 6.12000000e+03
  2.00000000e+00 6.40000000e+01]
 [6.76283071e+09 4.27340000e+04 4.00000000e+00 ... 6.63100000e+03
  3.00000000e+00 5.40000000e+01]
 [6.76283146e+09 4.27340000e+04 3.00000000e+00 ... 3.48000000e+03
  2.00000000e+00 5.50000000e+01]]
```