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


# Load the dataset
df = pd.read_csv('/content/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	condition of the house	...	Built Year	Renovation Year	Pos C
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5	...	1921	0	122
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5	...	1909	0	122
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3	...	1939	0	122
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3	...	2001	0	122
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	4	...	1929	0	122

5 rows × 23 columns

```
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
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
```

UNIVARIATE ANALYSIS

```
# for Waterfront present
df['waterfront present'].value_counts()
```

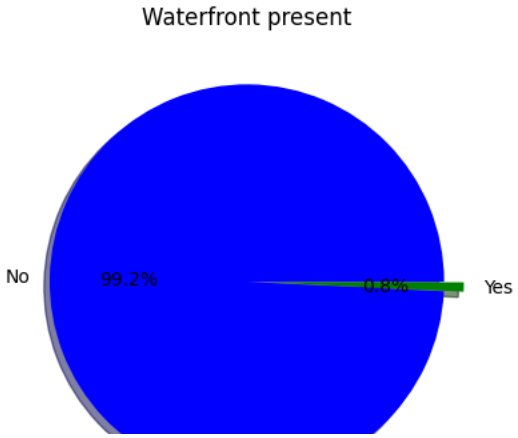
```
0    14508
1      112
Name: waterfront present, dtype: int64
```

```
df['waterfront present'].unique()
```

```
array([0, 1])
```

```
plt.pie(df['waterfront present'].value_counts(), [0,0.1],labels = ['No', 'Yes'], autopct = '%1.1f%%', shadow = True,colors = ['blue','green']
plt.title('Waterfront present')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```




```
#Grade of the house
df['grade of the house'].unique()
```

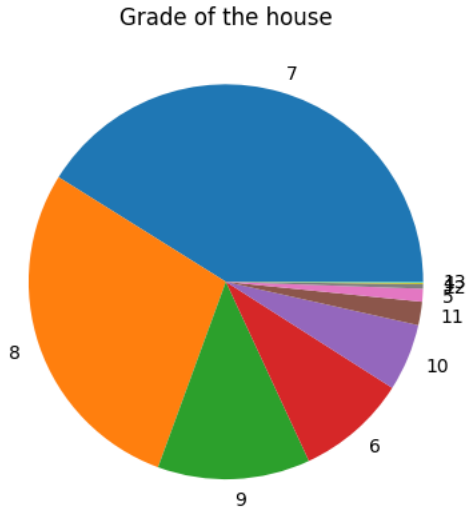
```
array([10,  8,  9,  7,  6, 12, 11,  5,  4, 13])
```

```
df['grade of the house'].value_counts()
```

```
7     6011
8     4137
9     1828
6     1324
10     804
11     280
5      154
12      55
4       17
13      10
Name: grade of the house, dtype: int64
```

```
plt.pie(df['grade of the house'].value_counts(), [0,0,0,0,0,0,0,0,0,0], labels = [7, 8, 9, 6, 10, 11, 5, 12, 4, 13])
plt.title('Grade of the house')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```




```
#condition of house
df['condition of the house'].unique()
```

```
array([5, 3, 4, 2, 1])
```

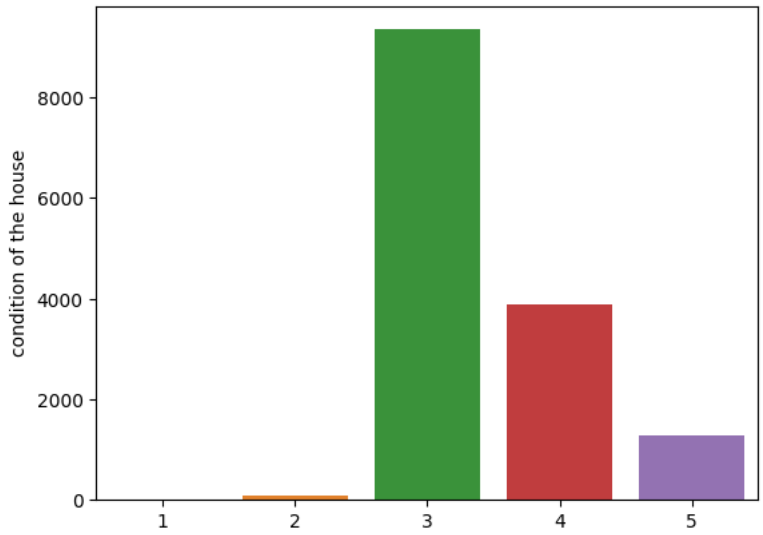
```
df['condition of the house'].value_counts()
```

```
3    9350
4    3874
5    1278
```

```
2     100
1      18
Name: condition of the house, dtype: int64
```

```
sns.barplot(x = df['condition of the house'].value_counts().index, y = df['condition of the house'].value_counts())
```

```
<Axes: ylabel='condition of the house'>
```




```
#BUilt year
df['Built Year'].unique()
```

```
array([1921, 1909, 1939, 2001, 1929, 1951, 2006, 1923, 1955, 1920, 1979,
       1945, 2000, 2005, 2014, 2007, 1948, 1991, 1995, 1980, 2012, 1976,
       2004, 1959, 1968, 1938, 1989, 2013, 1985, 1966, 1944, 1990, 1977,
       1954, 1963, 1956, 1996, 1957, 2008, 1967, 1997, 1978, 1950, 2009,
       1992, 1987, 1983, 1974, 1965, 1949, 1986, 1973, 1900, 1988, 1999,
       1971, 1928, 1998, 1960, 1982, 1908, 1994, 1961, 1902, 2003, 1924,
       1942, 1975, 2010, 1953, 1930, 1962, 1958, 1984, 1969, 1970, 1940,
       1916, 1926, 1964, 1903, 1905, 1912, 1947, 1952, 1910, 1914, 1937,
       1946, 2002, 2011, 1906, 1943, 1922, 1917, 1904, 1981, 1913, 1993,
       1932, 1941, 1918, 1925, 1972, 1919, 1911, 1936, 1927, 1931, 1907,
       1901, 1915, 2015, 1935, 1933, 1934])
```

```
df['Built Year'].value_counts()
```

```
2014    404
2005    319
2006    300
2004    296
2003    295
       ...
1902     20
1935     18
1933     17
1934     15
2015     12
Name: Built Year, Length: 116, dtype: int64
```

```
sns.distplot(df['Built Year'])
```

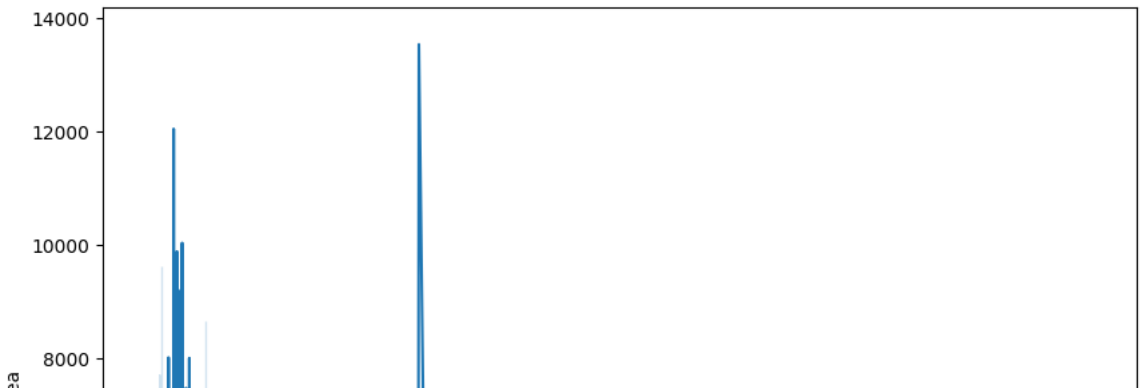
```
<ipython-input-45-f998117e4510>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Built Year'])
<Axes: xlabel='Built Year', ylabel='Density'>
```

BIVARIATE ANALYSIS

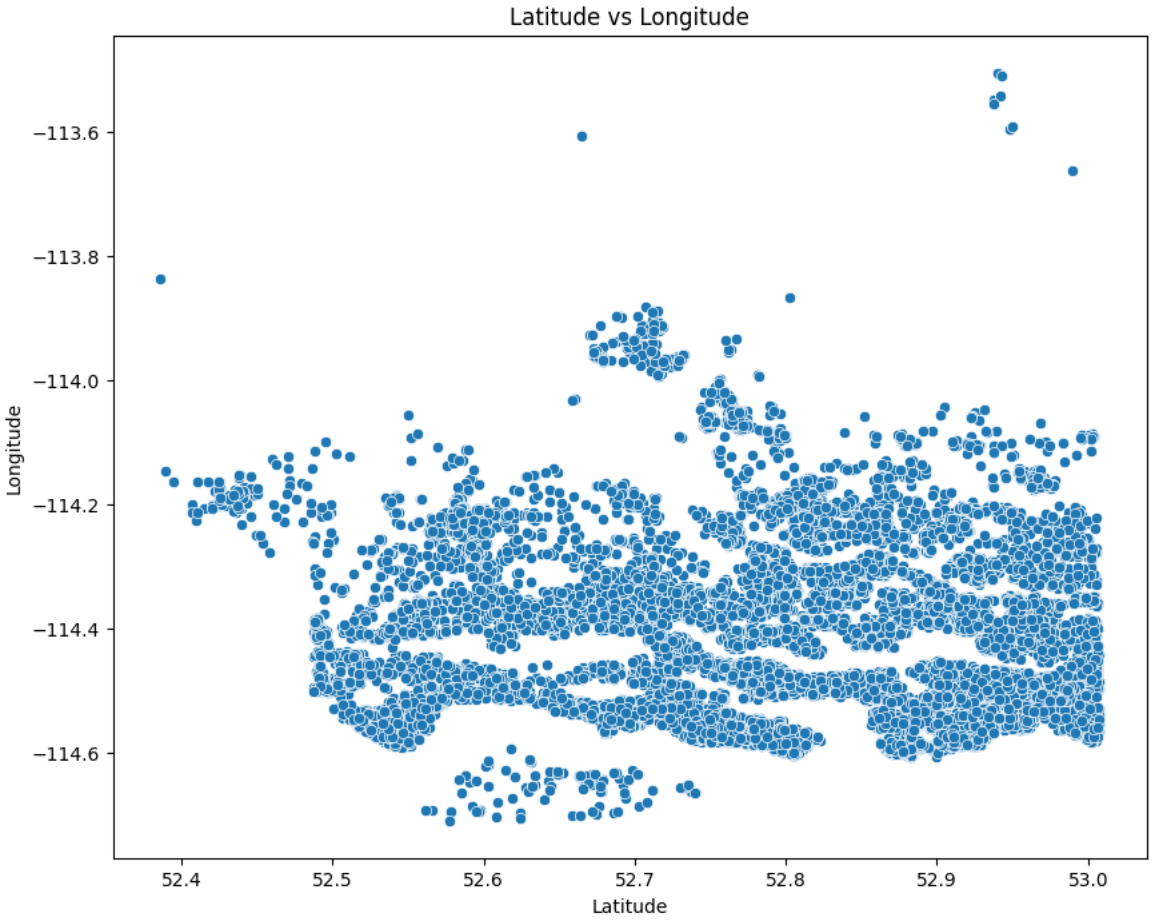
```
#lot area and living room area
print(df['living area'].value_counts())
print(df['lot area'].value_counts())
```

```
1400    93
1010    92
1320    91
1660    90
1820    88
        ..
2448     1
2846     1
5320     1
5930     1
1556     1
Name: living area, Length: 865, dtype: int64
5000     269
6000     176
4000     172
7200     149
7500      82
        ...
5022       1
10961      1
5823       1
11072      1
6621       1
Name: lot area, Length: 7451, dtype: int64
```

```
plt.figure(figsize = (10,8))
sns.lineplot(x = df['lot area'], y = df['living area'])
plt.show()
```

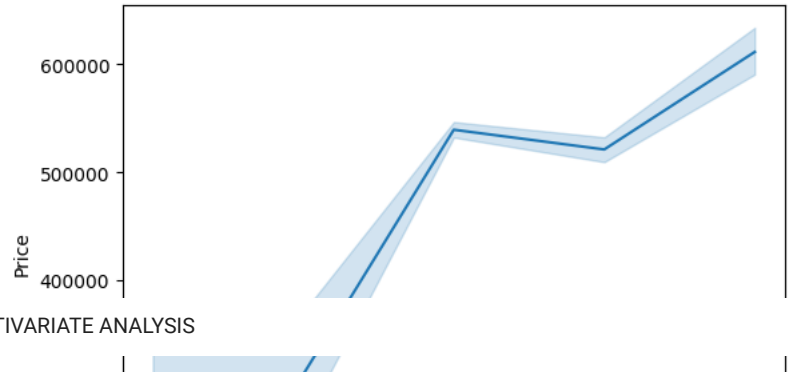



```
# Latitude and Longitude
plt.figure(figsize = (10,8))
sns.scatterplot(x = df['Lattitude'], y = df['Longitude'])
plt.xlabel('Latitude')
plt.ylabel('Longitude')
plt.title('Latitude vs Longitude')
plt.show()
```




```
#Condition of the house and price
sns.lineplot(x = df['condition of the house'], y = df['Price'])
```

```
<Axes: xlabel='condition of the house', ylabel='Price'>
```



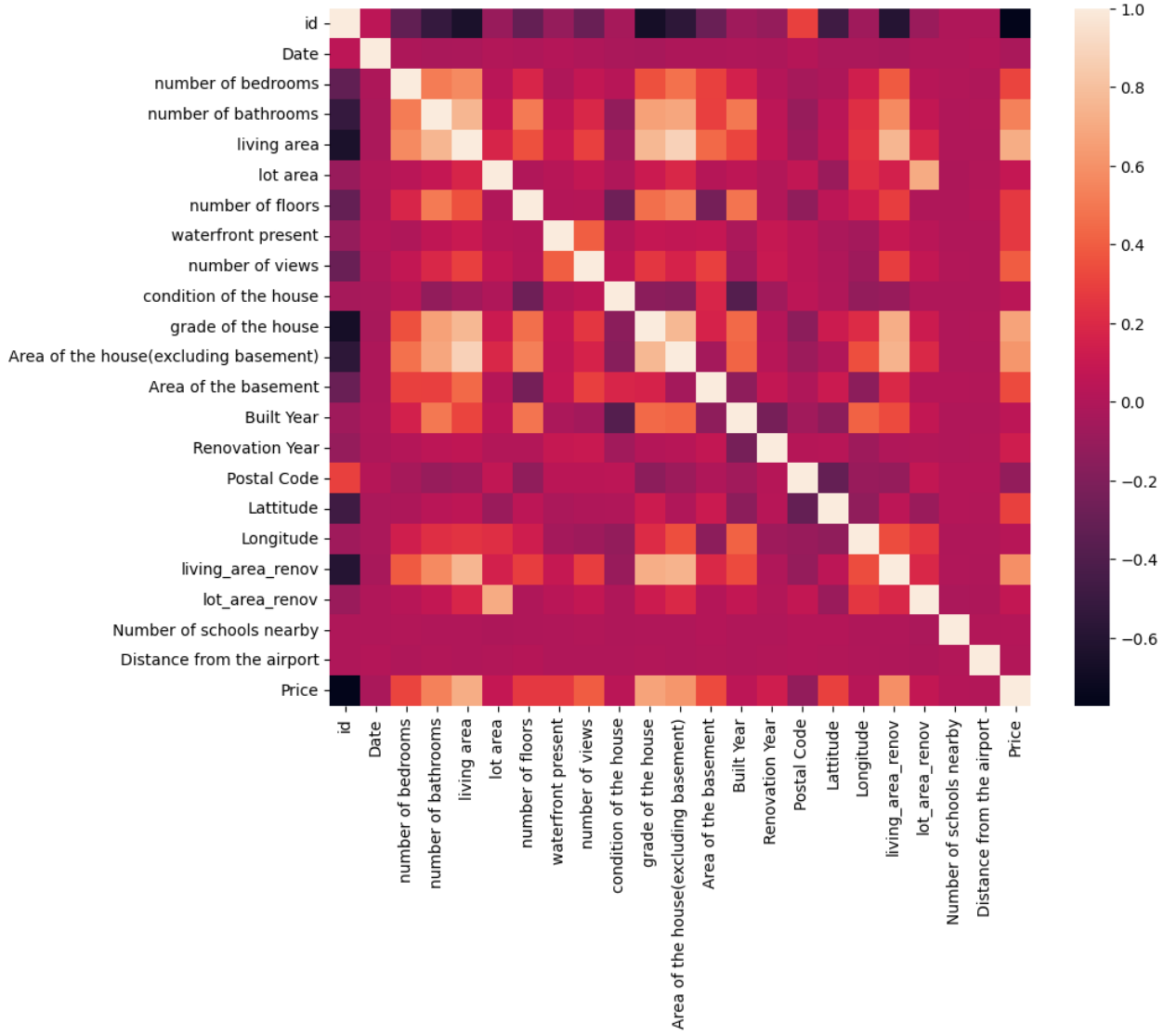
MULTIVARIATE ANALYSIS

```
df.size
```

```
336260
```

```
plt.figure(figsize = (10,8))
sns.heatmap(df.corr())
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



DESCRIPTIVE STATISTICS

```
df.describe()
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	numbe v
count	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.000000	1.462000e+04	14620.000000	14620.000000	14620.00
mean	6.762821e+09	42604.538646	3.379343	2.129583	2098.262996	1.509328e+04	1.502360	0.007661	0.23
std	6.237575e+03	67.347991	0.938719	0.769934	928.275721	3.791962e+04	0.540239	0.087193	0.76
min	6.762810e+09	42491.000000	1.000000	0.500000	370.000000	5.200000e+02	1.000000	0.000000	0.00
25%	6.762815e+09	42546.000000	3.000000	1.750000	1440.000000	5.010750e+03	1.000000	0.000000	0.00
50%	6.762821e+09	42600.000000	3.000000	2.250000	1930.000000	7.620000e+03	1.500000	0.000000	0.00
75%	6.762826e+09	42662.000000	4.000000	2.500000	2570.000000	1.080000e+04	2.000000	0.000000	0.00
max	6.762832e+09	42734.000000	33.000000	8.000000	13540.000000	1.074218e+06	3.500000	1.000000	4.00

8 rows × 23 columns

NULL VALUES

```
df.info() #shows non null count for each column along with data type
```

```
<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  Lattitude                             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
```

```
df.isnull().any() # shows true if null values are present else false
```

```
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
```

```
df.isnull().sum() #shows number of null values for each 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
```

✓ 0s completed at 11:33 PM