

```
#Pavithra S  
#21BIT0389  
#Assingment 2
```

1. Download the dataset: House Price India dataset is downloaded.
2. Load The dataset

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

```
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_pres
0	6762810145	42491		5	2.50	3650	9050		2.0
1	6762810635	42491		4	2.50	2920	4000		1.5
2	6762810998	42491		5	2.75	2910	9480		1.5
3	6762812605	42491		4	2.50	3310	42998		2.0
4	6762812919	42491		3	2.00	2710	4500		1.5

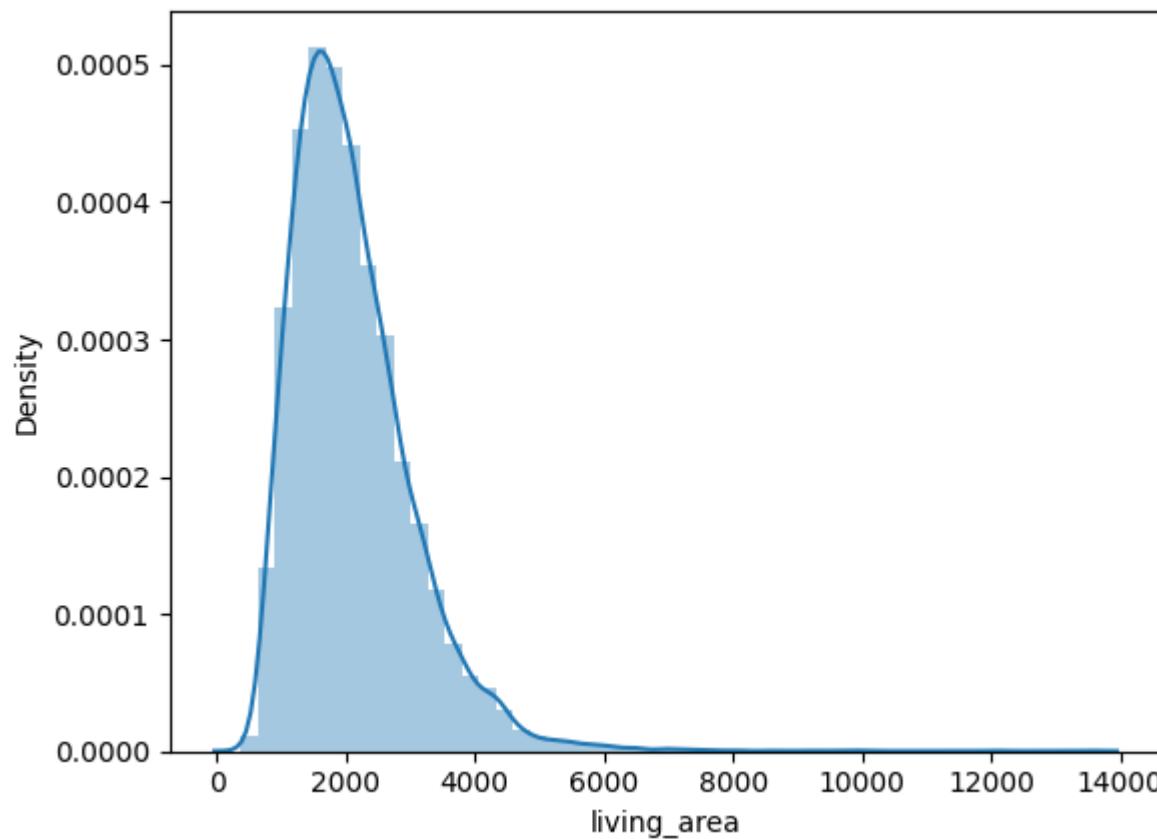
5 rows × 23 columns

Univariate Analysis (Analysis on single feature 'living area')

```
sns.distplot(df.living_area)
```

```
<ipython-input-10-2fe1fc3439c6>: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
```

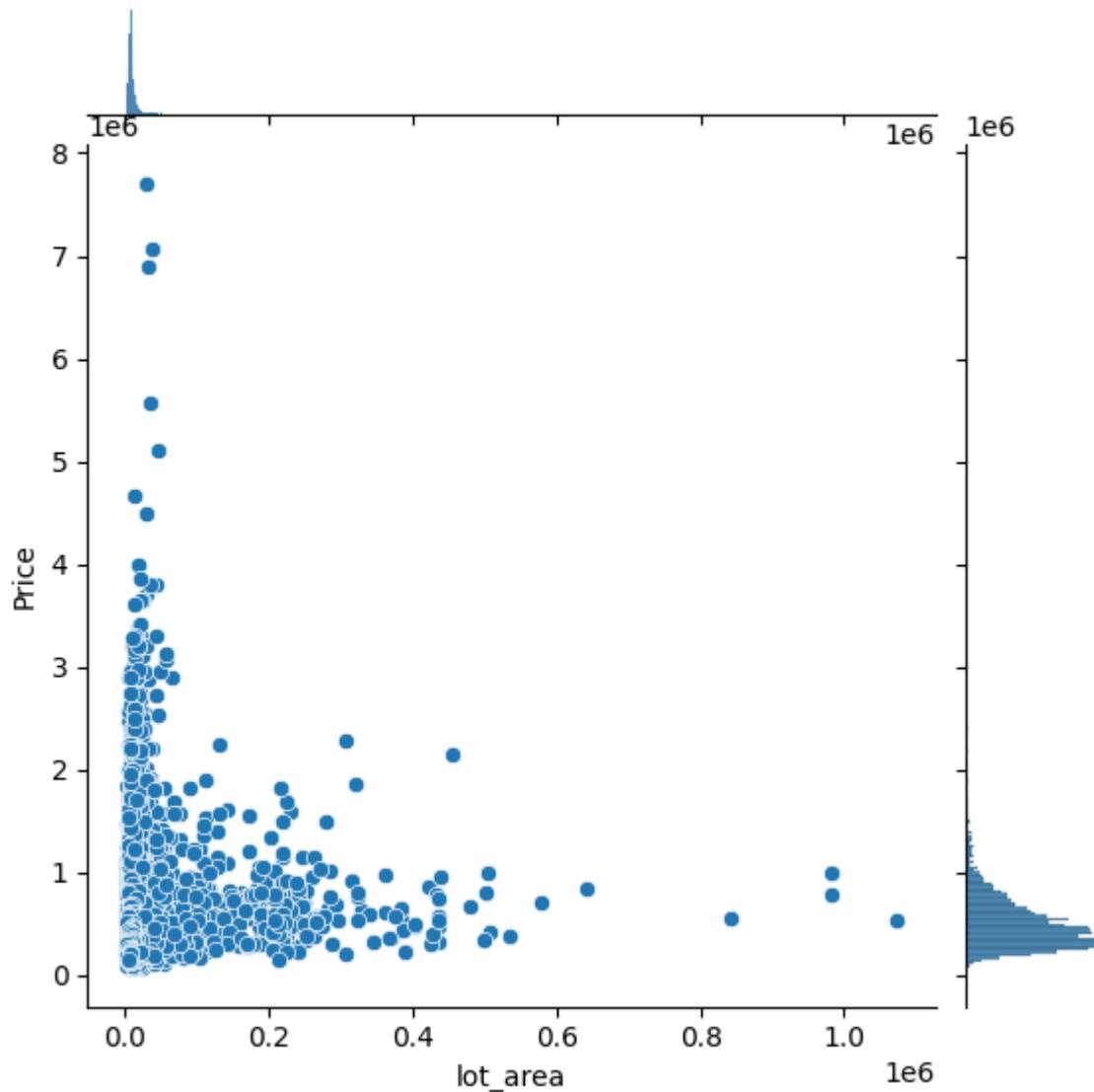
```
sns.distplot(df.living_area)  
<Axes: xlabel='living_area', ylabel='Density'>
```



Bivariate Analysis (Comparision between 'lot_area' feature and 'Price')

```
sns.jointplot(x='lot_area',y='Price',data=df)
```

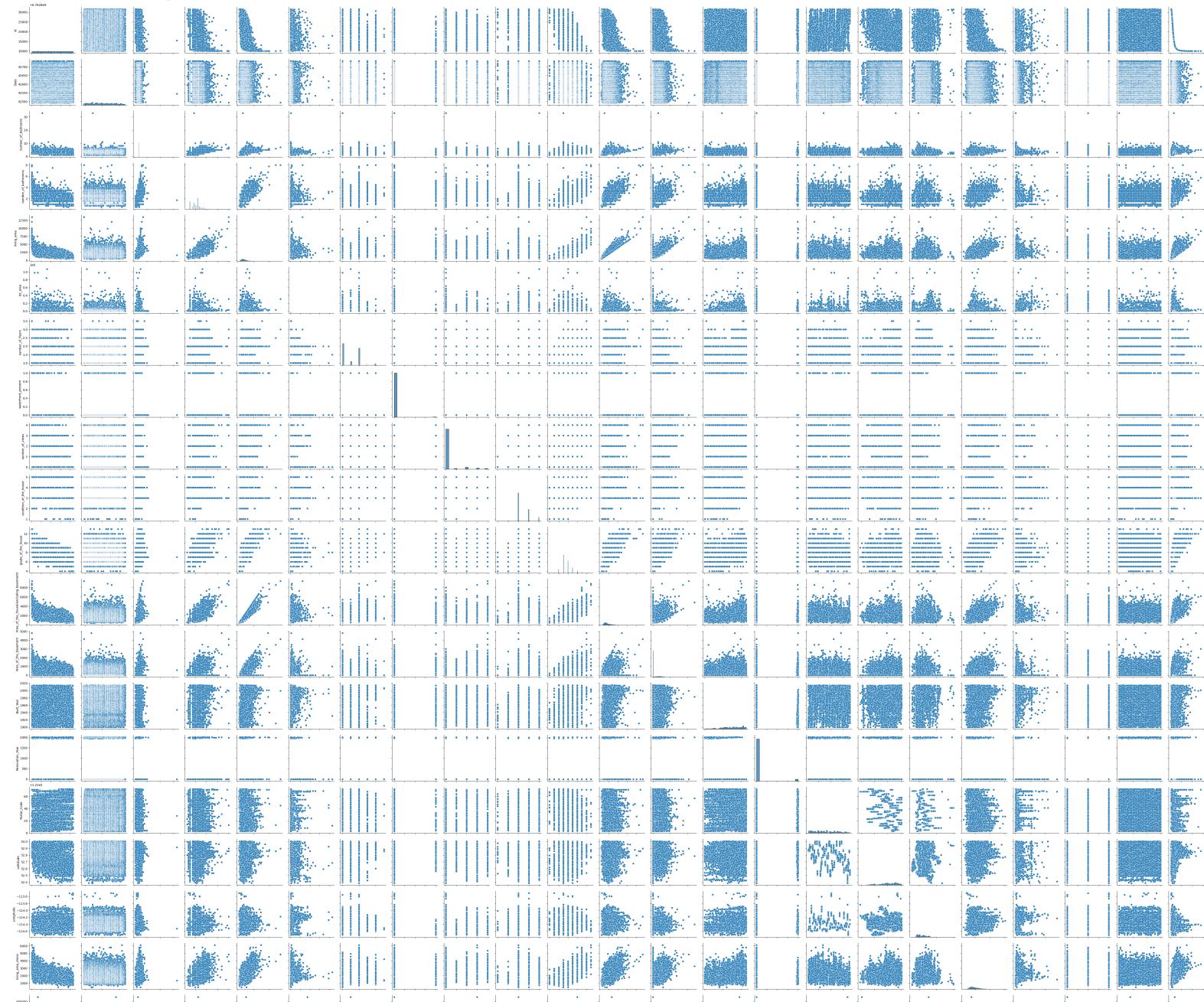
```
<seaborn.axisgrid.JointGrid at 0x7e5b487609a0>
```



Multivariate analysis

```
sns.pairplot(df)
```

<seaborn.axisgrid.PairGrid at 0x7e5b42cdfbe0>



4. Perform descriptive statistics on the dataset.

```
df.describe()
```

	id	Date	number_of_bedrooms	number_of_bathrooms	living_area	lot_area	...
count	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.000000	1.462000e+04	1.462000e+04
mean	6.762821e+09	42604.538646	3.379343	2.129583	2098.262996	1.509328e+04	1.509328e+04
std	6.237575e+03	67.347991	0.938719	0.769934	928.275721	3.791962e+04	3.791962e+04
min	6.762810e+09	42491.000000	1.000000	0.500000	370.000000	5.200000e+02	5.200000e+02
25%	6.762815e+09	42546.000000	3.000000	1.750000	1440.000000	5.010750e+03	5.010750e+03
50%	6.762821e+09	42600.000000	3.000000	2.250000	1930.000000	7.620000e+03	7.620000e+03
75%	6.762826e+09	42662.000000	4.000000	2.500000	2570.000000	1.080000e+04	1.080000e+04
max	6.762832e+09	42734.000000	33.000000	8.000000	13540.000000	1.074218e+06	1.074218e+06

8 rows × 23 columns

5. Handle the Missing values

```
df.isnull().any()
```

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
Latitude                         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
```

Since all the values are false for the null check function, there are no null values.

✓ 0s completed at 11:33 PM

