

# Joshua\_Jose

September 6, 2023

## ASSIGNMENT 2

### 1. Download the dataset:

### 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
```

```
[ ]:          id  Date  number_of_bedrooms  number_of_bathrooms  \
0    6762810145  42491                  5                  2.50
1    6762810635  42491                  4                  2.50
2    6762810998  42491                  5                  2.75
3    6762812605  42491                  4                  2.50
4    6762812919  42491                  3                  2.00
...
14615  6762830250  42734                  2                  1.50
14616  6762830339  42734                  3                  2.00
14617  6762830618  42734                  2                  1.00
14618  6762830709  42734                  4                  1.00
14619  6762831463  42734                  3                  1.00
```

```
          living_area  lot_area  number_of_floors  waterfront_present  \
0            3650      9050          2.0                  0
1            2920      4000          1.5                  0
2            2910      9480          1.5                  0
3            3310     42998          2.0                  0
4            2710      4500          1.5                  0
...
14615         1556     20000          1.0                  0
14616         1680      7000          1.5                  0
14617         1070      6120          1.0                  0
14618         1030      6621          1.0                  0
14619          900      4770          1.0                  0
```

	number_of_views	condition_of_the_house	...	Built_Year	\
0	4		5	...	1921
1	0		5	...	1909
2	0		3	...	1939
3	0		3	...	2001
4	0		4	...	1929
...	...	...	...	...	...
14615	0		4	...	1957
14616	0		4	...	1968
14617	0		3	...	1962
14618	0		4	...	1955
14619	0		3	...	1969
	Renovation_Year	Postal_Code	Lattitude	Longitude	living_area_renov \
0	0	122003	52.8645	-114.557	2880
1	0	122004	52.8878	-114.470	2470
2	0	122004	52.8852	-114.468	2940
3	0	122005	52.9532	-114.321	3350
4	0	122006	52.9047	-114.485	2060
...	...	...	...	...	...
14615	0	122066	52.6191	-114.472	2250
14616	0	122072	52.5075	-114.393	1540
14617	0	122056	52.7289	-114.507	1130
14618	0	122042	52.7157	-114.411	1420
14619	2009	122018	52.5338	-114.552	900
	lot_area_renov	Number_of_schools_nearby	Distance_from_the_airport	...	\
0	5400		2		58
1	4000		2		51
2	6600		1		53
3	42847		3		76
4	4500		1		51
...	...	...	...	...	...
14615	17286		3		76
14616	7480		3		59
14617	6120		2		64
14618	6631		3		54
14619	3480		2		55
	Price				
0	2380000				
1	1400000				
2	1200000				
3	838000				
4	805000				
...	...				

```
14615    221700
14616    219200
14617    209000
14618    205000
14619    146000
```

```
[14620 rows x 23 columns]
```

### 3. Perform the Below Visualizations.

Univariate Analysis

Bi - Variate Analysis

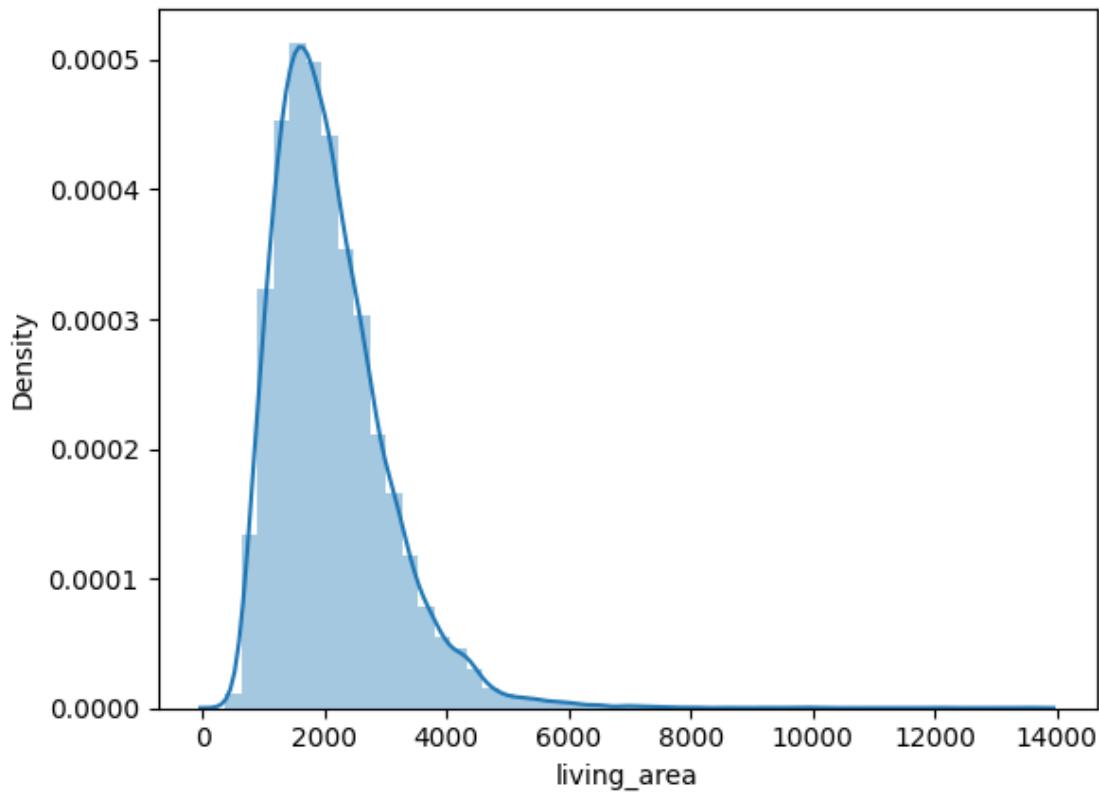
Multivariate Analysis

```
[ ]: # Univariate Analysis on "living area"
sns.distplot(df.living_area)
```

```
<ipython-input-9-0d6f0bee3138>:2: 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
```

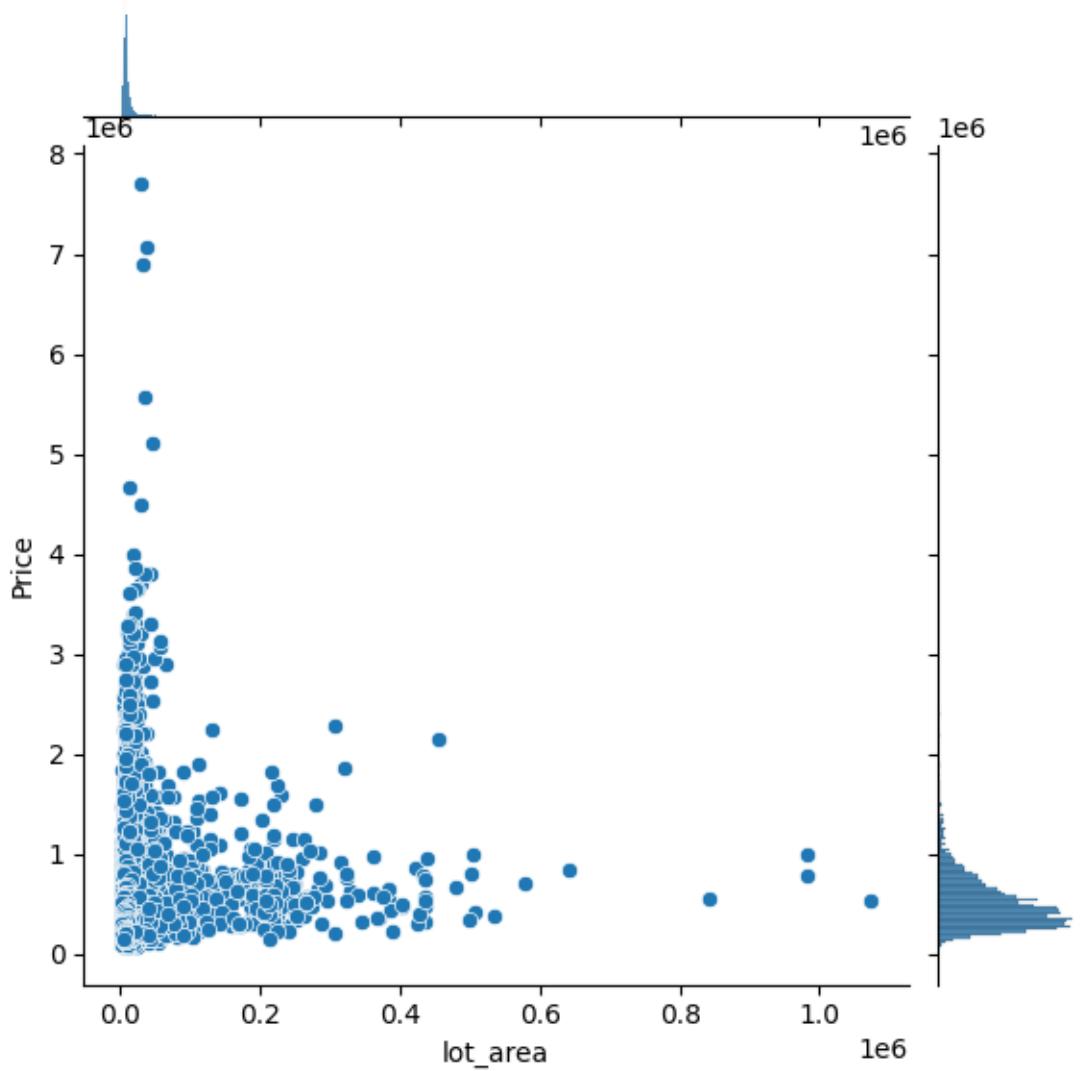
```
  sns.distplot(df.living_area)
```

```
[ ]: <Axes: xlabel='living_area', ylabel='Density'>
```



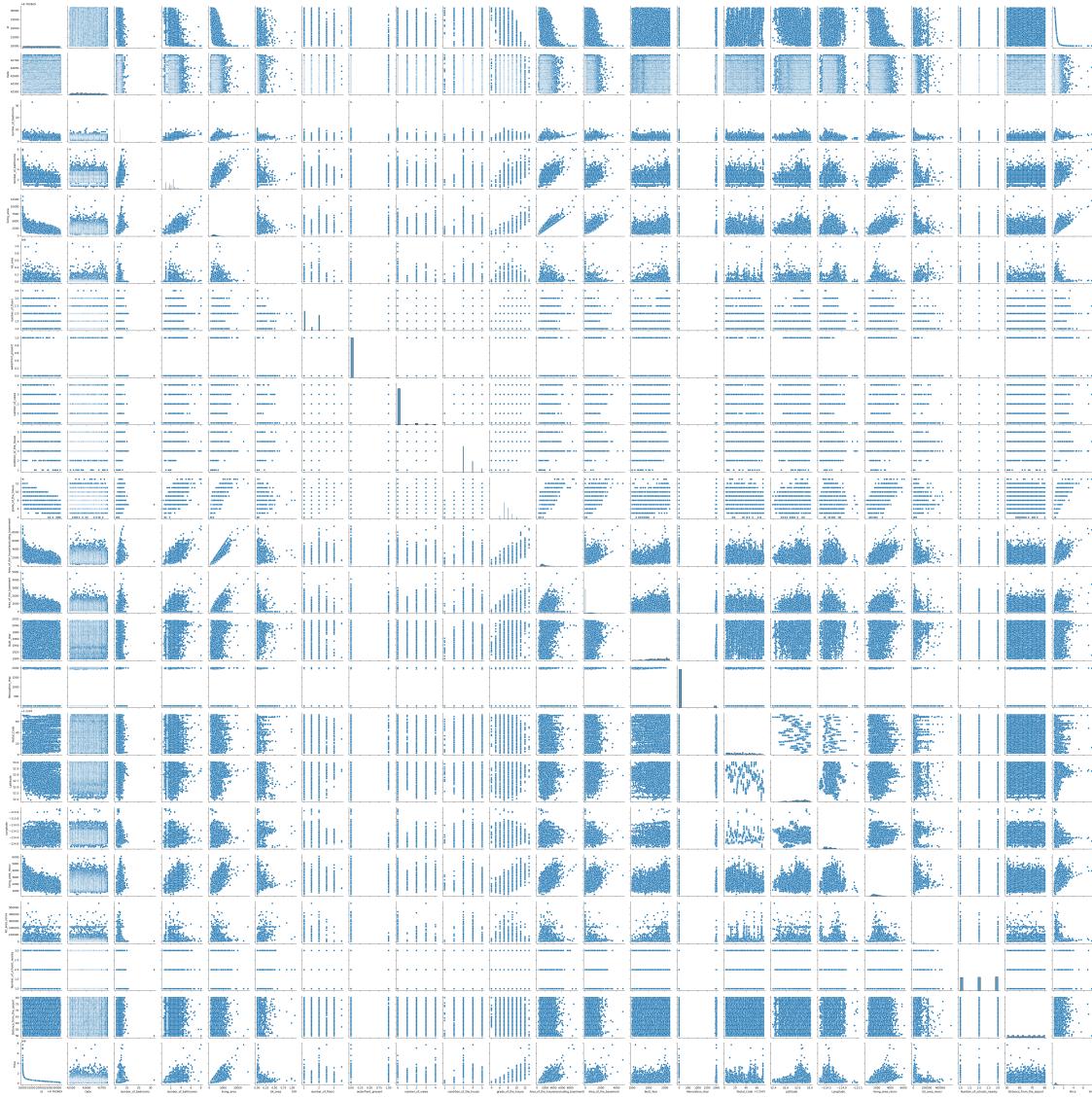
```
[ ]: # Bivariate Analysis (Comparision between 'lot_area' feature and 'Price')
sns.jointplot(x='lot_area',y='Price',data=df)
```

```
[ ]: <seaborn.axisgrid.JointGrid at 0x7caa6b0da230>
```



```
[ ]: # Multivariate analysis
sns.pairplot(df)
```

```
[ ]: <seaborn.axisgrid.PairGrid at 0x7caa657e3ca0>
```



#### 4. Perform descriptive statistics on the dataset.

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

	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]

## 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
```

There are no missing values in the dataset.