

GADDAM KRITHISHA

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

Load The Data

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

5 rows × 23 columns

```
df.shape
```

```
(14620, 23)
```

Checking Null 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
```

```
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

```

Descriptive statistics on the dataset.

```
df.info()
```

```

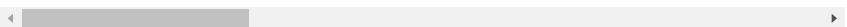
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14620 entries, 0 to 14619
Data columns (total 23 columns):
 #   Column           Non-Null Count Dtype  
 --- 
 0   id               14620 non-null  int64   
 1   Date              14620 non-null  int64   
 2   number of bedrooms 14620 non-null  int64   
 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  Latitude              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.describe()
```

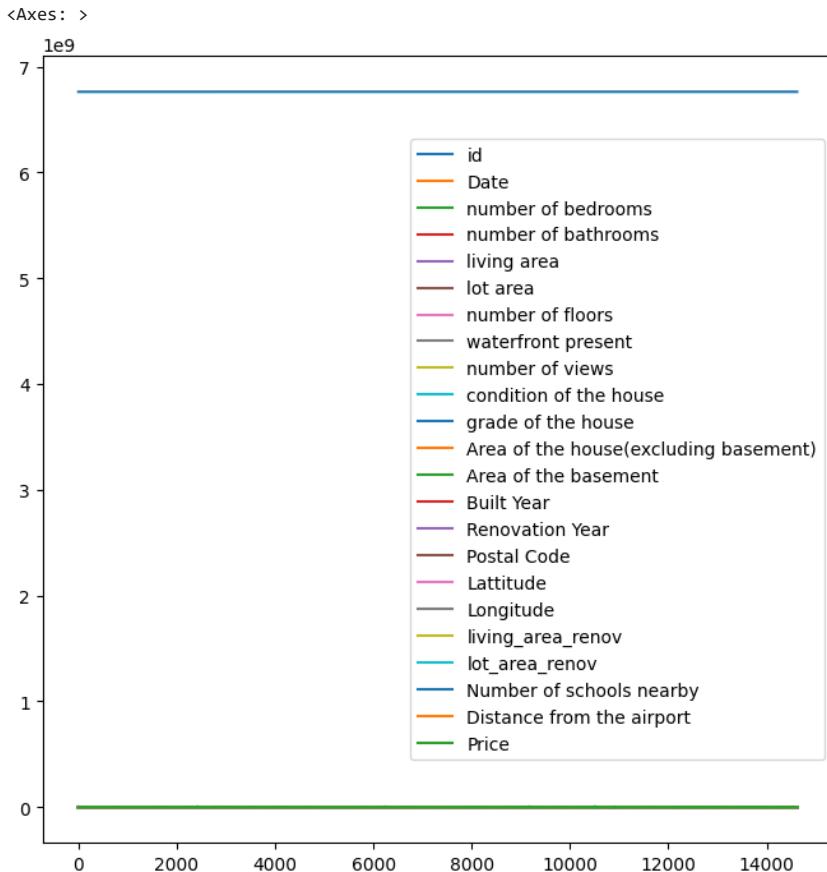
	id	Date	number of bedrooms	number of bathrooms	living area	lot
count	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.000000	1.462000
mean	6.762821e+09	42604.538646	3.379343	2.129583	2098.262996	1.50932
std	6.237575e+03	67.347991	0.938719	0.769934	928.275721	3.79196
min	6.762810e+09	42491.000000	1.000000	0.500000	370.000000	5.20000
25%	6.762815e+09	42546.000000	3.000000	1.750000	1440.000000	5.01075
50%	6.762821e+09	42600.000000	3.000000	2.250000	1930.000000	7.62000
75%	6.762826e+09	42662.000000	4.000000	2.500000	2570.000000	1.08000
max	6.762832e+09	42734.000000	33.000000	8.000000	13540.000000	1.07421

8 rows × 23 columns



Univariate Analysis

```
df.plot()
```

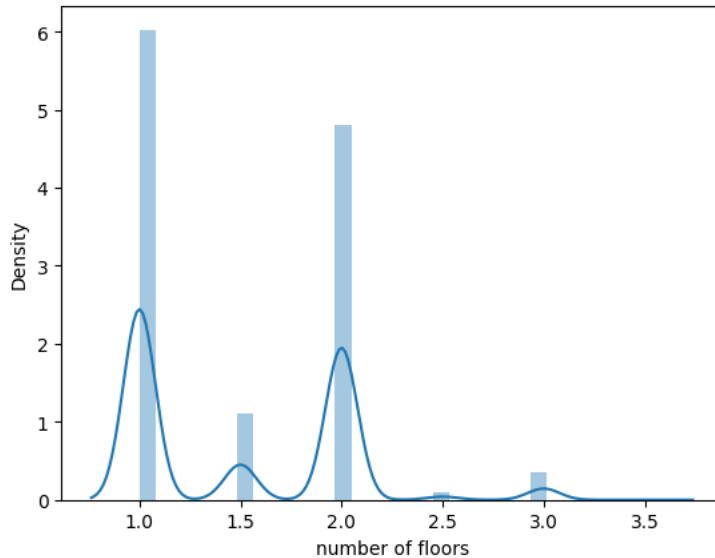


```
sns.distplot(df['number of floors'])
```

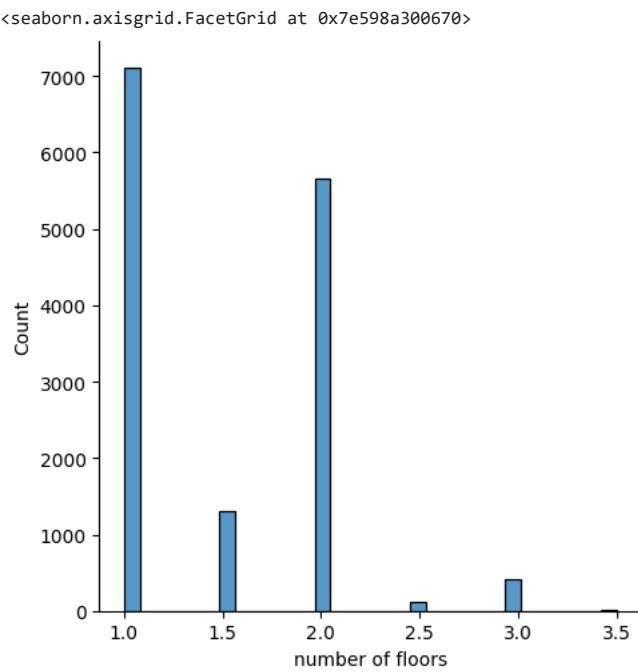
```
<ipython-input-13-4bce24ad8e73>:1: UserWarning:
  `distplot` is a deprecated function and will be removed in seaborn v0.14.0.
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
```

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

```
sns.distplot(df['number of floors'])
<Axes: xlabel='number of floors', ylabel='Density'>
```



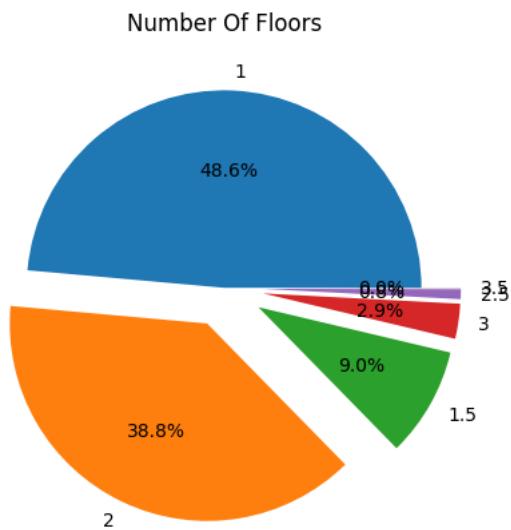
```
sns.displot(df['number of floors'])
```



```
df['number of floors'].value_counts()
```

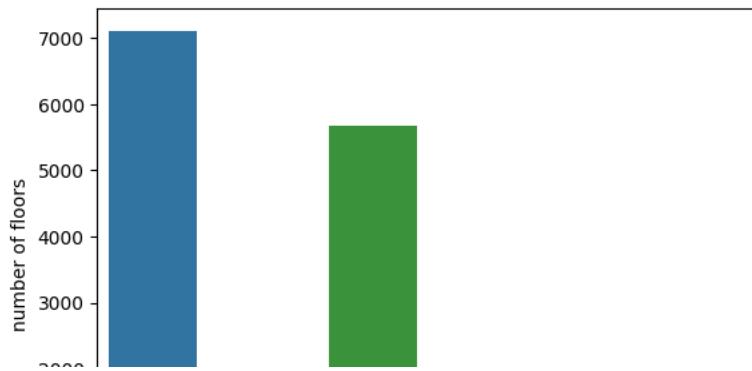
```
1.0    7103
2.0    5666
1.5    1311
3.0    418
2.5    118
3.5      4
Name: number of floors, dtype: int64
```

```
plt.pie(df['number of floors'].value_counts(),[0,0.2,0.2,0.2,0.2,0.2],labels=['1','2','1.5','3','2.5','3.5'],autopct="%1.1f%")
plt.title("Number Of Floors")
plt.show()
```



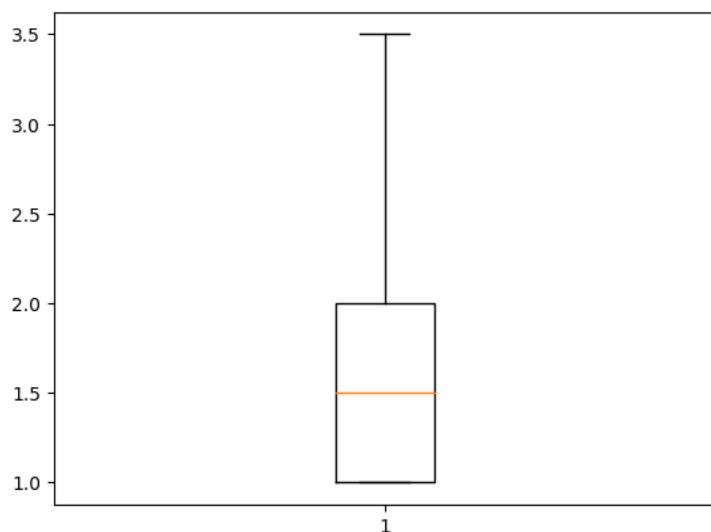
```
sns.barplot(x=df['number of floors'].value_counts().index,y=df['number of floors'].value_counts())
```

```
<Axes: ylabel='number of floors'>
```



```
plt.boxplot(df['number of floors'])
```

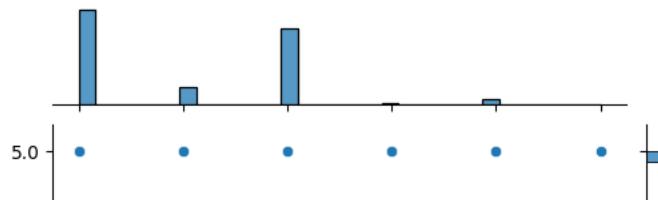
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7e5989e810f0>,
   <matplotlib.lines.Line2D at 0x7e5989e81390>],
 'caps': [<matplotlib.lines.Line2D at 0x7e5989e81630>,
   <matplotlib.lines.Line2D at 0x7e5989e818d0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7e5989e80e50>],
 'medians': [<matplotlib.lines.Line2D at 0x7e5989e81b70>],
 'fliers': [<matplotlib.lines.Line2D at 0x7e5989e81e10>],
 'means': []}
```



Bivariate Analysis

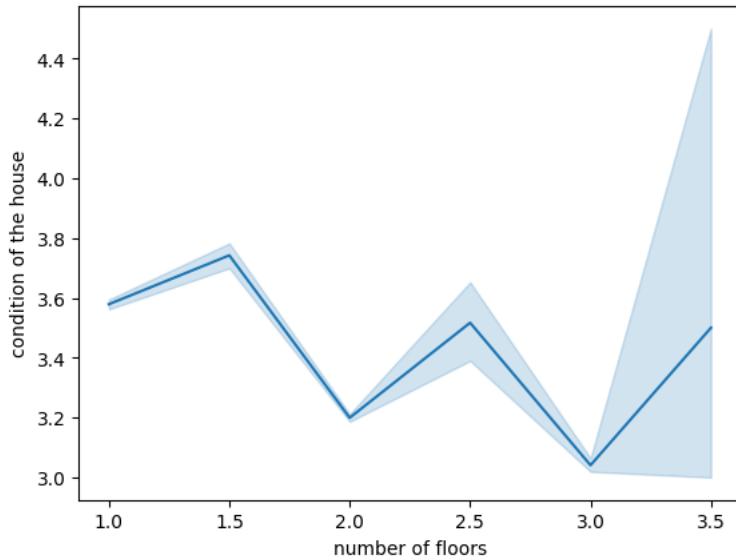
```
sns.jointplot(x = 'number of floors',y = 'condition of the house',data = df)
```

```
<seaborn.axisgrid.JointGrid at 0x7e5989a52e30>
```



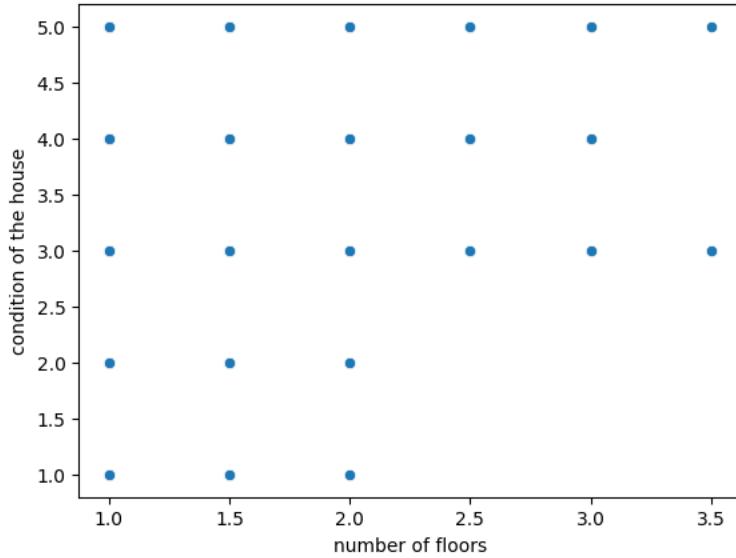
```
sns.lineplot(x=df['number of floors'],y=df['condition of the house'])
```

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



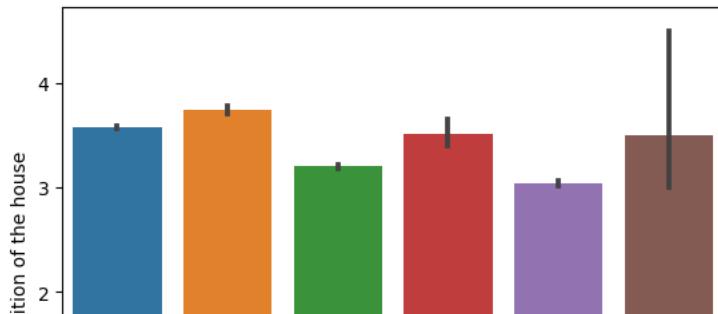
```
sns.scatterplot(x=df['number of floors'],y=df['condition of the house'])
```

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



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

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



Multivariate Analysis



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

```
sns.heatmap(df.corr())
```

<Axes: >

