NAME: N ANIRUDDHAN REG NO : 21BRS1682

AI-ML ASSIGNMENT WEEK 2

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

1. download the dataset
2. 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	..
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

```
df.columns
```

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

3. Perform the below visualisations

```
df.info()
```

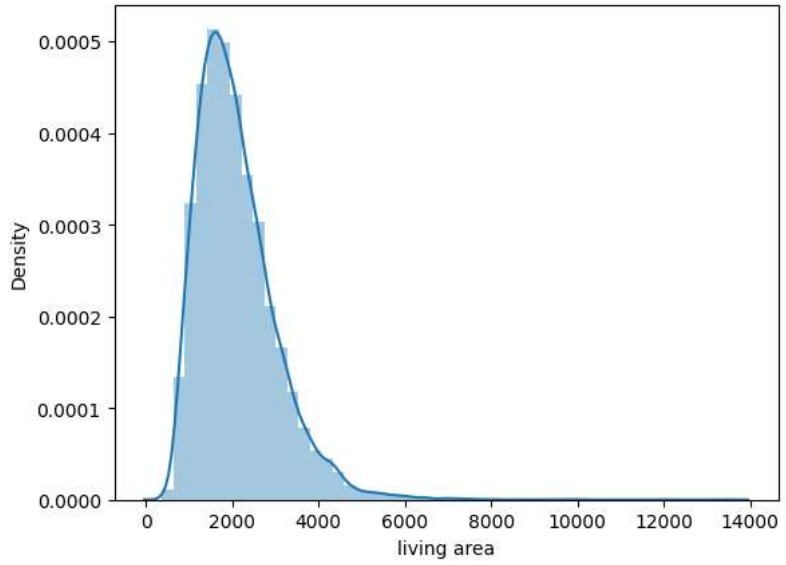
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14620 entries, 0 to 14619
Data columns (total 23 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   id                                     14620 non-null  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  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
```

UNIVARIATE ANALYSIS

```
sns.distplot(df['living area'])
#the graph is almost like a bell shape but it is right skewed
```

```
<ipython-input-9-30603c25a5d7>: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'>
```



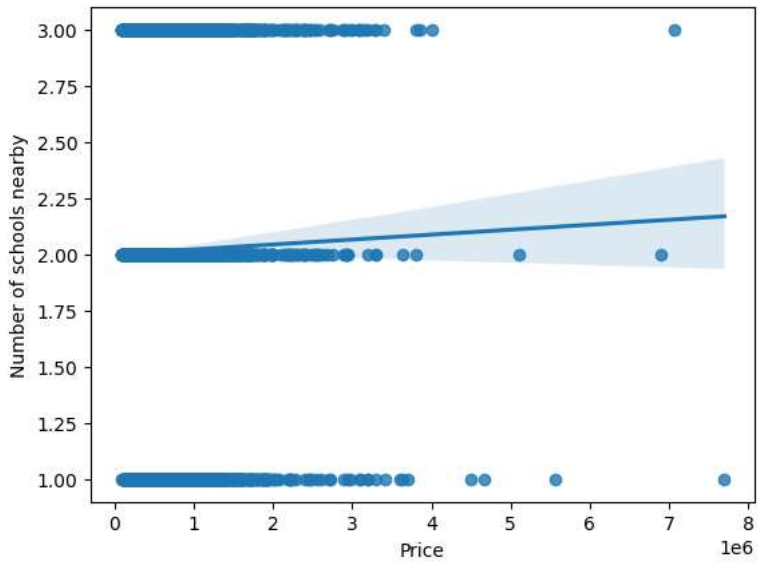

```
#visualise another variable price
sns.distplot(df['Price'])
```

```
<ipython-input-10-5ac5c1f4bc27>: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['Price'])
```

BIVARIATE ANALYSIS

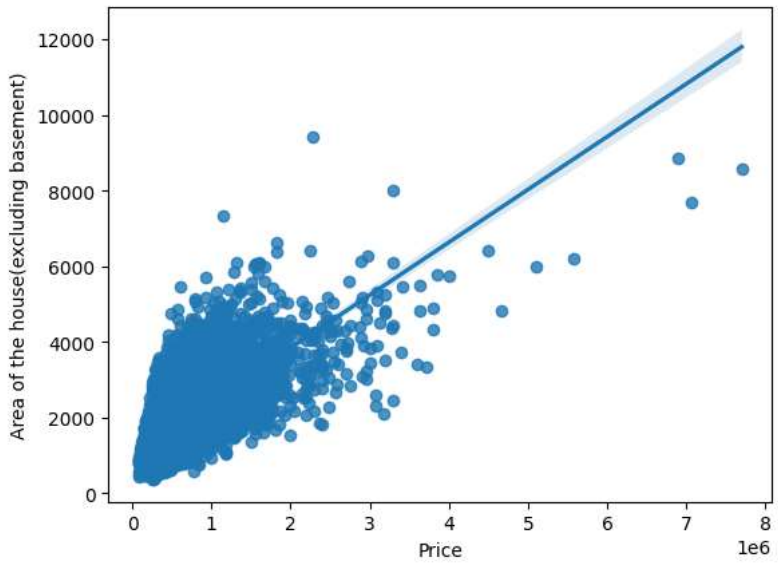
```
#no of schools vs price
sns.regplot(x='Price',y='Number of schools nearby',data=df)
#we observe the regression line slightly increase so we conclude price increases slightly and not too much
```

```
<Axes: xlabel='Price', ylabel='Number of schools nearby'>
```




```
sns.regplot(x='Price',y='Area of the house(excluding basement)', data=df)
```

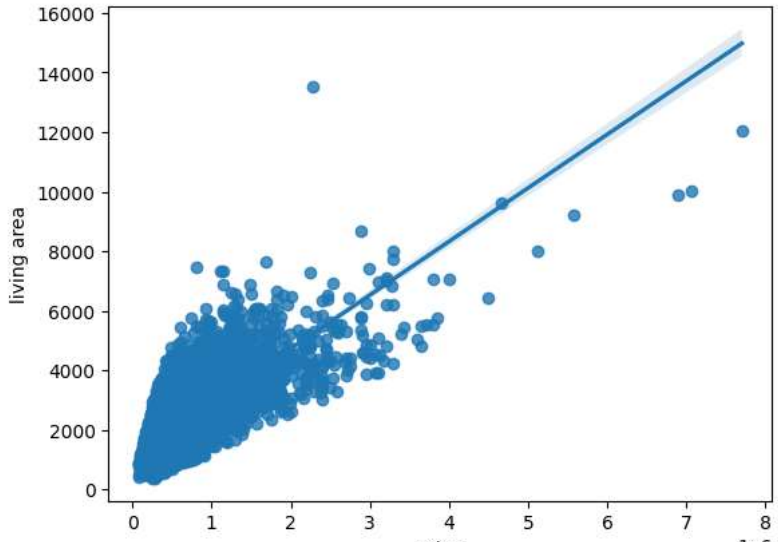
```
<Axes: xlabel='Price', ylabel='Area of the house(excluding basement)'>
```



we can conclude as area of the house increases the price increases

```
sns.regplot(x='Price',y='living area', data=df)
```

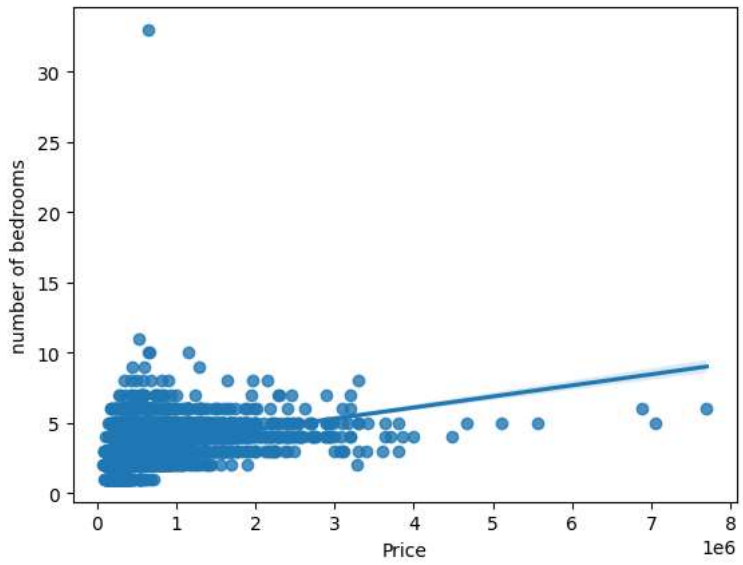
```
<Axes: xlabel='Price', ylabel='living area'>
```



steady increase which means As price increases for living area increases

```
sns.regplot(x='Price',y='number of bedrooms',data=df)
```

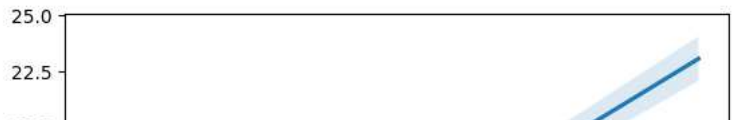
```
<Axes: xlabel='Price', ylabel='number of bedrooms'>
```



we could see no of bedrooms and price is stagnant in a particular area even though there is slight increase in the slope.

```
sns.regplot(x='Price',y='grade of the house',data=df)
```

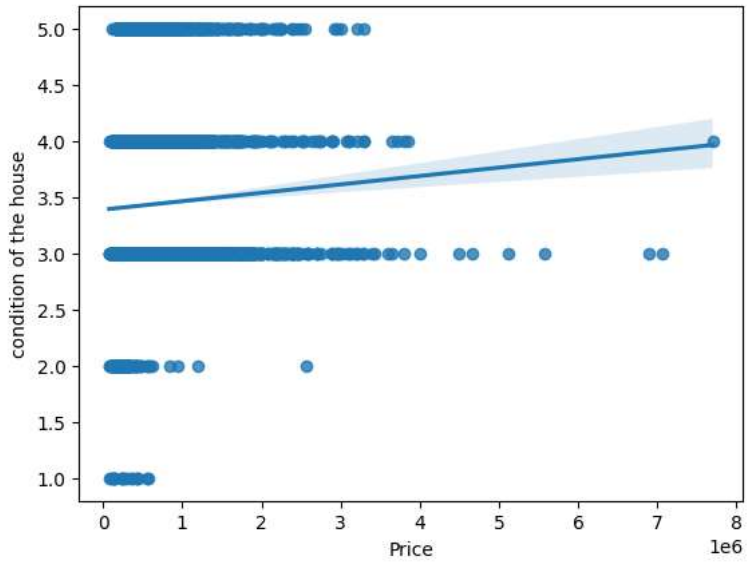
```
<Axes: xlabel='Price', ylabel='grade of the house'>
```



in Price vs grade of the house, Price and grade relation is stagnant even though there a steady increase in the regresion line

```
sns.regplot(x='Price',y='condition of the house',data=df)
```

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



There is not much relation to understand between price and condition of the house

3)MultiVariate Analysis

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

4. descriptive Statistics

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

8 rows × 23 columns

5.Handlinh 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
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()
```

```
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
```