assignment2

September 6, 2023

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[3]:
```
df = pd.read_csv('/content/House Price India.csv')
df.head(20)
```

[3]:

	id	Date	number_of_bedrooms	number_of_bathrooms	living_area	lot_area	number_of_floors	waterfront_present	number_of_views
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0
5	6762813105	42491	3	2.50	2600	4750	1.0	0	0
6	6762813157	42491	5	3.25	3660				
7	6762813599	42491	3	1.75	2240				
8	6762813600	42491	3	2.50	2390				
9	6762814461	42491	4	2.25	2200				
10	6762814787	42491	5	2.50	2820				
11	6762815225	42491	4	2.00	1820				
12	6762815461	42491	4	2.00	1520				
13	6762815654	42491	4	2.75	2710				
14	6762815737	42491	3	2.25	1750				
15	6762815748	42491	4	2.50	2820				
16	6762815749	42491	4	3.25	2730				
17	6762815986	42491	3	1.75	2360				
18	6762816059	42491	4	2.50	2730				
19	6762816236	42491	3	2.50	3240				

6	11995	2.0	0	2
7	10578	2.0	0	0
8	6550	1.0	0	2
9	11250	1.5	0	0
10	67518	2.0	0	0
11	5000	1.5	0	1
12	6200	1.5	0	0
13	37277	2.0	0	0
14	1572	2.5	0	0
15	8408	2.0	0	0
16	54014	1.0	0	0
17	7291	1.0	0	0
18	12261	2.0	0	0
19	33151	2.0	0	2

	condition_of_the_house	…	Built_Year	Renovation_Year	Postal_Code	\
0	5	…	1921	0	122003	
1	5	…	1909	0	122004	
2	3	…	1939	0	122004	
3	3	…	2001	0	122005	
4	4	…	1929	0	122006	
5	4	…	1951	0	122007	
6	3	…	2006	0	122008	
7	5	…	1923	0	122006	
8	4	…	1955	0	122009	
9	5	…	1920	0	122010	
10	3	…	1979	0	122011	
11	3	…	1945	0	122006	
12	3	…	1945	0	122006	
13	3	…	2000	0	122012	
14	3	…	2005	0	122013	
15	3	…	2014	0	122014	
16	3	…	2007	0	122015	
17	4	…	1948	0	122016	
18	3	…	1991	0	122017	
19	3	…	1995	0	122018	

	Lattitude	Longitude	living_area_renov	lot_area_renov	\
0	52.8645	-114.557	2880	5400	
1	52.8878	-114.470	2470	4000	
2	52.8852	-114.468	2940	6600	
3	52.9532	-114.321	3350	42847	
4	52.9047	-114.485	2060	4500	
5	52.9133	-114.590	2380	4750	
6	52.7637	-114.050	3320	11241	
7	52.9254	-114.482	1570	10578	
8	52.8014	-114.598	2010	6550	

```
9     52.9145   -114.391              2320           10814
10    52.8094   -114.215              2820           48351
11    52.9115   -114.459              2060            5000
12    52.9080   -114.459              1910            6200
13    52.6934   -114.177              2390           39299
14    52.8798   -114.511              2410            3050
15    52.9838   -114.515              1300            8408
16    52.7433   -114.300              2730          111274
17    52.7574   -114.574              1860            5499
18    52.9719   -114.395              2730           10872
19    52.5556   -114.568              4050           24967

    Number_of_schools_nearby  Distance_from_the_airport    Price
0                          2                         58  2380000
1                          2                         51  1400000
2                          1                         53  1200000
3                          3                         76   838000
4                          1                         51   805000
5                          1                         67   790000
6                          3                         72   785000
7                          3                         71   750000
8                          1                         73   750000
9                          2                         53   698000
10                         2                         51   675000
11                         2                         69   650000
12                         3                         80   640000
13                         1                         74   630000
14                         3                         51   626000
15                         3                         55   625000
16                         1                         70   625000
17                         2                         75   615000
18                         2                         70   612500
19                         1                         71   604000

[20 rows x 23 columns]
```

[4]:
```
df.shape
```

[4]:
```
(14620, 23)
```

##Visualization

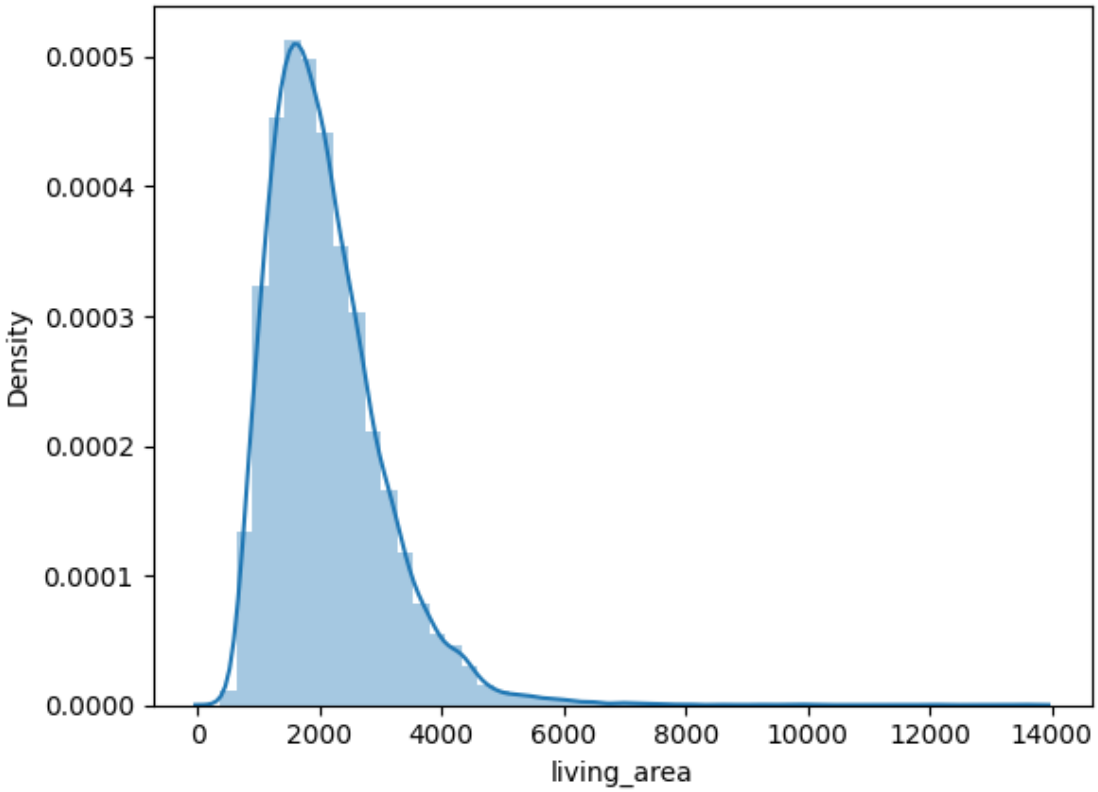
##univariate Analysis

[5]:
```
sns.distplot(df.living_area)
```

```
<ipython-input-5-2fe1fc3439c6>:1: UserWarning:
```

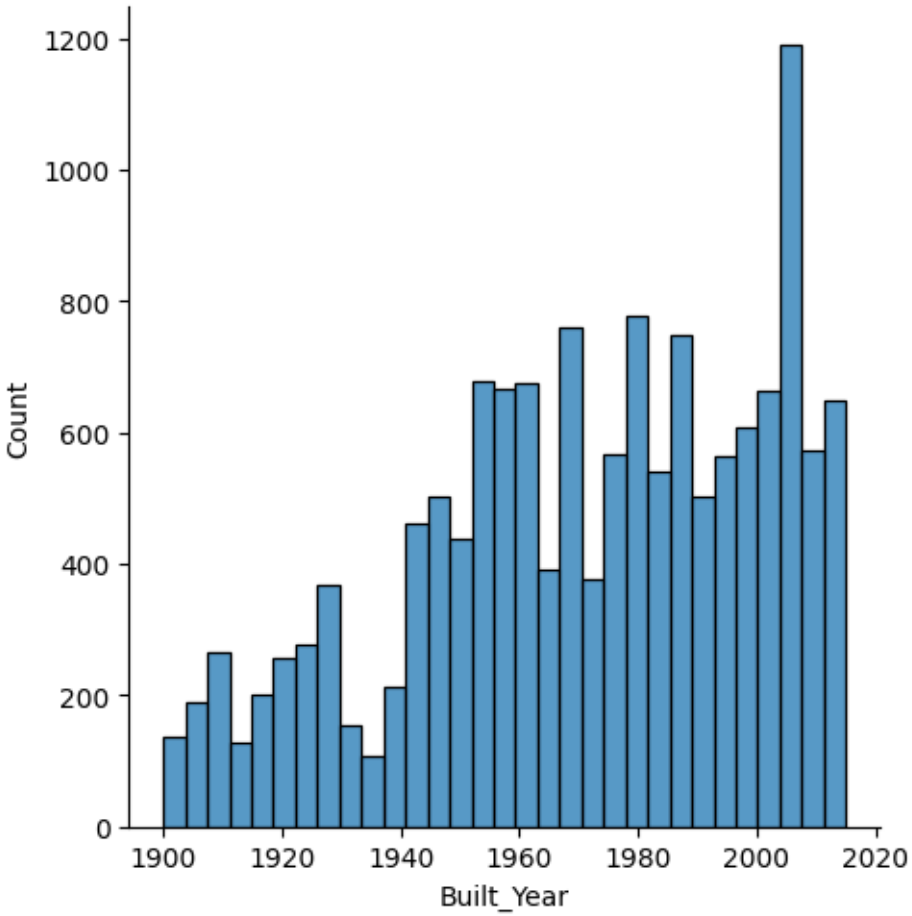
```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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)
```

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



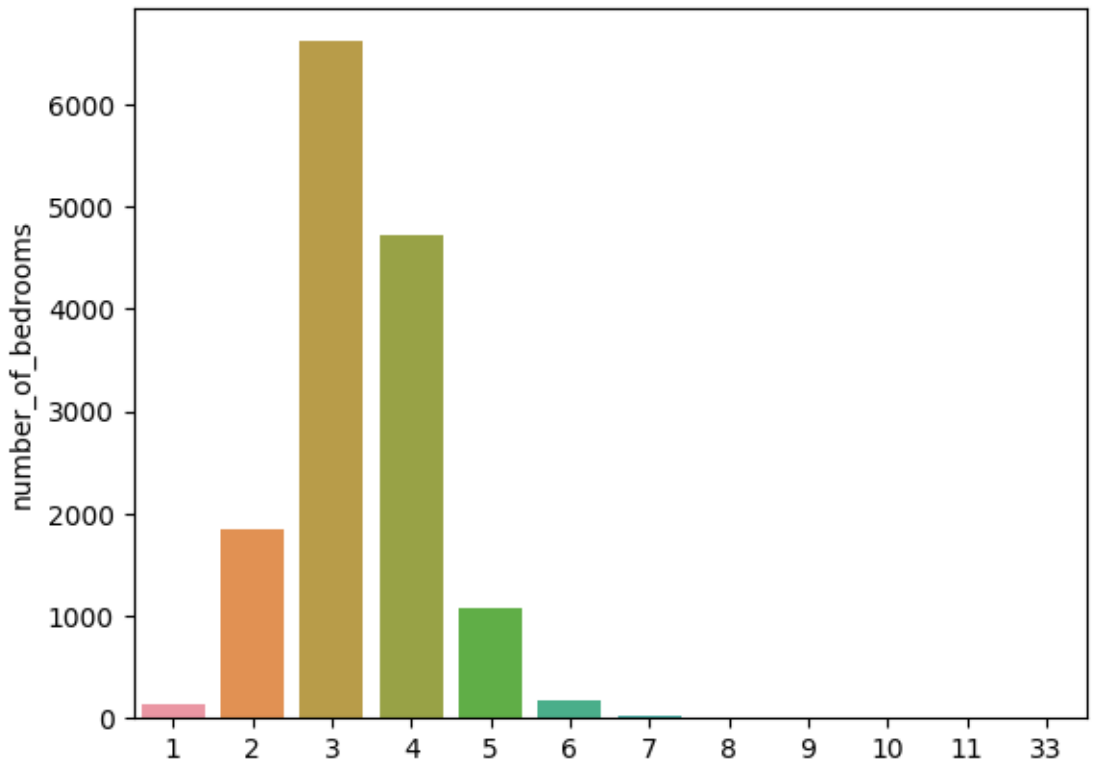
[6]:
```
sns.displot(df.Built_Year)
```

[6]: <seaborn.axisgrid.FacetGrid at 0x7f7a919d73d0>



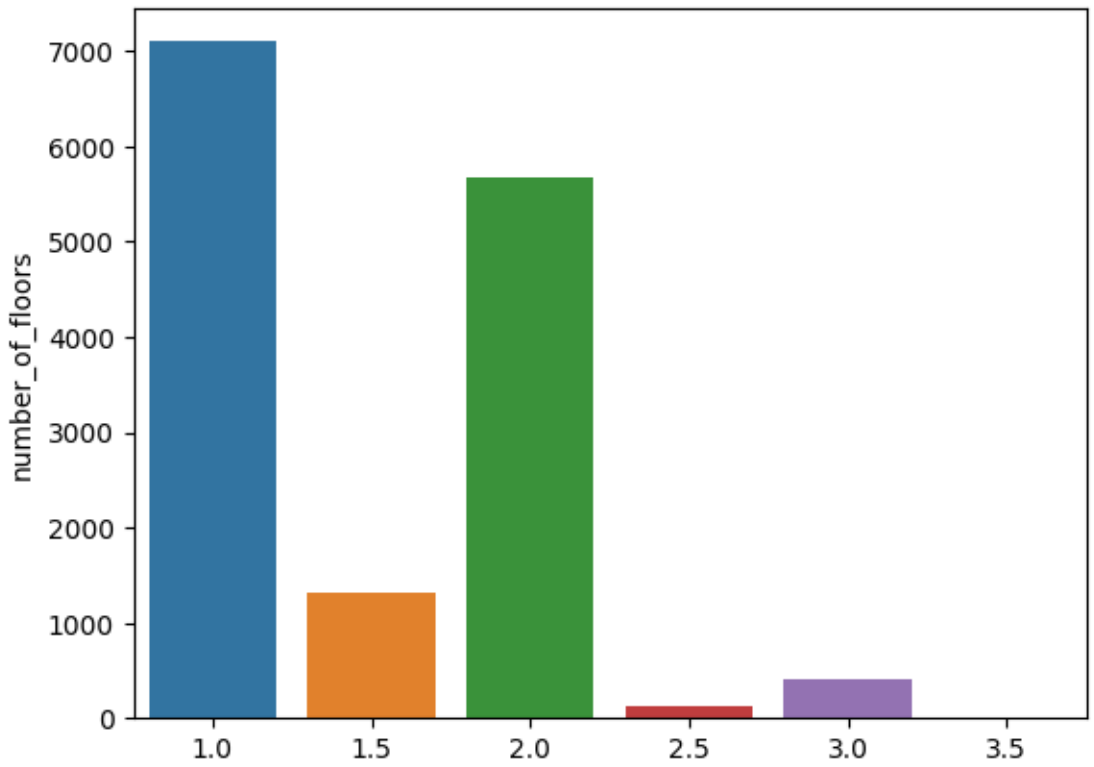
[7]:
```
sns.barplot(x=df.number_of_bedrooms        .value_counts().index,y=df.
↪number_of_bedrooms        .value_counts())
```

[7]: <Axes: ylabel='number_of_bedrooms'>



[8]:
```
sns.barplot(x=df.number_of_floors        .value_counts().index,y=df.
↪number_of_floors        .value_counts())
```

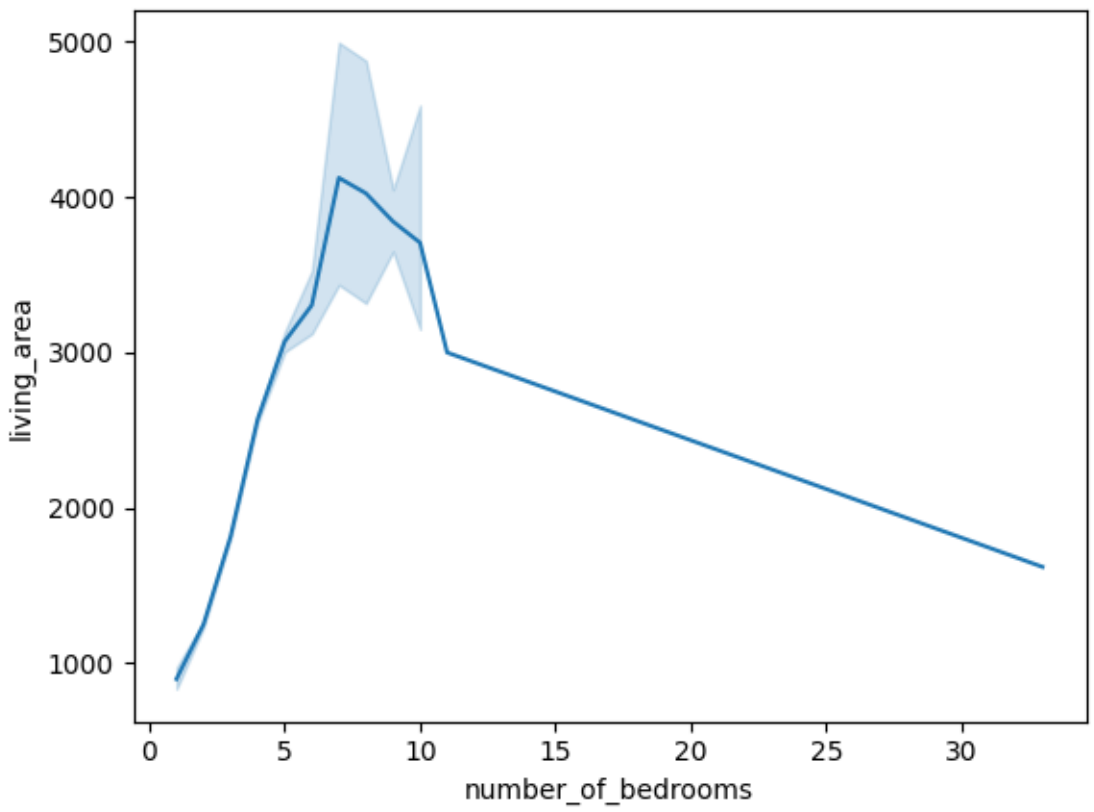
[8]:
```
<Axes: ylabel='number_of_floors'>
```



##Bi-variate Analysis

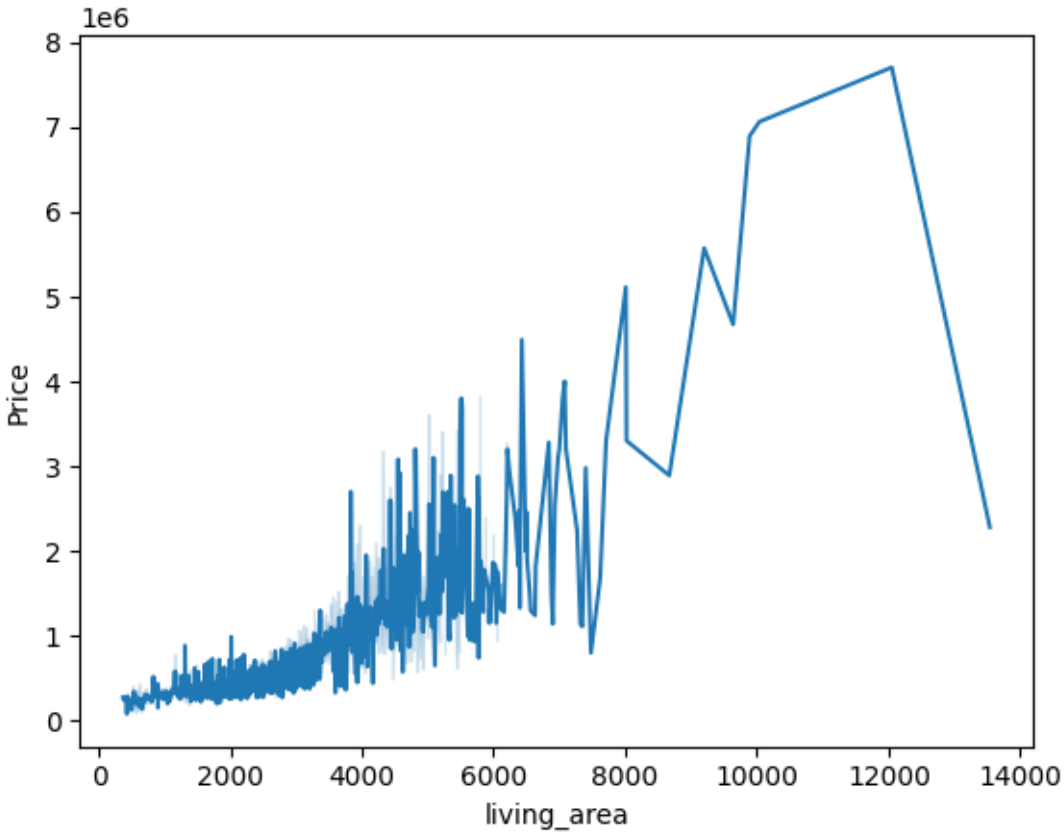
[9]:
```
sns.lineplot(x=df.number_of_bedrooms,y=df.living_area)
```

[9]:
```
<Axes: xlabel='number_of_bedrooms', ylabel='living_area'>
```



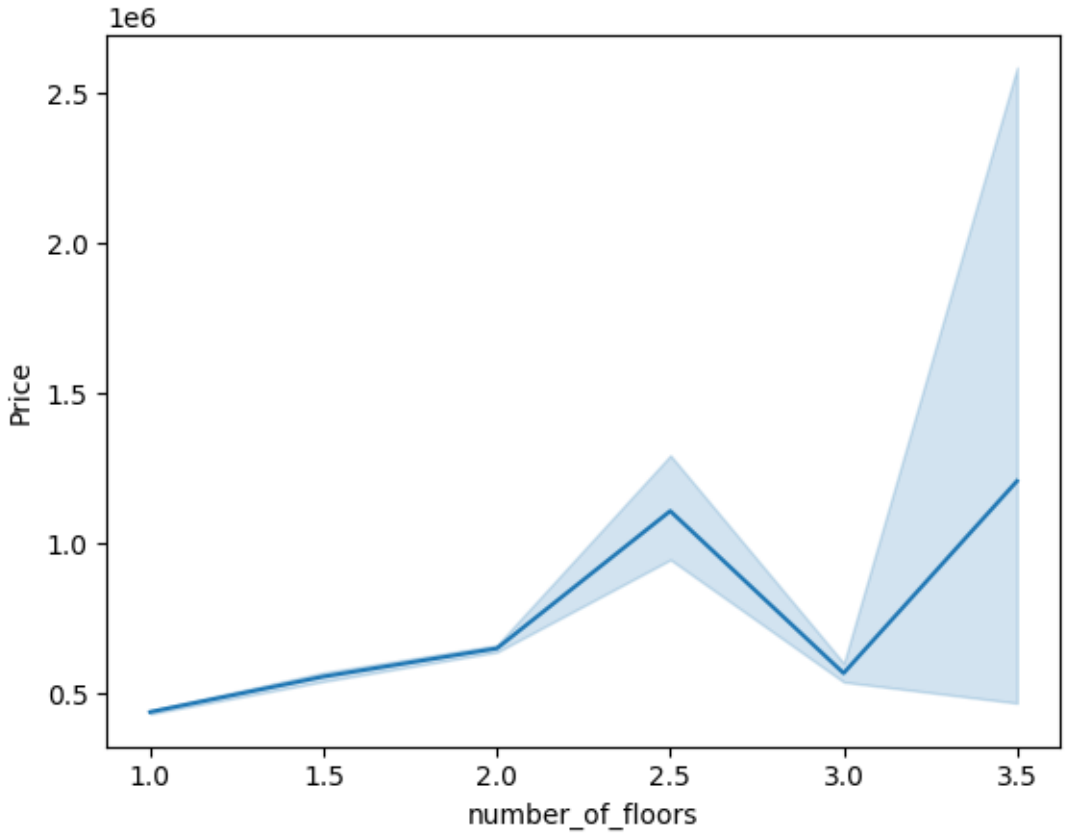
[10]:
```
sns.lineplot(x=df.living_area,y=df.Price)
```

[10]:
```
<Axes: xlabel='living_area', ylabel='Price'>
```



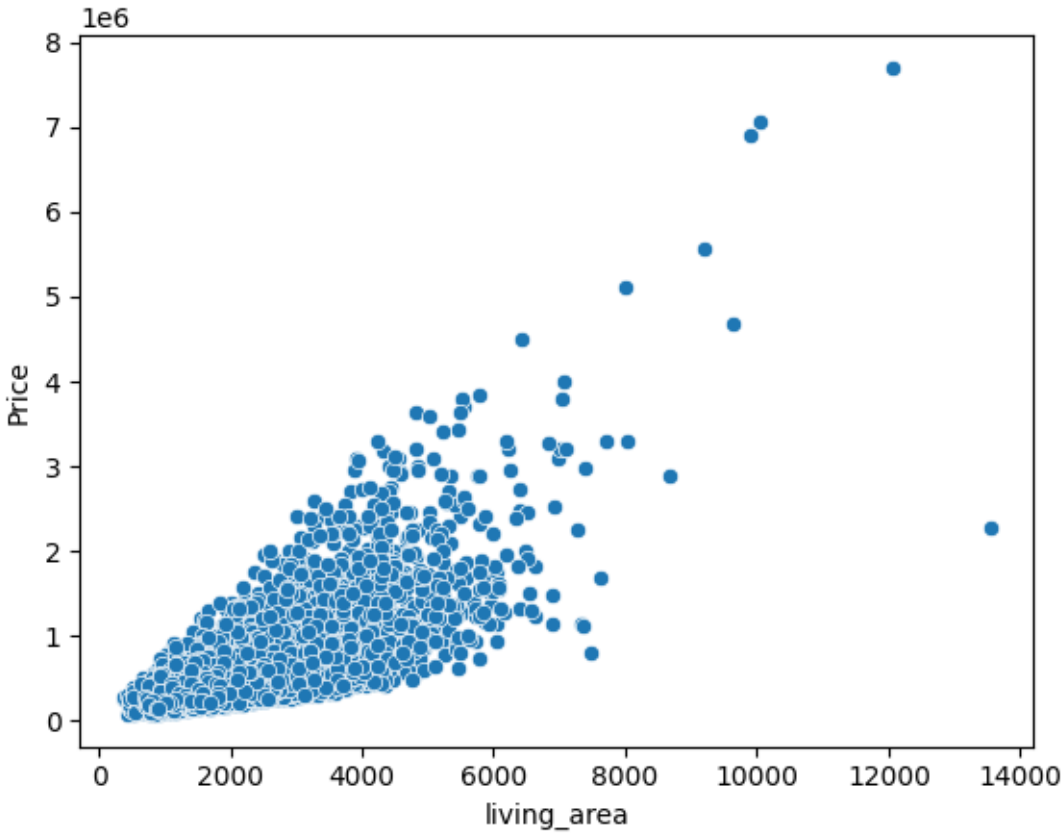
[11]:
```
sns.lineplot(x=df.number_of_floors,y=df.Price)
```

[11]:
```
<Axes: xlabel='number_of_floors', ylabel='Price'>
```



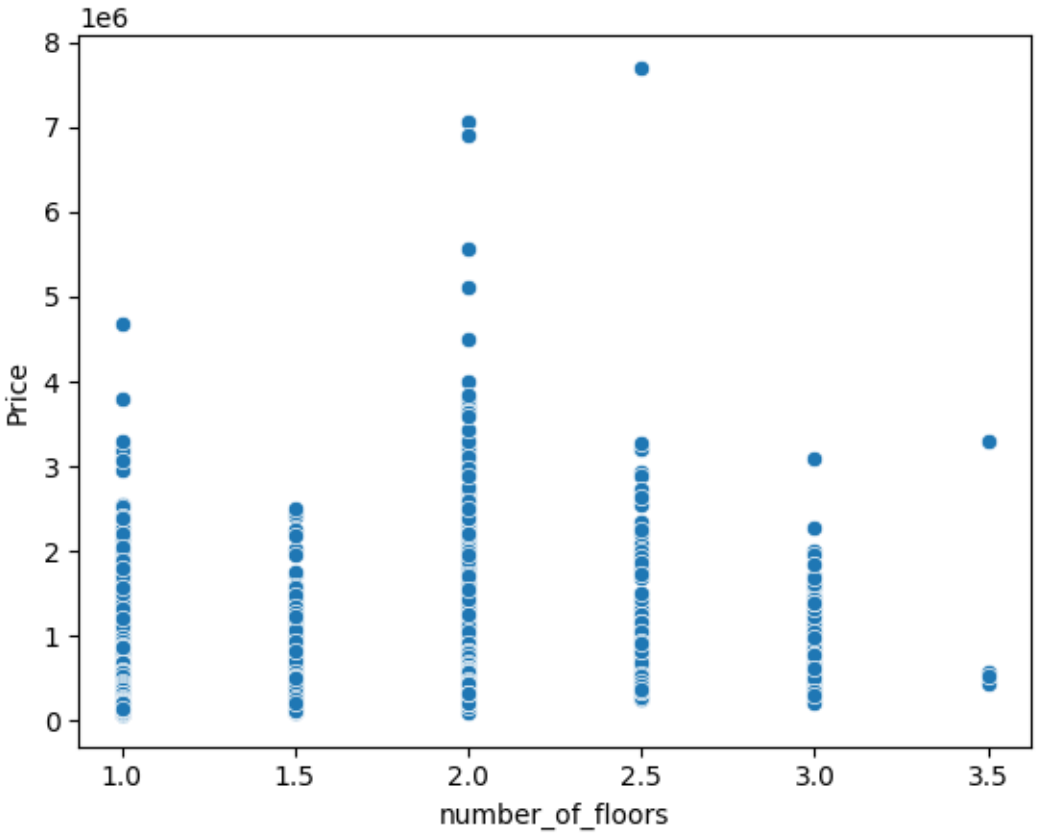
[12]:
```
sns.scatterplot(x=df.living_area,y=df.Price)
```

[12]:
```
<Axes: xlabel='living_area', ylabel='Price'>
```



[13]:
```
sns.scatterplot(x=df.number_of_floors,y=df.Price)
```

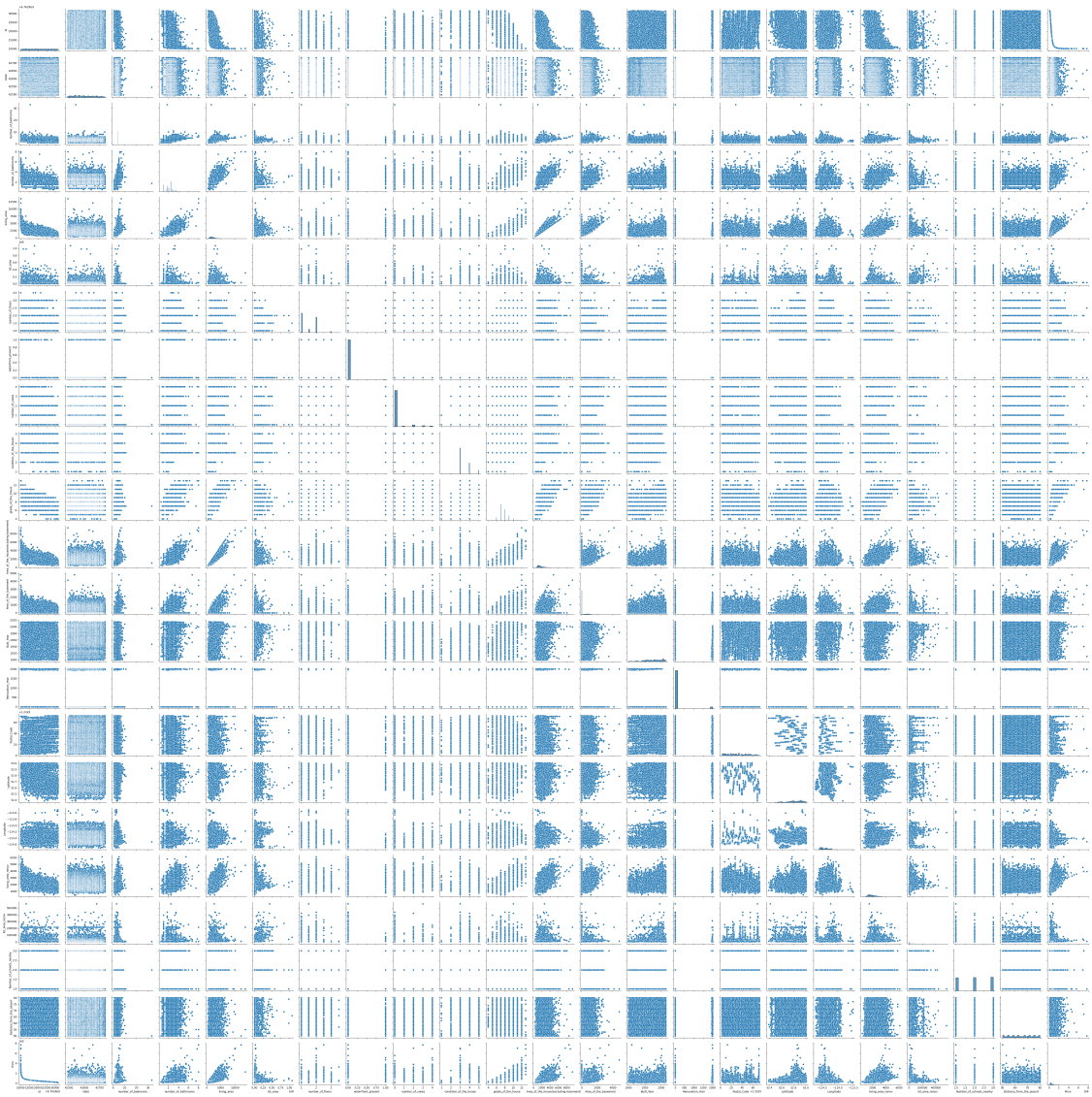
[13]: <Axes: xlabel='number_of_floors', ylabel='Price'>



##Multivariate Analysis

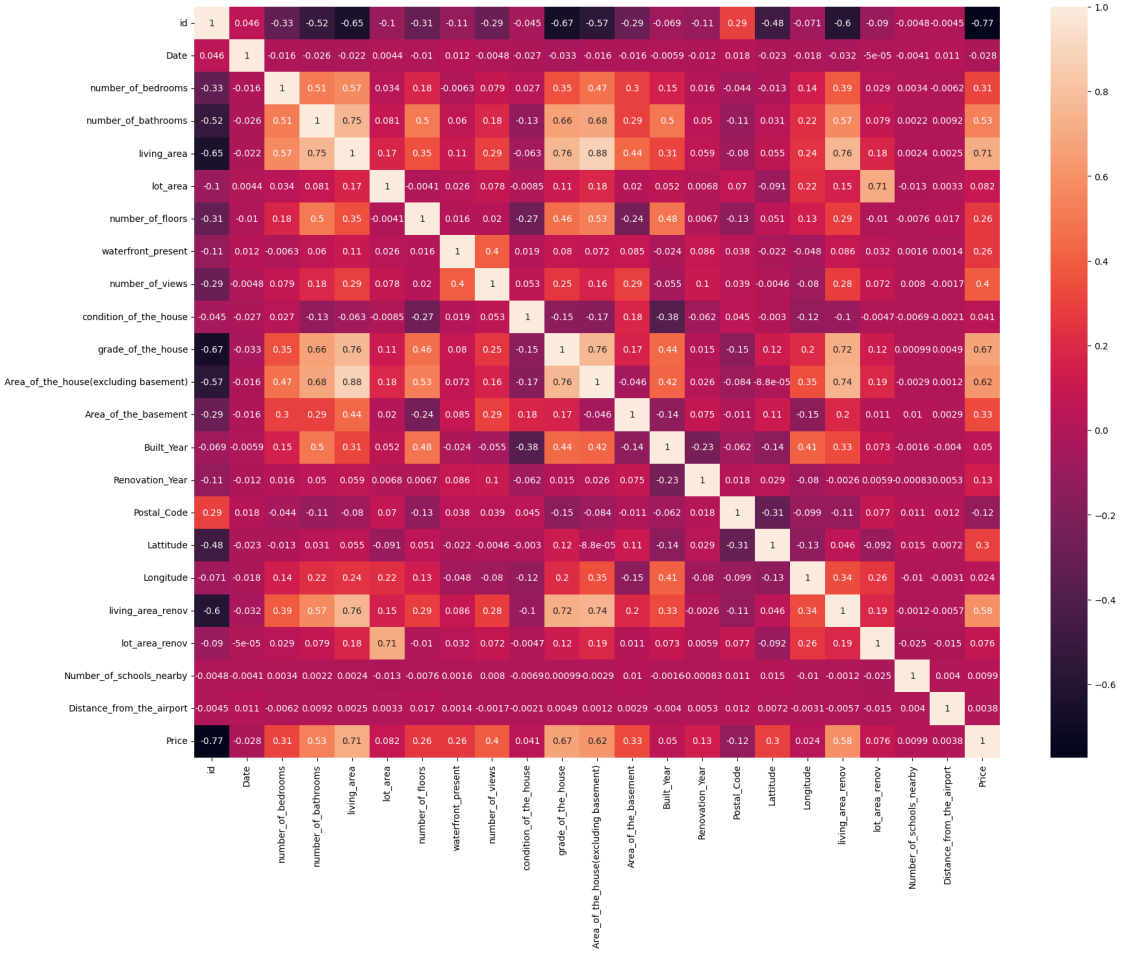
[14]:
```
sns.pairplot(df)
```

[14]:
```
<seaborn.axisgrid.PairGrid at 0x7f7a57693f40>
```



[17]:
```
plt.figure(figsize = (20,15))
sns.heatmap(df.corr(),annot =True,)
```

[17]: <Axes: >



4.Perform descriptive statistics on the dataset

[20]:
```
df.describe()
```

[20]:

	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.

[22]:
```
df.isnull().any()
```

[22]:
```
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
```

[23]:
```
df.isnull().sum()
```

[23]:
```
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
```

There are no null values in the above given dataset

[25]:
```
df = df.fillna(df.mean()) #this is used to fill the null vaulues but in the␣
↪above dataset there are no null values
```