NAME:SUCHANTH B

```
#import pandas required

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


df = pd.read_csv("/content/Houseprice Dataset.zip")
df.head()
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	co
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 rows × 23 columns

```
df.head()
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	co
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 rows × 23 columns

```
df.tail()
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condition of the house	...	Built Year	Renovation Year	Po
14615	6762830250	42734	2	1.5	1556	20000	1.0	0	0	4	...	1957	0	12
14616	6762830339	42734	3	2.0	1680	7000	1.5	0	0	4	...	1968	0	12
14617	6762830618	42734	2	1.0	1070	6120	1.0	0	0	3	...	1962	0	12
14618	6762830709	42734	4	1.0	1030	6621	1.0	0	0	4	...	1955	0	12
14619	6762831463	42734	3	1.0	900	4770	1.0	0	0	3	...	1969	2009	12

5 rows × 23 columns

```
df.shape
```

```
(14620, 23)
```

```
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
```

```
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
```

Univariate Analysis

```
df.describe()
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condi of h
count	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.000000	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.00
mean	6.762821e+09	42604.538646	3.379343	2.129583	2098.262996	1.509328e+04	1.502360	0.007661	0.233105	3.43
std	6.237575e+03	67.347991	0.938719	0.769934	928.275721	3.791962e+04	0.540239	0.087193	0.766259	0.66
min	6.762810e+09	42491.000000	1.000000	0.500000	370.000000	5.200000e+02	1.000000	0.000000	0.000000	1.00
25%	6.762815e+09	42546.000000	3.000000	1.750000	1440.000000	5.010750e+03	1.000000	0.000000	0.000000	3.00
50%	6.762821e+09	42600.000000	3.000000	2.250000	1930.000000	7.620000e+03	1.500000	0.000000	0.000000	3.00
75%	6.762826e+09	42662.000000	4.000000	2.500000	2570.000000	1.080000e+04	2.000000	0.000000	0.000000	4.00
max	6.762832e+09	42734.000000	33.000000	8.000000	13540.000000	1.074218e+06	3.500000	1.000000	4.000000	5.00

8 rows × 23 columns

```
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	...	Built Year	Renovation Year	Post Co
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5	...	1921	0	1220
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5	...	1909	0	1220
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3	...	1939	0	1220
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3	...	2001	0	1220
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	4	...	1929	0	1220

5 rows × 23 columns

```
df.living_area_renov.nunique()
```

```
665
```

```
df.living_area_renov.unique()
```

```
array([2880, 2470, 2940, 3350, 2060, 2380, 3320, 1570, 2010, 2320, 2820,
       1910, 2390, 2410, 1300, 2730, 1860, 4050, 2570, 2200, 2590, 2860,
       1090, 3000, 1340, 2780, 2080, 2260, 2990, 1560, 1320, 1850, 1150,
       1770, 2340, 1680, 1260, 1450, 2070, 2290, 1960, 2830, 1440, 1790,
       1160, 1480, 1100, 2280, 1590, 1410, 2310, 1750, 2130, 1400, 1380,
       1580, 3030, 1280, 1940, 1390, 2315, 2240, 2350, 2140, 4850, 1870,
       2610, 2720, 3100, 4420, 4530, 3430, 2550, 1670, 3070, 2020, 3180,
       2970, 1690, 2750, 2170, 3715, 1950, 2580, 1810, 3010, 1350, 1720,
       1800, 2840, 2330, 1060, 2160, 2030, 1880, 1520, 2500, 1290, 1470,
       1890, 1730, 2220, 1840, 2670, 1200, 1408, 1620, 1430, 1630, 1310,
       1760, 1820, 1220, 1980, 1130, 1170, 1510, 1240, 2488, 3510, 2490,
       2540, 2120, 2040, 3040, 3240, 3130, 3770, 2790, 2800, 2530, 2450,
       2520, 2770, 2000, 1780, 2210, 1420, 1660, 1970, 1270, 1460, 1500,
       1930, 1330, 1740, 1370, 2090, 1230, 2441,  840, 2360, 1650, 1490,
        900,  820, 1700, 4100, 2960, 3470, 3820, 2430, 4130, 2190, 1990,
       2250, 3200, 2850, 2560, 1640, 2870, 2510, 1180, 2600, 1540, 1250,
       1040, 1360, 1516, 2230, 2440, 2011, 1010, 1140, 1070,  910, 1326,
       3450, 2930, 2900, 3260, 2920, 2950, 3620, 1900, 1210, 3140, 2300,
       1190, 2527, 2150, 2980, 1920, 1600, 1357, 1572, 4460, 3890, 3660,
       3230, 3500, 3080, 3880, 2700, 2690, 2100, 2270, 1110, 1439,  998,
       1714, 1610, 1550, 1020, 3220, 4760, 2890, 3530, 2400, 3600, 2480,
       3170, 3640, 2370,  980, 1080, 1120, 1830,  890, 1710, 3740, 4040,
       4240, 4440, 3290, 2180, 3120,  990, 2650, 3060, 1364, 2420, 3480,
       4560, 3210, 3390, 3360, 2910,  950,  920, 1030, 1530, 3860, 4210,
       3700, 2740, 2810, 2460, 2660, 1232,  850, 3490, 3150, 1445, 2114,
       1404, 3910, 3160, 3580, 2760,  930, 3300, 5170, 4060, 3920, 3610,
       2303, 1862, 1050, 3850, 3840, 1000, 2110, 2680, 2050, 2620, 3790,
       2415, 3440, 2640, 3110, 2052, 2095, 3630, 2710, 3270, 5030, 3680,
```

```
 970, 1571, 1307, 1658, 3540, 4290, 2358, 3370, 1665, 3494, 2434,
 860,  880, 3930, 3710, 4140, 1365, 4020, 3690, 3750, 3590, 1346,
3330, 2630, 1518, 3190, 1495, 2305, 3730, 2037, 2363, 1765, 3810,
4090, 3280, 4390, 2027,  960, 2437,  770,  700, 4900, 3960, 3050,
2578, 1484, 2583, 1914, 4280, 2412, 4070, 3380, 1405, 1811, 3250,
3550, 2518, 3020, 2106, 2009, 1188, 4630, 3800, 4670, 3950, 1295,
2478,  740, 3310, 4180, 2683, 2955, 4000, 3400, 3900, 3670, 3780,
4400, 3420,  830,  460, 1256, 1494, 1098, 3720, 3560, 2028, 1459,
1584, 3340, 2496, 1934, 2456, 4470, 4170, 3980, 1798, 2376, 2594,
2214, 1768, 4550, 4010, 2554, 4950, 1277, 1156,  940, 2667, 5080,
5790, 3830, 3639, 1664, 1481, 4080, 2502, 4620, 3410, 3090, 3618,
2912, 2238, 1078, 5070, 3970, 4490, 3570, 2516,  780, 1767, 4160,
3760, 3520, 2566, 1678, 4920, 3650, 4510, 4030, 3625, 2165, 2156,
2641, 3460, 4340,  800, 4680, 4300, 2234,  760, 3990, 4640, 1746,
1569, 1696, 2815, 1309,  870, 2458, 4750, 3045, 1894, 2648, 1802,
2598, 2154, 2029, 1616, 2738, 2634, 2166, 2673, 1137, 4270, 4310,
1979, 1537, 1847, 4150, 2996, 1546, 1813, 2704, 5380, 3721, 4190,
2475,  790, 4362,  806, 4330, 2597, 1522, 1466, 1264, 2616, 1536,
4042, 4230, 2198, 2575, 4890, 3112, 1745, 1448, 2574, 2439, 1076,
 810, 4913, 2798, 2189, 1528, 3940, 2533, 2622, 5200, 2056, 1458,
1509, 2382, 1975, 4120, 4110, 4590, 4690, 2451, 1984, 2323, 1358,
5600, 2142, 3191, 1336, 4320, 4830, 4225, 2474, 3425, 2316, 2688,
2112, 3557, 5110, 1716, 2725, 2396, 1981, 4930, 3008, 1554, 1442,
1463, 4480, 1638, 3236, 1138, 2876, 3193,  750, 2424, 2901, 4540,
1303, 1919, 2049, 2077, 1381,  710, 1282, 2612, 1941, 2136, 4370,
2875, 2555, 2304, 1443, 3159, 2767, 4940, 4570, 2425, 1268, 1399,
1356, 2221,  720, 4770, 2665, 3078, 2344, 2246, 1639, 2724, 2092,
2389, 2406, 1566, 1168,  670, 2419, 2014, 2879, 2015, 3543, 2619,
1092, 1608, 1884, 1691, 2927, 4800, 2495, 1845, 1763, 4410, 2873,
2258, 1427,  690,  620, 2405, 4200, 1415, 2547, 3087, 2091, 4650,
```

```
df.living_area_renov.value_counts()
```

```
1440    136
1540    131
1560    127
1500    122
1510    117
       ...
2029      1
2634      1
1137      1
1537      1
1162      1
Name: living_area_renov, Length: 665, dtype: int64
```

```
df.lot_area_renov.value_counts()
```

```
5000      301
4000      256
6000      179
7200      138
4800      102
          ...
12068       1
185565      1
60112       1
14564       1
6631        1
Name: lot_area_renov, Length: 6835, dtype: int64
```

```
df.Price.value_counts()
```

```
450000    114
350000    113
400000    104
375000    103
550000    102
          ...
561600      1
856500      1
907687      1
307999      1
146000      1
Name: Price, Length: 2901, dtype: int64
```

```
plt.pie(df.Price.value_counts())
```

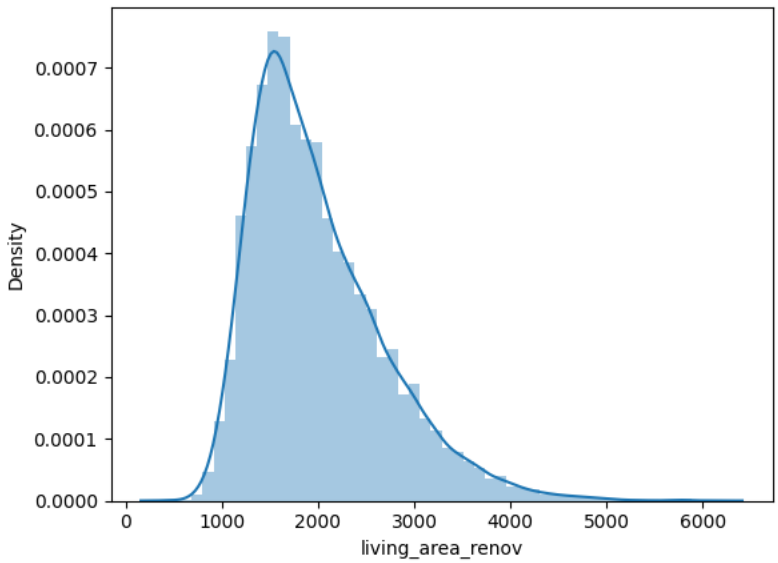
```
([<matplotlib.patches.Wedge at 0x7a41b7581f00>,
  <matplotlib.patches.Wedge at 0x7a41b7581e10>,
  <matplotlib.patches.Wedge at 0x7a41b75827d0>,
  <matplotlib.patches.Wedge at 0x7a41b7582c50>,
  <matplotlib.patches.Wedge at 0x7a41b75afac0>,
  <matplotlib.patches.Wedge at 0x7a41b75830d0>,
  <matplotlib.patches.Wedge at 0x7a41b7583550>,
  <matplotlib.patches.Wedge at 0x7a41b75839d0>,
  <matplotlib.patches.Wedge at 0x7a41b7583e50>,
  <matplotlib.patches.Wedge at 0x7a41b75d0310>,
  <matplotlib.patches.Wedge at 0x7a41b7581ed0>,
  <matplotlib.patches.Wedge at 0x7a41b75d0be0>,
  <matplotlib.patches.Wedge at 0x7a41b75d1060>,
  <matplotlib.patches.Wedge at 0x7a41b75d14e0>,
  <matplotlib.patches.Wedge at 0x7a41b75d1960>,
  <matplotlib.patches.Wedge at 0x7a41b75d1de0>,
  <matplotlib.patches.Wedge at 0x7a41b75d2260>,
  <matplotlib.patches.Wedge at 0x7a41b75d26e0>,
  <matplotlib.patches.Wedge at 0x7a41b75d2b60>,
  <matplotlib.patches.Wedge at 0x7a41b75d2fe0>,
  <matplotlib.patches.Wedge at 0x7a41b75d3460>,
  <matplotlib.patches.Wedge at 0x7a41b75d38e0>,
  <matplotlib.patches.Wedge at 0x7a41b75d3d60>,
  <matplotlib.patches.Wedge at 0x7a41b762c220>,
  <matplotlib.patches.Wedge at 0x7a41b762c6a0>,
  <matplotlib.patches.Wedge at 0x7a41b762cb20>,
  <matplotlib.patches.Wedge at 0x7a41b762cfa0>,
  <matplotlib.patches.Wedge at 0x7a41b762d420>,
  <matplotlib.patches.Wedge at 0x7a41b762d8a0>,
  <matplotlib.patches.Wedge at 0x7a41b762dd20>,
  <matplotlib.patches.Wedge at 0x7a41b762e1a0>,
  <matplotlib.patches.Wedge at 0x7a41b762e620>,
  <matplotlib.patches.Wedge at 0x7a41b762eaa0>,
  <matplotlib.patches.Wedge at 0x7a41b762ef20>,
  <matplotlib.patches.Wedge at 0x7a41b762f3a0>,
  <matplotlib.patches.Wedge at 0x7a41b762f820>,
  <matplotlib.patches.Wedge at 0x7a41b762fca0>,
  <matplotlib.patches.Wedge at 0x7a41b748c160>,
  <matplotlib.patches.Wedge at 0x7a41b748c5e0>,
  <matplotlib.patches.Wedge at 0x7a41b748ca60>,
  <matplotlib.patches.Wedge at 0x7a41b748cee0>,
  <matplotlib.patches.Wedge at 0x7a41b748d360>,
  <matplotlib.patches.Wedge at 0x7a41b748d7e0>,
  <matplotlib.patches.Wedge at 0x7a41b748dc60>,
  <matplotlib.patches.Wedge at 0x7a41b748e0e0>,
  <matplotlib.patches.Wedge at 0x7a41b748e560>,
  <matplotlib.patches.Wedge at 0x7a41b748e9e0>,
  <matplotlib.patches.Wedge at 0x7a41b748ee60>,
  <matplotlib.patches.Wedge at 0x7a41b748f2e0>,
  <matplotlib.patches.Wedge at 0x7a41b748f760>,
  <matplotlib.patches.Wedge at 0x7a41b748fbe0>,
  <matplotlib.patches.Wedge at 0x7a41b74d40a0>,
  <matplotlib.patches.Wedge at 0x7a41b74d4520>,
  <matplotlib.patches.Wedge at 0x7a41b74d49a0>,
```

```
sns.displot(df.living_area_renov)
```

```
<seaborn.axisgrid.FacetGrid at 0x7a41d37b65f0>
```

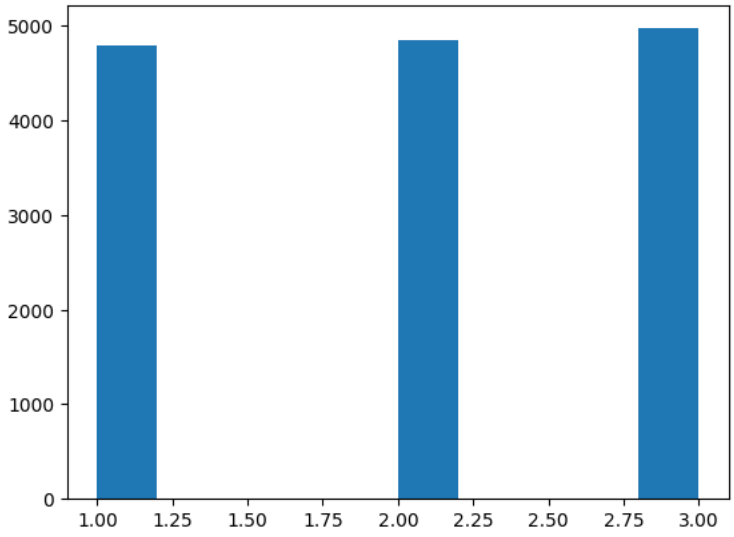
```
sns.distplot(df.living_area_renov)
```

```
<ipython-input-14-b944eacf0633>: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_renov)
<Axes: xlabel='living_area_renov', ylabel='Density'>
```




```
plt.hist(df['Number of schools nearby'])
```

```
(array([4794.,    0.,    0.,    0.,    0., 4853.,    0.,    0.,    0.,
        4973.]),
 array([1. , 1.2, 1.4, 1.6, 1.8, 2. , 2.2, 2.4, 2.6, 2.8, 3. ]),
 <BarContainer object of 10 artists>)
```




```
<matplotlib.patches.Wedge at 0x7a41b746e860>,
```

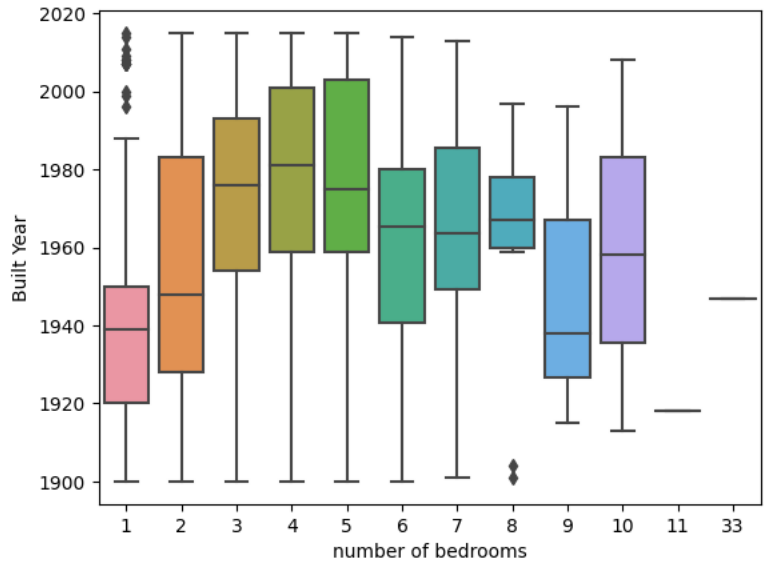
```
sns.countplot(df['number of bedrooms'])
```

```
<Axes: ylabel='count'>
```

Bivariant Analysis

```
sns.boxplot(x=df['number of bedrooms'],y=df['Built Year'])
```

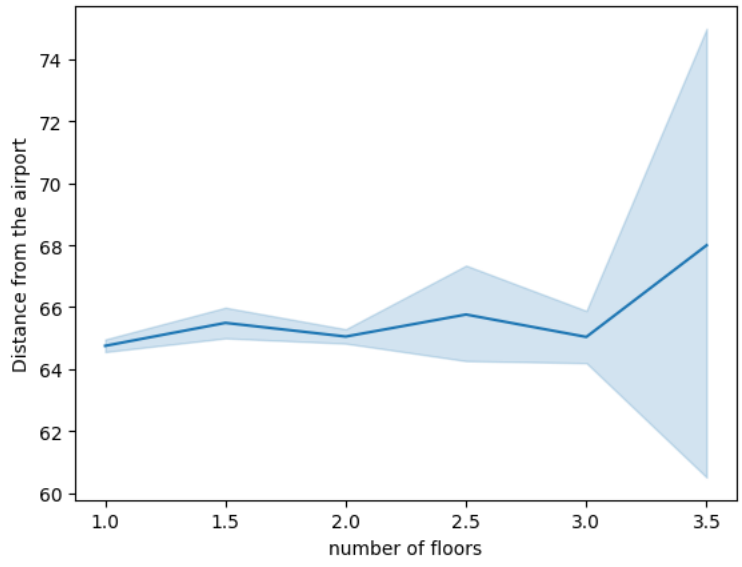
```
<Axes: xlabel='number of bedrooms', ylabel='Built Year'>
```




```
 <matplotlib.patches.Wedge at 0x7a41b71940a0>,
```

```
sns.lineplot(x=df['number of floors'],y=df['Distance from the airport'])
```

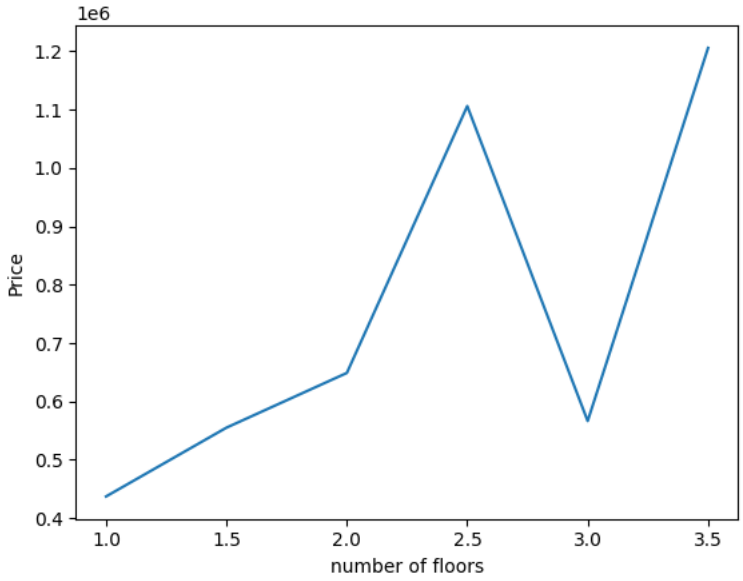
```
<Axes: xlabel='number of floors', ylabel='Distance from the airport'>
```




```
 <matplotlib.patches.Wedge at 0x7a41b7210820>,
```

```
sns.lineplot(x=df.groupby('number of floors').mean().index,y=df.groupby('number of floors').mean()['Price'])
plt.show
```

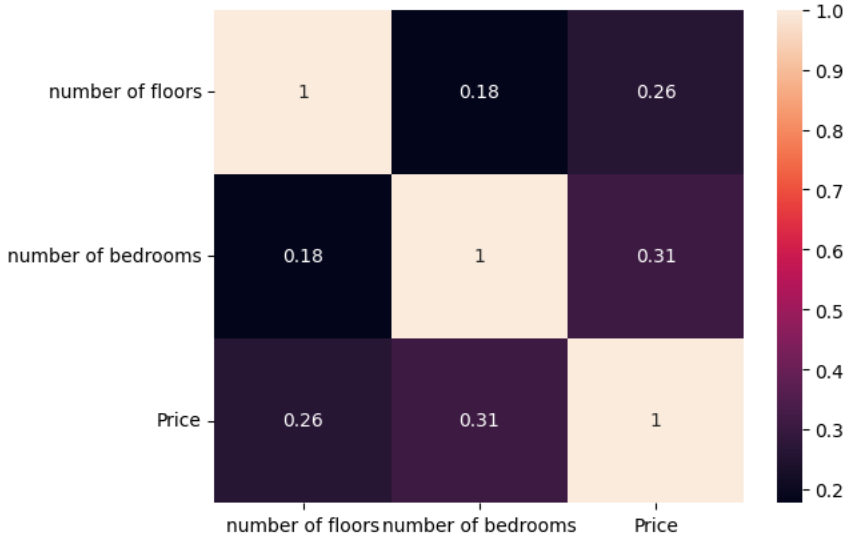
```
<function matplotlib.pyplot.show(close=None, block=None)>
```




```
<matplotlib.patches.Wedge at 0x7a41b7089ct0>,
```

```
sns.heatmap(df[['number of floors','number of bedrooms','Price']].corr(),annot=True)
```

```
<Axes: >
```

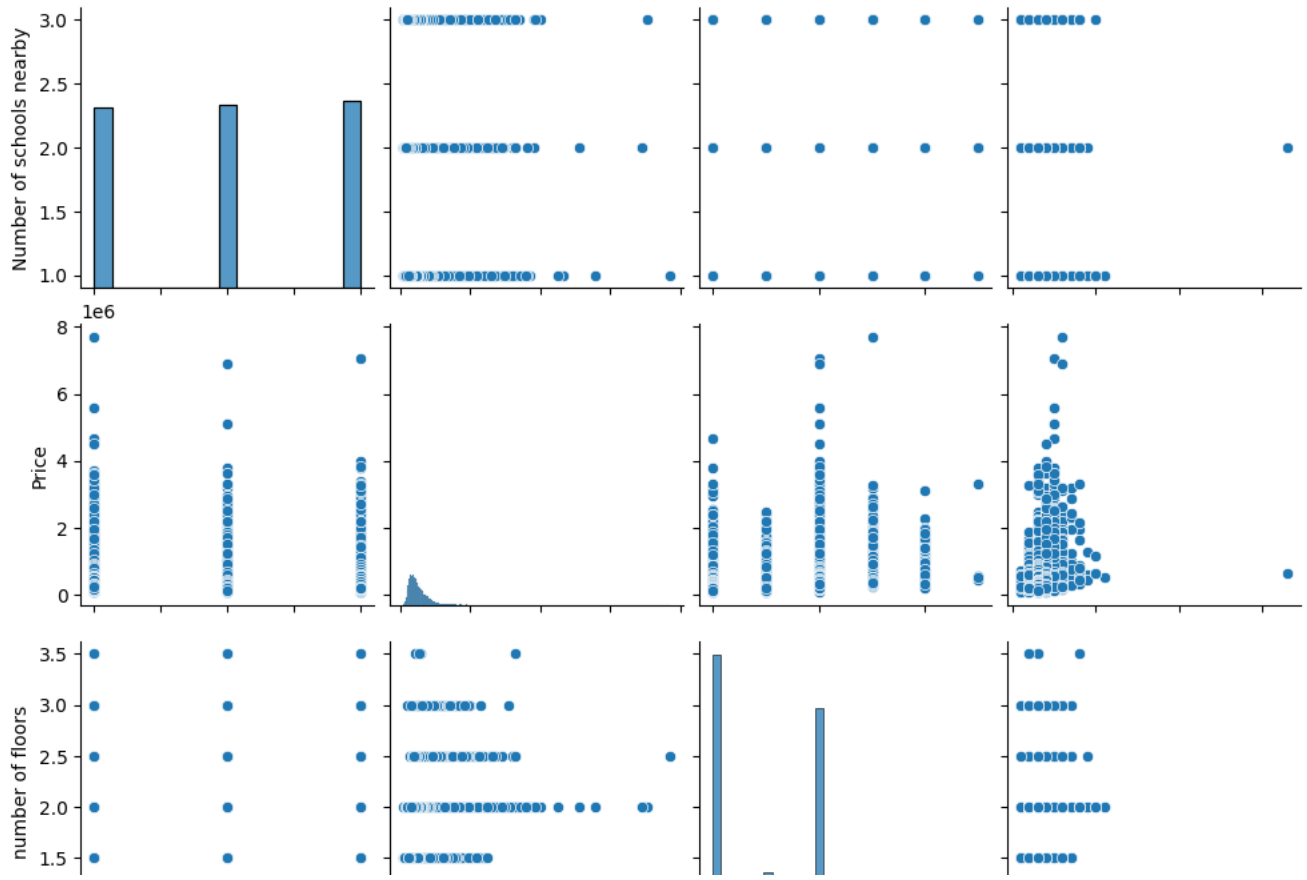



```
<matplotlib.patches.Wedge at 0x7a41b7165ff0>,
```

Multivariate Analysis

```
sns.pairplot(df[['Number of schools nearby','Price','number of floors','number of bedrooms']])
```

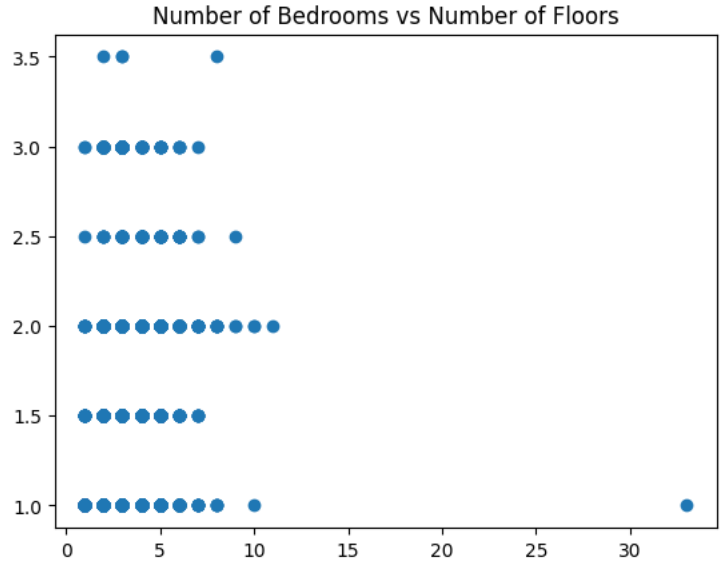
```
<seaborn.axisgrid.PairGrid at 0x7a41b424e3b0>
```




```
df.duplicated().sum()
```

```
0
```

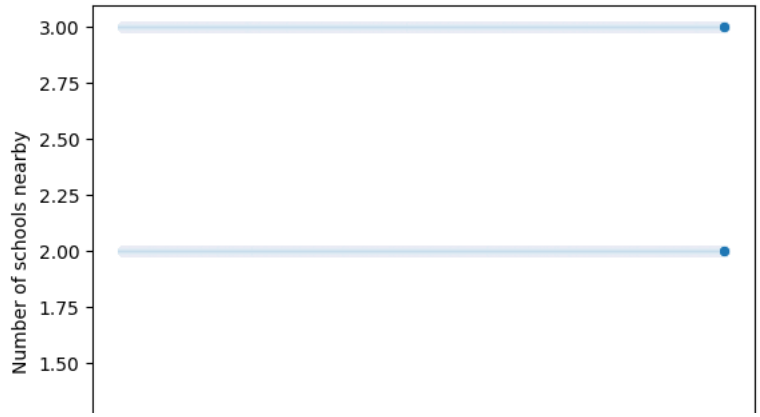
```
plt.scatter(df['number of bedrooms'],df['number of floors'])
plt.title("Number of Bedrooms vs Number of Floors")
plt.grid(linestyle='-', linewidth=0.)
```



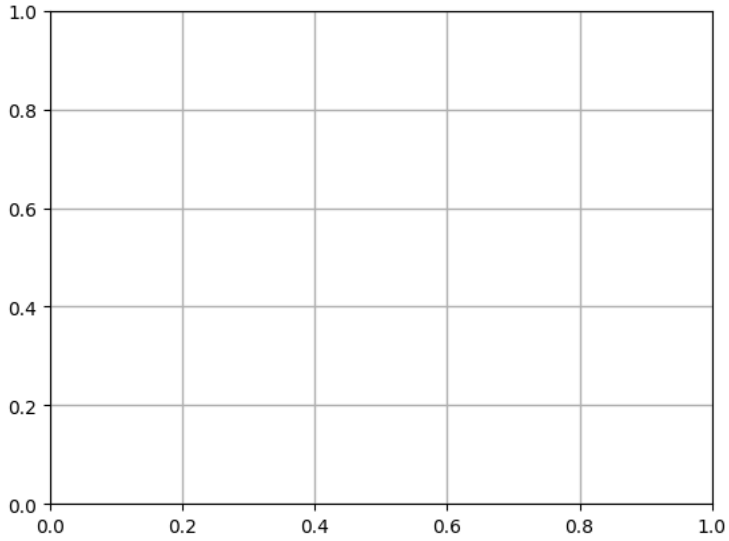

```
<matplotlib.patches.Wedge at 0x7a41b6e9e3e0>,
```

```
sns.scatterplot(df['Number of schools nearby'])
```

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

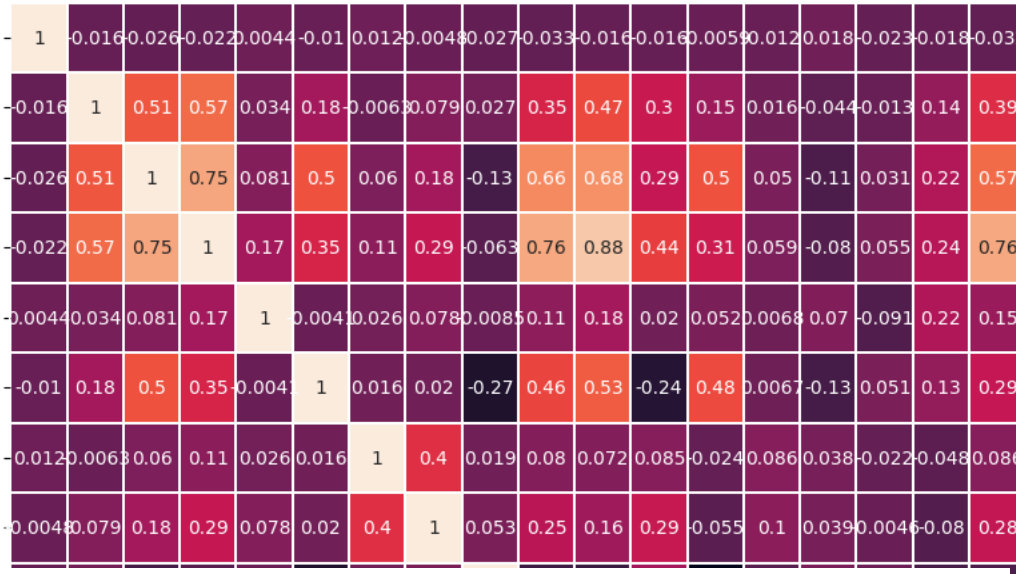



```
plt.grid(linestyle='-',linewidth=1.0)
```




```
<matplotlib.patches.Wedge at 0x7a41b6d8e560>,
```

```
plt.subplots(figsize=(15,15))
sns.heatmap(df.drop(['id'],axis=1).corr(),linewidth=0.3,annot=True)
plt.show()
```




```
print(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,
```

```
print(df.corr())
```

```
                                              id      Date  number of bedrooms  \
id                                     1.000000  0.045966           -0.329034
Date                                   0.045966  1.000000           -0.015663
number of bedrooms                    -0.329034 -0.015663            1.000000
number of bathrooms                   -0.516909 -0.026485            0.509784
living area                           -0.648127 -0.021958            0.570526
lot area                              -0.100269  0.004392            0.034416
number of floors                      -0.312305 -0.010335            0.177294
waterfront present                    -0.112937  0.012006           -0.006257
number of views                       -0.293004 -0.004782            0.078665
condition of the house                -0.045061 -0.027402            0.026597
grade of the house                    -0.673448 -0.033097            0.352945
Area of the house(excluding basement) -0.565116 -0.015994            0.473599
Area of the basement                  -0.290806 -0.015711            0.300332
Built Year                            -0.068645 -0.005869            0.152954
Renovation Year                       -0.109155 -0.011636            0.016132
Postal Code                            0.294709  0.018243           -0.044156
Lattitude                             -0.479334 -0.023327           -0.013163
Longitude                             -0.070841 -0.018231            0.135712
living_area_renov                     -0.599900 -0.032495            0.389855
lot_area_renov                        -0.089604 -0.000050            0.029400
Number of schools nearby              -0.004821 -0.004071            0.003397
Distance from the airport             -0.004542  0.011457           -0.006157
Price                                 -0.773114 -0.027919            0.308460

                                      number of bathrooms  living area  \
id                                              -0.516909    -0.648127
Date                                            -0.026485    -0.021958
number of bedrooms                               0.509784     0.570526
number of bathrooms                              1.000000     0.753517
living area                                      0.753517     1.000000
lot area                                         0.080806     0.174420
number of floors                                 0.502924     0.354743
waterfront present                               0.060104     0.105837
number of views                                  0.183789     0.287728
condition of the house                          -0.128232    -0.063358
grade of the house                               0.663054     0.761835
Area of the house(excluding basement)            0.684391     0.875793
Area of the basement                             0.287190     0.441491
Built Year                                       0.498127     0.309602
Renovation Year                                  0.049669     0.059400
Postal Code                                     -0.105546    -0.080303
Lattitude                                        0.031156     0.054518
Longitude                                        0.223904     0.240208
living_area_renov                                0.570530     0.757571
lot_area_renov                                   0.078627     0.180312
Number of schools nearby                         0.002180     0.002370
Distance from the airport                        0.009206     0.002511
Price                                            0.531735     0.712169

                                       lot area  number of floors  \
id                                    -0.100269         -0.312305
Date                                   0.004392         -0.010335
number of bedrooms                     0.034416          0.177294
number of bathrooms                    0.080806          0.502924
living area                            0.174420          0.354743
lot area                               1.000000         -0.004138
number of floors                      -0.004138          1.000000
  <matplotlib.patches.Wedge at 0x7a41b6a9eda0>,
```

```
print(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
  <matplotlib.patches.Wedge at 0x7a41b6ad6140>,
```

```
print('Mean:',df['number of bedrooms'].mean())
print('Median:',df['number of views'].median())
print('Mode:',df['number of bathrooms'].mode())
```

```
Mean: 3.379343365253078
Median: 0.0
Mode: 0    2.5
Name: number of bathrooms, dtype: float64
  <matplotlib.patches.Wedge at 0x7a41b6b1d6d0>,
```

<matplotlib.patches.Wedge at 0x7a41b60c670>,

Handle the Missing Values

<matplotlib.patches.Wedge at 0x7a41b60c1970>,

```
print(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
```

<matplotlib.patches.Wedge at 0x7a41b6098670>,

```
df.dropna(inplace=True)
df.fillna(0,inplace=True)
df.interpolate(inplace=True)
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
x=df.drop(['number of floors','number of bedrooms'],axis=1)
x.set_index(['Date'],inplace=True)
y=df[['id','Price']]
```

<matplotlib.patches.Wedge at 0x7a41b69bca30>,

```
x.head()
```

Date	id	number of bathrooms	living area	lot area	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	P
42491	6762810145	2.50	3650	9050	0	4	5	10	3370	280	1921	0	1
42491	6762810635	2.50	2920	4000	0	0	5	8	1910	1010	1909	0	1
42491	6762810998	2.75	2910	9480	0	0	3	8	2910	0	1939	0	1
42491	6762812605	2.50	3310	42998	0	0	3	9	3310	0	2001	0	1
42491	6762812919	2.00	2710	4500	0	0	4	8	1880	830	1929	0	1

<matplotlib.patches.Wedge at 0x7a41b6a06920>,

```
y.head()
```

	id	Price
0	6762810145	2380000
1	6762810635	1400000
2	6762810998	1200000
3	6762812605	838000
4	6762812919	805000

<matplotlib.patches.Wedge at 0x7a41b6a26ce0>,

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
```

```
from sklearn.metrics import r2_score
```

```
<matplotlib.patches.Wedge at 0x7a41b6a75120>,
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y['Price'],test_size =0.1,random_state=2)
model = GradientBoostingRegressor(n_estimators=400,max_depth=5,min_samples_split=2,learning_rate=0.1)
model.fit(x_train,y_train)
```

```
GradientBoostingRegressor
GradientBoostingRegressor(max_depth=5, n_estimators=400)
```

```
<matplotlib.patches.Wedge at 0x7a41b6...>,
```

```
y_pred = model.predict(x_test)
model.score(x_test,y_test)
```

```
0.9999938724950782
```

```
<matplotlib.patches.Wedge at 0x7a41b6...>,
```

```
r2_score(y_pred,y_test)
```

```
0.9999938674863043
```

```
<matplotlib.patches.Wedge at 0x7a41b6...>,
```

```
y_pred
```

```
array([467201.50782526, 244960.21568353, 250004.96234778, ...,
       667579.67630798, 230141.09447065, 208356.81611335])
```

```
<matplotlib.patches.Wedge at 0x7a41b6...>,
```

```
y_pred_list = y['Price'][-len(y_pred):].tolist()
```

```
<matplotlib.patches.Wedge at 0x7a41b6...>,
```

```
y_pred_df=pd.DataFrame(y_pred_list,columns=['Date'])
y_pred_df["number of floors"]= y_pred.round(2)
```

```
<matplotlib.patches.Wedge at 0x7a41b6...>,
```

```
y_pred_df
```

	Date	number of floors
0	1100000	467201.51
1	1040000	244960.22
2	950000	250004.96
3	932990	284580.81
4	910000	485186.77
...	...	...
1457	221700	1009906.80
1458	219200	302627.98
1459	209000	667579.68
1460	205000	230141.09
1461	146000	208356.82

1462 rows × 2 columns

```
<matplotlib.patches.Wedge at 0x7a41b69624a0>,
<matplotlib.patches.Wedge at 0x7a41b6962920>,
<matplotlib.patches.Wedge at 0x7a41b6962da0>,
<matplotlib.patches.Wedge at 0x7a41b6963220>,
<matplotlib.patches.Wedge at 0x7a41b69636a0>,
<matplotlib.patches.Wedge at 0x7a41b6963b20>,
<matplotlib.patches.Wedge at 0x7a41b6963fa0>,
<matplotlib.patches.Wedge at 0x7a41b6798460>,
<matplotlib.patches.Wedge at 0x7a41b67988e0>,
<matplotlib.patches.Wedge at 0x7a41b6798d60>,
<matplotlib.patches.Wedge at 0x7a41b67991e0>,
<matplotlib.patches.Wedge at 0x7a41b6799660>,
<matplotlib.patches.Wedge at 0x7a41b6799ae0>,
<matplotlib.patches.Wedge at 0x7a41b6799f60>,
<matplotlib.patches.Wedge at 0x7a41b679a3e0>,
<matplotlib.patches.Wedge at 0x7a41b679a860>,
<matplotlib.patches.Wedge at 0x7a41b679ace0>,
<matplotlib.patches.Wedge at 0x7a41b679b160>,
<matplotlib.patches.Wedge at 0x7a41b679b5e0>,
<matplotlib.patches.Wedge at 0x7a41b679ba60>,
<matplotlib.patches.Wedge at 0x7a41b679bee0>,
```