

Load the dataset.

```
In [60]: import numpy as np
import pandas as pd
import seaborn as sns
import random as random
import matplotlib.pyplot as plt
sns.set(color_codes=True)

In [4]: df = pd.read_csv("../Dataset/House Price India.csv")
df

Out[4]:
```

	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	Postal Code	Latitude	Longitude	living_area_renov	lot_area_renov
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5	...	1921	0	122003	52.8645	-114.557	2880	5
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5	...	1909	0	122004	52.8878	-114.470	2470	4
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3	...	1939	0	122004	52.8852	-114.468	2940	6
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3	...	2001	0	122005	52.9532	-114.321	3350	42
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	4	...	1929	0	122006	52.9047	-114.485	2060	4
...
14615	6762830250	42734	2	1.50	1556	20000	1.0	0	0	4	...	1957	0	122066	52.6191	-114.472	2250	17
14616	6762830339	42734	3	2.00	1680	7000	1.5	0	0	4	...	1968	0	122072	52.5075	-114.393	1540	7
14617	6762830618	42734	2	1.00	1070	6120	1.0	0	0	3	...	1962	0	122056	52.7289	-114.507	1130	6
14618	6762830709	42734	4	1.00	1030	6621	1.0	0	0	4	...	1955	0	122042	52.7157	-114.411	1420	6
14619	6762831463	42734	3	1.00	900	4770	1.0	0	0	3	...	1969	2009	122018	52.5338	-114.552	900	3

14620 rows × 23 columns

```
In [8]: df.head()

Out[8]:
```

	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	Postal Code	Latitude	Longitude	living_area_renov	lot_area_renov
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5	...	1921	0	122003	52.8645	-114.557	2880	5400
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5	...	1909	0	122004	52.8878	-114.470	2470	4000
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3	...	1939	0	122004	52.8852	-114.468	2940	6600
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3	...	2001	0	122005	52.9532	-114.321	3350	42847
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	4	...	1929	0	122006	52.9047	-114.485	2060	4500

5 rows × 23 columns

```
In [5]: 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
```

Perform the Below Visualizations.

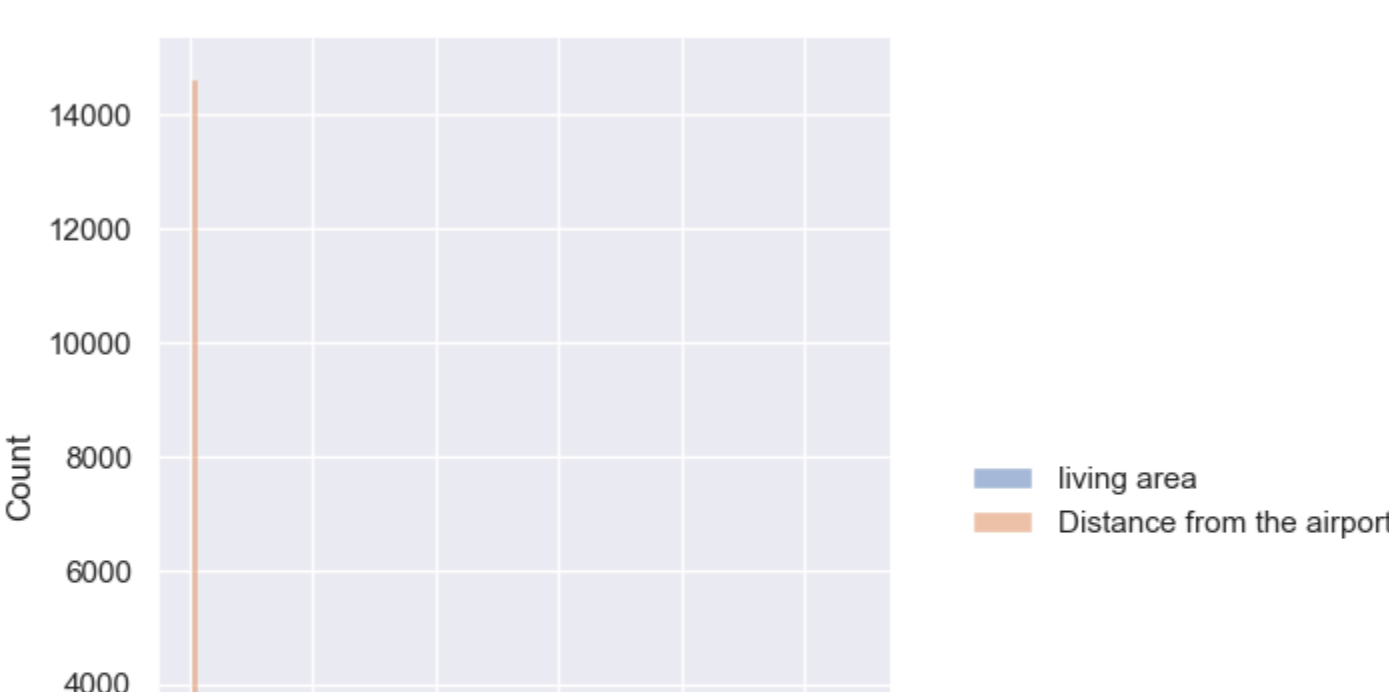
1.Univariate Analysis

2.Bi - Variate Analysis

3.Multi-Variate Analysis

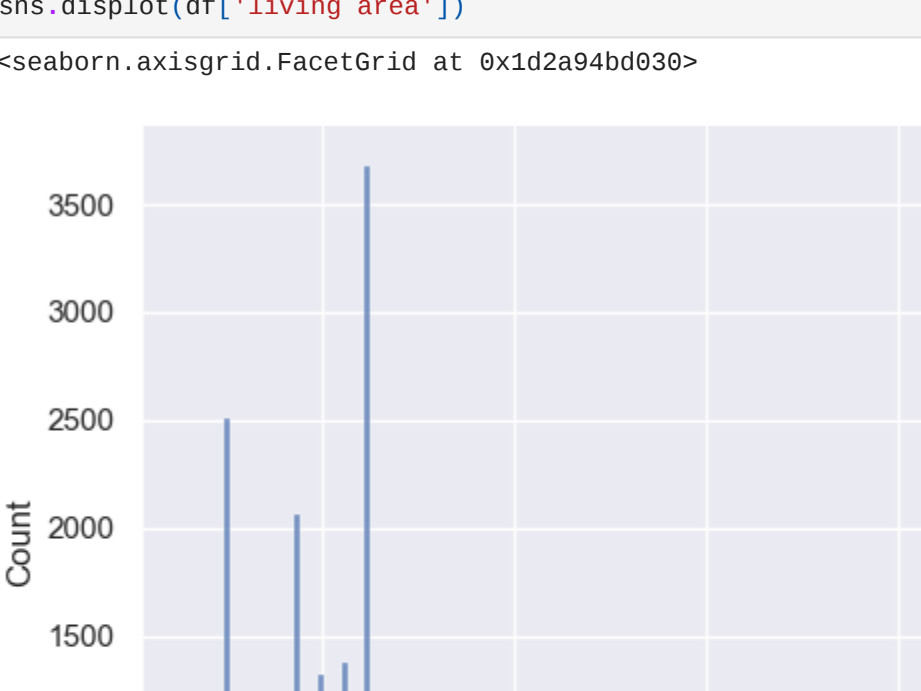
```
In [55]: sns.displot(df[['living area', 'Distance from the airport']])

Out[55]: <seaborn.axisgrid.FacetGrid at 0x1d2ba5dd429>
```

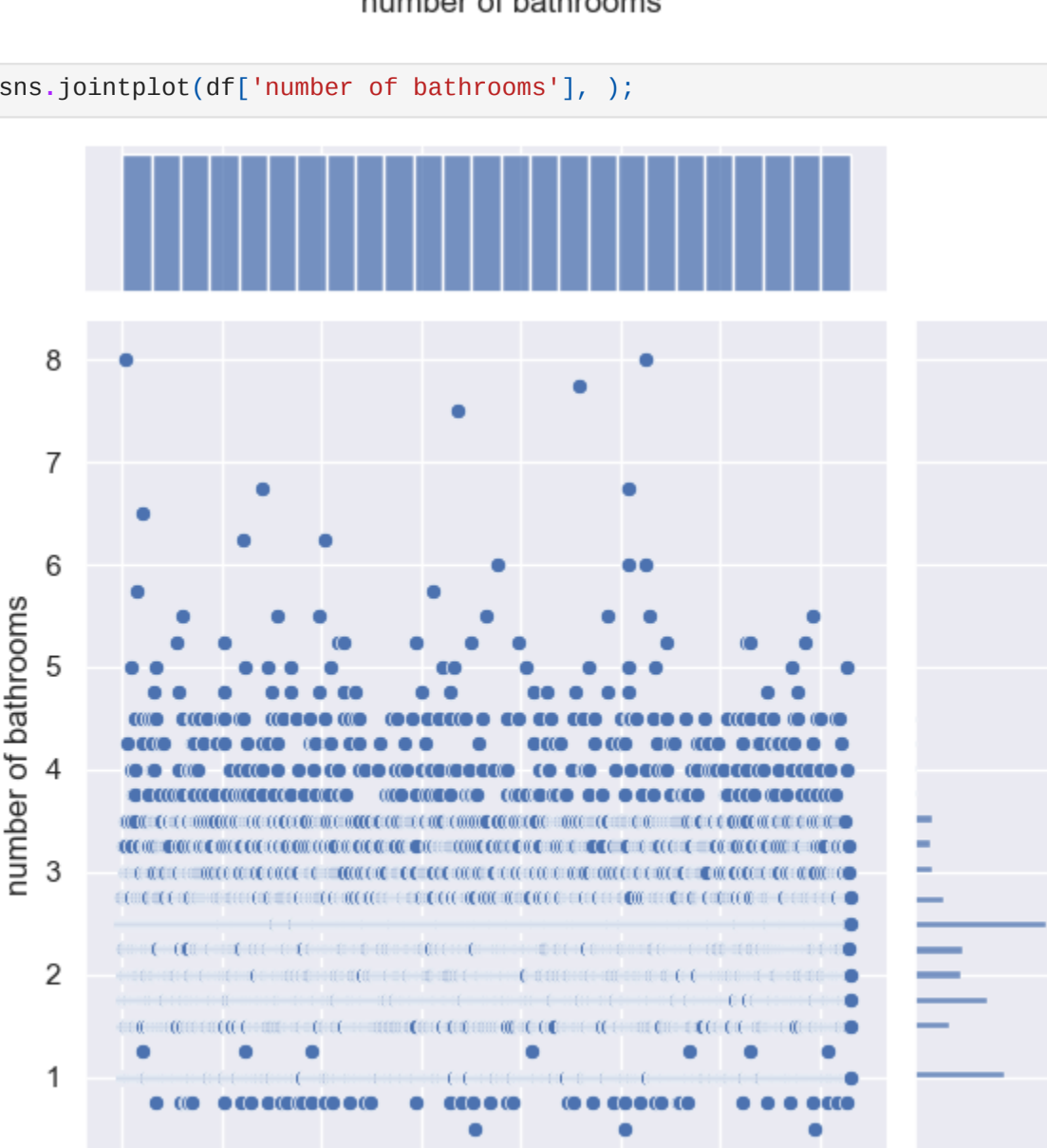


```
In [15]: sns.displot(df['living area'])

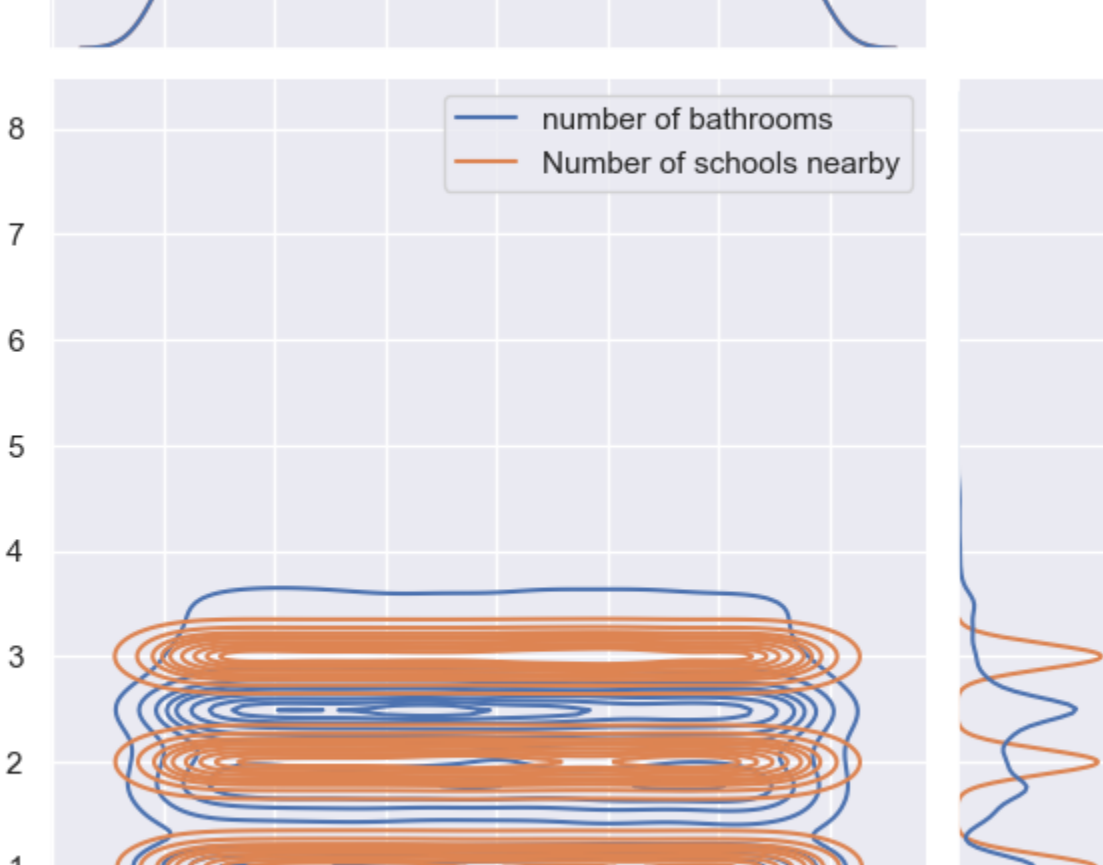
Out[15]: <seaborn.axisgrid.FacetGrid at 0x1d2a94bd830>
```



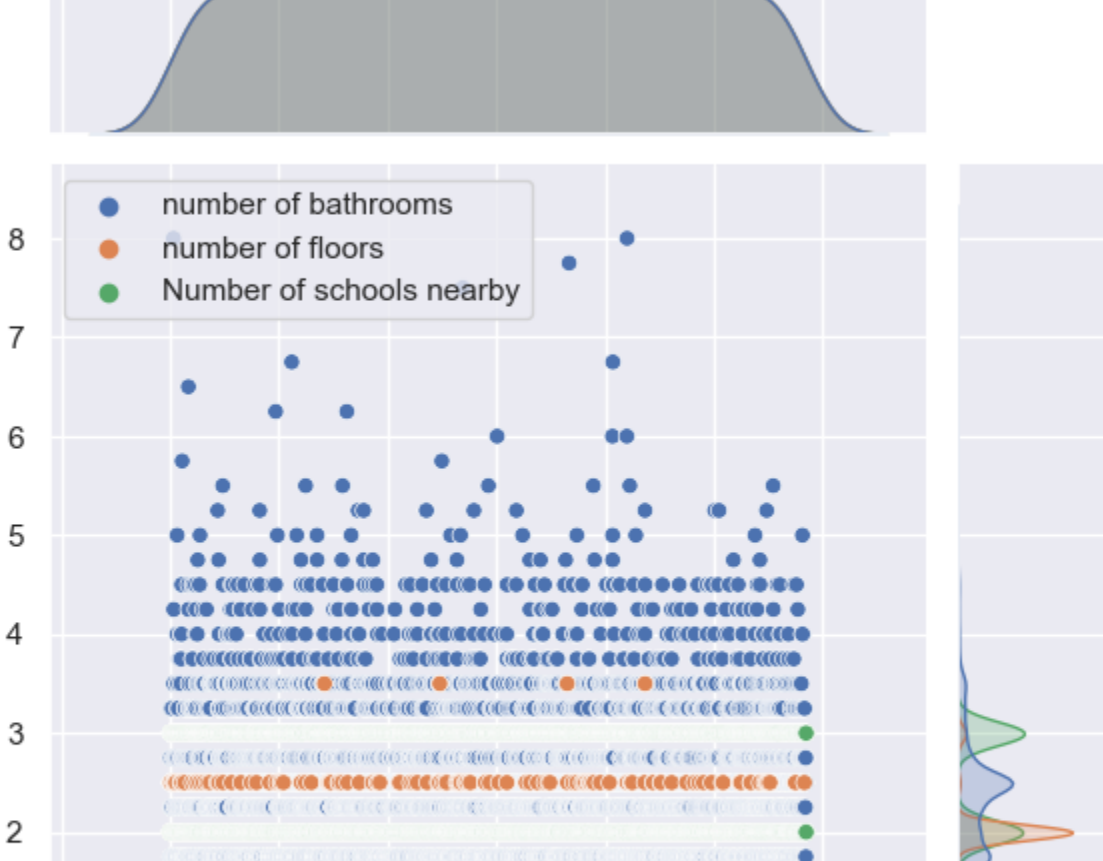
```
In [34]: sns.jointplot(df['number of bathrooms'], );
```



```
In [41]: sns.jointplot(df[['number of bathrooms', 'Number of schools nearby']], kind="kde");
```

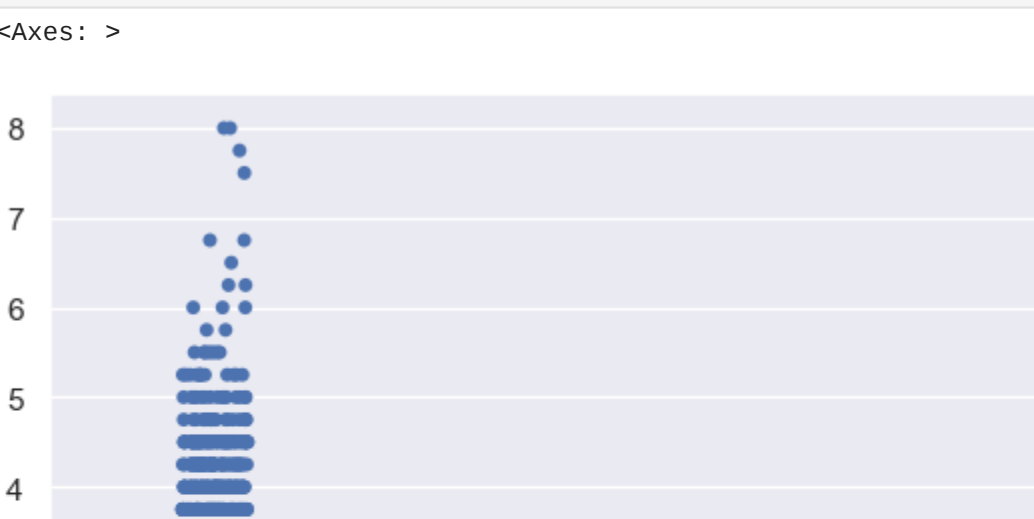


```
In [39]: sns.jointplot(df[['number of bathrooms', 'number of floors', 'Number of schools nearby'] );
```



```
In [50]: sns.stripplot(df[['number of bathrooms', 'Number of schools nearby', 'number of floors']])

Out[50]: <Axes: >
```



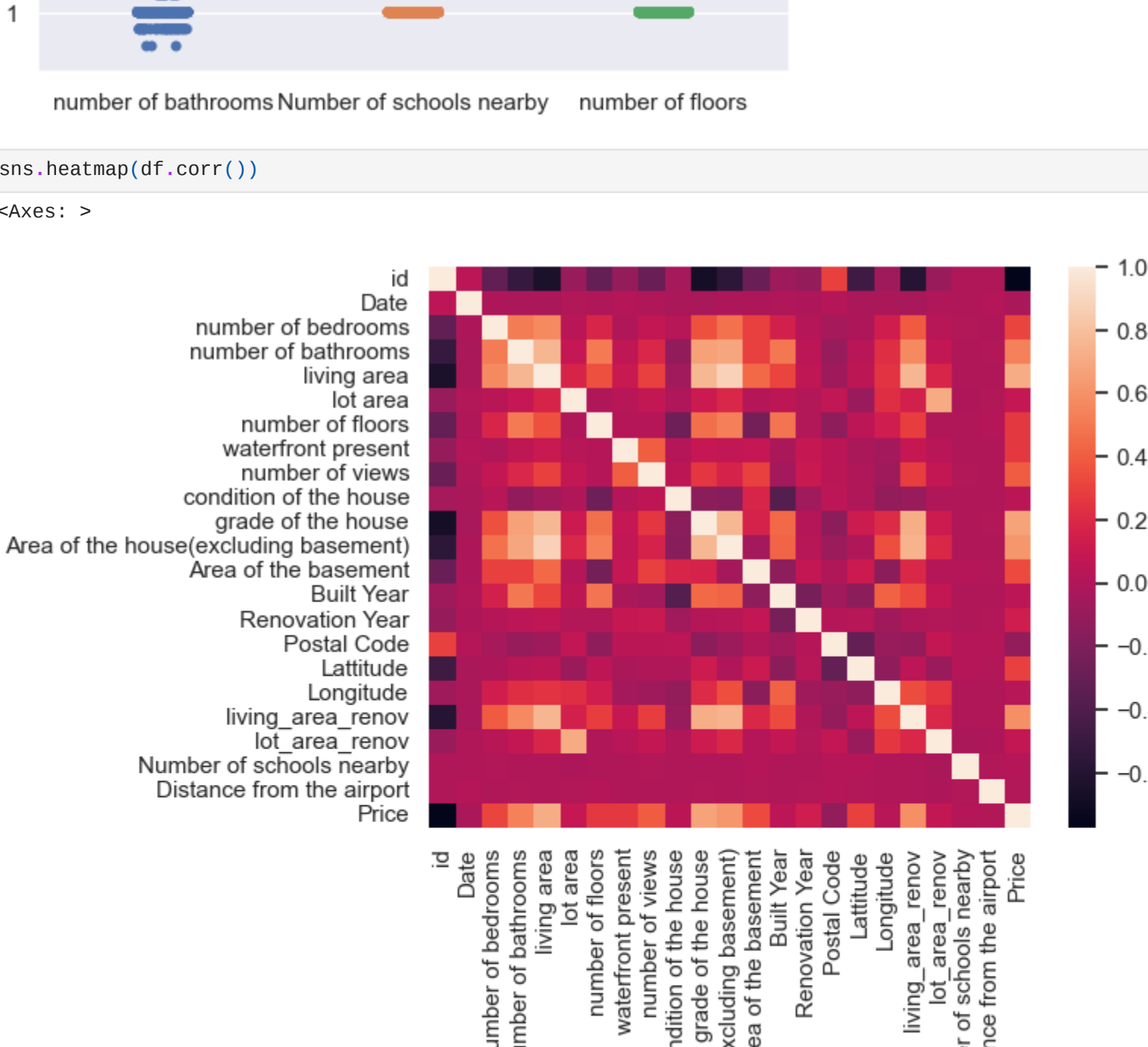
```
In [54]: sns.stripplot(df[['number of bathrooms', 'Number of schools nearby', 'number of floors']], jitter = True)

Out[54]: <Axes: >
```



```
In [61]: sns.heatmap(df.corr())

Out[61]: <Axes: >
```



Perform descriptive statistics on the dataset.

```
In [62]: random.seed(42)
values = [random.randrange(1,1001,1) for _ in range(10000)]
values[0:10]

Out[62]: [655, 115, 26, 760, 282, 251, 229, 143, 755, 105]
```

```
In [64]: len(values)

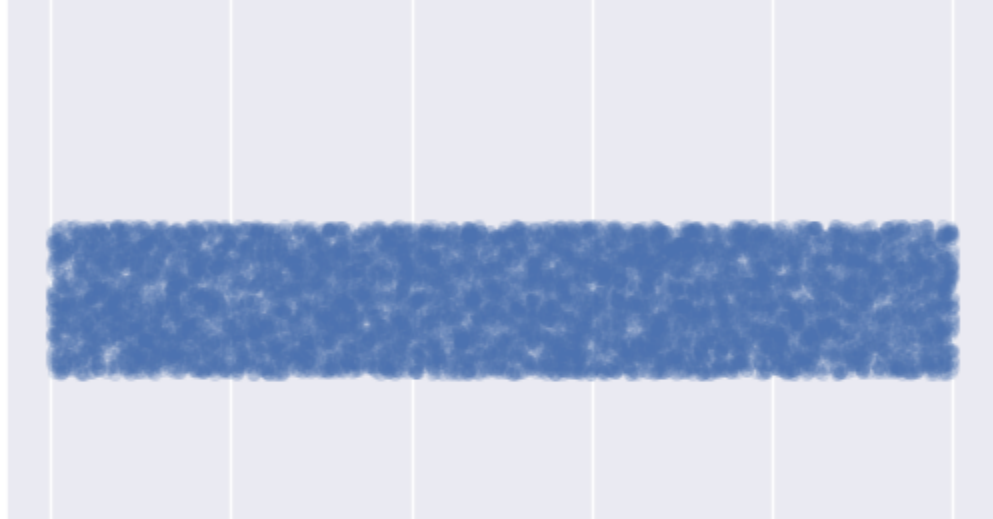
Out[64]: 10000

In [65]: sum(values)

Out[65]: 5021696
```

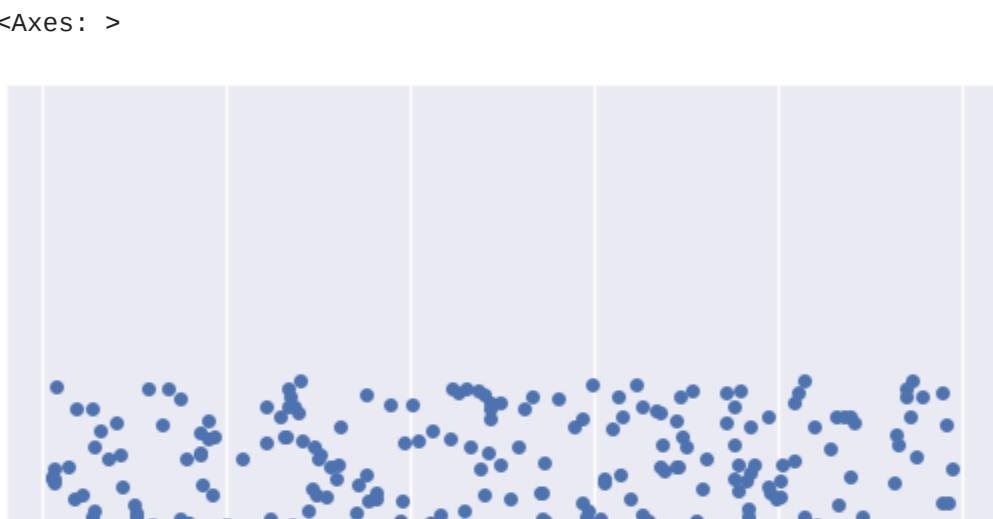
```
In [66]: sns.stripplot(x=values, jitter=True, alpha=0.2)

Out[66]: <Axes: >
```



```
In [69]: sparse_values = [random.randrange(1,1001) for _ in range(1200)]
sns.stripplot(x=sparse_values, jitter=True)

Out[69]: <Axes: >
```



```
In [70]: print("Max value: {}".format(max(values), min(values)))
Max value: 1000
Min value: 1

In [71]: sorted_vals = sorted(values)
sorted_vals[0:20]

Out[71]: [1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3]
```

Handle the Missing values

```
In [58]: df.isnull()

Out[58]:
```

	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	Postal Code	Latitude	Longitude	living_area_renov	lot_area_renov	sc	n
0	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
...	
14615	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
14616	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
14617	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
14618	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
14619	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	

14620 rows × 23 columns

```
In [ ]:
```