

# r6ilb7fnk

May 19, 2023

```
[2]: import kaggle
[3]: datasets=kaggle.api.dataset_list()
[4]: print(datasets)

[salvatorerastelli/spotify-and-youtube, arnabchaki/data-science-salaries-2023,
erdemtaha/cancer-data, evangower/premier-league-2022-2023, lokeshparab/amazon-
products-dataset, iammustafatz/diabetes-prediction-dataset,
ulrikthygepedersen/fastfood-nutrition, mikoajfish99/us-recession-and-financial-
indicators, desalegngeb/students-exam-scores, rkiattisak/student-performance-in-
mathematics, ppb00x/credit-risk-customers, ritwikb3/heart-disease-cleveland,
ppb00x/country-gdp, dansbecker/melbourne-housing-snapshot, rishabhgupta/google-
stock-price-daily-weekly-and-monthly-2023, omartorres25/honda-data,
harshghadiya/kidneystone, ashishraut64/internet-users, andrezaza/clapper-
massive-rotten-tomatoes-movies-and-reviews, richardson/the-world-university-
rankings-2011-2023]

[5]: dataset=kaggle.api.datasets_list(search="House Price dataset of India")
[6]: import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
[7]: df=pd.DataFrame(dataset)
[8]: df.head()

[8]:          subtitleNullable  creatorNameNullable  \
0  This data Set contains 23 columns and 14620 Ro...      Mohamed Afsal
1                  Predict the house prices in India      Anmol Kumar
2                  Prices of new and resale houses      Ruchi Bhatia
3  Rental prices of ~2Lakh houses (NO DUPLICATED ...  Sai Saathvik Domala
4                  Machine Hack, Problem for beginners      Jaswinder Singh

creatorUrlNullable  totalBytesNullable  \
```

0	mohamedafsal007	491826
1	anmolkumar	2233190
2	ruchi798	275305
3	saisaathvik	2391227
4	jassican	2233190

	urlNullable	\
0	https://www.kaggle.com/datasets/mohamedafsal007	
1	https://www.kaggle.com/datasets/anmolkumar/hou...	
2	https://www.kaggle.com/datasets/ruchi798/housi...	
3	https://www.kaggle.com/datasets/saisaathvik/ho...	
4	https://www.kaggle.com/datasets/jassican/house...	

	licenseNameNullable	descriptionNullable	ownerNameNullable	\
0	Other (specified in description)	None	Mohamed Afsal	
1	GPL 2	None	Anmol Kumar	
2	CC0: Public Domain	None	Ruchi Bhatia	
3	CC0: Public Domain	None	Sai Saathvik Domala	
4	Other (specified in description)	None	Jaswinder Singh	

	ownerRefNullable	titleNullable	\
0	mohamedafsal007	House Price dataset of India	...
1	anmolkumar	House Price Prediction Challenge	...
2	ruchi798	Housing Prices in Metropolitan Areas of India	...
3	saisaathvik	House Rent Prices of Metropolitan Cities in India	...
4	jassican	House Price Prediction Challenge	...

	topicCount	viewCount	voteCount	currentVersionNumber	\
0	0	16560	57	1	
1	0	114023	188	1	
2	0	39145	189	4	
3	0	7400	15	3	
4	0	8688	21	1	

	hasCurrentVersionNumber	usabilityRating	hasUsabilityRating	\
0	True	0.882353	True	
1	True	1.000000	True	
2	True	0.970588	True	
3	True	0.970588	True	
4	True	0.941176	True	

	tags	files	versions
0	[{'nameNullable': 'india', 'descriptionNullabl...']	[]	[]
1	[{'nameNullable': 'india', 'descriptionNullabl...']	[]	[]
2	[{'nameNullable': 'cities and urban areas', 'd...']	[]	[]
3	[{'nameNullable': 'india', 'descriptionNullabl...']	[]	[]
4	[{'nameNullable': 'housing', 'descriptionNulla...']	[]	[]

[5 rows x 49 columns]

[9]: !kaggle datasets list -s "House Price dataset of India"

ref	title			
size	lastUpdated	downloadCount	voteCount	usabilityRating
<hr/>				
<hr/>				
mohamedafsal007/house-price-dataset-of-india	House Price			
dataset of India	480KB	2023-03-27	06:22:55	
3917	57	0.88235295		
anmolkumar/house-price-prediction-challenge	House Price			
Prediction Challenge	2MB	2020-10-01	19:37:13	
14071	188	1.0		
ruchi798/housing-prices-in-metropolitan-areas-of-india	Housing Prices in			
Metropolitan Areas of India	269KB	2020-09-05	08:53:24	7469
189	0.9705882			
saisaathvik/house-rent-prices-of-metropolitan-cities-in-india	House Rent Prices			
of Metropolitan Cities in India	2MB	2021-06-01	08:35:34	1114
15	0.9705882			
jassican/house-price-prediction-challenge-machine-hack	House Price			
Prediction Challenge	2MB	2020-10-01	02:33:40	
694	21	0.9411765		
anantsakhare/rental-price-of-indias-it-capital-pune-mh-ind	Rental price of			
India's IT Capital - Pune, MH, IND	497KB	2021-07-08	10:45:08	638
24	0.9705882			
akash14/house-price-dataset	House Price			
Dataset	2MB	2020-09-27	09:59:55	
2007	32	0.9411765		
ishandutta/machine-hack-housing-price-prediction	Machine Hack			
Housing Price Prediction	2MB	2020-09-26	13:02:19	312
11	1.0			
sonukiller99/indian-house-price-combined	Indian House			
Prices	147KB	2021-06-07	17:01:03	394
11	0.7352941			
andynath/new-delhi-rental-listings	New Delhi Rental			
Listings	748KB	2020-08-11	16:05:38	336
13	0.9411765			
architsingh15/bengaluru-house-prices	Bengaluru House			
Prices	215KB	2020-10-10	18:59:31	577
14	0.88235295			
amaanafif/chennai-house-price	Chennai House			
Price	21KB	2021-11-02	07:28:56	853
14	1.0			
promptcloud/product-listing-on-amazon-india	Product Listing			

```

On Amazon India 8MB 2020-05-26 07:54:29 292
9 0.9411765
shantanudhakadd/house-prediction-dataset House Prediction
Dataset 195KB 2022-03-01 16:54:58 1012
27 0.9411765
dravidvaishnav/mumbai-house-prices Mumbai House
Prices 889KB 2023-01-08 04:36:51 194
6 0.7058824
goelyash/housing-price-dataset-of-delhiindia Housing Price
Dataset of Delhi(India) 944KB 2021-11-23 14:07:35 385
18 0.9705882
ahmedashrafahmed/houseprice House-Price
24KB 2022-11-29 20:55:49 55 5 0.5294118
promptcloud/flipkart-product-listing-2020 Flipkart Product
Listing 2020 16MB 2021-03-05 20:24:26 128
5 0.9411765
abhinaykumarsingh/bengaluru-house-price-data Bengaluru House
price data 195KB 2020-08-23 14:06:14 157
4 0.3529412
prathameshf/housing-properties-in-maharashtra-big-cities Housing
properties in Maharashtra (Big cities) 2MB 2022-05-01 07:54:07
81 5 0.4117647

```

```
[10]: !kaggle datasets download -d mohamedafsal007/house-price-dataset-of-india
```

Downloading house-price-dataset-of-india.zip to c:\Users\balam\Desktop

```

0%|          | 0.00/480k [00:00<?, ?B/s]
100%|     | 480k/480k [00:02<00:00, 223kB/s]
100%|     | 480k/480k [00:02<00:00, 221kB/s]

```

```
[11]: import zipfile
import pandas as pd

with zipfile.ZipFile("house-price-dataset-of-india.zip", "r") as zip_ref:

    # extracting content in the zipfile
    zip_ref.extractall()

    # reading the contents of the "NetFlix.csv"
    df = pd.read_csv("House Price India.csv")

    # printing first 6 rows of the dataframe
    df.head(6)
```

```
[11]:
```

	id	Date	number of bedrooms	number of bathrooms	living area	\
0	6762810145	42491		5	2.50	3650
1	6762810635	42491		4	2.50	2920
2	6762810998	42491		5	2.75	2910
3	6762812605	42491		4	2.50	3310
4	6762812919	42491		3	2.00	2710
5	6762813105	42491		3	2.50	2600

	lot area	number of floors	waterfront present	number of views	\
0	9050	2.0	0	4	
1	4000	1.5	0	0	
2	9480	1.5	0	0	
3	42998	2.0	0	0	
4	4500	1.5	0	0	
5	4750	1.0	0	0	

	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

	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	

	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 rows x 23 columns]

```
[12]: 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

[13]: df.head()

	id	Date	number of bedrooms	number of bathrooms	living area	\
0	6762810145	42491	5	2.50	3650	
1	6762810635	42491	4	2.50	2920	
2	6762810998	42491	5	2.75	2910	
3	6762812605	42491	4	2.50	3310	
4	6762812919	42491	3	2.00	2710	

	lot area	number of floors	waterfront present	number of views	\
0	9050	2.0	0	4	
1	4000	1.5	0	0	
2	9480	1.5	0	0	
3	42998	2.0	0	0	
4	4500	1.5	0	0	

	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

   Latitude  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

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

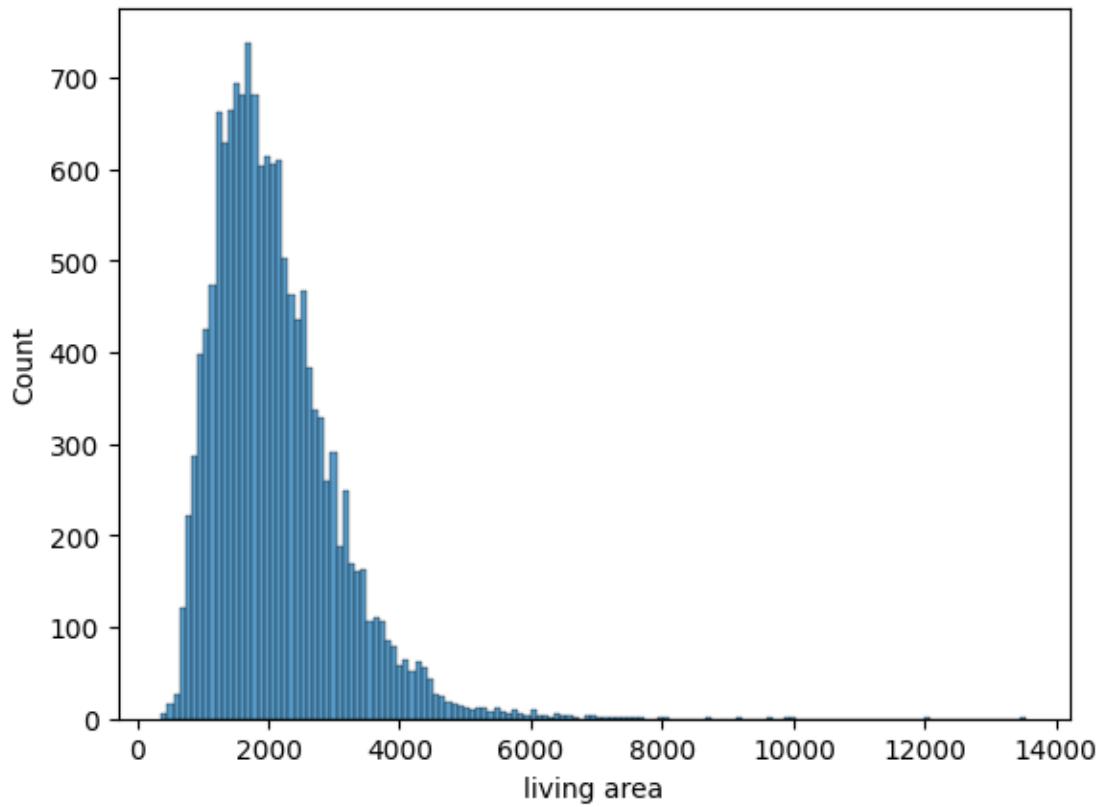
[5 rows x 23 columns]

```

Univariate

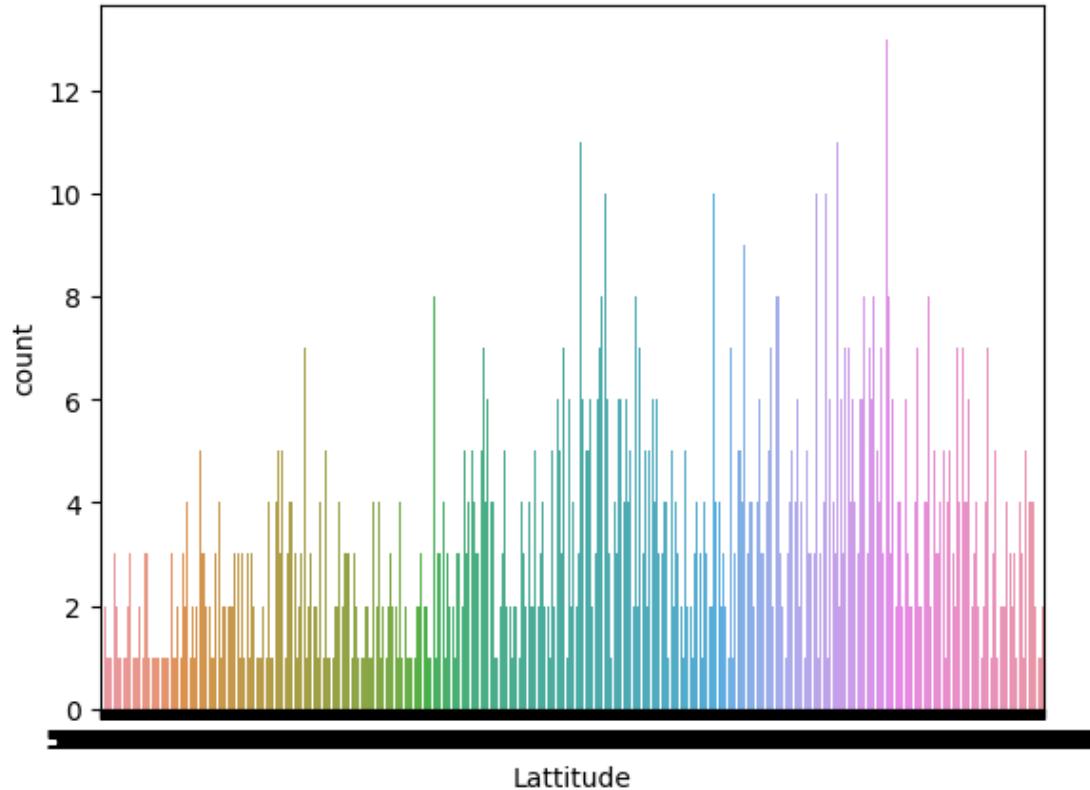
```
[16]: sns.histplot(df["living area"])
```

```
[16]: <Axes: xlabel='living area', ylabel='Count'>
```



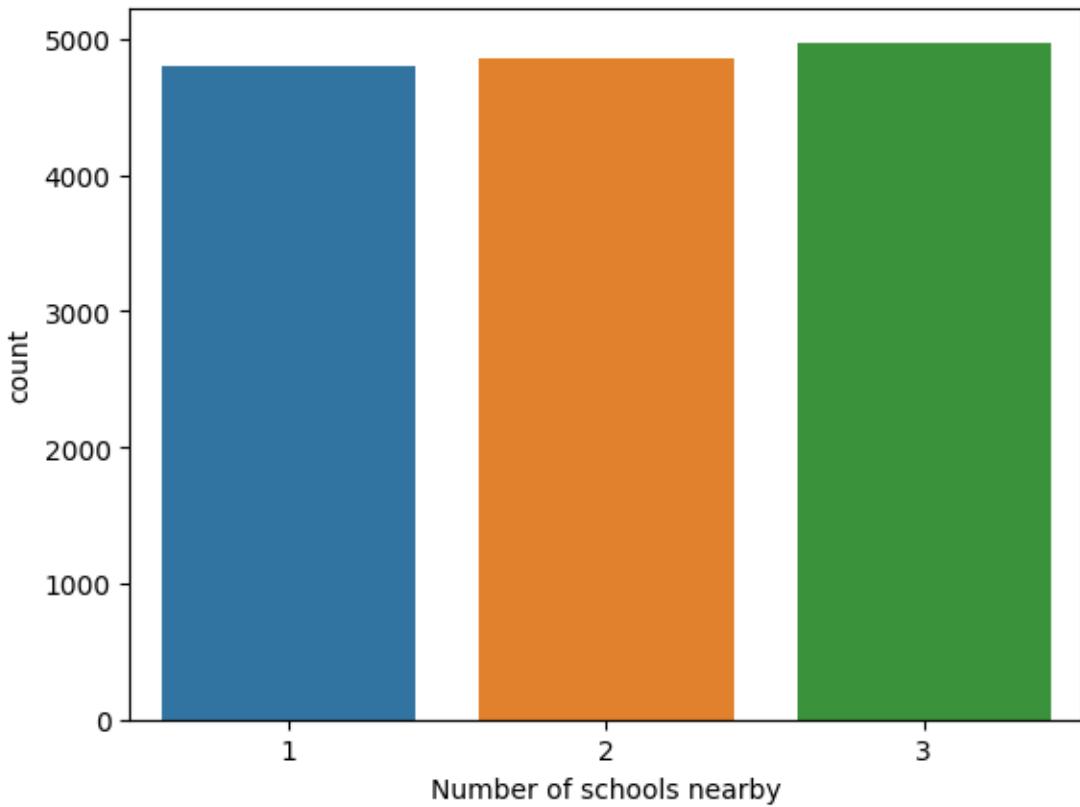
```
[17]: sns.countplot(x=df['Latitude'])
```

```
[17]: <Axes: xlabel='Latitude', ylabel='count'>
```



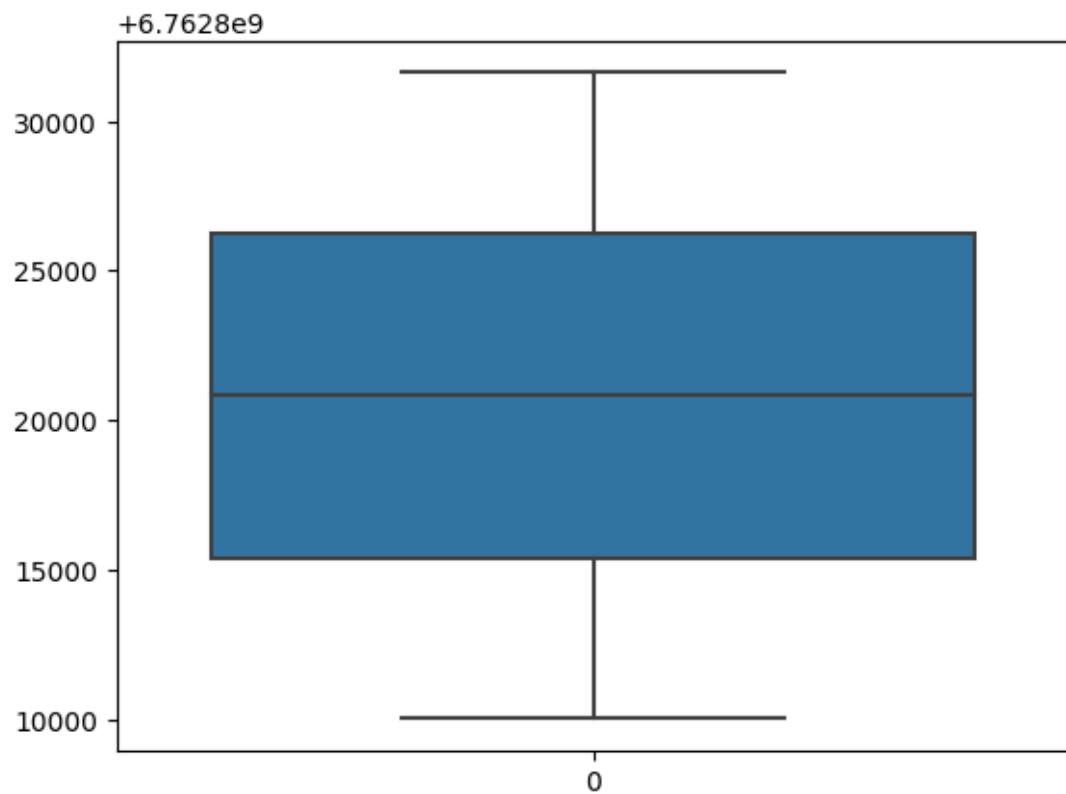
```
[18]: sns.countplot(x=df['Number of schools nearby'])
```

```
[18]: <Axes: xlabel='Number of schools nearby', ylabel='count'>
```

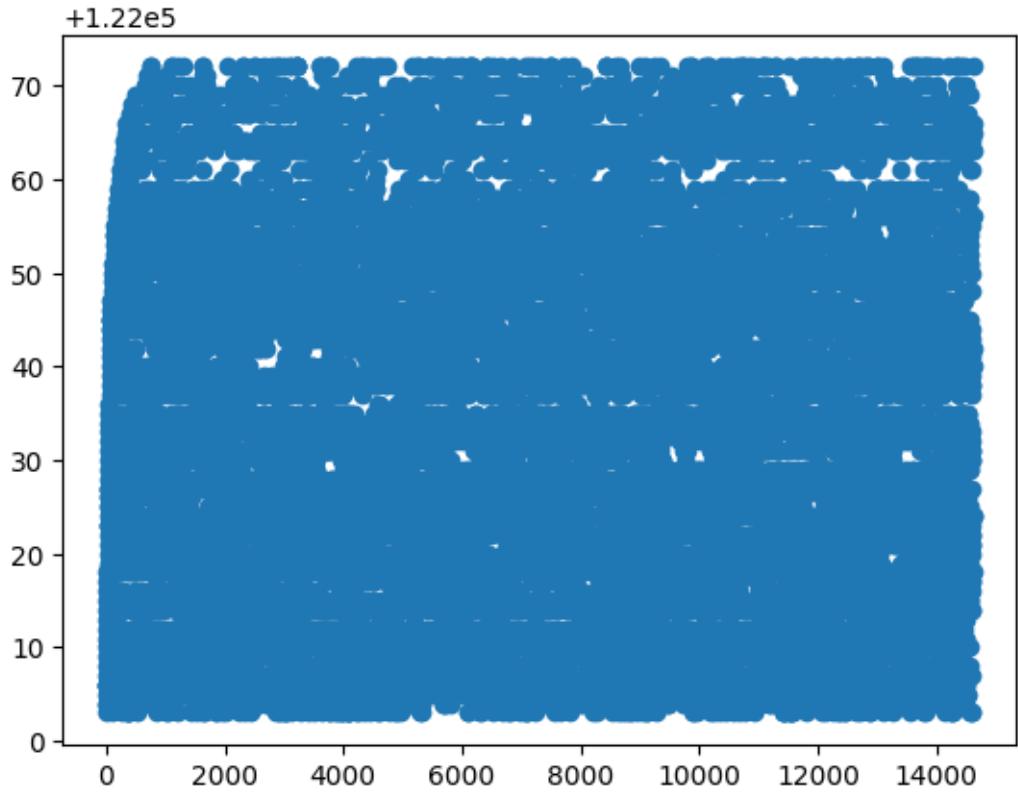


```
[19]: sns.boxplot(df.id)
```

```
[19]: <Axes: >
```

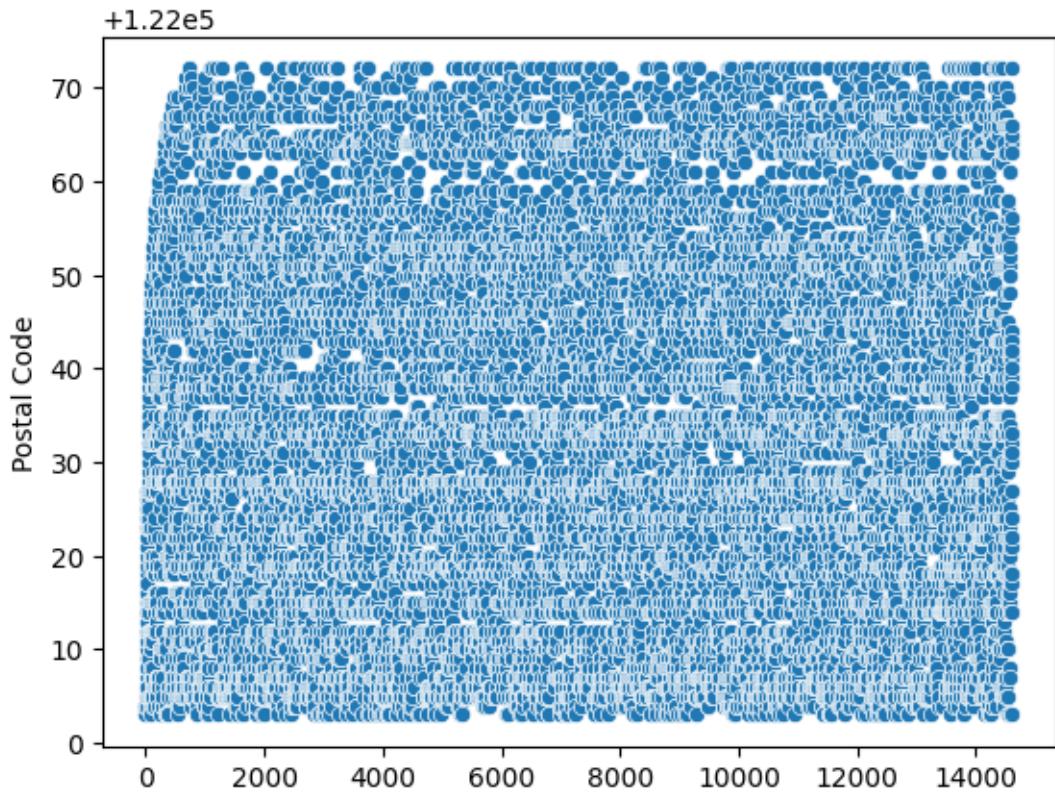


```
[ ]: plt.scatter(df.index,df['Postal Code'])
plt.show()
```



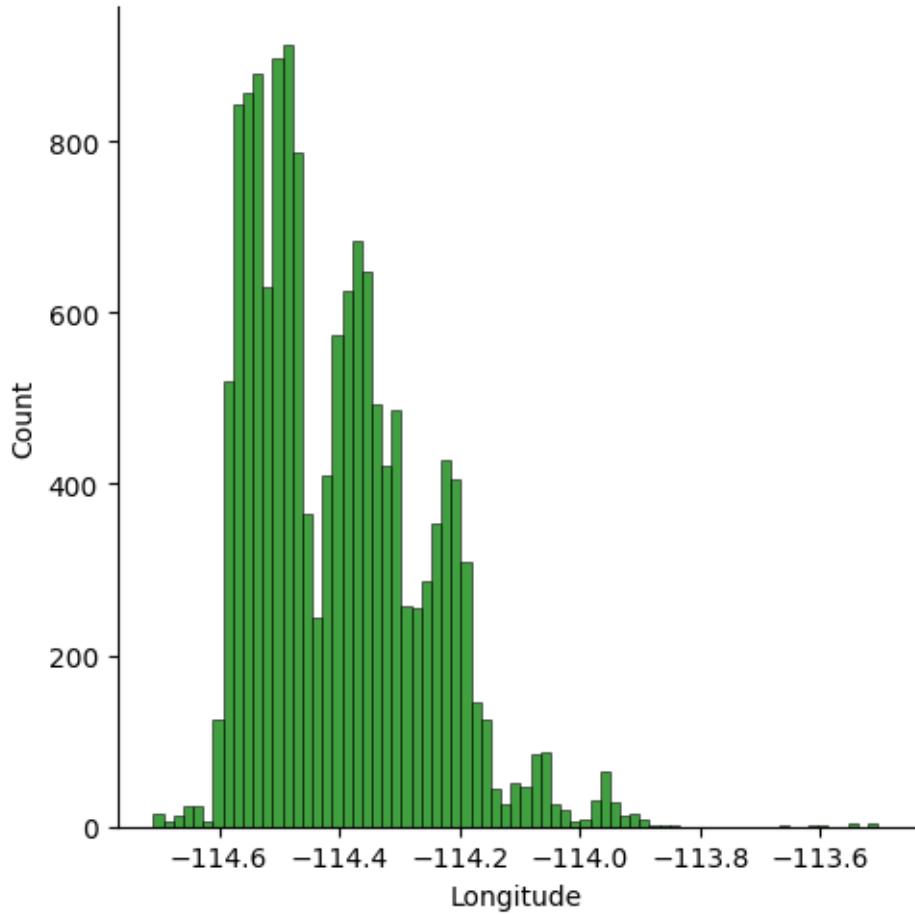
```
[ ]: sns.scatterplot(data=df,x=df.index, y=df['Postal Code'])
```

```
[ ]: <Axes: ylabel='Postal Code'>
```



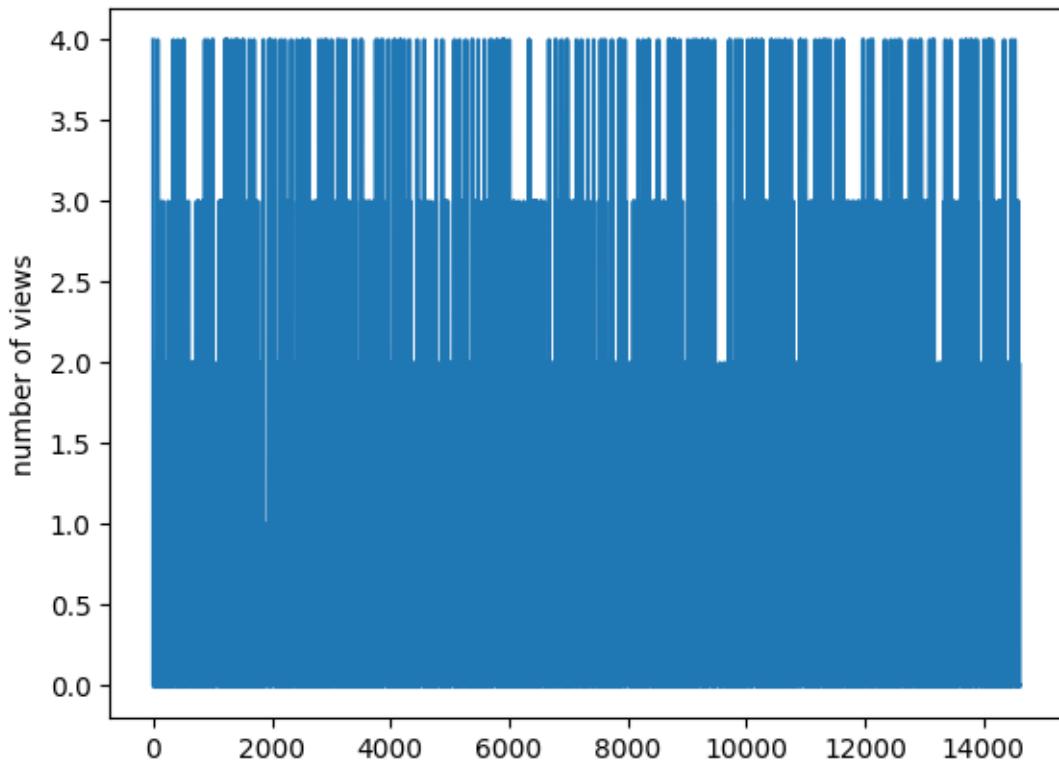
```
[22]: sns.displot(df['Longitude'], color='green')
```

```
[22]: <seaborn.axisgrid.FacetGrid at 0x1d9278b4310>
```



```
[25]: sns.lineplot(df['number of views'])
```

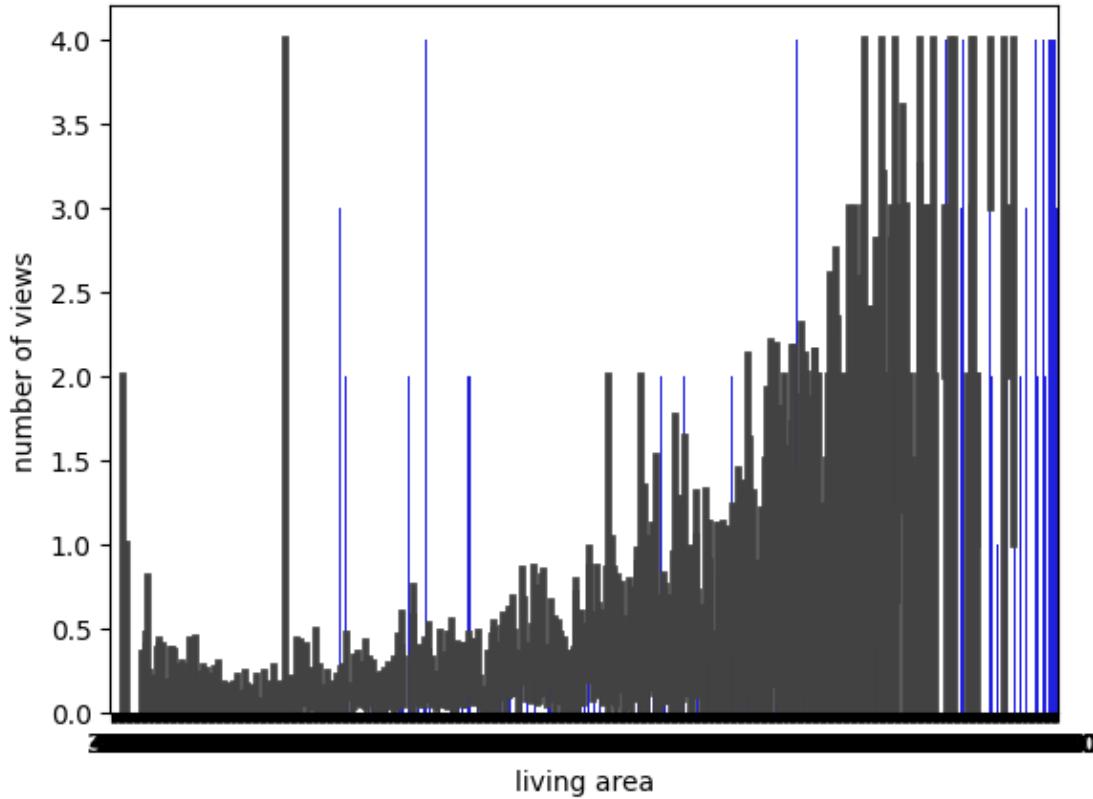
```
[25]: <Axes: ylabel='number of views'>
```



Bivariate

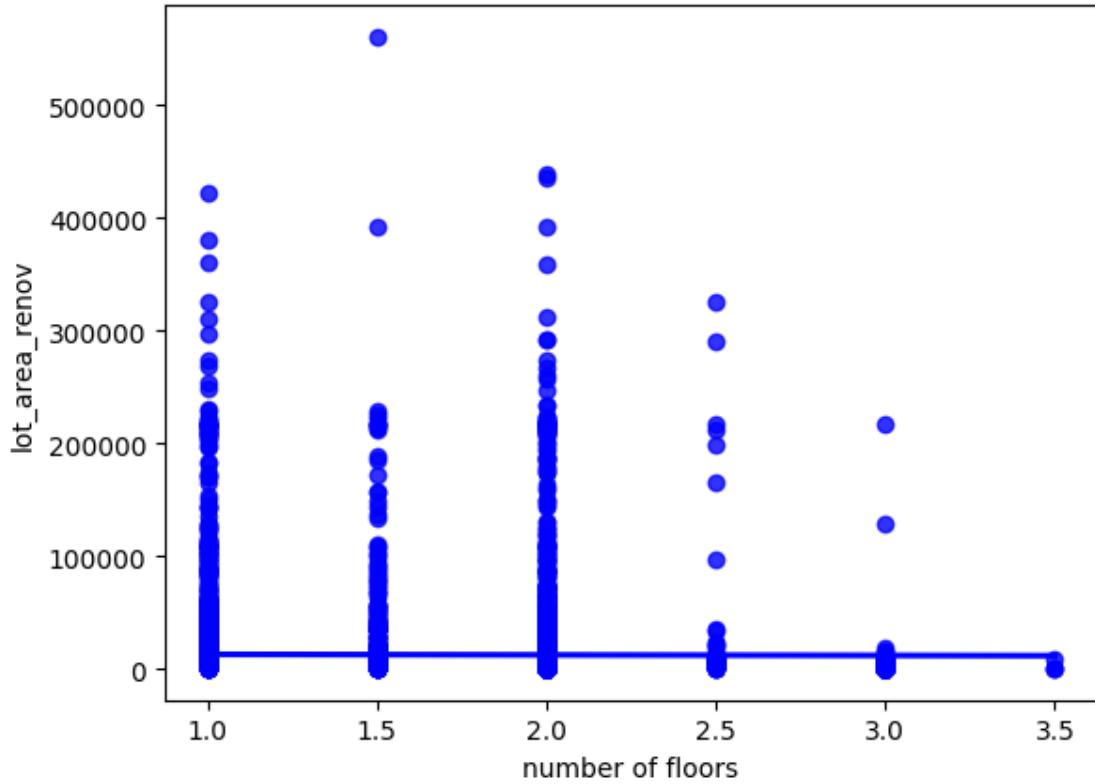
```
[23]: sns.barplot(x='living area',y='number of views',data=df,color='blue')
```

```
[23]: <Axes: xlabel='living area', ylabel='number of views'>
```



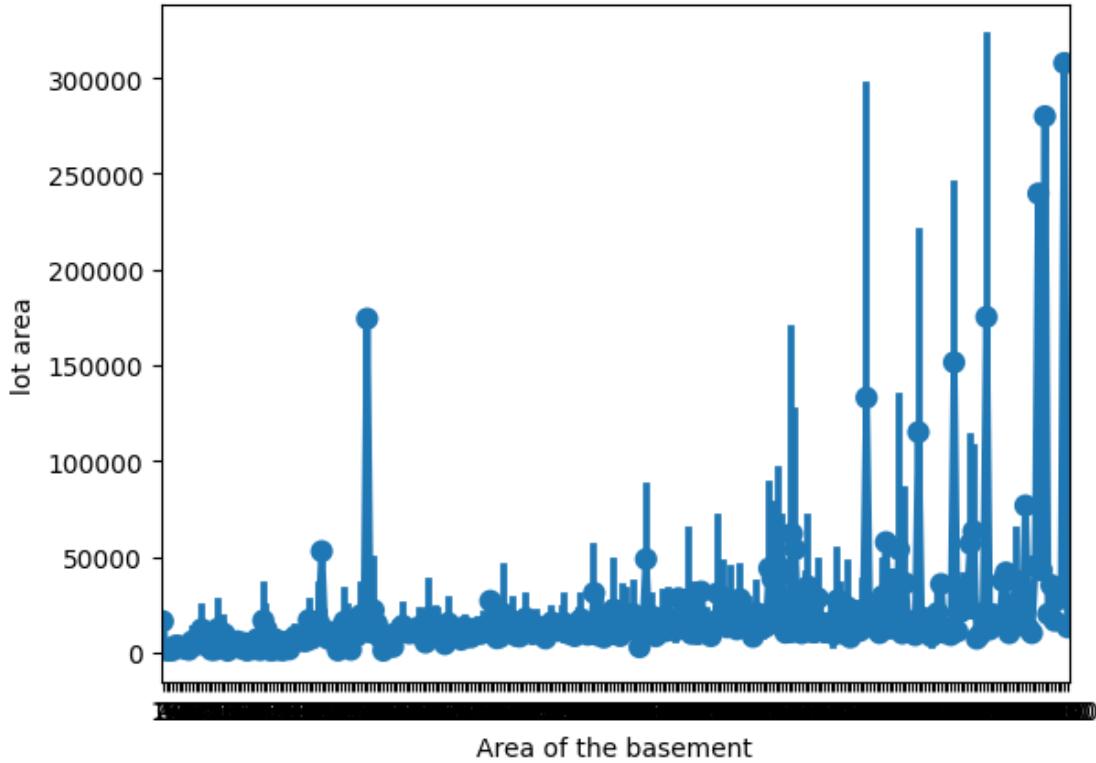
```
[26]: sns.regplot(x=df['number of floors'],y=df['lot_area_renov'],color='blue')
```

```
[26]: <Axes: xlabel='number of floors', ylabel='lot_area_renov'>
```



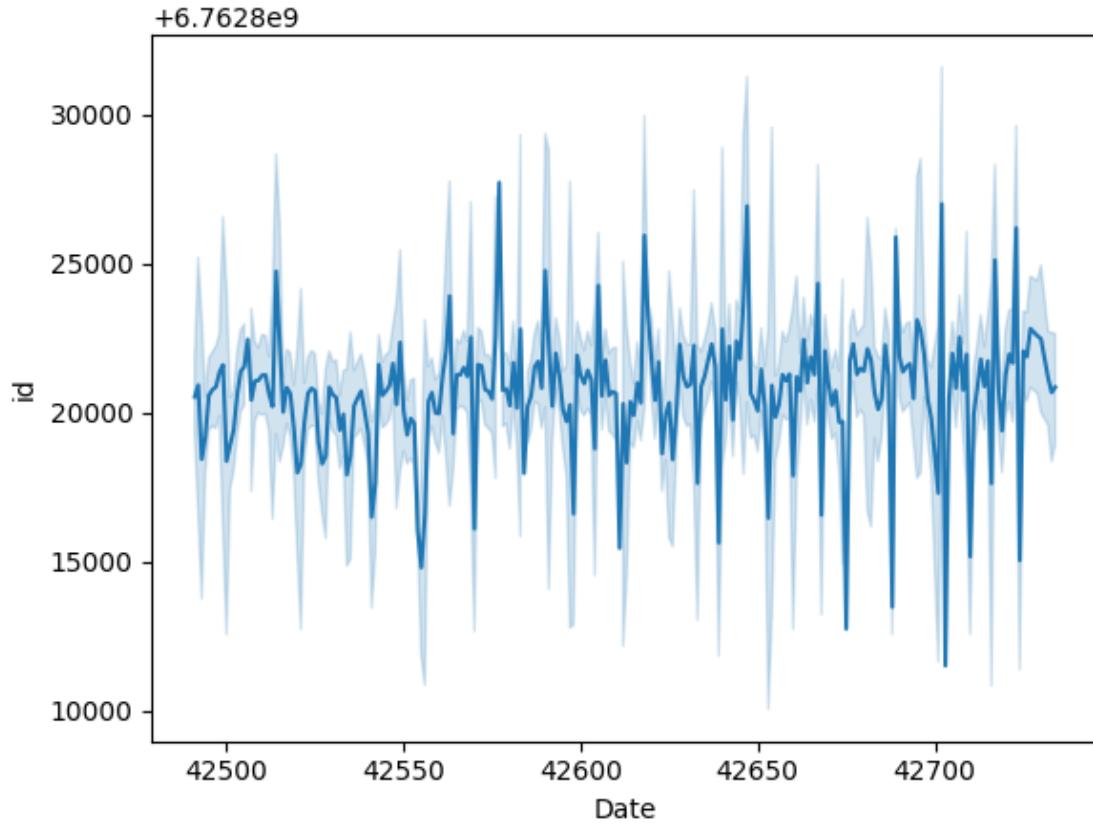
```
[27]: sns.pointplot(x=df['Area of the basement'],y=df['lot area'])
```

```
[27]: <Axes: xlabel='Area of the basement', ylabel='lot area'>
```



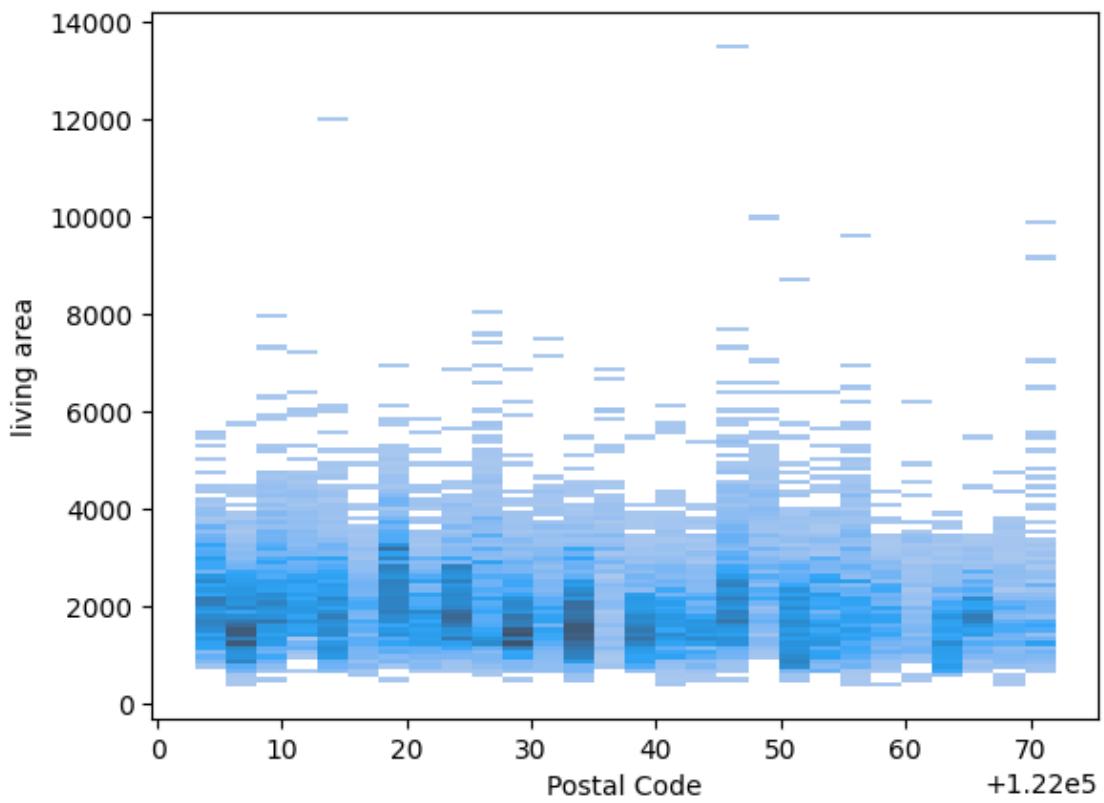
```
[24]: sns.lineplot(x=df['Date'], y=df['id'])
```

```
[24]: <Axes: xlabel='Date', ylabel='id'>
```



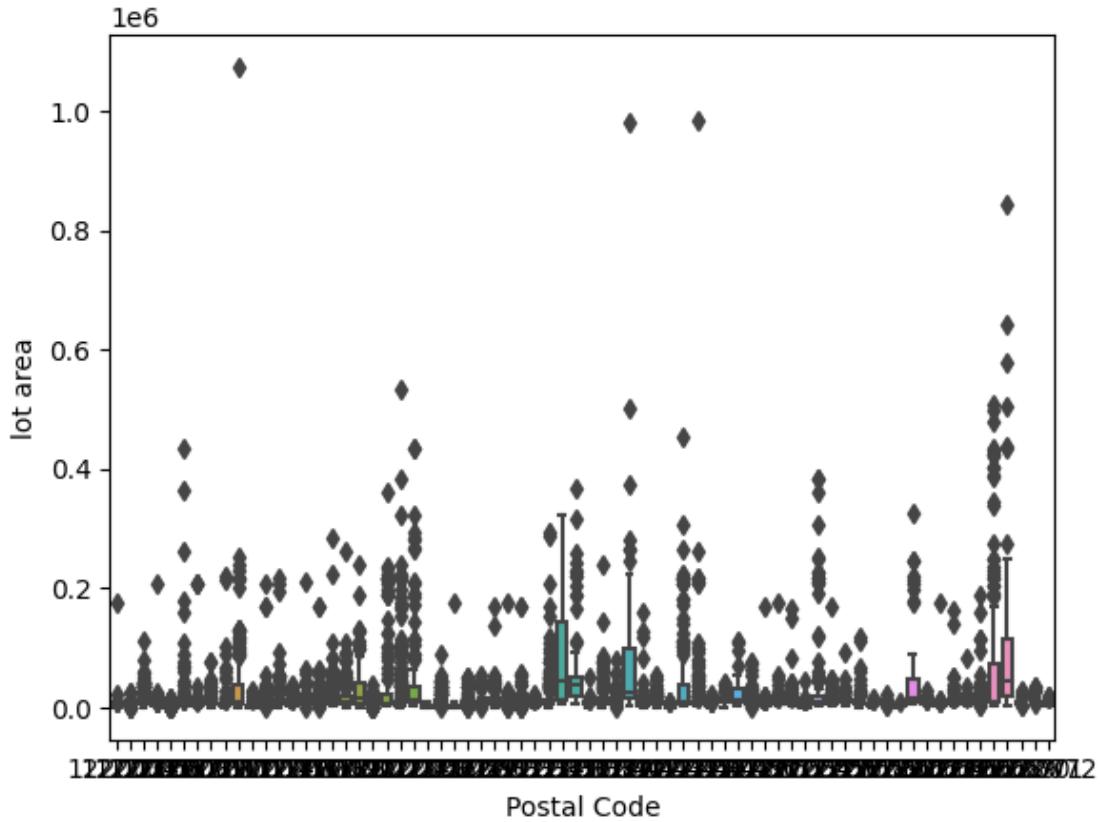
```
[ ]: sns.histplot(data=df,x=df['Postal Code'],y=df['living area'])
```

```
[ ]: <Axes: xlabel='Postal Code', ylabel='living area'>
```



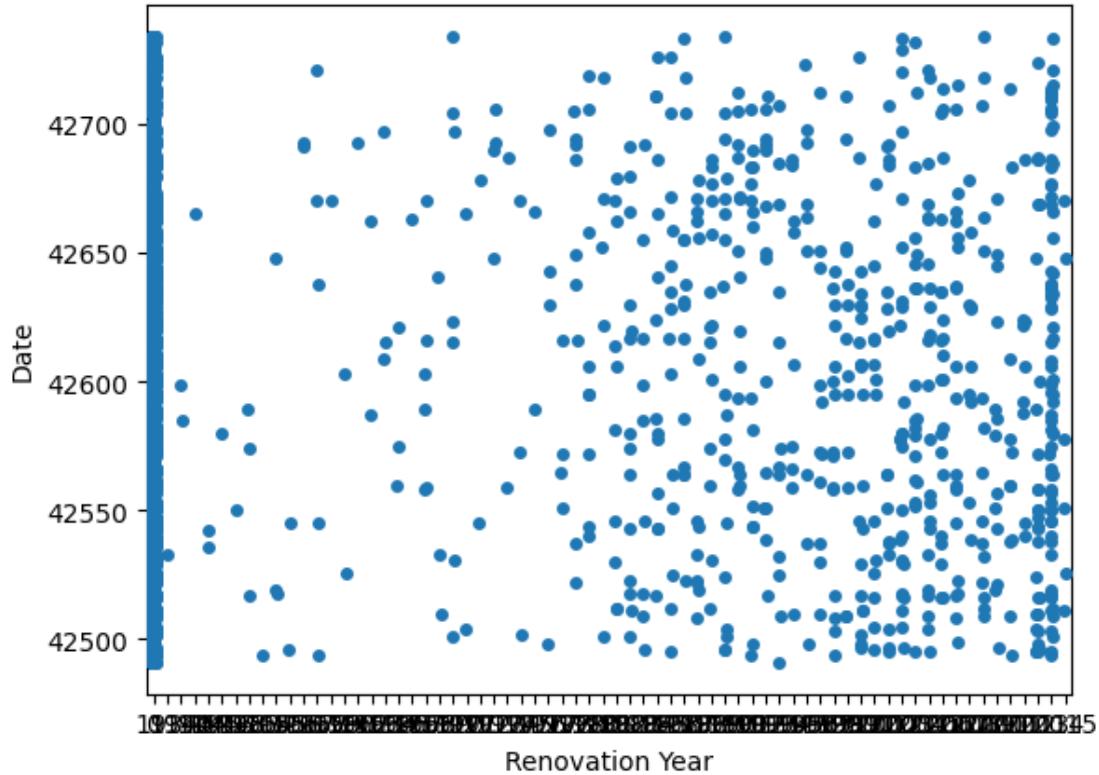
```
[20]: sns.boxplot(x=df['Postal Code'],y=df['lot area'])
```

```
[20]: <Axes: xlabel='Postal Code', ylabel='lot area'>
```



```
[21]: sns.stripplot(x=df['Renovation Year'], y=df['Date'])
```

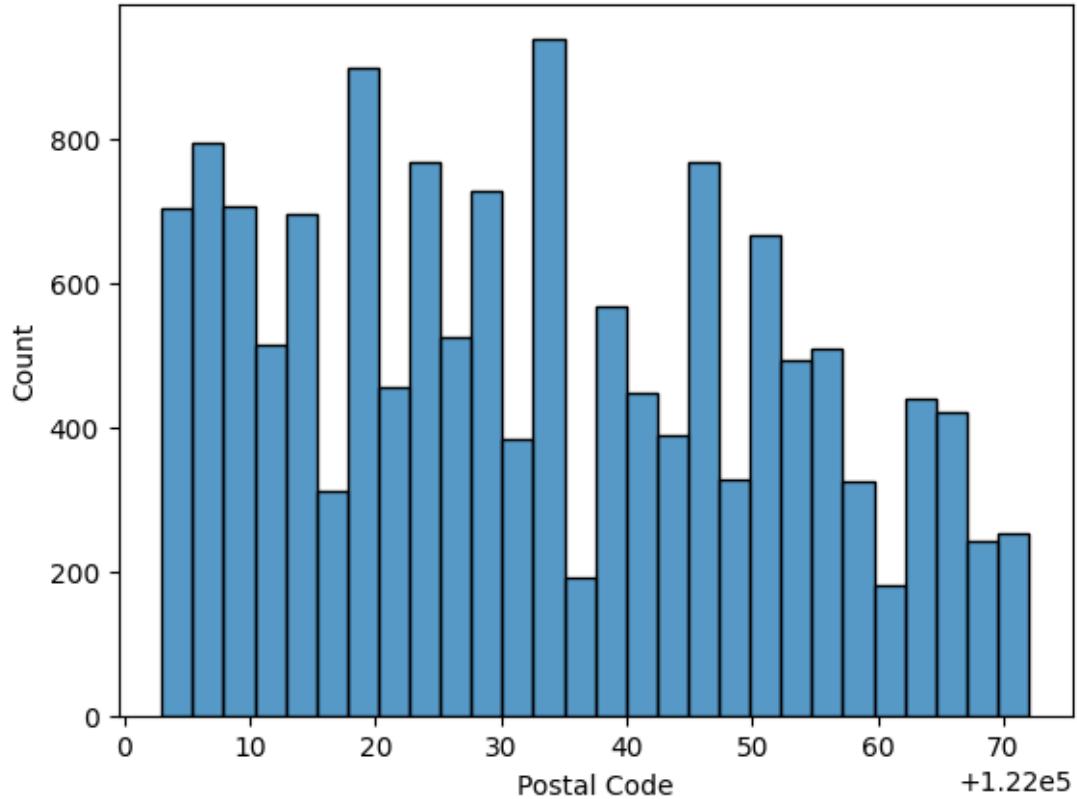
```
[21]: <Axes: xlabel='Renovation Year', ylabel='Date'>
```



```
[29]: ##Univariate
```

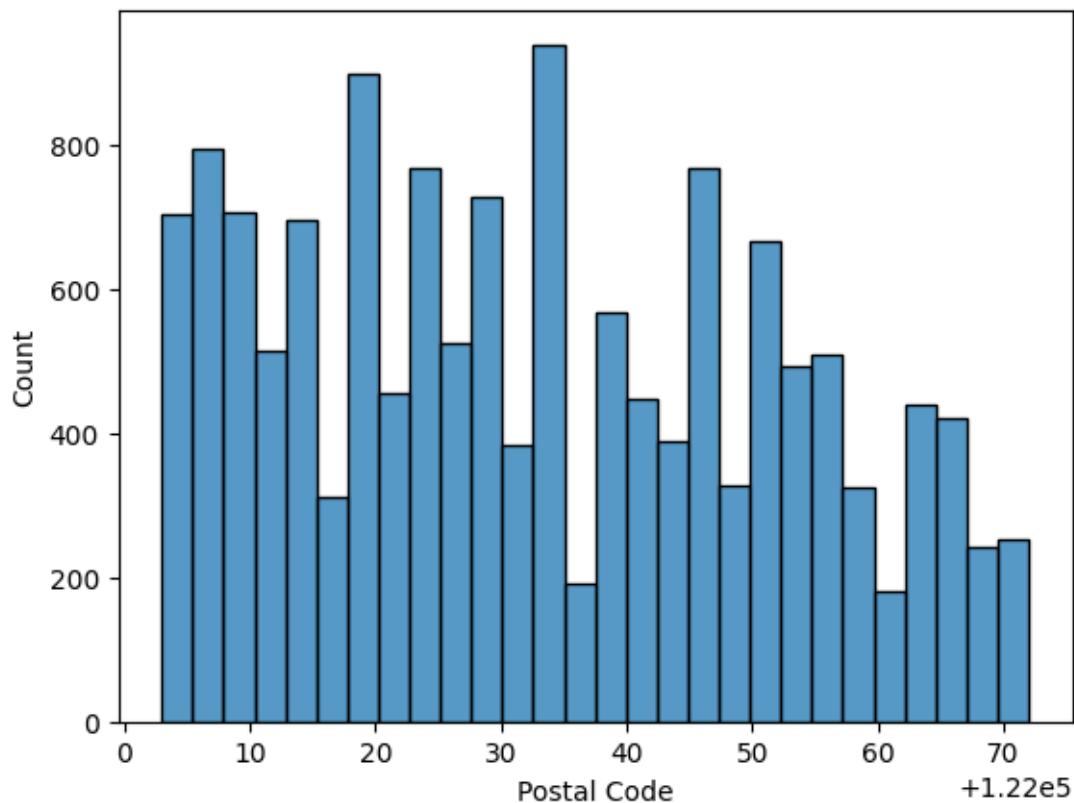
```
[ ]: sns.histplot(df['Postal Code'])
```

```
[ ]: <Axes: xlabel='Postal Code', ylabel='Count'>
```



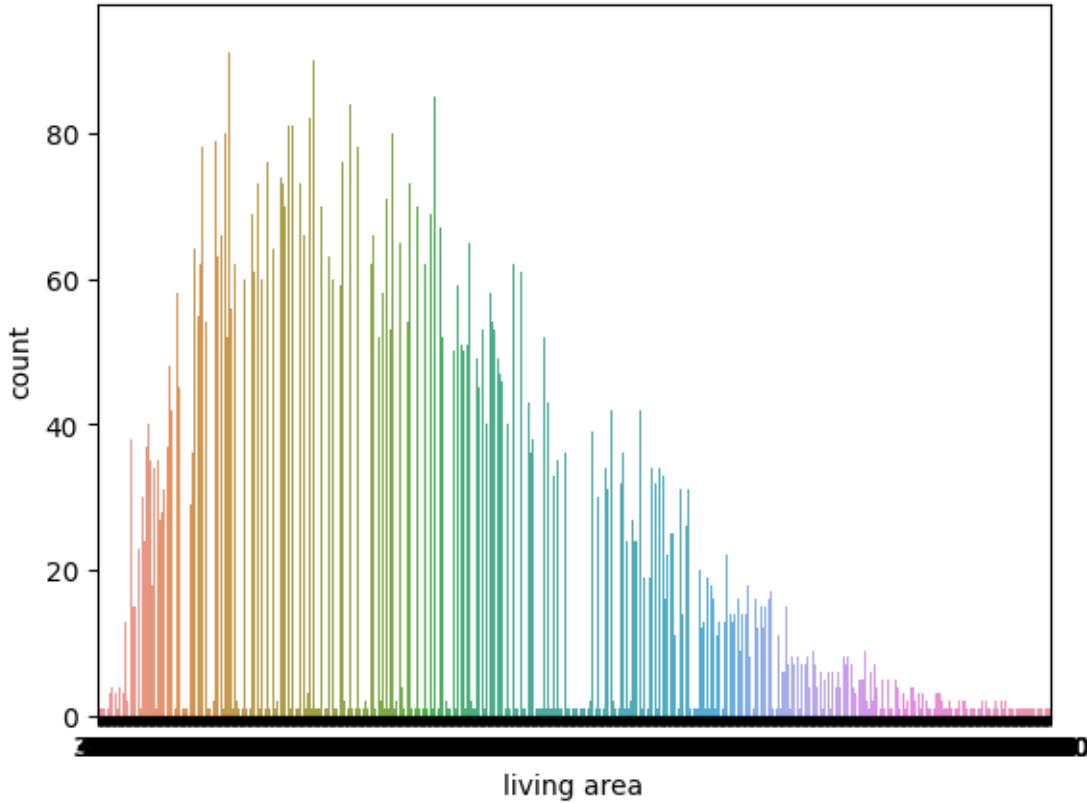
```
[ ]: sns.histplot(x='Postal Code', data=df)
```

```
[ ]: <Axes: xlabel='Postal Code', ylabel='Count'>
```



```
[ ]: sns.countplot(x='living area', data=df)
```

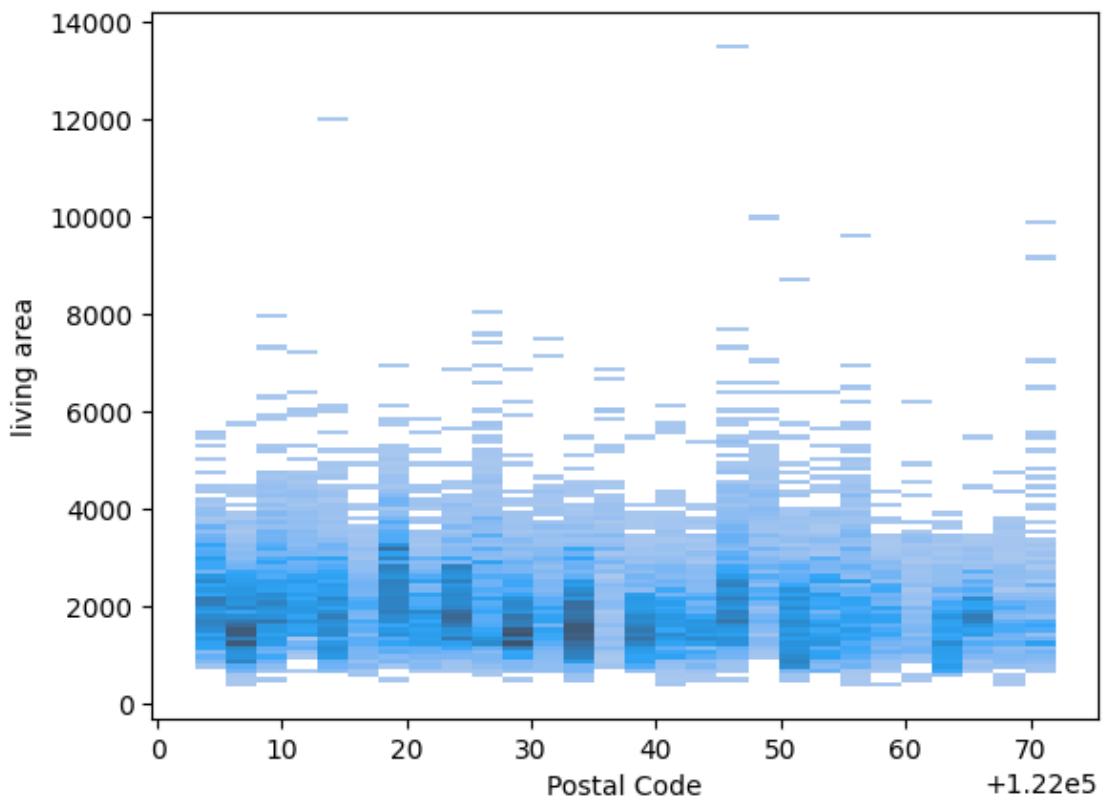
```
[ ]: <Axes: xlabel='living area', ylabel='count'>
```



```
[30]: ##Bivariate
```

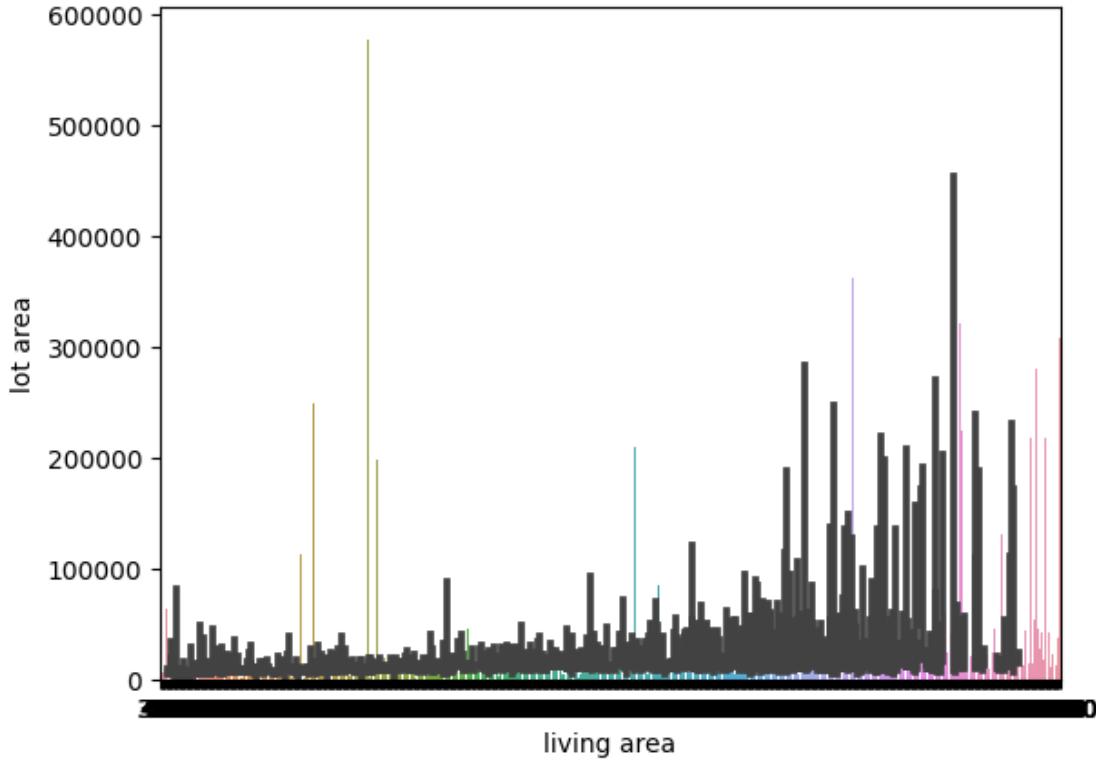
```
[ ]: sns.histplot(data=df,x=df['Postal Code'],y=df['living area'])
```

```
[ ]: <Axes: xlabel='Postal Code', ylabel='living area'>
```



```
[ ]: sns.barplot(x='living area',y='lot area',data=df)
```

```
[ ]: <Axes: xlabel='living area', ylabel='lot area'>
```



```
[ ]: 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
max	80.000000	7.700000e+06

[8 rows x 23 columns]

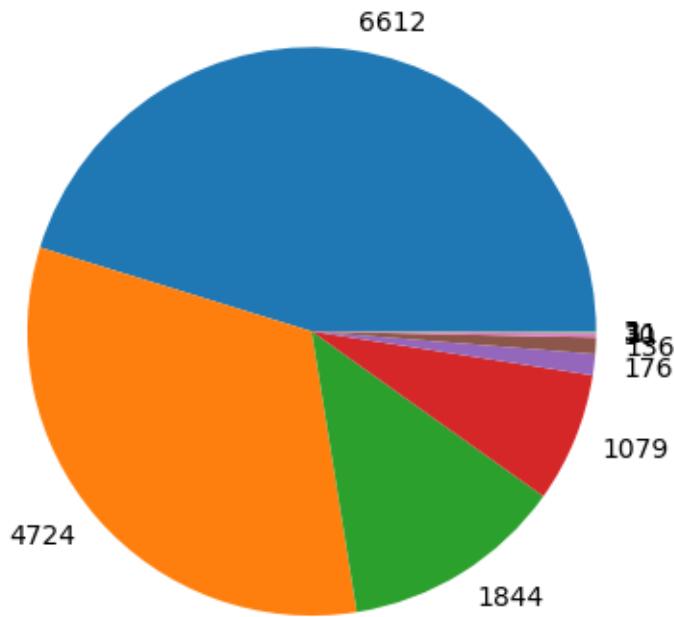
```
[ ]: plt.pie(df['number of bedrooms'].value_counts(), labels=df['number of bedrooms'].value_counts())
```

```
[ ]: ([<matplotlib.patches.Wedge at 0x2c6e67fda10>,
<matplotlib.patches.Wedge at 0x2c6e6854890>,
```

```

<matplotlib.patches.Wedge at 0x2c6e6857610>,
<matplotlib.patches.Wedge at 0x2c6e688dc10>,
<matplotlib.patches.Wedge at 0x2c6e6878a50>,
<matplotlib.patches.Wedge at 0x2c6e687b910>,
<matplotlib.patches.Wedge at 0x2c6e686ac10>,
<matplotlib.patches.Wedge at 0x2c6e687dfd0>,
<matplotlib.patches.Wedge at 0x2c6e6878d50>,
<matplotlib.patches.Wedge at 0x2c6e688b4d0>,
<matplotlib.patches.Wedge at 0x2c6e67a5a10>,
<matplotlib.patches.Wedge at 0x2c6e576f710>],
[Text(0.16436940858108168, 1.0876500804591087, '6612'),
Text(-0.8305078126467117, -0.7212882732533329, '4724'),
Text(0.5802807019422348, -0.934491469706079, '1844'),
Text(1.0186480552346253, -0.4151579694125067, '1079'),
Text(1.0924376813878238, -0.12876300821274694, '176'),
Text(1.0986097001624984, -0.0552876723046412, '136'),
Text(1.0998825698043138, -0.01607272972021759, '30'),
Text(1.0999814860504655, -0.006382033078075224, '11'),
Text(1.0999957079511045, -0.0030728633469087586, '3'),
Text(1.0999987555333315, -0.0016546374594783731, '3'),
Text(1.0999997714107155, -0.0007091518693299728, '1'),
Text(1.099999974595844, -0.00023640884670048204, '1')])

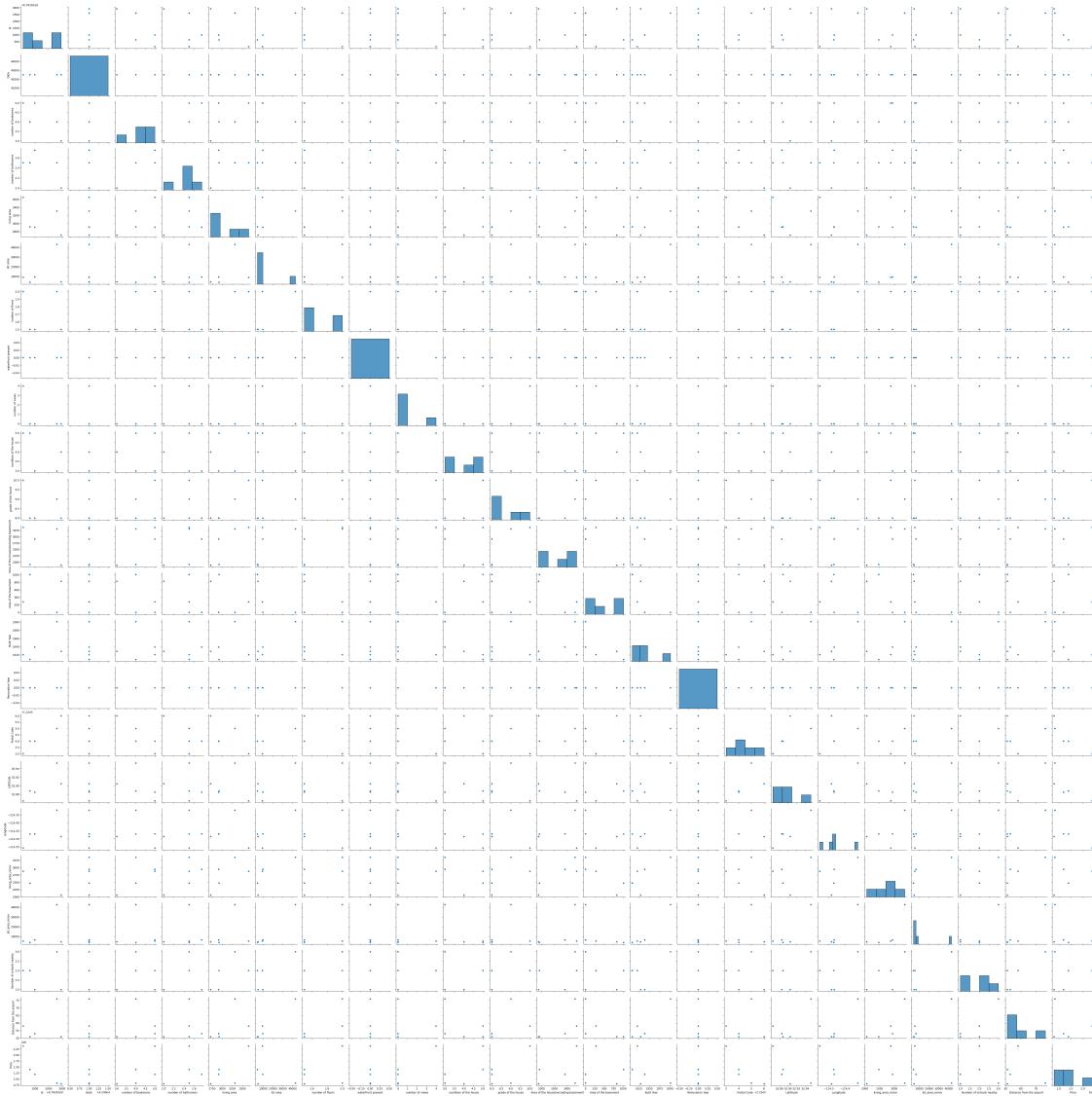
```



Multivariate

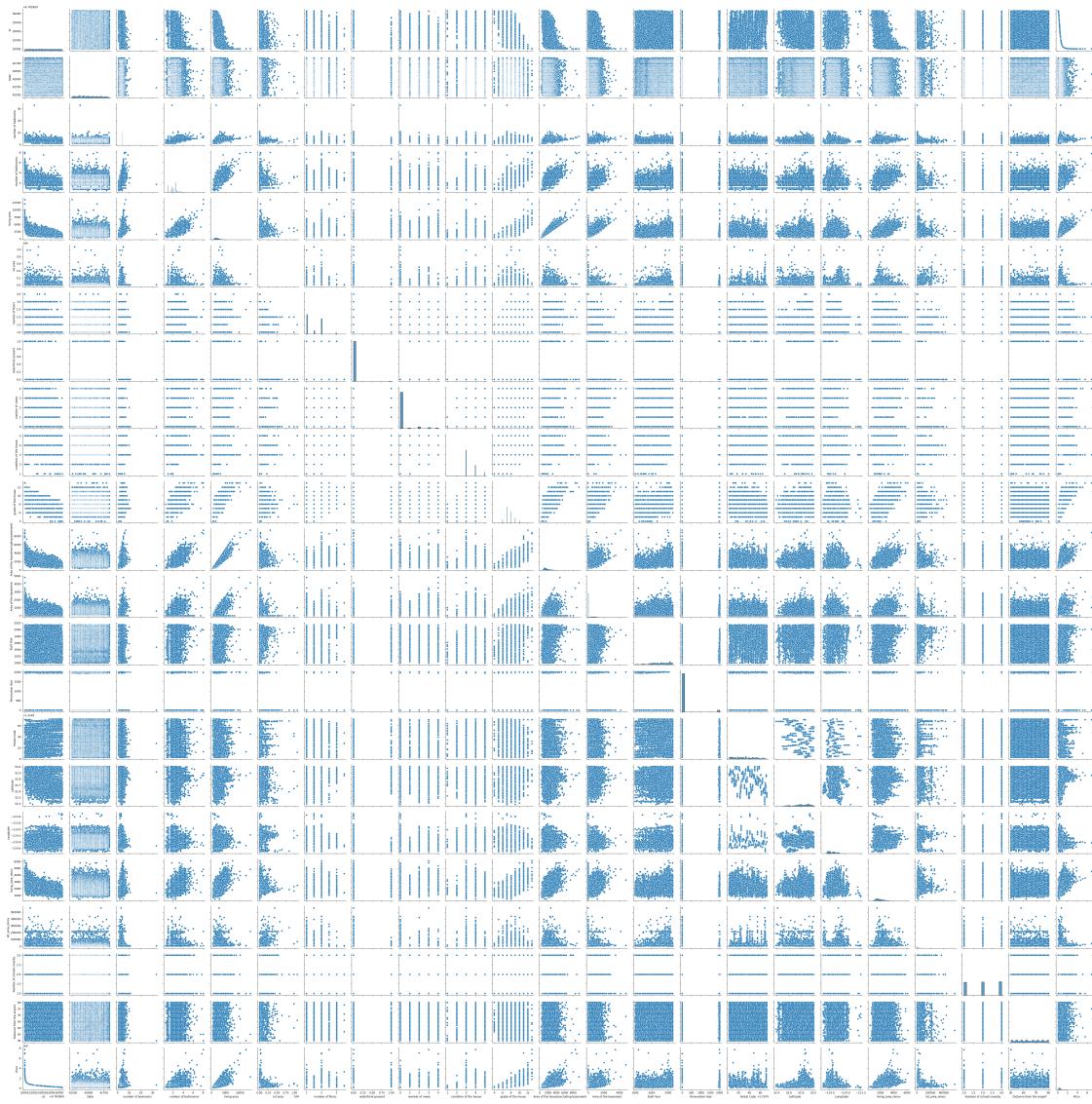
```
[ ]: sns.pairplot(df.head())
```

```
[ ]: <seaborn.axisgrid.PairGrid at 0x2c6e4a9b090>
```



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

```
[ ]: <seaborn.axisgrid.PairGrid at 0x2c68057b090>
```



```
[28]: sns.heatmap(df.corr(), annot=True)
```

```
[28]: <Axes: >
```



[31]: df.isnull()

```
[31]:      id   Date  number of bedrooms  number of bathrooms  living area \
0    False  False           False           False       False
1    False  False           False           False       False
2    False  False           False           False       False
3    False  False           False           False       False
4    False  False           False           False       False
...
14615  False  False           ...           False       False
14616  False  False           False           False       False
14617  False  False           False           False       False
14618  False  False           False           False       False
14619  False  False           False           False       False

      lot area  number of floors  waterfront present  number of views \
0        False           False           False           False
```

1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
14615	False	False	False	False	False
14616	False	False	False	False	False
14617	False	False	False	False	False
14618	False	False	False	False	False
14619	False	False	False	False	False
condition of the house ... Built Year Renovation Year Postal Code \					
0	False	...	False	False	False
1	False	...	False	False	False
2	False	...	False	False	False
3	False	...	False	False	False
4	False	...	False	False	False
...	...	...	...	...	...
14615	False	...	False	False	False
14616	False	...	False	False	False
14617	False	...	False	False	False
14618	False	...	False	False	False
14619	False	...	False	False	False
Lattitude Longitude living_area_renov lot_area_renov \					
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
14615	False	False	False	False	False
14616	False	False	False	False	False
14617	False	False	False	False	False
14618	False	False	False	False	False
14619	False	False	False	False	False
Number of schools nearby Distance from the airport Price					
0	False	...	False	False	False
1	False	...	False	False	False
2	False	...	False	False	False
3	False	...	False	False	False
4	False	...	False	False	False
...	...	...	...	...	...
14615	False	...	False	False	False
14616	False	...	False	False	False
14617	False	...	False	False	False

```

14618          False      False  False
14619          False      False  False

```

[14620 rows x 23 columns]

[32]: df.notnull()

```

[32]:      id  Date  number of bedrooms  number of bathrooms  living area \
0      True  True                  True                  True  True
1      True  True                  True                  True  True
2      True  True                  True                  True  True
3      True  True                  True                  True  True
4      True  True                  True                  True  True
...
14615  True  True                  ...                  True  True
14616  True  True                  True                  True  True
14617  True  True                  True                  True  True
14618  True  True                  True                  True  True
14619  True  True                  True                  True  True

      lot area  number of floors  waterfront present  number of views \
0      True        True          True          True  True
1      True        True          True          True  True
2      True        True          True          True  True
3      True        True          True          True  True
4      True        True          True          True  True
...
14615  True        ...          ...          ...  ...
14616  True        ...          ...          ...  ...
14617  True        ...          ...          ...  ...
14618  True        ...          ...          ...  ...
14619  True        ...          ...          ...  ...

      condition of the house  ...  Built Year  Renovation Year  Postal Code \
0                      True  ...  True    True      True  True
1                      True  ...  True    True      True  True
2                      True  ...  True    True      True  True
3                      True  ...  True    True      True  True
4                      True  ...  True    True      True  True
...
14615                  ...  ...  ...  ...
14616                  ...  ...  ...  ...
14617                  ...  ...  ...  ...
14618                  ...  ...  ...  ...
14619                  ...  ...  ...  ...

      Latitude  Longitude  living_area_renov  lot_area_renov \

```

0	True	True	True	True
1	True	True	True	True
2	True	True	True	True
3	True	True	True	True
4	True	True	True	True
...	...	...	...	...
14615	True	True	True	True
14616	True	True	True	True
14617	True	True	True	True
14618	True	True	True	True
14619	True	True	True	True
Number of schools nearby Distance from the airport Price				
0		True	True	True
1		True	True	True
2		True	True	True
3		True	True	True
4		True	True	True
...	...	...	...	...
14615		True	True	True
14616		True	True	True
14617		True	True	True
14618		True	True	True
14619		True	True	True

[14620 rows x 23 columns]

[33]: df.fillna(0)

	id	Date	number of bedrooms	number of bathrooms	\
0	6762810145	42491	5	2.50	
1	6762810635	42491	4	2.50	
2	6762810998	42491	5	2.75	
3	6762812605	42491	4	2.50	
4	6762812919	42491	3	2.00	
...	...	...	...	...	
14615	6762830250	42734	2	1.50	
14616	6762830339	42734	3	2.00	
14617	6762830618	42734	2	1.00	
14618	6762830709	42734	4	1.00	
14619	6762831463	42734	3	1.00	
living area lot area number of floors waterfront present \					
0	3650	9050	2.0	0	
1	2920	4000	1.5	0	
2	2910	9480	1.5	0	
3	3310	42998	2.0	0	

4	2710	4500	1.5	0
...	...	...	...	...
14615	1556	20000	1.0	0
14616	1680	7000	1.5	0
14617	1070	6120	1.0	0
14618	1030	6621	1.0	0
14619	900	4770	1.0	0
number of views   condition of the house ... Built Year \				
0	4		5	... 1921
1	0		5	... 1909
2	0		3	... 1939
3	0		3	... 2001
4	0		4	... 1929
...	...	...	...	...
14615	0		4	... 1957
14616	0		4	... 1968
14617	0		3	... 1962
14618	0		4	... 1955
14619	0		3	... 1969
Renovation Year Postal Code Lattitude Longitude living_area_renov \				
0	0	122003	52.8645	-114.557 2880
1	0	122004	52.8878	-114.470 2470
2	0	122004	52.8852	-114.468 2940
3	0	122005	52.9532	-114.321 3350
4	0	122006	52.9047	-114.485 2060
...	...	...	...	...
14615	0	122066	52.6191	-114.472 2250
14616	0	122072	52.5075	-114.393 1540
14617	0	122056	52.7289	-114.507 1130
14618	0	122042	52.7157	-114.411 1420
14619	2009	122018	52.5338	-114.552 900
lot_area_renov Number of schools nearby Distance from the airport \				
0	5400		2	58
1	4000		2	51
2	6600		1	53
3	42847		3	76
4	4500		1	51
...	...	...	...	...
14615	17286		3	76
14616	7480		3	59
14617	6120		2	64
14618	6631		3	54
14619	3480		2	55

	Price
0	2380000
1	1400000
2	1200000
3	838000
4	805000
...	...
14615	221700
14616	219200
14617	209000
14618	205000
14619	146000

[14620 rows x 23 columns]