

In [1]:

```
import pandas as pd
```

In [10]:

```
df=pd.read_csv(r"C:\Users\sadiy\Downloads\Datascience\pandas\dataframe\home_data.csv")
```

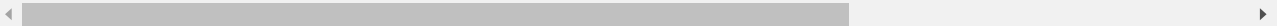
In [11]:

```
df
```

Out[11]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement
0	7129300520	20141013T000000	221900	3	1.00	1180	5650	1.0	0	0	...	7	1180	0
1	6414100192	20141209T000000	538000	3	2.25	2570	7242	2.0	0	0	...	7	2170	400
2	5631500400	20150225T000000	180000	2	1.00	770	10000	1.0	0	0	...	6	770	0
3	2487200875	20141209T000000	604000	4	3.00	1960	5000	1.0	0	0	...	7	1050	910
4	1954400510	20150218T000000	510000	3	2.00	1680	8080	1.0	0	0	...	8	1680	0
...
21608	263000018	20140521T000000	360000	3	2.50	1530	1131	3.0	0	0	...	8	1530	0
21609	6600060120	20150223T000000	400000	4	2.50	2310	5813	2.0	0	0	...	8	2310	0
21610	1523300141	20140623T000000	402101	2	0.75	1020	1350	2.0	0	0	...	7	1020	0
21611	291310100	20150116T000000	400000	3	2.50	1600	2388	2.0	0	0	...	8	1600	0
21612	1523300157	20141015T000000	325000	2	0.75	1020	1076	2.0	0	0	...	7	1020	0

21613 rows × 15 columns



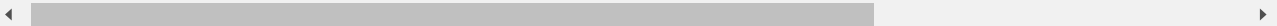
In [12]:

```
df.head()
```

Out[12]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement	yr_bui
0	7129300520	20141013T000000	221900	3	1.00	1180	5650	1.0	0	0	...	7	1180	0	195
1	6414100192	20141209T000000	538000	3	2.25	2570	7242	2.0	0	0	...	7	2170	400	195
2	5631500400	20150225T000000	180000	2	1.00	770	10000	1.0	0	0	...	6	770	0	193
3	2487200875	20141209T000000	604000	4	3.00	1960	5000	1.0	0	0	...	7	1050	910	196
4	1954400510	20150218T000000	510000	3	2.00	1680	8080	1.0	0	0	...	8	1680	0	198

5 rows × 16 columns



In [14]:

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                    21613 non-null  int64
1   date                 21613 non-null  object
2   price                21613 non-null  int64
3   bedrooms             21613 non-null  int64
4   bathrooms            21613 non-null  float64
5   sqft_living          21613 non-null  int64
6   sqft_lot             21613 non-null  int64
7   floors               21613 non-null  float64
8   waterfront           21613 non-null  int64
9   view                 21613 non-null  int64
10  condition            21613 non-null  int64
11  grade                21613 non-null  int64
12  sqft_above           21613 non-null  int64
13  sqft_basement        21613 non-null  int64
14  yr_built             21613 non-null  int64
15  yr_renovated         21613 non-null  int64
16  zipcode              21613 non-null  int64
17  lat                  21613 non-null  float64
18  long                 21613 non-null  float64
19  sqft_living15        21613 non-null  int64
20  sqft_lot15           21613 non-null  int64
dtypes: float64(4), int64(16), object(1)
memory usage: 3.5+ MB
```

In [16]:

df.shape

Out[16]:

(21613, 21)

no of records

df.shape[0]

In [18]:

df.bathrooms

Out[18]:

```
0      1.00
1      2.25
2      1.00
3      3.00
4      2.00
...
21608   2.50
21609   2.50
21610   0.75
21611   2.50
21612   0.75
Name: bathrooms, Length: 21613, dtype: float64
```

changing the values in numeric

In [27]:

df[['bathrooms', 'floors']] = df[['bathrooms', 'floors']].apply(pd.to_numeric)

In [28]:

df.dtypes

Out[28]:

```
id                int64
date              object
price            int64
bedrooms         int64
bathrooms        float64
sqft_living      int64
sqft_lot         int64
floors           float64
waterfront       int64
view             int64
condition        int64
grade            int64
sqft_above       int64
sqft_basement    int64
yr_built         int64
yr_renovated     int64
zipcode          int64
lat              float64
long             float64
sqft_living15    int64
sqft_lot15       int64
dtype: object
```

In [29]:

df

Out[29]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement
0	7129300520	20141013T000000	221900	3	1.00	1180	5650	1.0	0	0	...	7	1180	0
1	6414100192	20141209T000000	538000	3	2.25	2570	7242	2.0	0	0	...	7	2170	400
2	5631500400	20150225T000000	180000	2	1.00	770	10000	1.0	0	0	...	6	770	0
3	2487200875	20141209T000000	604000	4	3.00	1960	5000	1.0	0	0	...	7	1050	910
4	1954400510	20150218T000000	510000	3	2.00	1680	8080	1.0	0	0	...	8	1680	0
...
21608	2630000018	20140521T000000	360000	3	2.50	1530	1131	3.0	0	0	...	8	1530	0
21609	6600060120	20150223T000000	400000	4	2.50	2310	5813	2.0	0	0	...	8	2310	0
21610	1523300141	20140623T000000	402101	2	0.75	1020	1350	2.0	0	0	...	7	1020	0
21611	291310100	20150116T000000	400000	3	2.50	1600	2388	2.0	0	0	...	8	1600	0
21612	1523300157	20141015T000000	325000	2	0.75	1020	1076	2.0	0	0	...	7	1020	0

21613 rows × 21 columns

In [21]:

df.columns

Out[21]:

```
Index(['id', 'date', 'price', 'bedrooms', 'bathrooms', 'sqft_living',
       'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade',
       'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode',
       'lat', 'long', 'sqft_living15', 'sqft_lot15'],
      dtype='object')
```

In [40]:

In [39]:

df

Out[39]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement	yr_built	yr_r
0	7129300520	20141013T000000	221900	3	1.00	1180	5650	1.0	0	0	...	7	1180	0	1955	
1	6414100192	20141209T000000	538000	3	2.25	2570	7242	2.0	0	0	...	7	2170	400	1951	
2	5631500400	20150225T000000	180000	2	1.00	770	10000	1.0	0	0	...	6	770	0	1933	
3	2487200875	20141209T000000	604000	4	3.00	1960	5000	1.0	0	0	...	7	1050	910	1965	
4	1954400510	20150218T000000	510000	3	2.00	1680	8080	1.0	0	0	...	8	1680	0	1987	
...
21608	263000018	20140521T000000	360000	3	2.50	1530	1131	3.0	0	0	...	8	1530	0	2009	
21609	6600060120	20150223T000000	400000	4	2.50	2310	5813	2.0	0	0	...	8	2310	0	2014	
21610	1523300141	20140623T000000	402101	2	0.75	1020	1350	2.0	0	0	...	7	1020	0	2009	

In [42]:

df["bathrooms"]=df["bathrooms"].astype(int)

In [43]:

df

Out[43]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement
0	7129300520	20141013T000000	221900	3	1	1180	5650	1.0	0	0	...	7	1180	0
1	6414100192	20141209T000000	538000	3	2	2570	7242	2.0	0	0	...	7	2170	400
2	5631500400	20150225T000000	180000	2	1	770	10000	1.0	0	0	...	6	770	0
3	2487200875	20141209T000000	604000	4	3	1960	5000	1.0	0	0	...	7	1050	910
4	1954400510	20150218T000000	510000	3	2	1680	8080	1.0	0	0	...	8	1680	0
...
21608	263000018	20140521T000000	360000	3	2	1530	1131	3.0	0	0	...	8	1530	0
21609	6600060120	20150223T000000	400000	4	2	2310	5813	2.0	0	0	...	8	2310	0
21610	1523300141	20140623T000000	402101	2	0	1020	1350	2.0	0	0	...	7	1020	0
21611	291310100	20150116T000000	400000	3	2	1600	2388	2.0	0	0	...	8	1600	0
21612	1523300157	20141015T000000	325000	2	0	1020	1076	2.0	0	0	...	7	1020	0

21613 rows × 15 columns

In [44]:

df["floors"]=df["floors"].astype(int)

In [45]:

df

Out[45]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement
0	7129300520	20141013T000000	221900	3	1	1180	5650	1	0	0	...	7	1180	0
1	6414100192	20141209T000000	538000	3	2	2570	7242	2	0	0	...	7	2170	400
2	5631500400	20150225T000000	180000	2	1	770	10000	1	0	0	...	6	770	0
3	2487200875	20141209T000000	604000	4	3	1960	5000	1	0	0	...	7	1050	910
4	1954400510	20150218T000000	510000	3	2	1680	8080	1	0	0	...	8	1680	0
...
21608	263000018	20140521T000000	360000	3	2	1530	1131	3	0	0	...	8	1530	0
21609	6600060120	20150223T000000	400000	4	2	2310	5813	2	0	0	...	8	2310	0
21610	1523300141	20140623T000000	402101	2	0	1020	1350	2	0	0	...	7	1020	0
21611	291310100	20150116T000000	400000	3	2	1600	2388	2	0	0	...	8	1600	0
21612	1523300157	20141015T000000	325000	2	0	1020	1076	2	0	0	...	7	1020	0

21613 rows × 15 columns

In [47]:

df["bedrooms"].value_counts()

Out[47]:

3	9824
4	6882
2	2760
5	1601
6	272
1	199
7	38
0	13
8	13
9	6
10	3
11	1
33	1

Name: bedrooms, dtype: int64

In [48]:

df["floors"].unique()

Out[48]:

array([1, 2, 3])

In [49]:

df["floors"].nunique()

Out[49]:

3

total no of bedrooms

In [50]:

df["bedrooms"].unique()

Out[50]:

array([3, 2, 4, 5, 1, 6, 7, 0, 8, 9, 11, 10, 33])

In [52]:

df["bedrooms"].nunique()

Out[52]:

13

In [56]:

```
df.isna().sum()
```

Out[56]:

```
id          0
date        0
price       0
bedrooms   0
bathrooms  0
sqft_living 0
sqft_lot   0
floors     0
waterfront 0
view       0
condition  0
grade      0
sqft_above 0
sqft_basement 0
yr_built   0
yr_renovated 0
zipcode    0
lat        0
long       0
sqft_living15 0
sqft_lot15 0
dtype: int64
```

In [57]:

```
df
```

Out[57]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement
0	7129300520	20141013T000000	221900	3	1	1180	5650	1	0	0	...	7	1180	0
1	6414100192	20141209T000000	538000	3	2	2570	7242	2	0	0	...	7	2170	400
2	5631500400	20150225T000000	180000	2	1	770	10000	1	0	0	...	6	770	0
3	2487200875	20141209T000000	604000	4	3	1960	5000	1	0	0	...	7	1050	910
4	1954400510	20150218T000000	510000	3	2	1680	8080	1	0	0	...	8	1680	0
...
21608	2630000018	20140521T000000	360000	3	2	1530	1131	3	0	0	...	8	1530	0
21609	6600060120	20150223T000000	400000	4	2	2310	5813	2	0	0	...	8	2310	0
21610	1523300141	20140623T000000	402101	2	0	1020	1350	2	0	0	...	7	1020	0
21611	291310100	20150116T000000	400000	3	2	1600	2388	2	0	0	...	8	1600	0
21612	1523300157	20141015T000000	325000	2	0	1020	1076	2	0	0	...	7	1020	0

21613 rows × 21 columns

In [59]:

```
s=df["date"]
```

In [60]:

```
s
```

Out[60]:

```
0      20141013T000000
1      20141209T000000
2      20150225T000000
3      20141209T000000
4      20150218T000000
...
21608   20140521T000000
21609   20150223T000000
21610   20140623T000000
21611   20150116T000000
21612   20141015T000000
Name: date, Length: 21613, dtype: object
```

In [67]:

```
s=s.str.replace("000000"," ")
```

In [68]:

s

Out[68]:

```
0      20141013T
1      20141209T
2      20150225T
3      20141209T
4      20150218T
...
21608   20140521T
21609   20150223T
21610   20140623T
21611   20150116T
21612   20141015T
Name: date, Length: 21613, dtype: object
```

In [69]:

df

Out[69]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement
0	7129300520	20141013T000000	221900	3	1	1180	5650	1	0	0	...	7	1180	0
1	6414100192	20141209T000000	538000	3	2	2570	7242	2	0	0	...	7	2170	400
2	5631500400	20150225T000000	180000	2	1	770	10000	1	0	0	...	6	770	0
3	2487200875	20141209T000000	604000	4	3	1960	5000	1	0	0	...	7	1050	910
4	1954400510	20150218T000000	510000	3	2	1680	8080	1	0	0	...	8	1680	0
...
21608	2630000018	20140521T000000	360000	3	2	1530	1131	3	0	0	...	8	1530	0
21609	6600060120	20150223T000000	400000	4	2	2310	5813	2	0	0	...	8	2310	0
21610	1523300141	20140623T000000	402101	2	0	1020	1350	2	0	0	...	7	1020	0
21611	291310100	20150116T000000	400000	3	2	1600	2388	2	0	0	...	8	1600	0
21612	1523300157	20141015T000000	325000	2	0	1020	1076	2	0	0	...	7	1020	0

21613 rows × 21 columns

In [70]:

df['date']=s

In [71]:

df

Out[71]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement	yr_bui
0	7129300520	20141013T	221900	3	1	1180	5650	1	0	0	...	7	1180	0	195
1	6414100192	20141209T	538000	3	2	2570	7242	2	0	0	...	7	2170	400	195
2	5631500400	20150225T	180000	2	1	770	10000	1	0	0	...	6	770	0	193
3	2487200875	20141209T	604000	4	3	1960	5000	1	0	0	...	7	1050	910	196
4	1954400510	20150218T	510000	3	2	1680	8080	1	0	0	...	8	1680	0	198
...
21608	2630000018	20140521T	360000	3	2	1530	1131	3	0	0	...	8	1530	0	200
21609	6600060120	20150223T	400000	4	2	2310	5813	2	0	0	...	8	2310	0	201
21610	1523300141	20140623T	402101	2	0	1020	1350	2	0	0	...	7	1020	0	200
21611	291310100	20150116T	400000	3	2	1600	2388	2	0	0	...	8	1600	0	200
21612	1523300157	20141015T	325000	2	0	1020	1076	2	0	0	...	7	1020	0	200

21613 rows × 21 columns

In [74]:

df.drop(['waterfront', 'view'],axis=1,inplace=True)

In [75]:

df

Out[75]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated
0	7129300520	20141013T	221900	3	1	1180	5650	1	3	7	1180	0	1955	
1	6414100192	20141209T	538000	3	2	2570	7242	2	3	7	2170	400	1951	
2	5631500400	20150225T	180000	2	1	770	10000	1	3	6	770	0	1933	
3	2487200875	20141209T	604000	4	3	1960	5000	1	5	7	1050	910	1965	
4	1954400510	20150218T	510000	3	2	1680	8080	1	3	8	1680	0	1987	
...
21608	263000018	20140521T	360000	3	2	1530	1131	3	3	8	1530	0	2009	
21609	6600060120	20150223T	400000	4	2	2310	5813	2	3	8	2310	0	2014	
21610	1523300141	20140623T	402101	2	0	1020	1350	2	3	7	1020	0	2009	
21611	291310100	20150116T	400000	3	2	1600	2388	2	3	8	1600	0	2004	
21612	1523300157	20141015T	325000	2	0	1020	1076	2	3	7	1020	0	2008	

21613 rows × 19 columns

In [76]:

s

Out[76]:

0	20141013T
1	20141209T
2	20150225T
3	20141209T
4	20150218T
...	...
21608	20140521T
21609	20150223T
21610	20140623T
21611	20150116T
21612	20141015T

Name: date, Length: 21613, dtype: object

In [77]:

s=s.str.replace("T", " ")

In [78]:

s

Out[78]:

0	20141013
1	20141209
2	20150225
3	20141209
4	20150218
...	...
21608	20140521
21609	20150223
21610	20140623
21611	20150116
21612	20141015

Name: date, Length: 21613, dtype: object

In [79]:

df["date"]=s

In [80]:

s

Out[80]:

```
0      20141013
1      20141209
2      20150225
3      20141209
4      20150218
...
21608   20140521
21609   20150223
21610   20140623
21611   20150116
21612   20141015
Name: date, Length: 21613, dtype: object
```

In [81]:

df

Out[81]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated
0	7129300520	20141013	221900	3	1	1180	5650	1	3	7	1180	0	1955	
1	6414100192	20141209	538000	3	2	2570	7242	2	3	7	2170	400	1951	1
2	5631500400	20150225	180000	2	1	770	10000	1	3	6	770	0	1933	
3	2487200875	20141209	604000	4	3	1960	5000	1	5	7	1050	910	1965	
4	1954400510	20150218	510000	3	2	1680	8080	1	3	8	1680	0	1987	
...
21608	2630000018	20140521	360000	3	2	1530	1131	3	3	8	1530	0	2009	
21609	6600060120	20150223	400000	4	2	2310	5813	2	3	8	2310	0	2014	
21610	1523300141	20140623	402101	2	0	1020	1350	2	3	7	1020	0	2009	
21611	291310100	20150116	400000	3	2	1600	2388	2	3	8	1600	0	2004	
21612	1523300157	20141015	325000	2	0	1020	1076	2	3	7	1020	0	2008	

21613 rows × 19 columns

In [82]:

s

Out[82]:

```
0      20141013
1      20141209
2      20150225
3      20141209
4      20150218
...
21608   20140521
21609   20150223
21610   20140623
21611   20150116
21612   20141015
Name: date, Length: 21613, dtype: object
```

In [83]:

s.str.split()

Out[83]:

```
0      [20141013]
1      [20141209]
2      [20150225]
3      [20141209]
4      [20150218]
...
21608   [20140521]
21609   [20150223]
21610   [20140623]
21611   [20150116]
21612   [20141015]
Name: date, Length: 21613, dtype: object
```

In [87]:

```
m="1235"  
print(list(m.split))
```

```
-----  
TypeError                                Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_12536\4284156713.py in <module>  
      1 m="1235"  
----> 2 print(list(m.split))  
  
TypeError: 'builtin_function_or_method' object is not iterable
```

In [88]:

```
m
```

Out[88]:

```
'1235'
```

In [90]:

```
str=s.str.split()
```

In [91]:

```
str
```

Out[91]:

```
0      [20141013]  
1      [20141209]  
2      [20150225]  
3      [20141209]  
4      [20150218]  
...  
21608   [20140521]  
21609   [20150223]  
21610   [20140623]  
21611   [20150116]  
21612   [20141015]  
Name: date, Length: 21613, dtype: object
```

In [92]:

```
str[0]
```

Out[92]:

```
['20141013']
```

In [95]:

```
str[0]=str[0][-1]
```

In [96]:

```
str[0]
```

Out[96]:

```
'20141013'
```

In [99]:

```
File "C:\Users\sadiy\AppData\Local\Temp\ipykernel_12536\1640769444.py", line 1  
    for i in str[0]  
            ^  
SyntaxError: invalid syntax
```

In [100]:

df

Out[100]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovate
0	7129300520	20141013	221900	3	1	1180	5650	1	3	7	1180	0	1955	
1	6414100192	20141209	538000	3	2	2570	7242	2	3	7	2170	400	1951	1
2	5631500400	20150225	180000	2	1	770	10000	1	3	6	770	0	1933	
3	2487200875	20141209	604000	4	3	1960	5000	1	5	7	1050	910	1965	
4	1954400510	20150218	510000	3	2	1680	8080	1	3	8	1680	0	1987	
...
21608	263000018	20140521	360000	3	2	1530	1131	3	3	8	1530	0	2009	
21609	6600060120	20150223	400000	4	2	2310	5813	2	3	8	2310	0	2014	
21610	1523300141	20140623	402101	2	0	1020	1350	2	3	7	1020	0	2009	
21611	291310100	20150116	400000	3	2	1600	2388	2	3	8	1600	0	2004	
21612	1523300157	20141015	325000	2	0	1020	1076	2	3	7	1020	0	2008	

21613 rows × 19 columns

sort according to price

In [102]:

df.sort_values(["price"],inplace=True)

In [103]:

df

Out[103]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovate
1149	3421079032	20150217	75000	1	0	670	43377	1	3	3	670	0	1966	
15293	40000362	20140506	78000	2	1	780	16344	1	1	5	780	0	1942	
465	8658300340	20140523	80000	1	0	430	5050	1	2	4	430	0	1912	
16198	3028200080	20150324	81000	2	1	730	9975	1	1	5	730	0	1943	
8274	3883800011	20141105	82000	3	1	860	10426	1	3	6	860	0	1954	
...
1448	8907500070	20150413	5350000	5	5	8000	23985	2	3	12	6720	1280	2009	
4411	2470100110	20140804	5570000	5	5	9200	35069	2	3	13	6200	3000	2001	
9254	9208900037	20140919	6885000	6	7	9890	31374	2	3	13	8860	1030	2001	
3914	9808700762	20140611	7062500	5	4	10040	37325	2	3	11	7680	2360	1940	
7252	6762700020	20141013	7700000	6	8	12050	27600	2	4	13	8570	3480	1910	

21613 rows × 19 columns

In [104]:

#Last records
df.tail()

Out[104]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovate
1448	8907500070	20150413	5350000	5	5	8000	23985	2	3	12	6720	1280	2009	
4411	2470100110	20140804	5570000	5	5	9200	35069	2	3	13	6200	3000	2001	
9254	9208900037	20140919	6885000	6	7	9890	31374	2	3	13	8860	1030	2001	
3914	9808700762	20140611	7062500	5	4	10040	37325	2	3	11	7680	2360	1940	2
7252	6762700020	20141013	7700000	6	8	12050	27600	2	4	13	8570	3480	1910	1

In [106]:

```
#descriptive info
df.describe()
```

Out[106]:

	id	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	grade	sqft_above	sqft
count	2.161300e+04	2.161300e+04	21613.000000	21613.000000	21613.000000	2.161300e+04	21613.000000	21613.000000	21613.000000	21613.000000	21613.000000
mean	4.580302e+09	5.400881e+05	3.370842	1.749734	2079.899736	1.510697e+04	1.446213	3.409430	7.656873	1788.390691	1788.390691
std	2.876566e+09	3.671272e+05	0.930062	0.734873	918.440897	4.142051e+04	0.551894	0.650743	1.175459	828.090978	828.090978
min	1.000102e+06	7.500000e+04	0.000000	0.000000	290.000000	5.200000e+02	1.000000	1.000000	1.000000	290.000000	290.000000
25%	2.123049e+09	3.219500e+05	3.000000	1.000000	1427.000000	5.040000e+03	1.000000	3.000000	7.000000	1190.000000	1190.000000
50%	3.904930e+09	4.500000e+05	3.000000	2.000000	1910.000000	7.618000e+03	1.000000	3.000000	7.000000	1560.000000	1560.000000
75%	7.308900e+09	6.450000e+05	4.000000	2.000000	2550.000000	1.068800e+04	2.000000	4.000000	8.000000	2210.000000	2210.000000
max	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e+06	3.000000	5.000000	13.000000	9410.000000	9410.000000

In [112]:

```
#show the highest price of room
df["price"].max()
```

Out[112]:

7700000

In [116]:

```
#coverting grade into int
df['grade']=df["grade"].astype(int)
```

In [117]:

```
df
```

Out[117]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated
1149	3421079032	20150217	75000	1	0	670	43377	1	3	3	670	0	1966	
15293	40000362	20140506	78000	2	1	780	16344	1	1	5	780	0	1942	
465	8658300340	20140523	80000	1	0	430	5050	1	2	4	430	0	1912	
16198	3028200080	20150324	81000	2	1	730	9975	1	1	5	730	0	1943	
8274	3883800011	20141105	82000	3	1	860	10426	1	3	6	860	0	1954	
...
1448	8907500070	20150413	5350000	5	5	8000	23985	2	3	12	6720	1280	2009	
4411	2470100110	20140804	5570000	5	5	9200	35069	2	3	13	6200	3000	2001	
9254	9208900037	20140919	6885000	6	7	9890	31374	2	3	13	8860	1030	2001	
3914	9808700762	20140611	7062500	5	4	10040	37325	2	3	11	7680	2360	1940	
7252	6762700020	20141013	7700000	6	8	12050	27600	2	4	13	8570	3480	1910	

21613 rows × 19 columns

In [118]:

```
import matplotlib.pyplot as plt
```

In [121]:

```
x=df['yr_built']
```

In [122]:

x

Out[122]:

```

1149    1966
15293   1942
  465    1912
16198   1943
 8274   1954
      ...
 1448   2009
 4411   2001
 9254   2001
 3914   1940
 7252   1910
Name: yr_built, Length: 21613, dtype: int64

```

In [124]:

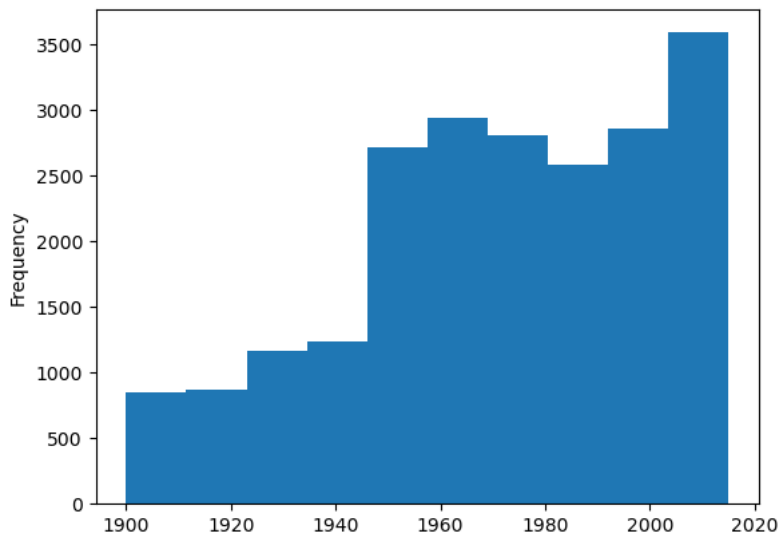
```

#data of year of construction:
x.plot(kind='hist')

```

Out[124]:

<AxesSubplot:ylabel='Frequency'>



In [125]:

df.head()

Out[125]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovat
1149	3421079032	20150217	75000	1	0	670	43377	1	3	3	670	0	1966	
15293	40000362	20140506	78000	2	1	780	16344	1	1	5	780	0	1942	
465	8658300340	20140523	80000	1	0	430	5050	1	2	4	430	0	1912	
16198	3028200080	20150324	81000	2	1	730	9975	1	1	5	730	0	1943	
8274	3883800011	20141105	82000	3	1	860	10426	1	3	6	860	0	1954	

In [128]:

g1=list(df.groupby(["price", "sqft_lot"]))

In [132]:

g1

Out[132]:

```
[((75000, 43377),
  id      date  price bedrooms bathrooms sqft_living \
1149 3421079032 20150217 75000      1          0      670

  sqft_lot floors condition grade sqft_above sqft_basement yr_built \
1149   43377      1          3      3      670          0    1966

  yr_renovated zipcode      lat      long sqft_living15 sqft_lot15
1149      0    98022 47.2638 -121.906      1160      42882 ),
((78000, 16344),
  id      date  price bedrooms bathrooms sqft_living \
15293 40000362 20140506 78000      2          1      780

  sqft_lot floors condition grade sqft_above sqft_basement \
15293   16344      1          1      5      780          0

  yr_built yr_renovated zipcode      lat      long sqft_living15 \
```

In [135]:

piel=df["bedrooms"]

In [136]:

piel

Out[136]:

```
1149    1
15293   2
465     1
16198   2
8274    3
..
1448    5
4411    5
9254    6
3914    5
7252    6
Name: bedrooms, Length: 21613, dtype: int32
```

In [142]:

df

Out[142]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated
1149	3421079032	20150217	75000	1	0	670	43377	1	3	3	670	0	1966	
15293	40000362	20140506	78000	2	1	780	16344	1	1	5	780	0	1942	
465	8658300340	20140523	80000	1	0	430	5050	1	2	4	430	0	1912	
16198	3028200080	20150324	81000	2	1	730	9975	1	1	5	730	0	1943	
8274	3883800011	20141105	82000	3	1	860	10426	1	3	6	860	0	1954	
...
1448	8907500070	20150413	5350000	5	5	8000	23985	2	3	12	6720	1280	2009	
4411	2470100110	20140804	5570000	5	5	9200	35069	2	3	13	6200	3000	2001	
9254	9208900037	20140919	6885000	6	7	9890	31374	2	3	13	8860	1030	2001	
3914	9808700762	20140611	7062500	5	4	10040	37325	2	3	11	7680	2360	1940	
7252	6762700020	20141013	7700000	6	8	12050	27600	2	4	13	8570	3480	1910	

21613 rows × 19 columns

In []:

In []:

