In [1]:
```python
import pandas as pd
```

In [2]:
```python
data=pd.read_csv(r"E:\new download\5. London Housing Data.csv")
```

In [3]:
```python
data
```

Out[3]:

	date	area	average_price	code	houses_sold	no_of_crimes
0	1/1/1995	city of london	91449	E09000001	17.0	NaN
1	2/1/1995	city of london	82203	E09000001	7.0	NaN
2	3/1/1995	city of london	79121	E09000001	14.0	NaN
3	4/1/1995	city of london	77101	E09000001	7.0	NaN
4	5/1/1995	city of london	84409	E09000001	10.0	NaN
...	...	...	...	...	...	...
13544	9/1/2019	england	249942	E92000001	64605.0	NaN
13545	10/1/2019	england	249376	E92000001	68677.0	NaN
13546	11/1/2019	england	248515	E92000001	67814.0	NaN
13547	12/1/2019	england	250410	E92000001	NaN	NaN
13548	1/1/2020	england	247355	E92000001	NaN	NaN

13549 rows × 6 columns

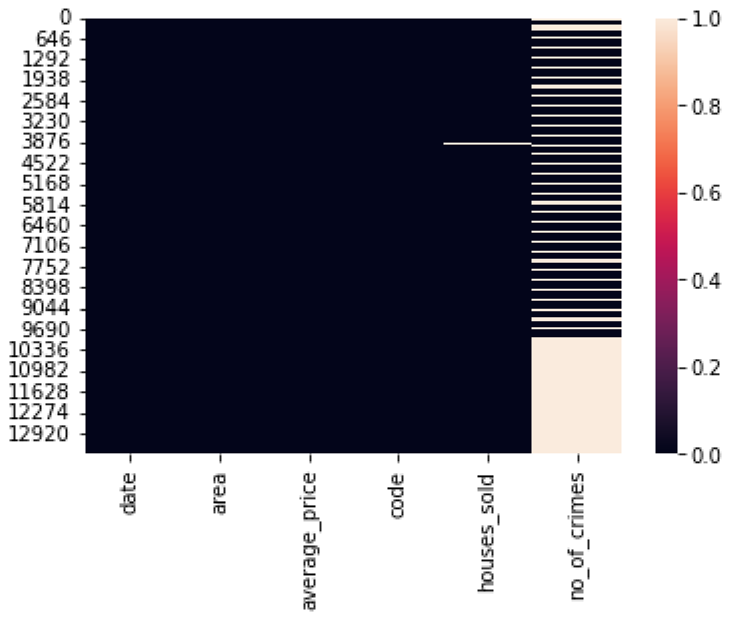
In [4]:
```python
data.isnull().sum()
```

Out[4]:
```
date                0
area                0
average_price       0
code                0
houses_sold        94
no_of_crimes     6110
dtype: int64
```

In [5]:
```python
import seaborn as sns
import matplotlib.pyplot as plt
```

In [6]:
```python
sns.heatmap(data.isnull())
plt.show()
```

Loading [MathJax]/extensions/Safe.js



In [7]:
```
data.dtypes
```

Out[7]:
```
date                object
area                object
average_price        int64
code                object
houses_sold        float64
no_of_crimes       float64
dtype: object
```

In [8]:
```
data.date=pd.to_datetime(data.date)
```

Add a new column "Year" in the dataframe, which contains years only.

In [9]:
```
data['Year']= data.date.dt.year
```

In [10]:
```
data.head()
```

Out[10]:

	date	area	average_price	code	houses_sold	no_of_crimes	Year
0	1995-01-01	city of london	91449	E09000001	17.0	NaN	1995
1	1995-02-01	city of london	82203	E09000001	7.0	NaN	1995
2	1995-03-01	city of london	79121	E09000001	14.0	NaN	1995
3	1995-04-01	city of london	77101	E09000001	7.0	NaN	1995
4	1995-05-01	city of london	84409	E09000001	10.0	NaN	1995

Add a new column "month" as second column. Which contains month only.

In [11]:
```
data.insert(1, 'month', data.date.dt.month)
```

In [12]:
```
data.tail()
```

Loading [MathJax]/extensions/Safe.js

Out[12]:

	date	month	area	average_price	code	houses_sold	no_of_crimes	Year
13544	2019-09-01	9	england	249942	E92000001	64605.0	NaN	2019
13545	2019-10-01	10	england	249376	E92000001	68677.0	NaN	2019
13546	2019-11-01	11	england	248515	E92000001	67814.0	NaN	2019
13547	2019-12-01	12	england	250410	E92000001	NaN	NaN	2019
13548	2020-01-01	1	england	247355	E92000001	NaN	NaN	2020

Remove the columns "Year" and 'month'.

In [13]:
```
data.drop(['Year', 'month'], axis=1, inplace=True)
```

Show all the records where 'no. of crime' is 0. and how many such records are there.

In [14]:
```
len(data[data['no_of_crimes']==0])
```

Out[14]:
```
104
```

What is the maximum and minimum 'average_price' per year in england.

In [15]:
```
data['year']=data.date.dt.year
```

In [16]:
```
data
```

Out[16]:

	date	area	average_price	code	houses_sold	no_of_crimes	year
0	1995-01-01	city of london	91449	E09000001	17.0	NaN	1995
1	1995-02-01	city of london	82203	E09000001	7.0	NaN	1995
2	1995-03-01	city of london	79121	E09000001	14.0	NaN	1995
3	1995-04-01	city of london	77101	E09000001	7.0	NaN	1995
4	1995-05-01	city of london	84409	E09000001	10.0	NaN	1995
...	...	...	...	...	...	...	...
13544	2019-09-01	england	249942	E92000001	64605.0	NaN	2019
13545	2019-10-01	england	249376	E92000001	68677.0	NaN	2019
13546	2019-11-01	england	248515	E92000001	67814.0	NaN	2019
13547	2019-12-01	england	250410	E92000001	NaN	NaN	2019
13548	2020-01-01	england	247355	E92000001	NaN	NaN	2020

13549 rows × 7 columns

In [17]:
```
df1=data[data.area=='england']
```

Loading [MathJax]/extensions/Safe.js

```
In [18]: df1.groupby('year').average_price.min()
```

```
Out[18]: year
1995     52788
1996     52333
1997     55789
1998     61659
1999     65522
2000     75219
2001     84245
2002     96215
2003    121610
2004    139719
2005    158572
2006    166544
2007    181824
2008    165795
2009    159340
2010    174458
2011    173046
2012    174161
2013    176816
2014    188265
2015    202856
2016    220361
2017    231593
2018    240428
2019    243281
2020    247355
Name: average_price, dtype: int64
```

```
In [19]: df1.groupby('year').average_price.mean()
```

```
Out[19]: year
1995     53322.416667
1996     54151.500000
1997     59160.666667
1998     64301.666667
1999     70070.750000
2000     80814.333333
2001     90306.750000
2002    107981.500000
2003    130218.583333
2004    152314.416667
2005    163570.000000
2006    174351.500000
2007    190025.583333
2008    182379.916667
2009    166558.666667
2010    177472.666667
2011    175230.000000
2012    177488.000000
2013    182581.416667
2014    197771.083333
2015    211174.750000
2016    227337.166667
2017    238161.166667
2018    245018.333333
2019    247101.083333
2020    247355.000000
Name: average_price, dtype: float64
```

What is the maximum and minimum number of

Loading [MathJax]/extensions/Safe.js

crimes recorded per area.

In [20]:
```
data
```

Out[20]:

	date	area	average_price	code	houses_sold	no_of_crimes	year
0	1995-01-01	city of london	91449	E09000001	17.0	NaN	1995
1	1995-02-01	city of london	82203	E09000001	7.0	NaN	1995
2	1995-03-01	city of london	79121	E09000001	14.0	NaN	1995
3	1995-04-01	city of london	77101	E09000001	7.0	NaN	1995
4	1995-05-01	city of london	84409	E09000001	10.0	NaN	1995
...	...	...	...	...	...	...	...
13544	2019-09-01	england	249942	E92000001	64605.0	NaN	2019
13545	2019-10-01	england	249376	E92000001	68677.0	NaN	2019
13546	2019-11-01	england	248515	E92000001	67814.0	NaN	2019
13547	2019-12-01	england	250410	E92000001	NaN	NaN	2019
13548	2020-01-01	england	247355	E92000001	NaN	NaN	2020

13549 rows × 7 columns

In [23]:
```
data.groupby('area').no_of_crimes.max().sort_values(ascending= False)
```

```
Out[23]: area
         westminster               7461.0
         lambeth                   4701.0
         camden                    4558.0
         southwark                 3821.0
         newham                    3668.0
         hackney                   3466.0
         ealing                    3401.0
         islington                 3384.0
         tower hamlets             3316.0
         croydon                   3263.0
         haringey                  3199.0
         wandsworth                3051.0
         waltham forest            2941.0
         brent                     2937.0
         barnet                    2893.0
         greenwich                 2853.0
         hillingdon                2819.0
         hounslow                  2817.0
         lewisham                  2813.0
         enfield                   2798.0
         kensington and chelsea    2778.0
         hammersmith and fulham    2645.0
         bromley                   2637.0
         redbridge                 2560.0
         barking and dagenham      2049.0
         havering                  1956.0
         bexley                    1914.0
         harrow                    1763.0
         merton                    1623.0
         richmond upon thames      1551.0
         sutton                    1425.0
         kingston upon thames      1379.0
         city of london              10.0
         east midlands                NaN
         east of england              NaN
         england                      NaN
         inner london                 NaN
         london                       NaN
         north east                   NaN
         north west                   NaN
         outer london                 NaN
         south east                   NaN
         south west                   NaN
         west midlands                NaN
         yorks and the humber         NaN
         Name: no_of_crimes, dtype: float64
```

Show the total count of the records of each area, where average price is less than 1,00,000.

```
In [24]: data
```

Loading [MathJax]/extensions/Safe.js

Out[24]:

	date	area	average_price	code	houses_sold	no_of_crimes	year
0	1995-01-01	city of london	91449	E09000001	17.0	NaN	1995
1	1995-02-01	city of london	82203	E09000001	7.0	NaN	1995
2	1995-03-01	city of london	79121	E09000001	14.0	NaN	1995
3	1995-04-01	city of london	77101	E09000001	7.0	NaN	1995
4	1995-05-01	city of london	84409	E09000001	10.0	NaN	1995
...	...	...	...	...	...	...	...
13544	2019-09-01	england	249942	E92000001	64605.0	NaN	2019
13545	2019-10-01	england	249376	E92000001	68677.0	NaN	2019
13546	2019-11-01	england	248515	E92000001	67814.0	NaN	2019
13547	2019-12-01	england	250410	E92000001	NaN	NaN	2019
13548	2020-01-01	england	247355	E92000001	NaN	NaN	2020

13549 rows × 7 columns

In [29]:

```
data[data['average_price']<100000].value_counts('area')
```

```
Out[29]: area
         north east                112
         north west                111
         yorks and the humber      110
         east midlands              96
         west midlands              94
         england                    87
         barking and dagenham       85
         south west                 78
         east of england            76
         newham                     72
         waltham forest             64
         bexley                     64
         lewisham                   62
         havering                   60
         south east                 59
         greenwich                  59
         croydon                    57
         enfield                    54
         sutton                     54
         hackney                    53
         redbridge                  52
         southwark                  48
         tower hamlets              47
         outer london               46
         hillingdon                 44
         lambeth                    41
         hounslow                   41
         brent                      40
         london                     39
         merton                     35
         bromley                    33
         haringey                   33
         ealing                     31
         inner london               31
         kingston upon thames       30
         harrow                     30
         wandsworth                 26
         barnet                     25
         islington                  19
         city of london             11
         dtype: int64
```

```
In [ ]:
```

Loading [MathJax]/extensions/Safe.js