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

import seaborn as sns

import matplotlib.pyplot as plt
```

In [2]:
```
df = pd.read_csv(r'London_Housing (2).csv')

df
```

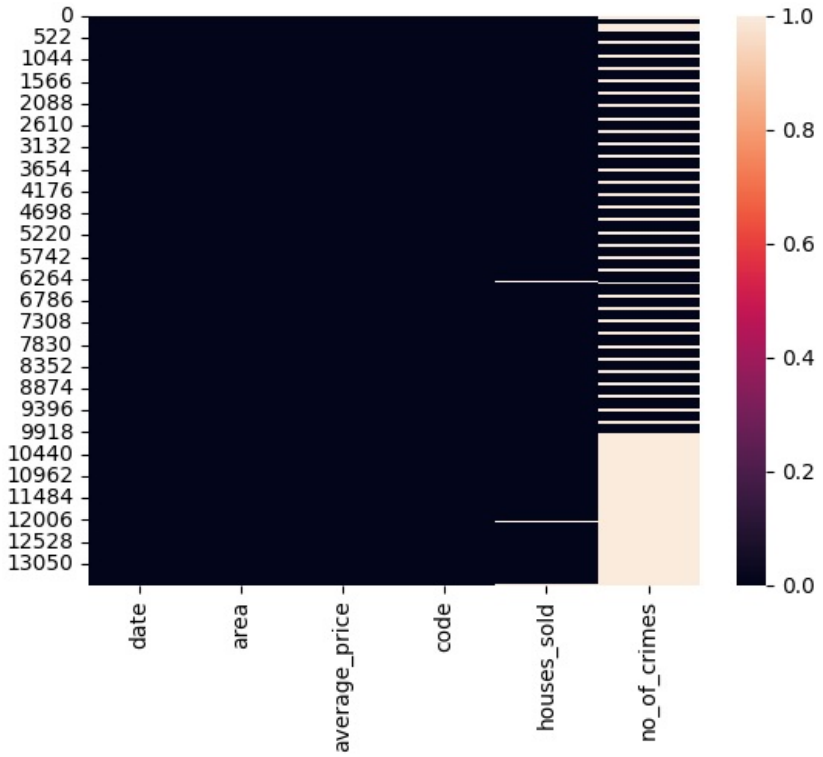
Out[2]:

	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 [3]:
```
sns.heatmap(df.isnull())

plt.show()
```



In [4]:
```
df.dtypes
```

Out[4]:
```
date              object
area              object
average_price      int64
code              object
houses_sold      float64
no_of_crimes     float64
dtype: object
```

In [5]:
```
df['date'] = pd.to_datetime(df.date)
df.dtypes
```

Out[5]:
```
date             datetime64[ns]
area                     object
average_price             int64
code                     object
houses_sold             float64
no_of_crimes            float64
dtype: object
```

In [6]:
```
df['year'] = df.date.dt.year
df
```

Out[6]:

	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 [7]:
```
df.insert(1, 'month', df.date.dt.month)
df
```

Out[7]:

	date	month	area	average_price	code	houses_sold	no_of_crimes	year
0	1995-01-01	1	city of london	91449	E09000001	17.0	NaN	1995
1	1995-02-01	2	city of london	82203	E09000001	7.0	NaN	1995
2	1995-03-01	3	city of london	79121	E09000001	14.0	NaN	1995
3	1995-04-01	4	city of london	77101	E09000001	7.0	NaN	1995
4	1995-05-01	5	city of london	84409	E09000001	10.0	NaN	1995
...	...	...	...	...	...	...	...	...
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

13549 rows × 8 columns

In [8]:
```
df.drop(['month', 'year'], axis = 1, inplace = True)
df
```

Out[8]:

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

13549 rows × 6 columns

In [9]:
```
df[df.no_of_crimes == 0]
```

Out[9]:

	date	area	average_price	code	houses_sold	no_of_crimes
72	2001-01-01	city of london	284262	E09000001	24.0	0.0
73	2001-02-01	city of london	198137	E09000001	37.0	0.0
74	2001-03-01	city of london	189033	E09000001	44.0	0.0
75	2001-04-01	city of london	205494	E09000001	38.0	0.0
76	2001-05-01	city of london	223459	E09000001	30.0	0.0
...	...	...	...	...	...	...
178	2009-11-01	city of london	397909	E09000001	11.0	0.0
179	2009-12-01	city of london	411955	E09000001	16.0	0.0
180	2010-01-01	city of london	464436	E09000001	20.0	0.0
181	2010-02-01	city of london	490525	E09000001	9.0	0.0
182	2010-03-01	city of london	498241	E09000001	15.0	0.0

104 rows × 6 columns

In [10]:
```
# Maximum average Price

df.groupby(by = df.date.dt.year).average_price.max().sort_values(ascending = False).head(1)
```

Out[10]:
```
date
2018    1463378
Name: average_price, dtype: int64
```

In [11]:
```
# Minimum Average Price

df.groupby(by = df.date.dt.year).average_price.min().sort_values(ascending = True).head(1)
```

Out[11]:
```
date
1996    40722
Name: average_price, dtype: int64
```

In [12]:
```
# Maximum No_of_crimes

df.groupby(by = 'area').no_of_crimes.max().sort_values(ascending = False).head(1)
```

Out[12]:
```
area
westminster    7461.0
Name: no_of_crimes, dtype: float64
```

In [13]:
```
df[df.average_price < 100000].groupby('area').count()
```

Out[13]:

area	date	average_price	code	houses_sold	no_of_crimes
barking and dagenham	85	85	85	85	13
barnet	25	25	25	25	0
bexley	64	64	64	64	0
brent	40	40	40	40	0
bromley	33	33	33	33	0
city of london	11	11	11	11	0
croydon	57	57	57	57	0
ealing	31	31	31	31	0
east midlands	96	96	96	96	0
east of england	76	76	76	76	0
enfield	54	54	54	53	0
england	87	87	87	87	0
greenwich	59	59	59	59	0
hackney	53	53	53	52	0
haringey	33	33	33	33	0
harrow	30	30	30	30	0
havering	60	60	60	60	0
hillingdon	44	44	44	44	0
hounslow	41	41	41	41	0
inner london	31	31	31	31	0
islington	19	19	19	19	0
kingston upon thames	30	30	30	30	0
lambeth	41	41	41	41	0
lewisham	62	62	62	62	0
london	39	39	39	39	0
merton	35	35	35	35	0
newham	72	72	72	72	0
north east	112	112	112	112	0
north west	111	111	111	111	0
outer london	46	46	46	46	0
redbridge	52	52	52	52	0
south east	59	59	59	58	0
south west	78	78	78	78	0
southwark	48	48	48	48	0
sutton	54	54	54	54	0
tower hamlets	47	47	47	46	0
waltham forest	64	64	64	64	0
wandsworth	26	26	26	26	0
west midlands	94	94	94	94	0
yorks and the humber	110	110	110	110	0

In [ ]: