

London Housing Data

```
In [11]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

In [3]: df = pd.read_csv('D:\Python Projects\London Housing Data.csv')

In [4]: df

Out[4]:
```

	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 [5]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13549 entries, 0 to 13548
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   date             13549 non-null  object
1   area             13549 non-null  object
2   average_price    13549 non-null  int64
3   code             13549 non-null  object
4   houses_sold      13455 non-null  float64
5   no_of_crimes     7439 non-null   float64
dtypes: float64(2), int64(1), object(3)
memory usage: 635.2+ KB

In [7]: df.count()

Out[7]:
date            13549
area            13549
average_price    13549
code            13549
houses_sold      13455
no_of_crimes     7439
dtype: int64

In [8]: df.isnull().sum()

Out[8]:
date            0
area            0
average_price    0
code            0
houses_sold      94
no_of_crimes    6110
dtype: int64
```

null values using heat map

```
In [13]: sns.heatmap(df.isnull())
plt.show()
```

```
In [14]: # convert datatype 'date'column into 'date-time' format

In [15]: df.head()
```

```
Out[15]:
```

	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

```
In [16]: df.dtypes

Out[16]:
date            object
area            object
average_price    int64
code            object
houses_sold      float64
no_of_crimes     float64
dtype: object

In [17]: df['date'] = pd.to_datetime(df['date'])

In [18]: df.dtypes

Out[18]:
date            datetime64[ns]
area            object
average_price    int64
code            object
houses_sold      float64
no_of_crimes     float64
dtype: object
```

```
In [19]: df.head()

Out[19]:
```

	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

```
In [22]: df['year'] = df.date.dt.year
df['month'] = df.date.dt.month

In [23]: df

Out[23]:
```

	date	area	average_price	code	houses_sold	no_of_crimes	year	month
0	1995-01-01	city of london	91449	E09000001	17.0	NaN	1995	1
1	1995-02-01	city of london	82203	E09000001	7.0	NaN	1995	2
2	1995-03-01	city of london	79121	E09000001	14.0	NaN	1995	3
3	1995-04-01	city of london	77101	E09000001	7.0	NaN	1995	4
4	1995-05-01	city of london	84409	E09000001	10.0	NaN	1995	5
...
13544	2019-09-01	england	249942	E92000001	64605.0	NaN	2019	9
13545	2019-10-01	england	249376	E92000001	68677.0	NaN	2019	10
13546	2019-11-01	england	248515	E92000001	67814.0	NaN	2019	11
13547	2019-12-01	england	250410	E92000001	NaN	NaN	2019	12
13548	2020-01-01	england	247355	E92000001	NaN	NaN	2020	1

13549 rows × 8 columns

```
In [25]: df.drop(['month','year'], axis=1, inplace=True)

In [26]: df.head()

Out[26]:
```

	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

no of crimes is 0

```
In [28]: df[df.no_of_crimes == 0]

Out[28]:
```

	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 [29]: len(df[df.no_of_crimes == 0])

Out[29]: 104
```

max and min 'average price' per year in england

```
In [34]: df['year'] = df.date.dt.year

In [35]: df

Out[35]:
```

	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 [36]: df1 = df[df.area == 'england']
df1

Out[36]:
```

	date	area	average_price	code	houses_sold	no_of_crimes	year
13248	1995-01-01	england	53203	E92000001	47639.0	NaN	1995
13249	1995-02-01	england	53096	E92000001	47880.0	NaN	1995
13250	1995-03-01	england	53201	E92000001	67025.0	NaN	1995
13251	1995-04-01	england	53591	E92000001	56925.0	NaN	1995
13252	1995-05-01	england	53678	E92000001	64192.0	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

301 rows × 7 columns

```
In [39]: # df1.groupby('year').average_price.max()
df1.groupby('year').average_price.min()

Out[39]:
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    173646
2012    174161
2013    178616
2014    188265
2015    202856
2016    220361
2017    231593
2018    248428
2019    243281
2020    247355
Name: average_price, dtype: int64
```

max and min no of crimes recorded per area

```
In [52]: df.groupby('area').no_of_crimes.max().sort_values(ascending=False)
# df.groupby('area').no_of_crimes.min().sort_values(ascending=True)

Out[52]:
area
westminster    7461.0
lambeth        4761.0
camden         4558.0
southwark      3821.0
newham         3668.0
hackney        3466.0
ealing         3461.0
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
```

Total count of record of each area, where average price is less than 100000

```
In [56]: df[df.average_price<100000].area.value_counts().sort_values(ascending=True)

Out[56]:
city of london    11
islington         19
barnet            25
wandsworth       26
harrow            30
kingston upon thames 30
ealing            31
inner london     31
bromley          33
haringey         33
merton           35
london           39
brent            40
hounslow         41
lambeth          41
hillingdon       44
outer london     46
tower hamlets    47
southwark        48
redbridge        52
hackney          53
enfield          54
sutton           54
croydon          57
greenwich        59
south east       59
haringey         60
lewisham         62
waltham forest   64
bexley           64
newham           72
east of england  76
south west       78
barking and dagenham 85
england          87
west midlands    94
east midlands    96
yorks and the humber 110
north west       111
north east       112
Name: area, dtype: int64

In [ ]:
```