

```
In [1]: import pandas as pd
        lha=pd.read_csv(r"E:\Personal Python Projects\London Housing Analysis\Housing+Data.
        lha.head()
```

```
Out[1]:
```

	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

## Exploring Size of DF

```
In [2]: lha.size
```

```
Out[2]: 81294
```

## Finding rows and columns

```
In [5]: lha.shape
```

```
Out[5]: (13549, 6)
```

## Data type and columns of the DF

```
In [8]: lha.dtypes
```

```
Out[8]: date           object
        area           object
        average_price  int64
        code           object
        houses_sold    float64
        no_of_crimes   float64
        dtype: object
```

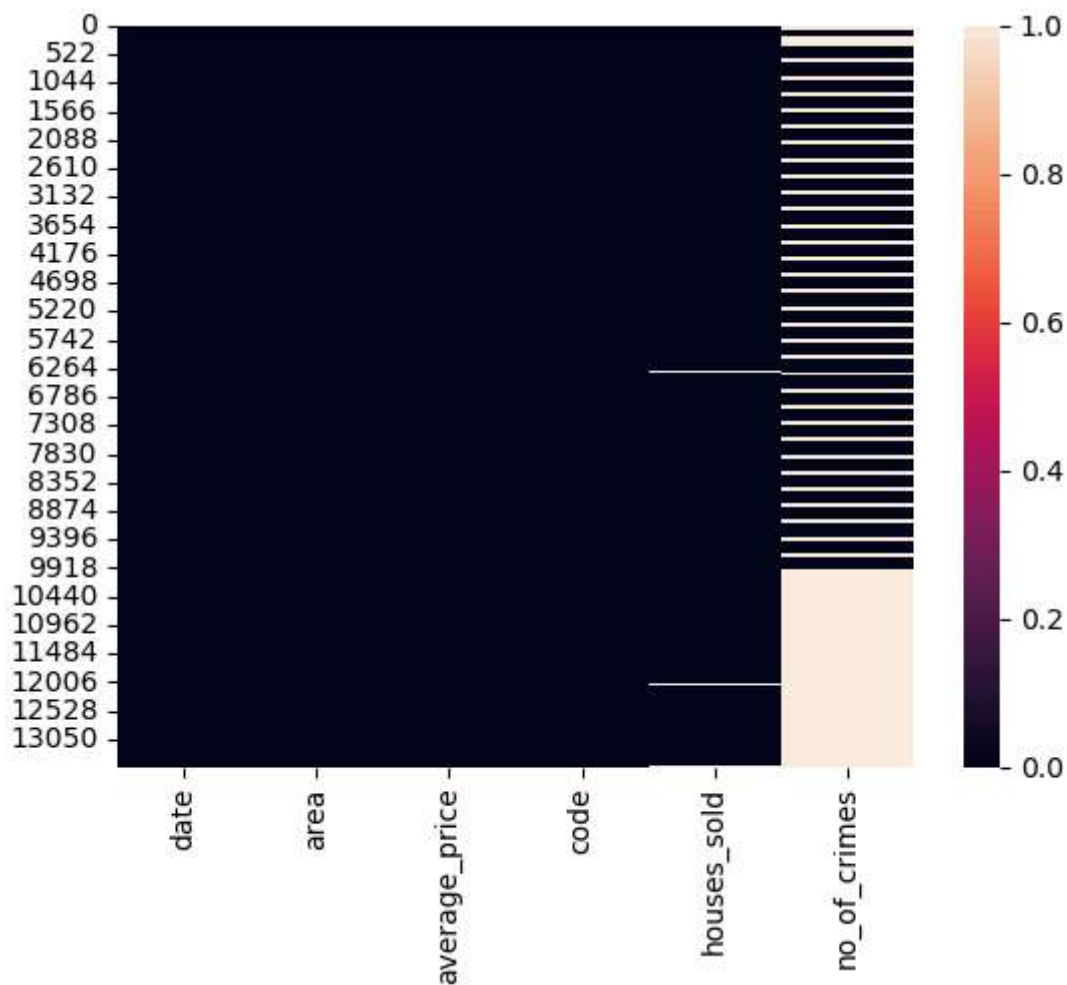
## Lets find out the missing values inside the DF columns (if any)

```
In [13]: lha.isna().sum()
```

```
Out[13]: date           0
        area           0
        average_price    0
        code           0
        houses_sold      94
        no_of_crimes    6110
        dtype: int64
```

```
In [15]: import seaborn as sns
sns.heatmap(lha.isnull())
```

Out[15]: <Axes: >



### Changing date column dtype from object to date

```
In [19]: lha['date'] = pd.to_datetime(lha['date'])
lha
```

Out[19]:

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

13549 rows × 6 columns

### Adding 'year' as a new column in DF

```
In [23]: lha['year']=lha['date'].dt.year
lha
```

Out[23]:

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

13549 rows × 7 columns

## Adding month column after the date column in the DF

```
In [24]: lha.insert(1, 'month', lha.date.dt.month)
         lha
```

Out[24]:

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

13549 rows × 8 columns



'Year and 'month' columns are not required for this analysis, removing them from the DF

```
In [27]: lha.drop(['year', 'month'], axis=1, inplace=True)
lha
```

Out[27]:

	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

### Showing all records where no. of crimes is nil

```
In [46]: lha_crime_nil=lha[lha['no_of_crimes']==0]['no_of_crimes'].count()
print(lha_crime_nil)
# OR
len(lha[lha['no_of_crimes']==0])
```

104

Out[46]: 104

### Max (in descending order) and Min (in ascending order) avg price per year

```
In [47]: lha['year']=lha.date.dt.year
lha.head(2)
```

Out[47]:

	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

```
In [59]: lha_max_avg_price=lha[lha['area']=='england'].groupby('year')['average_price'].max()
lha_max_avg_price
```

```
Out[59]: year
2019    250410
2018    248620
2020    247355
2017    242628
2016    231922
2015    219582
2014    203639
2007    194764
2008    191750
2013    188544
2006    182031
2010    180807
2012    180129
2011    177335
2009    174136
2005    167244
2004    160330
2003    138985
2002    119982
2001     95992
2000     84191
1999     75071
1998     65743
1997     61564
1996     55755
1995     53901
Name: average_price, dtype: int64
```

```
In [55]: lha_min_avg_price=lha[lha['area']=='england'].groupby('year')['average_price'].min(
lha_min_avg_price
```

```
Out[55]: 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 [66]: lha_count_avgPrice_less100000=lha[lha.average_price<100000]['area'].value_counts()
        lha_count_avgPrice_less100000
```



```
Out[66]: 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
bexley          64
waltham forest  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
hillington      44
lambeth         41
hounslow        41
brent           40
london          39
merton          35
haringey        33
bromley         33
inner london    31
ealing          31
kingston upon thames 30
harrow          30
wandsworth      26
barnet          25
islington       19
city of london  11
Name: count, dtype: int64
```

```
In [68]: dict={}
for i in range(1,10):
    for j in range(2,11):
        dict[i]=j
print(dict)
```

```
{1: 10, 2: 10, 3: 10, 4: 10, 5: 10, 6: 10, 7: 10, 8: 10, 9: 10}
```

```
In [ ]:
```