london-housing-data

January 30, 2024

0.1 Working on Real Project with Python on 'London Housing Dataset'

0.1.1 London HOusing Dataset

This dataset is primarily centered around th housing market of london. It contains a lot of additonal relevant data. - Monthly average houses prices - Yearly number of houses sold - Monthly number of crimes committed The data used here is from year 1995 to 2019 of each different area.

[19]:
```
import pandas as pd
```

[20]:
```
data = pd.read_csv("D:/data analystics/Python for data analytics/Python␣
↪Projects/London Housing Data/file.csv")
```

[21]:
```
data
```

[21]:

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

	no_of_crimes
0	NaN
1	NaN
2	NaN
3	NaN
4	NaN
…	…
13544	NaN
13545	NaN
13546	NaN
13547	NaN

```
13548           NaN

[13549 rows x 6 columns]
```

[22]:
```
data.count()
```

[22]:
```
date             13549
area             13549
average_price    13549
code             13549
houses_sold      13455
no_of_crimes      7439
dtype: int64
```

[23]:
```
data.isnull().sum()
```

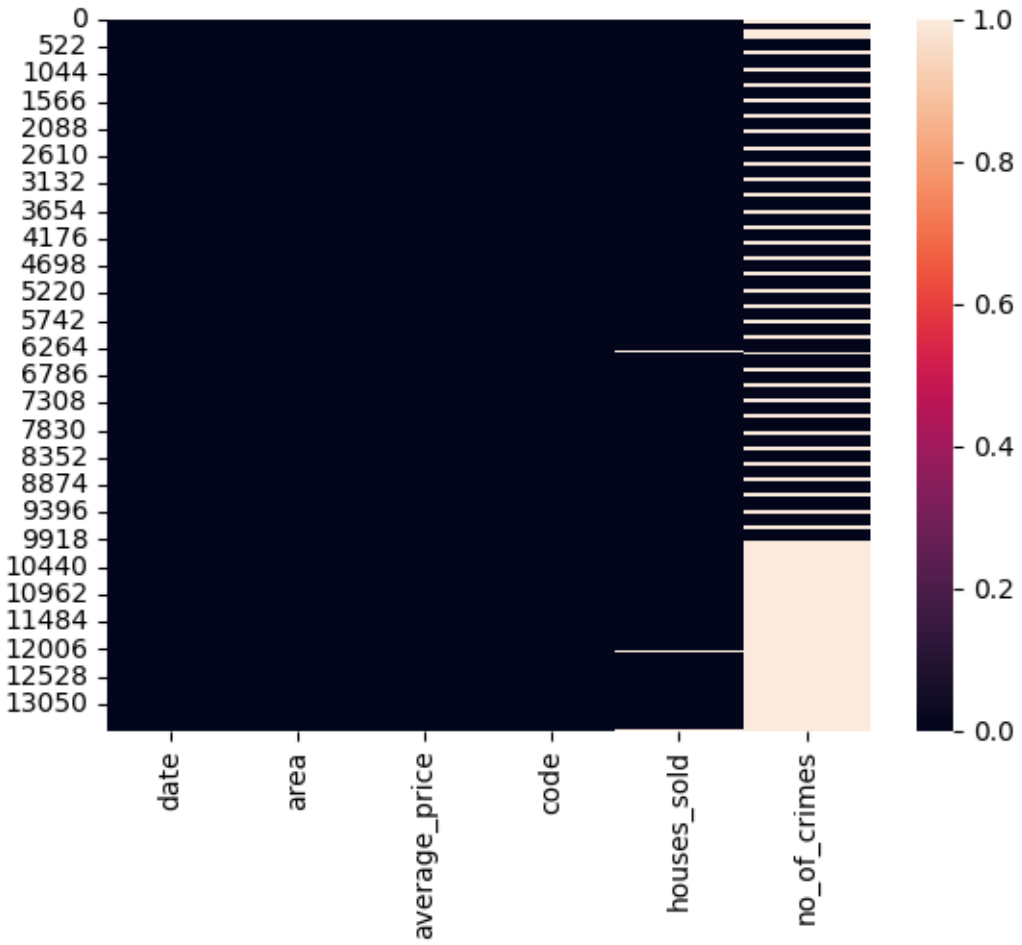
[23]:
```
date                0
area                0
average_price       0
code                0
houses_sold        94
no_of_crimes     6110
dtype: int64
```

[24]:
```
import seaborn as sns
```

[25]:
```
import matplotlib.pyplot as plt
```

[26]:
```
sns.heatmap(data.isnull())
plt.show
```

[26]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



0.1.2 Question.1. Convert the datatype of 'Date' column to date-time format.

[27]:
```
data.date = pd.to_datetime(data.date)
```

[28]:
```
data.dtypes
```

[28]:
```
date             datetime64[ns]
area                     object
average_price             int64
code                     object
houses_sold             float64
no_of_crimes            float64
dtype: object
```

0.1.3 Question.2A. Add a new Column 'Year' in the dataframe, which contains years only.

[29]:
```
data['year'] = data.date.dt.year
```

[30]:
```
data.head()
```

[30]:
```
        date            area  average_price       code  houses_sold  \
0 1995-01-01  city of london          91449  E09000001         17.0
1 1995-02-01  city of london          82203  E09000001          7.0
2 1995-03-01  city of london          79121  E09000001         14.0
3 1995-04-01  city of london          77101  E09000001          7.0
4 1995-05-01  city of london          84409  E09000001         10.0

   no_of_crimes  year
0           NaN  1995
1           NaN  1995
2           NaN  1995
3           NaN  1995
4           NaN  1995
```

0.1.4 Question.2B. Add a new column 'month' as 2nd column in the dataframe, which contains month only.

[31]:
```
#data['month'] = data.date.dt.month
```

[32]:
```
data.insert(1, 'month', data.date.dt.month)
```

0.1.5 Question.3. Remove the columns 'year' and 'month' from the dataframe

[33]:
```
data.drop(['month', 'year'], axis=1, inplace= True)
```

0.1.6 Question.4. Show all the records where 'No. of Crimes' is 0. And, How many such records are there?

[35]:
```
#data[data.no_of_crimes == 0]
len(data[data.no_of_crimes == 0])
```

[35]:
```
104
```

0.1.7 Question.5. What is the maximum & minimum 'average_price' per year in england?

[36]:
```
data['year'] = data.date.dt.year
```

[39]:
```
df1 = data[data.area == 'england']
df1
```

[39]:
```
            date     area  average_price       code  houses_sold  \
13248 1995-01-01  england          53203  E92000001      47639.0
13249 1995-02-01  england          53096  E92000001      47880.0
13250 1995-03-01  england          53201  E92000001      67025.0
13251 1995-04-01  england          53591  E92000001      56925.0
13252 1995-05-01  england          53678  E92000001      64192.0
…          …      …              …         …           …
13544 2019-09-01  england         249942  E92000001      64605.0
13545 2019-10-01  england         249376  E92000001      68677.0
13546 2019-11-01  england         248515  E92000001      67814.0
13547 2019-12-01  england         250410  E92000001          NaN
13548 2020-01-01  england         247355  E92000001          NaN

       no_of_crimes  year
13248           NaN  1995
13249           NaN  1995
13250           NaN  1995
13251           NaN  1995
13252           NaN  1995
…              …   …
13544           NaN  2019
13545           NaN  2019
13546           NaN  2019
13547           NaN  2019
13548           NaN  2020

[301 rows x 7 columns]
```

[41]:
```
df1.groupby('year').average_price.max()
```

[41]:
```
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
```

[42]:
```
df1.groupby('year').average_price.min()
```

[42]:
```
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
```

0.1.8 Question.6. What is Maximum & Minimum No. of Crimes recorded per area?

[45]:
```
data.groupby('area').no_of_crimes.max().sort_values(ascending= False)
#data.groupby('area').no_of_crimes.min().sort_values(ascending= True)
```

[45]:
```
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
```

0.1.9 Question.7. Show total count of records of each area, where average price is less than 100000.

[46]:
```
data[data.average_price < 100000].area.value_counts()
```

[46]:
```
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
hillingdon               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
```