London House Analysis Markdown Script

September 27, 2023

1 DATA ANALYSIS PROJECT.

1.1 The Housing Data Set

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

[2]:
```
House = pd.read_csv(r"C:\Users\user\Projects\Data-Analysis-Projects\Housing Data Analysis\HouseData.csv")
```

[3]:
```
House
```

[3]:
```
            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]
```

1.2 How to Analyze DataFrames

[4]:
```
House.head() # for the head of the data
```

[4]:
```
        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

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

[5]:
```
House.shape # for the total number of Rows and Columns
```

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

[6]:
```
House.index # for the Range of the Data
```

[6]:
```
RangeIndex(start=0, stop=13549, step=1)
```

[7]:
```
House.columns # for the names of the Columns
```

[7]:
```
Index(['date', 'area', 'average_price', 'code', 'houses_sold', 'no_of_crimes'],
dtype='object')
```

[8]:
```
House.dtypes #this show the Kind of data a particular column is
```

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

[9]:
```
House.nunique() #shows the number of unique values in a table column
```

[9]:
```
date              301
area               45
average_price    13343
code               45
houses_sold      3946
```

```
no_of_crimes      2669
dtype: int64
```

Lets see the Basic information about the Data

[10]:
```
House.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
```

1.2.1 Lets look at the Data to see the number of null values in each column

[11]:
```
House.isnull().sum()
```

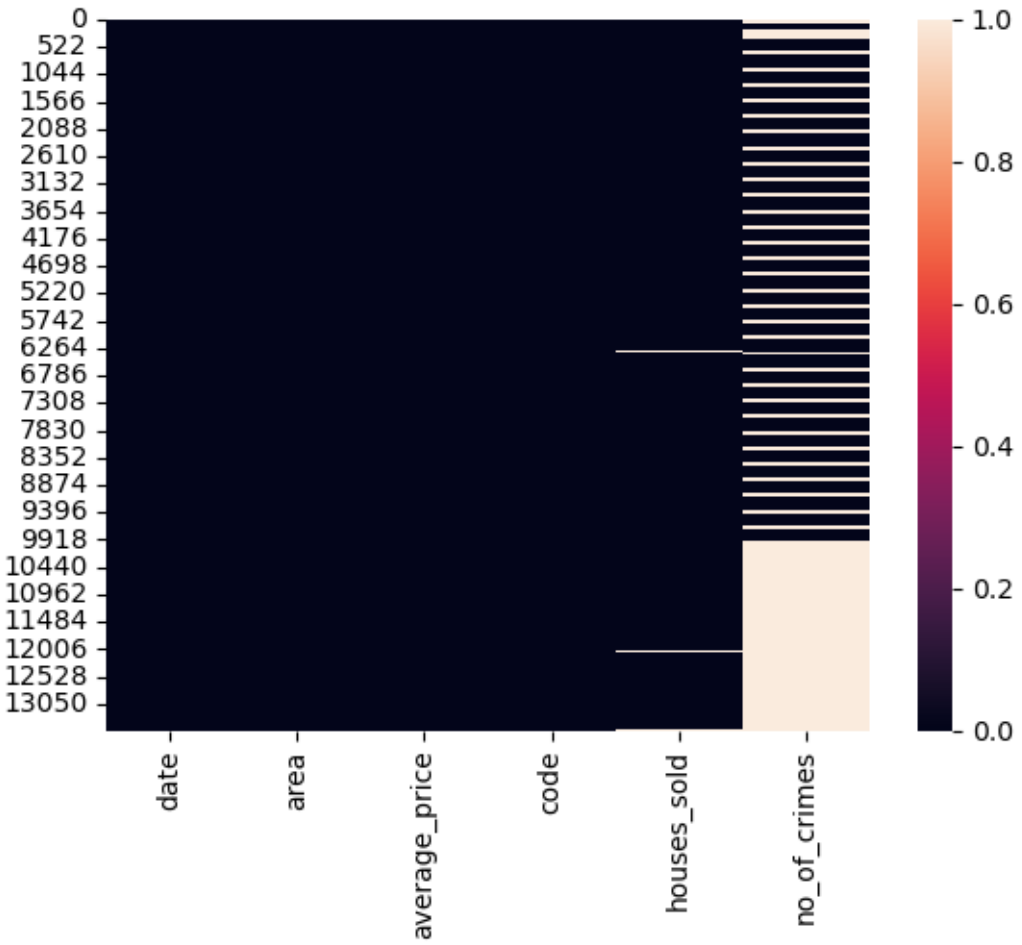
[11]:
```
date               0
area               0
average_price      0
code               0
houses_sold       94
no_of_crimes    6110
dtype: int64
```

1.3 Lets run some Analysis to view somw certain Updates

1.3.1 Firstly we would want to see the heatmap of the null values

[12]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

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



1.4 For the Questions Part

1 Convert the Datatype of 'Date' column to Date-Time format

[14]:
```
House.date = pd.to_datetime(House.date) # To convert a date Column TO a DateTime
```

[15]:
```
House.head()
```

[15]:

	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	
3	1995-04-01	city of london	77101	E09000001	7.0	
4	1995-05-01	city of london	84409	E09000001	10.0	

```
2           NaN
3           NaN
4           NaN
```

2 Add a new column ‘’year” in the dataframe, which contains years only.

[24]:
```
House['Year'] = House.date.dt.year
```

[25]:
```
House.head()
```

[25]:
```
        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
```

3 Add a new column ‘’month” as 2nd column in the dataframe, which contains month only.

[18]:
```
House['Month'] = House.date.dt.month
```

[19]:
```
House
```

[19]:
```
            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
…            …               …            …          …            …
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  Month
0               NaN  1995      1
1               NaN  1995      2
2               NaN  1995      3
3               NaN  1995      4
```

```
4               NaN  1995      5
…               …    …         …
13544           NaN  2019      9
13545           NaN  2019     10
13546           NaN  2019     11
13547           NaN  2019     12
13548           NaN  2020      1

[13549 rows x 8 columns]
```

[ ]:
```
#House.insert(1, 'Month', House.date.dt.month)
```

4 Remove the columns 'year' and 'month' from the dataframe.

[20]:
```
House.drop(['Month', 'Year'], axis = 1, inplace = True)
```

[21]:
```
House
```

[21]:
```
            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
…            …               …              …          …            …
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
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]
```

5 Show all the records where 'No. of Crimes' is 0. And, how many such records are there?

[22]:
```
House[House.no_of_crimes == 0]
```

[22]:

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

	no_of_crimes
72	0.0
73	0.0
74	0.0
75	0.0
76	0.0
..	…
178	0.0
179	0.0
180	0.0
181	0.0
182	0.0

[104 rows x 6 columns]

6 What is the maximum & minimum 'average_price' per year in england?

[26]:
```
england = House[House.area == 'england']
england.groupby('Year').average_price.max()
```

[26]:

Year	
1995	53901
1996	55755
1997	61564
1998	65743
1999	75071
2000	84191
2001	95992
2002	119982
2003	138985
2004	160330
2005	167244
2006	182031

```
2007    194764
2008    191750
2009    174136
2010    180807
2011    177335
2012    180129
2013    188544
2014    203639
2015    219582
2016    231922
2017    242628
2018    248620
2019    250410
2020    247355
Name: average_price, dtype: int64
```

[27]:
```
england = House[House.area == 'england']
england.groupby('Year').average_price.min()
```

[27]:
```
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
```

[28]:
```
england = House[House.area == 'england']
england.groupby('Year').average_price.mean() # For the Mean
```

[28]:
```
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 & Minimum No. of Crimes recorded per area?

[29]:
```
House.groupby('area').no_of_crimes.min().sort_values(ascending = False)
```

[29]:
```
area
westminster              3504.0
lambeth                  2381.0
southwark                2267.0
newham                   2130.0
camden                   2079.0
croydon                  2031.0
islington                1871.0
ealing                   1871.0
hackney                  1870.0
brent                    1850.0
barnet                   1703.0
```

```
lewisham                  1675.0
tower hamlets             1646.0
enfield                   1635.0
wandsworth                1582.0
waltham forest            1575.0
haringey                  1536.0
hounslow                  1529.0
greenwich                 1513.0
redbridge                 1487.0
hillingdon                1445.0
bromley                   1441.0
kensington and chelsea    1347.0
hammersmith and fulham    1323.0
barking and dagenham      1217.0
havering                  1130.0
harrow                     937.0
bexley                     860.0
merton                     819.0
sutton                     787.0
richmond upon thames       700.0
kingston upon thames       692.0
city of london               0.0
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
```

[30]:
```
House.groupby('area').no_of_crimes.max().sort_values(ascending = False)
```

[30]:
```
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
```

[31]:
```
House.groupby('area').no_of_crimes.mean().sort_values(ascending = False)
```

[31]:
```
area
westminster               5291.454148
lambeth                   3141.720524
camden                    3056.572052
southwark                 3028.563319
newham                    2851.432314
```

```
croydon                   2652.943231
ealing                    2607.061135
hackney                   2575.751092
tower hamlets             2542.253275
islington                 2515.991266
brent                     2415.602620
haringey                  2406.427948
lewisham                  2296.458515
barnet                    2278.441048
wandsworth                2209.716157
greenwich                 2154.899563
hillingdon                2141.768559
waltham forest            2113.598253
enfield                   2077.921397
hounslow                  2040.231441
bromley                   2020.131004
redbridge                 1940.480349
hammersmith and fulham    1914.676856
kensington and chelsea    1833.222707
barking and dagenham      1599.275109
havering                  1530.358079
bexley                    1299.458515
harrow                    1256.598253
merton                    1208.615721
sutton                    1078.213974
richmond upon thames      1041.620087
kingston upon thames       991.790393
city of london               0.423423
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
```

8 Show the total count of records of each area, where average price is less than 100000.

[32]:
```
House[House.average_price < 100000].area.value_counts()
```

[32]:
```
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: area, dtype: int64
```

We have come to the End
