In [3]: data head() Out[3]: date area average_price code houses_sold no_of_crimes Out[1]: city of london 91449 E0900001 17.0 NaN 1 2/1/1995 city of london 82203 E0900001 7.0 NaN 2 3/1/1995 city of london 79121 E0900001 14.0 NaN 3 4/1/1995 city of london 77101 E0900001 7.0 NaN 4 5/1/1995 city of london 84409 E0900001 10.0 NaN	
In [4]: date average_price code house_sold no_fcrimes 0 1/1/1995 city of london 91449 E09000001 17.0 NaN 1 2/1/1995 city of london 92000001 7.0 NaN 2 3/1/1995 city of london 77101 E09000001 1.0 NaN 4 5/1/1995 city of london 84409 E09000001 7.0 NaN 13548 9/1/2019 england 249376 E9200001 66677.0 NaN 13548 11/1/2019 england 249376 E9200001 66677.0 NaN 13548 11/1/2019 england 249376 E9200001 66677.0 NaN 13548 11/1/2019 england 249376 E920	
13548 1/1/2020 england 247355 E9200001 NaN NaN 13549 rows × 6 columns	
1 False False False False False False False True 2 False False False False False False True 3 False False False False False False True 4 False False False False False False True	
True shows null value are present and False show there are no null value. In [7]: data.isnull().sum() Out[7]: date	
Out[10]: <axessubplot:> </axessubplot:>	
In [11]: sns.heatmap(data.isnull()) plt.show()	
Convert the data types of 'Date' column to Date-time formate. In [12]: # data.dtypes # data.date=pd.to_datetime(data.date) In [13]: date	
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 [14]: data.dtypes #date: object Dut[14]: date object area object average_price int64 code object houses_sold float64 mo_of_crimes float64 dtype: object In [15]: data.date=pd.to_datetime(data.date) In [16]: data.dtypes #date: datetime64[ns]	
Add a new column 'year' in the dataframe which contains year only In [17]: # data['New_column']=df.Date_column.dt.year In [18]: 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 [19]: data['year']=data.date.dt.year Out[20]: data Out[20]: 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 8203 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	
# data['month']=data.date.dt.year 1995-05-01 city of london 84409 E09000001 10.0 NaN 1995	
Add a new column 'month' as 2nd column in the dataframe, which contains month only. If we add any column by default it goes to the end of the dataframe. In [22]: # df.insert(index, 'new_column_name', 'new_column_values') # df.insert(1, 'month', data.date.dt.month) In [23]: data_head() Out[23]: 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 79121 E09000001 14.0 NaN 1995 2 1995-03-01 city of london 77101 E09000001 7.0 NaN 1995 3 1995-04-01 city of london 77101 E09000001 10.0 NaN 1995 4 1995-05-01 city of london 84409 E09000001 10.0 NaN 1995	
data.insert(1, 'month', data.date.dt.month)	
Dut[27]:	
To remove permanently In [28]: data.drop(['month',],axis=1,inplace=True) In [29]: data.head(1) Out[29]: date area average_price code houses_sold no_of_crimes year 0 1995-01-01 city of london 91449 E09000001 17.0 NaN 1995 Show all the records where no_of_crimes is 0 and how many such records are there? In [30]: # df[df.no_of_crimes==0] # len(df[df.no_of_crimes==0])	
Dut[31]: data.no_of_crimes==0 Dut[31]: 0	
73 2001-02-01 city of london 198137 E09000001 37.0 0.0 2001 74 2001-03-01 city of london 189033 E09000001 34.0 0.0 2001 75 2001-04-01 city of london 205494 E09000001 38.0 0.0 2001 76 2001-05-01 city of london 223459 E09000001 30.0 0.0 2001	
<pre>In [33]: len(data[data.no_of_crimes==0]) Dut[33]: 104 What is the max. and min. 'average_price' per year in england? We have to make another df In [34]: #df1= data[data.area=='england'] In [35]: #df1.groupby('year').average_price.max()/min()/mean() In [36]: df1=data[data.area=='england'] In [37]: date area average_price code houses_sold no_of_crimes year</pre>	
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	
Unt[38]: df1.groupby('year').average_price.max() Dut[38]: year	
2013 188544 2014 203639 2015 219582 2016 231922 2017 242628 2018 248620 2019 250410 2020 247355 Name: average_price, dtype: int64 In [39]: df1.groupby('year').average_price.min() Dut[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 173046 2012 174161 2013 176816 2014 188265 2015 202856 2016 220361 2017 231593 2018 240428 2019 243281 2020 247355 Name: average_price, dtype: int64	
Dut[40]: year	
2016	
barking and dagenham 2049.0 barnet 2893.0 bexley 1914.0 brent 2937.0 bromley 2637.0 camden 4558.0 city of london 10.0 croydon 3263.0 ealing 3401.0 east midlands NaN east of england NaN enfield 2798.0 england NaN greenwich 2853.0 hackney 3466.0 hammersmith and fulham 2645.0 haringey 3199.0 harrow 1763.0 havering 1956.0 hillingdon 2819.0 hounslow 2817.0 inner london NaN islington 3384.0	
kensington and chelsea 2778.0 kingston upon thames 1379.0 lambeth 4701.0 lewisham 2813.0 london NaN merton 1623.0 newham 3668.0 north east NaN north west NaN outer london NaN redbridge 2560.0 richmond upon thames 1551.0 south east NaN south west NaN southwark 3821.0 sutton 1425.0 tower hamlets 3316.0 waltham forest 2941.0 wandsworth 3051.0 west midlands NaN verst midlands NaN yorks and the humber NaN	
Name: no_of_crimes, dtype: float64 In [43]: data.groupby('area').no_of_crimes.min() Dut[43]: area barking and dagenham 1217.0 barnet 1703.0 bexley 860.0 brent 1850.0 bromley 1441.0 camden 2079.0 city of london 0.0 croydon 2031.0 ealing 1871.0 east midlands NaN east of england NaN enfield 1635.0 england NaN greenwich 1513.0 hackney 1870.0 hammersmith and fulham 1323.0	
haringey 1536.0 harrow 937.0 havering 1130.0 hillingdon 1445.0 hounslow 1529.0 inner london NaN islington 1871.0 kensington and chelsea 1347.0 kingston upon thames 692.0 lambeth 2381.0 lewisham 1675.0 london NaN merton 819.0 nowth east NaN north west NaN outer london NaN redbridge 1487.0 richmond upon thames 700.0 south east NaN south west NaN south	
tower hamlets 1646.0 waltham forest 1575.0 wandsworth 1582.0 west midlands NaN westminster 3504.0 yorks and the humber NaN Name: no_of_crimes, dtype: float64 In [44]: data.groupby('area').no_of_crimes.min().sort_values(ascending=False) Dut[44]: 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 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 havering 1130.0 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 inner london NaN inner london NaN london NaN london NaN north east NaN north west NaN south east NaN west midlands NaN west midlands NaN west midlands Name: no_of_crimes, dtype: float64 Show the total count of records of each area, where average price is less than 100000 In [45]: date area average_price code houses_sold no_of_crimes year of 1995-01-01 city of london 91449 E0900001 17.0 NaN 1995	
0 1995-01-01 city of london 91449 E0900001 17.0 NaN 1995 In [46]: # data[data.average_price<100000].area.value_counts Dut[47]: data [data.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 E0900001 17.0 NaN 1995 2 1995-03-01 city of london 79121 E09000001 14.0 NaN 1995 3 1995-04-01 city of london 79121 E09000001 7.0 NaN 1995 4 1995-05-01 city of london 84409 E09000001 10.0 NaN 1995 - NaN 1995 3 1995-04-01 city of london 84409 E09000001 10.0 NaN 1995 - NaN 1995	
1331 2001-12-01 england 9599 E9200001 93329.0 NaN 2001 1332 2002-01-01 england 96215 E92000001 71678.0 NaN 2002 1333 2002-02-01 england 96676 E9200001 77131.0 NaN 2002 1334 2002-03-01 england 98962 E9200001 102828.0 NaN 2002 2209 rows × 7 columns In [48]: data[data.average_price<100000].area.value_counts() Dut[48]: north east 112 north west 111 yorks and the humber 110 east midlands 96 west midlands 96 west midlands 94 england 87 barking and dagenham 85 south west 78 east of england 76	
east of england 76 newham 72 bexley 64 waltham forest 64 lewisham 62 havering 60 greenwich 59 south east 59 croydon 57 sutton 54 enfield 54 hackney 53 redbridge 52 southwark 48 tower hamlets 47 outer london 46 hillingdon 44 hounslow 41 lambeth 41 brent 40 london 39 merton 35 haringey 33	
Naringey 33 bromley 33 ealing 31 inner london 31 harrow 30 kingston upon thames 30 wandsworth 26 barnet 25 islington 19 city of london 11 Name: area, dtype: int64 In []:	