1.9

January 11, 2024

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

#
qq = 'data/10-17homeless .csv'
aa = 'data/CRIME.csv'
homeless = pd.read_csv(qq)
crime1 = pd.read_csv(aa)
population=pd.read_csv('data/population.csv')
homeless20=pd.read_csv('data/20  .csv')

#
homeless = homeless[['Year', 'London ']].copy()
homeless.columns = ['Year', 'Homeless_Count']
homeless['Year'] = homeless['Year'].astype(int)
homeless['Homeless_Count'] = homeless['Homeless_Count'].str.replace(',', '').
    ↪astype(int)
homeless_london_2010_2017_newest = homeless[homeless['Year'].between(2010,␣
    ↪2017)]

#
crime = crime1.iloc[:, 3:].sum()
crime = crime.reset_index()
crime.columns = ['Year', 'Total_Crimes']
crime['Year'] = crime['Year'].astype(int)

#
df_analysis_2010_2017_newest = pd.merge(homeless_london_2010_2017_newest,␣
    ↪crime, on='Year', how='inner')

df_analysis_2010_2017_newest.shape
df_analysis_2010_2017_newest.head()
```

[1]:

	Year	Homeless_Count	Total_Crimes
0	2010	10180	614376.0
1	2011	12720	556881.0
2	2012	15660	787392.0
3	2013	17030	708584.0

```
4  2014           17530      701361.0
```

[2]:
```
crime1.head()
```

[2]:
```
                   MajorText                          MinorText  \
0  Arson and Criminal Damage                              Arson
1  Arson and Criminal Damage                    Criminal Damage
2                   Burglary  Burglary Business and Community
3                   Burglary                 Domestic Burglary
4              Drug Offences                  Drug Trafficking

            BoroughName  2010    2011  2012  2013  2014  2015  2016  2017  \
0  Barking and Dagenham    84   132.0    75    52    65    77    74    78
1  Barking and Dagenham  1730  2017.0  1613  1550  1553  1678  1816  1631
2  Barking and Dagenham   473   740.0   665   574   541   448   454   364
3  Barking and Dagenham  1131  1625.0  1795  1669  1369  1184   840  1229
4  Barking and Dagenham    74     NaN    76    69    89    53    79    64

   2018  2019  2020  2021
0    58    68    48    65
1  1392  1435  1276  1272
2   387   361   282   253
3  1285  1185   935   833
4    76    88   163   126
```

[2]:
```
#
#  'Area Name'  borough  'population'
df_population_cleaned = population[['Area Name', 'population']].dropna()
df_population_cleaned.columns = ['Borough', 'Population']
df_population_cleaned['Population'] = df_population_cleaned['Population'].
 ↪astype(int)

#
year_columns = crime1.columns[3:]

#
df_crime_borough_yearly = crime1.melt(id_vars=['BoroughName'],
                                                value_vars=year_columns,
                                                var_name='Year',
                                                value_name='Crime_Count')
##.melt():   Pandas         wide format     long format

#id_vars=['LookUp_BoroughName']          melting

#value_vars=year_columns          year_columns

#var_name='Year'            year_columns
```

```
#value_name='Crime_Count'



df_crime_borough_yearly['Year'] = df_crime_borough_yearly['Year'].astype(int)

#   borough
df_crime_borough_yearly_total = df_crime_borough_yearly.groupby(['BoroughName', 'Year']).sum().reset_index()
df_crime_borough_yearly_total.columns = ['Borough', 'Year', 'Crime_Count']

#
df_crime_population_merged = pd.merge(df_crime_borough_yearly_total, df_population_cleaned, on='Borough', how='inner')

#      / *1000
df_crime_population_merged['Crime_Rate'] = (df_crime_population_merged['Crime_Count'] / df_crime_population_merged['Population']) * 1000


#
df_crime_population_merged.head()
# df_crime_population_merged:   borough
# Crime_Rate:  borough         / *1000
```

[2]:

	Borough	Year	Crime_Count	Population	Crime_Rate
0	Barking and Dagenham	2010	14454.0	219000	66.000000
1	Barking and Dagenham	2011	12732.0	219000	58.136986
2	Barking and Dagenham	2012	17297.0	219000	78.981735
3	Barking and Dagenham	2013	16500.0	219000	75.342466
4	Barking and Dagenham	2014	16210.0	219000	74.018265

[ ]:

[3]:
```
df_crime_yearly_total = df_crime_borough_yearly.groupby('Year')['Crime_Count'].sum().reset_index()
df_crime_yearly_total['crime_rate']=df_crime_yearly_total['Crime_Count']/8992000*1000
#
df_crime_yearly_total
# df_crime_yearly_total:
```

[3]:

	Year	Crime_Count	crime_rate
0	2010	614376.0	68.324733
1	2011	556881.0	61.930716

```
2   2012     787392.0   87.565836
3   2013     708584.0   78.801601
4   2014     701361.0   77.998332
5   2015     734664.0   81.701957
6   2016     763392.0   84.896797
7   2017     823381.0   91.568172
8   2018     840589.0   93.481873
9   2019     920619.0  102.382006
10  2020     791491.0   88.021686
11  2021     726630.0   80.808496
```

[4]:
```
import numpy as np
import scipy.stats as stats
df_analysis_merged = pd.
 ↪merge(df_analysis_2010_2017_newest,df_crime_yearly_total, on='Year',␣
 ↪how='inner')

#
pearson_corr_avg_rate, p_value_avg_rate = stats.
 ↪pearsonr(df_analysis_merged['Homeless_Count'],␣
 ↪df_analysis_merged['crime_rate'])

#     p
pearson_corr_avg_rate, p_value_avg_rate
# pearson_corr_avg_rate:
# p_value_avg_rate:     p

#
#     0.622   2010-2017

#   p 0.10                         p  0.05
```

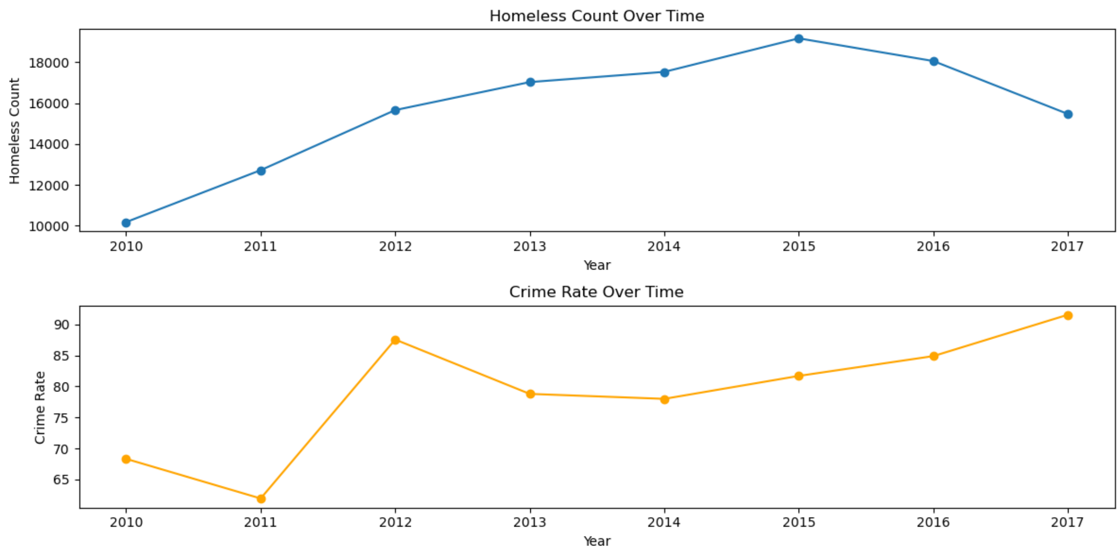
[4]:
```
(0.6215973820711973, 0.09992364018841875)
```

[6]:
```
df_analysis_merged
```

[6]:

	Year	Homeless_Count	Total_Crimes	Crime_Count	crime_rate
0	2010	10180	614376.0	614376.0	68.324733
1	2011	12720	556881.0	556881.0	61.930716
2	2012	15660	787392.0	787392.0	87.565836
3	2013	17030	708584.0	708584.0	78.801601
4	2014	17530	701361.0	701361.0	77.998332
5	2015	19170	734664.0	734664.0	81.701957
6	2016	18060	763392.0	763392.0	84.896797
7	2017	15470	823381.0	823381.0	91.568172

[5]:
```python
##
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm




#
# df_analysis:

#
plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 1)
plt.plot(df_analysis_merged['Year'], df_analysis_merged['Homeless_Count'],␣
↪marker='o')
plt.title('Homeless Count Over Time')
plt.xlabel('Year')
plt.ylabel('Homeless Count')

plt.subplot(2, 1, 2)
plt.plot(df_analysis_merged['Year'], df_analysis_merged['crime_rate'],␣
↪marker='o', color='orange')
plt.title('Crime Rate Over Time')
plt.xlabel('Year')
plt.ylabel('Crime Rate')
plt.tight_layout()
plt.savefig('Homeless Count Over Time.png', dpi=300, bbox_inches='tight')
plt.show()


#
#  ADF

#
correlation = df_analysis_merged['Homeless_Count'].
↪corr(df_analysis_merged['crime_rate'])

#  VAR
model = sm.tsa.VAR(df_analysis_merged[['Homeless_Count', 'crime_rate']])
results = model.fit()
#
results.summary()
```



[5]:
```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Wed, 10, Jan, 2024
Time:                     18:06:02
--------------------------------------------------------------------
No. of Equations:         2.00000    BIC:                    19.2284
Nobs:                     7.00000    HQIC:                   18.7017
Log likelihood:          -81.3269    FPE:                2.64426e+08
AIC:                      19.2748    Det(Omega_mle):     1.29569e+08
--------------------------------------------------------------------
Results for equation Homeless_Count
================================================================================
====
                       coefficient       std. error           t-stat
prob
--------------------------------------------------------------------------------
----
const                 11043.760759      5344.987252            2.066
0.039
L1.Homeless_Count         0.659958         0.284364            2.321
0.020
L1.crime_rate           -63.731300        99.978338           -0.637
0.524
================================================================================
====

Results for equation crime_rate
```

```
================================================================================
====
                        coefficient       std. error         t-stat
prob
--------------------------------------------------------------------------------
----
const                     72.614492        28.120245          2.582
0.010
L1.Homeless_Count          0.003065         0.001496          2.049
0.040
L1.crime_rate             -0.521133         0.525991         -0.991
0.322
================================================================================
====

Correlation matrix of residuals
                  Homeless_Count  crime_rate
Homeless_Count          1.000000   -0.252769
crime_rate             -0.252769    1.000000
```

[6]:

```
##GWR
import esda
import pandas as pd
import geopandas as gpd
from geopandas import GeoDataFrame
import libpysal as lps
import numpy as np
import matplotlib.pyplot as plt
from shapely.geometry import Point
import contextily as ctx
from pylab import figure, scatter, show
%matplotlib inline
from splot.esda import plot_moran
from splot.esda import moran_scatterplot
from esda.moran import Moran_Local
from splot.esda import lisa_cluster
from splot.esda import plot_local_autocorrelation
from mgwr.sel_bw import Sel_BW
from mgwr.gwr import GWR, MGWR
```

[7]:

```
map=gpd.read_file('data/statistical-gis-boundaries-london/ESRI/
 ↪London_Borough_Excluding_MHW.shp')
display(map)
```

```
                     NAME    GSS_CODE   HECTARES  NONLD_AREA ONS_INNER  \
```

```
0     Kingston upon Thames  E09000021   3726.117      0.000         F
1                  Croydon  E09000008   8649.441      0.000         F
2                  Bromley  E09000006  15013.487      0.000         F
3                 Hounslow  E09000018   5658.541     60.755         F
4                   Ealing  E09000009   5554.428      0.000         F
5                 Havering  E09000016  11445.735    210.763         F
6               Hillingdon  E09000017  11570.063      0.000         F
7                   Harrow  E09000015   5046.330      0.000         F
8                    Brent  E09000005   4323.270      0.000         F
9                   Barnet  E09000003   8674.837      0.000         F
10                 Lambeth  E09000022   2724.940     43.927         T
11               Southwark  E09000028   2991.340    105.139         T
12                Lewisham  E09000023   3531.706     16.795         T
13               Greenwich  E09000011   5044.190    310.785         F
14                  Bexley  E09000004   6428.649    370.619         F
15                 Enfield  E09000010   8220.025      0.000         F
16          Waltham Forest  E09000031   3880.793      0.000         F
17               Redbridge  E09000026   5644.225      2.300         F
18                  Sutton  E09000029   4384.698      0.000         F
19    Richmond upon Thames  E09000027   5876.111    135.443         F
20                  Merton  E09000024   3762.466      0.000         F
21              Wandsworth  E09000032   3522.022     95.600         T
22  Hammersmith and Fulham  E09000013   1715.409     75.648         T
23  Kensington and Chelsea  E09000020   1238.379     25.994         T
24             Westminster  E09000033   2203.005     54.308         T
25                  Camden  E09000007   2178.932      0.000         T
26           Tower Hamlets  E09000030   2157.501    179.707         T
27               Islington  E09000019   1485.664      0.000         T
28                 Hackney  E09000012   1904.902      0.000         T
29                Haringey  E09000014   2959.837      0.000         T
30                  Newham  E09000025   3857.806    237.637         T
31    Barking and Dagenham  E09000002   3779.934    169.150         F
32          City of London  E09000001    314.942     24.546         T

   SUB_2009 SUB_2006                                           geometry
0      None     None  POLYGON ((516401.600 160201.800, 516407.300 16…
1      None     None  POLYGON ((535009.200 159504.700, 535005.500 15…
2      None     None  POLYGON ((540373.600 157530.400, 540361.200 15…
3      None     None  POLYGON ((521975.800 178100.000, 521967.700 17…
4      None     None  POLYGON ((510253.500 182881.600, 510249.900 18…
5      None     None  POLYGON ((549893.900 181459.800, 549894.600 18…
6      None     None  POLYGON ((510599.800 191689.500, 510615.200 19…
7      None     None  POLYGON ((510599.800 191689.500, 510660.000 19…
8      None     None  POLYGON ((525201.000 182512.600, 525181.500 18…
9      None     None  POLYGON ((524579.900 198355.200, 524594.300 19…
10     None     None  POLYGON ((530046.800 177893.400, 530048.400 17…
11     None     None  POLYGON ((531335.600 180529.500, 531337.700 18…
12     None     None  POLYGON ((536691.000 178958.600, 536691.900 17…
```

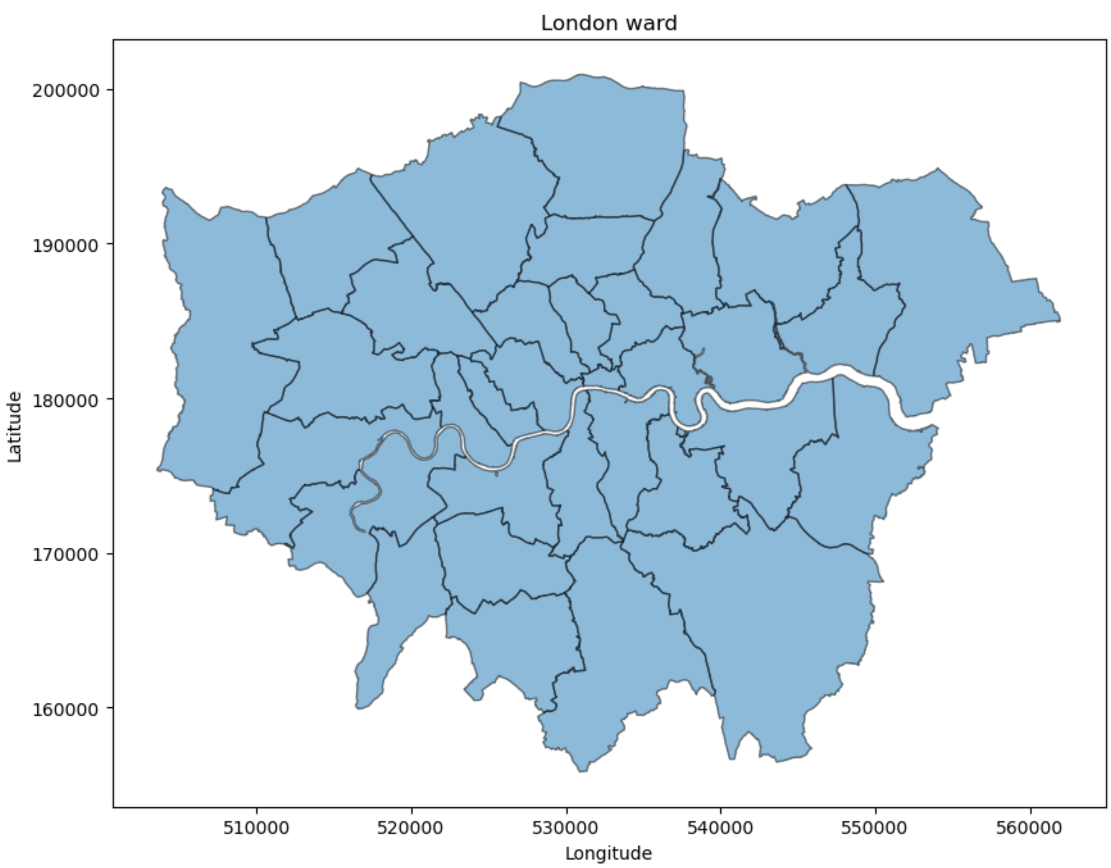
```
13      None     None  MULTIPOLYGON (((537238.700 178137.700, 537242…
14      None     None  POLYGON ((547226.200 181299.300, 547320.900 18…
15      None     None  POLYGON ((531023.500 200933.600, 531039.900 20…
16      None     None  POLYGON ((539923.100 191863.100, 539928.100 19…
17      None     None  POLYGON ((543595.500 184832.800, 543577.100 18…
18      None     None  POLYGON ((528552.300 159658.100, 528399.700 15…
19      None     None  POLYGON ((516677.500 175383.800, 516678.600 17…
20      None     None  POLYGON ((529906.200 167417.300, 529902.200 16…
21      None     None  POLYGON ((523489.600 176224.800, 523500.600 17…
22      None     None  POLYGON ((521975.800 178100.000, 521973.000 17…
23      None     None  POLYGON ((526219.700 176948.000, 526208.800 17…
24      None     None  POLYGON ((528549.500 177903.800, 528542.600 17…
25      None     None  POLYGON ((528840.200 187217.800, 528834.600 18…
26      None     None  POLYGON ((533387.600 180516.400, 533389.800 18…
27      None     None  POLYGON ((529153.600 185861.400, 529144.800 18…
28      None     None  POLYGON ((531928.400 187801.500, 531935.700 18…
29      None     None  POLYGON ((531928.400 187801.500, 531919.200 18…
30      None     None  MULTIPOLYGON (((544065.000 183254.100, 544062…
31      None     None  MULTIPOLYGON (((543905.400 183199.100, 543905…
32      None     None  POLYGON ((531145.100 180782.100, 531143.800 18…
```

[8]:

```python
# Load & plot: London ward (city merged)

map=gpd.read_file('data/statistical-gis-boundaries-london/ESRI/
 ↪London_Borough_Excluding_MHW.shp')[['NAME','geometry']]

display(map)
map.plot(figsize=(10, 8), alpha=0.5, edgecolor='k')
plt.title('London ward')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.show()
```

```
                    NAME                                          geometry
0     Kingston upon Thames  POLYGON ((516401.600 160201.800, 516407.300 16…
1                  Croydon  POLYGON ((535009.200 159504.700, 535005.500 15…
2                  Bromley  POLYGON ((540373.600 157530.400, 540361.200 15…
3                 Hounslow  POLYGON ((521975.800 178100.000, 521967.700 17…
4                   Ealing  POLYGON ((510253.500 182881.600, 510249.900 18…
5                 Havering  POLYGON ((549893.900 181459.800, 549894.600 18…
6               Hillingdon  POLYGON ((510599.800 191689.500, 510615.200 19…
7                   Harrow  POLYGON ((510599.800 191689.500, 510660.000 19…
8                    Brent  POLYGON ((525201.000 182512.600, 525181.500 18…
9                   Barnet  POLYGON ((524579.900 198355.200, 524594.300 19…
10                 Lambeth  POLYGON ((530046.800 177893.400, 530048.400 17…
11               Southwark  POLYGON ((531335.600 180529.500, 531337.700 18…
12                Lewisham  POLYGON ((536691.000 178958.600, 536691.900 17…
13               Greenwich  MULTIPOLYGON (((537238.700 178137.700, 537242…
```

```
14                  Bexley  POLYGON ((547226.200 181299.300, 547320.900 18…
15                 Enfield  POLYGON ((531023.500 200933.600, 531039.900 20…
16          Waltham Forest  POLYGON ((539923.100 191863.100, 539928.100 19…
17               Redbridge  POLYGON ((543595.500 184832.800, 543577.100 18…
18                  Sutton  POLYGON ((528552.300 159658.100, 528399.700 15…
19    Richmond upon Thames  POLYGON ((516677.500 175383.800, 516678.600 17…
20                  Merton  POLYGON ((529906.200 167417.300, 529902.200 16…
21              Wandsworth  POLYGON ((523489.600 176224.800, 523500.600 17…
22  Hammersmith and Fulham  POLYGON ((521975.800 178100.000, 521973.000 17…
23  Kensington and Chelsea  POLYGON ((526219.700 176948.000, 526208.800 17…
24             Westminster  POLYGON ((528549.500 177903.800, 528542.600 17…
25                  Camden  POLYGON ((528840.200 187217.800, 528834.600 18…
26           Tower Hamlets  POLYGON ((533387.600 180516.400, 533389.800 18…
27               Islington  POLYGON ((529153.600 185861.400, 529144.800 18…
28                 Hackney  POLYGON ((531928.400 187801.500, 531935.700 18…
29                Haringey  POLYGON ((531928.400 187801.500, 531919.200 18…
30                  Newham  MULTIPOLYGON (((544065.000 183254.100, 544062…
31    Barking and Dagenham  MULTIPOLYGON (((543905.400 183199.100, 543905…
32          City of London  POLYGON ((531145.100 180782.100, 531143.800 18…
```



[11]:
```
homeless20
```

[11]:
```
    Unnamed: 0                                   Unnamed: 1  Unnamed: 2  \
0    E92000001                                      ENGLAND         NaN
1    E12000007                                       London         NaN
2            -                              Rest of England         NaN
3          NaN                                          NaN         NaN
4    E12000001                                   North East         NaN
..         …                                             …          …
339        NaN                                          NaN         NaN
340     Source  DLUHC H-CLIC Homelessness returns (quarterly)         NaN
341        NaN    email: homelessnessstats@levellingup.gov.uk         NaN
342        NaN                      Latest update: September 2022         NaN
343        NaN                           Next update: Autumn 2022         NaN

     Unnamed: 3 Total initial assessments1,2  Unnamed: 5  \
0           NaN                      284,330         NaN
1           NaN                       55,350         NaN
2           NaN                      228,980         NaN
3           NaN                          NaN         NaN
4           NaN                       15,370         NaN
..          …                           …          …
339         NaN                          NaN         NaN
340         NaN                          NaN         NaN
341         NaN                          NaN         NaN
342         NaN                          NaN         NaN
343         NaN                          NaN         NaN

    Total owed a prevention or relief duty\n  \
0                                    270,560
1                                     52,210
2                                    218,350
3                                        NaN
4                                     14,790
..                                       …
339                                      NaN
340                                      NaN
341                                      NaN
342                                      NaN
343                                      NaN

    Threatened with homelessness within 56 days - \nPrevention duty owed  \
0                                             119,890
1                                              24,000
2                                              95,880
3                                                 NaN
4                                               7,180
```

```
..                                                 …
339                                              NaN
340                                              NaN
341                                              NaN
342                                              NaN
343                                              NaN

    due to service of valid Section 21 Notice3 Homeless - \nRelief duty owed4  \
0                                          8,960                       150,670
1                                          1,600                        28,200
2                                          7,360                       122,470
3                                            NaN                           NaN
4                                            350                         7,600
..                                           …                           …
339                                          NaN                           NaN
340                                          NaN                           NaN
341                                          NaN                           NaN
342                                          NaN                           NaN
343                                          NaN                           NaN

     Unnamed: 10  \
0            NaN
1            NaN
2            NaN
3            NaN
4            NaN
..           …
339          NaN
340          NaN
341          NaN
342          NaN
343          NaN

    Not homeless nor threatened with homelessness within 56 days - no duty owed
\
0                                                 13,770
1                                                  3,140
2                                                 10,630
3                                                    NaN
4                                                    590
..                                                 …
339                                                  NaN
340                                                  NaN
341                                                  NaN
342                                                  NaN
343                                                  NaN
```

```
     Unnamed: 12 Number of households\n in area4 (000s)  \
0            NaN                                23,688.89
1            NaN                                 3,563.45
2            NaN                                20,125.44
3            NaN                                      NaN
4            NaN                                 1,182.19
..           …                                     …
339          NaN                                      NaN
340          NaN                                      NaN
341          NaN                                      NaN
342          NaN                                      NaN
343          NaN                                      NaN

    Households assessed as threatened with homelessness\nper (000s)  \
0                                                 5.06
1                                                 6.74
2                                                 4.76
3                                                  NaN
4                                                 6.07
..                                                 …
339                                                NaN
340                                                NaN
341                                                NaN
342                                                NaN
343                                                NaN

    Households assessed as homeless\nper (000s)  \
0                                          6.36
1                                          7.91
2                                          6.09
3                                           NaN
4                                          6.43
..                                          …
339                                         NaN
340                                         NaN
341                                         NaN
342                                         NaN
343                                         NaN

    homeless end prevention duty dindt get  \
0                                      NaN
1                                     9870
2                                    37390
3                                        0
4                                     2560
..                                     …
339                                    NaN
```

```
340                                         NaN
341                                         NaN
342                                         NaN
343                                         NaN

    homeless with relief duty didn’t get homeless didn’t get accomadation
0                                       NaN                           NaN
1                                      7570                         17440
2                                     37070                         74460
3                                         0                             0
4                                      2970                          5530
..                                      …                             …
339                                     NaN                           NaN
340                                     NaN                           NaN
341                                     NaN                           NaN
342                                     NaN                           NaN
343                                     NaN                           NaN

[344 rows x 19 columns]
```

[12]:
```
homeless20.columns
```

[12]:
```
Index(['Unnamed: 0', 'Unnamed: 1', 'Unnamed: 2', 'Unnamed: 3',
       'Total initial assessments1,2', 'Unnamed: 5',
       'Total owed a prevention or relief duty\n',
       'Threatened with homelessness within 56 days - \nPrevention duty owed',
       'due to service of valid Section 21 Notice3',
       'Homeless - \nRelief duty owed4', 'Unnamed: 10',
       'Not homeless nor threatened with homelessness within 56 days - no duty
owed',
       'Unnamed: 12', 'Number of households\n in area4 (000s)',
       'Households assessed as threatened with homelessness\nper (000s)',
       'Households assessed as homeless\nper (000s)',
       'homeless end prevention duty dindt get',
       'homeless with relief duty didn’t get',
       'homeless didn’t get accomadation'],
      dtype='object')
```

[9]:
```
homeless2020=homeless20[['Unnamed: 1','Total owed a prevention or relief␣
↪duty\n','Households assessed as homeless\nper (000s)','homeless end␣
↪prevention duty dindt get',
       'homeless with relief duty didn’t get',
       'homeless didn’t get accomadation']]
homeless2020_copy = homeless2020.copy()
homeless2020_copy.rename(columns={'Unnamed: 1': 'Borough'}, inplace=True)
homeless2020=homeless2020_copy
```

[14]:
```
homeless2020.head()
```

[14]:
```
           Borough Total owed a prevention or relief duty\n  \
0          ENGLAND                                 270,560
1           London                                  52,210
2  Rest of England                                 218,350
3              NaN                                     NaN
4       North East                                  14,790

  Households assessed as homeless\nper (000s)  \
0                                        6.36
1                                        7.91
2                                        6.09
3                                         NaN
4                                        6.43

  homeless end prevention duty dindt get homeless with relief duty didn't get  \
0                                    NaN                                    NaN
1                                   9870                                   7570
2                                  37390                                  37070
3                                      0                                      0
4                                   2560                                   2970

  homeless didn't get accomadation
0                              NaN
1                            17440
2                            74460
3                                0
4                             5530
```

[10]:
```
#      2020
df_crime_2020 = df_crime_population_merged[df_crime_population_merged['Year']␣
 ↪== 2020].dropna()



#  2020 borough
df_crime_2020

#
new_row = {'Borough':'City of London', 'Year': '2020','Crime_Count':␣
 ↪'908','Population': '9000','Crime_Rate': '100.9'}

#  append
df_crime_2020 = pd.concat([df_crime_2020, pd.DataFrame([new_row])],␣
 ↪ignore_index=True)
```

[11]:
```
# Merge & View: data & map
Data=pd.merge(df_crime_2020, homeless2020, on='Borough', how='inner')
map.rename(columns={'NAME': 'Borough'}, inplace=True)

columnsto=[ 'Year', 'Crime_Count', 'Population', 'Crime_Rate',
       'Households assessed as homeless\nper (000s)',
       'homeless end prevention duty dindt get',
       'homeless with relief duty didn't get',
       'homeless didn't get accomadation']
Data['Total owed a prevention or relief duty\n'] = Data['Total owed a␣
 ↪prevention or relief duty\n'].str.replace(',', '').astype(float)
Data[columnsto] = Data[columnsto].apply(pd.to_numeric, errors='coerce')


Data
```

[11]:

	Borough	Year	Crime_Count	Population	Crime_Rate	\
0	Barking and Dagenham	2020	19187.0	219000	87.611872	
1	Barnet	2020	27388.0	396000	69.161616	
2	Bexley	2020	15601.0	251000	62.155378	
3	Brent	2020	28799.0	330000	87.269697	
4	Bromley	2020	22013.0	335000	65.710448	
5	Camden	2020	29574.0	260000	113.746154	
6	Croydon	2020	33313.0	390000	85.417949	
7	Ealing	2020	29361.0	344000	85.351744	
8	Enfield	2020	28862.0	340000	84.888235	
9	Greenwich	2020	25752.0	289000	89.107266	
10	Hackney	2020	30718.0	285000	107.782456	
11	Hammersmith and Fulham	2020	19018.0	180000	105.655556	
12	Haringey	2020	28855.0	279000	103.422939	
13	Harrow	2020	15861.0	250000	63.444000	
14	Havering	2020	16831.0	261000	64.486590	
15	Hillingdon	2020	24954.0	310000	80.496774	
16	Hounslow	2020	24314.0	273000	89.062271	
17	Islington	2020	25456.0	240000	106.066667	
18	Kensington and Chelsea	2020	17935.0	154000	116.461039	
19	Kingston upon Thames	2020	11354.0	179000	63.430168	
20	Lambeth	2020	31014.0	328000	94.554878	
21	Lewisham	2020	27578.0	309000	89.249191	
22	Merton	2020	13588.0	209000	65.014354	
23	Newham	2020	33219.0	358000	92.790503	
24	Redbridge	2020	23146.0	311000	74.424437	
25	Richmond upon Thames	2020	12275.0	198000	61.994949	
26	Southwark	2020	32246.0	321000	100.454829	
27	Sutton	2020	13059.0	206000	63.393204	
28	Tower Hamlets	2020	31614.0	320000	98.793750	
29	Waltham Forest	2020	23559.0	280000	84.139286	

```
30             Wandsworth  2020      24918.0      325000   76.670769
31            Westminster  2020      48558.0      253000  191.928854
32         City of London  2020        908.0        9000  100.900000

    Total owed a prevention or relief duty\n  \
0                                     1434.0
1                                     2035.0
2                                      794.0
3                                     2934.0
4                                     1170.0
5                                     1104.0
6                                     2424.0
7                                     2442.0
8                                     1930.0
9                                     1556.0
10                                    2166.0
11                                    1065.0
12                                    2386.0
13                                     646.0
14                                    1735.0
15                                    1731.0
16                                    1556.0
17                                    1787.0
18                                    1061.0
19                                     433.0
20                                    3228.0
21                                    3158.0
22                                     551.0
23                                    3228.0
24                                    1731.0
25                                     297.0
26                                    3413.0
27                                     808.0
28                                    1940.0
29                                    1940.0
30                                    1940.0
31                                    1604.0
32                                      12.0

    Households assessed as homeless\nper (000s)  \
0                                          7.07
1                                          5.98
2                                          5.20
3                                         14.11
4                                          4.67
5                                          6.63
6                                         11.40
```

```
7                                               9.81
8                                               7.35
9                                               8.50
10                                             11.93
11                                              8.12
12                                              8.29
13                                              5.01
14                                              7.15
15                                              3.99
16                                              8.50
17                                              7.37
18                                              8.64
19                                              3.20
20                                             12.16
21                                             10.09
22                                              1.96
23                                             12.16
24                                              3.99
25                                              2.37
26                                             17.87
27                                              5.46
28                                              8.64
29                                              9.07
30                                              8.64
31                                             10.20
32                                              2.33
```

```
    homeless end prevention duty dindt get  \
0                                      500.0
1                                      429.0
2                                        NaN
3                                      426.0
4                                        NaN
5                                      166.0
6                                      266.0
7                                      405.0
8                                      443.0
9                                      252.0
10                                     427.0
11                                     171.0
12                                     644.0
13                                     113.0
14                                     538.0
15                                     380.0
16                                     252.0
17                                      67.0
18                                      97.0
```

19	87.0
20	548.0
21	868.0
22	84.0
23	548.0
24	380.0
25	51.0
26	151.0
27	84.0
28	309.0
29	613.0
30	157.0
31	82.0
32	1.0

	homeless with relief duty didn't get	homeless didn't get accomadation
0	320.0	820
1	513.0	942
2	NaN	550
3	481.0	907
4	NaN	260
5	120.0	286
6	137.0	403
7	444.0	849
8	430.0	873
9	164.0	416
10	135.0	562
11	177.0	348
12	664.0	1308
13	83.0	196
14	233.0	771
15	386.0	766
16	164.0	416
17	435.0	502
18	140.0	237
19	78.0	165
20	404.0	952
21	443.0	1311
22	203.0	287
23	404.0	952
24	386.0	766
25	39.0	90
26	68.0	219
27	175.0	259
28	187.0	496
29	108.0	721
30	102.0	259

```
31                                    73.0                                   155
32                                     1.0                                     2
```

[12]:
```python
myData=pd.merge(Data, map, on='Borough', how='inner')
myData=gpd.GeoDataFrame(myData)
display(myData.describe())
myData.head()
# myData.to_csv('data.csv',index=False)
```

```
         Year   Crime_Count     Population  Crime_Rate  \
count    33.0     33.000000      33.000000   33.000000
mean   2020.0  23964.484848  272484.848485   88.637510
std       0.0   8826.486897   76806.624556   24.706462
min    2020.0    908.000000    9000.000000   61.994949
25%    2020.0  17935.000000  240000.000000   69.161616
50%    2020.0  24954.000000  280000.000000   87.269697
75%    2020.0  29361.000000  325000.000000  100.454829
max    2020.0  48558.000000  396000.000000  191.928854

       Total owed a prevention or relief duty\n  \
count                                 33.000000
mean                                1704.212121
std                                  890.975054
min                                   12.000000
25%                                 1065.000000
50%                                 1731.000000
75%                                 2166.000000
max                                 3413.000000

       Households assessed as homeless\nper (000s)  \
count                                    33.000000
mean                                      7.813939
std                                       3.574613
min                                       1.960000
25%                                       5.200000
50%                                       8.120000
75%                                       9.810000
max                                      17.870000

       homeless end prevention duty dindt get  \
count                               31.000000
mean                               307.709677
std                                216.088283
min                                  1.000000
25%                                105.000000
50%                                266.000000
75%                                436.000000
```

```
max                                 868.000000

       homeless with relief duty didn’t get  homeless didn’t get accomadation
count                              31.000000                         33.000000
mean                              248.290323                        546.848485
std                               171.502419                        346.442557
min                                 1.000000                          2.000000
25%                               114.000000                        259.000000
50%                               177.000000                        496.000000
75%                               404.000000                        820.000000
max                               664.000000                       1311.000000
```

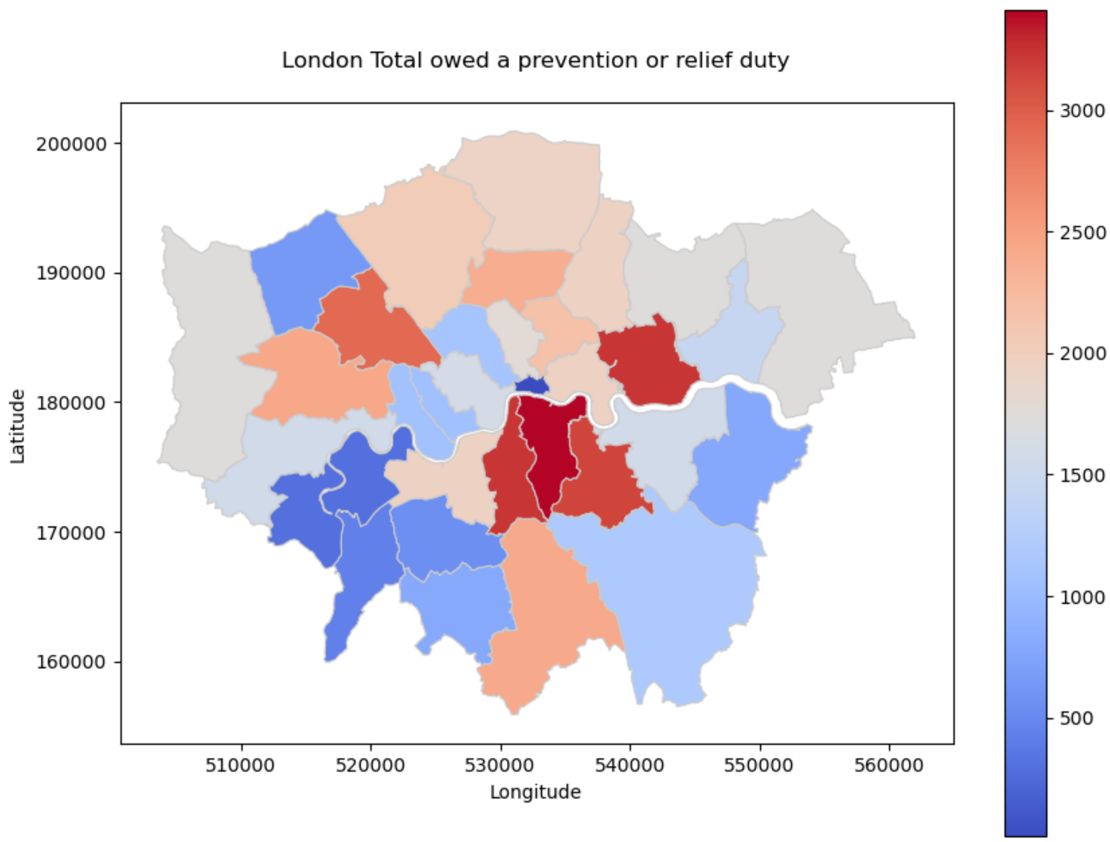
[12]:

```
               Borough  Year  Crime_Count  Population  Crime_Rate  \
0  Barking and Dagenham  2020      19187.0      219000   87.611872
1                Barnet  2020      27388.0      396000   69.161616
2                Bexley  2020      15601.0      251000   62.155378
3                 Brent  2020      28799.0      330000   87.269697
4               Bromley  2020      22013.0      335000   65.710448

   Total owed a prevention or relief duty\n  \
0                                    1434.0
1                                    2035.0
2                                     794.0
3                                    2934.0
4                                    1170.0

   Households assessed as homeless\nper (000s)  \
0                                         7.07
1                                         5.98
2                                         5.20
3                                        14.11
4                                         4.67

   homeless end prevention duty dindt get  \
0                                   500.0
1                                   429.0
2                                     NaN
3                                   426.0
4                                     NaN

   homeless with relief duty didn’t get  homeless didn’t get accomadation  \
0                                 320.0                               820
1                                 513.0                               942
2                                   NaN                               550
3                                 481.0                               907
4                                   NaN                               260
```

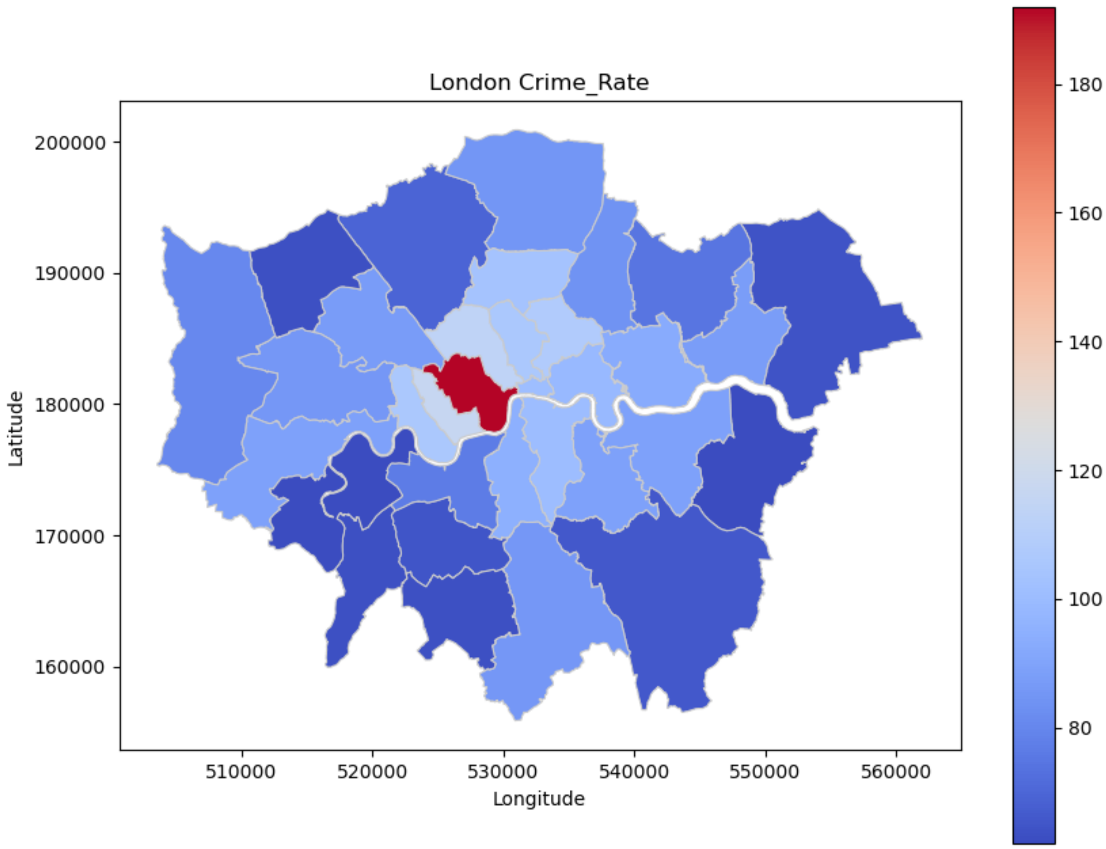
```
                                           geometry
0  MULTIPOLYGON (((543905.400 183199.100, 543905…
1  POLYGON ((524579.900 198355.200, 524594.300 19…
2  POLYGON ((547226.200 181299.300, 547320.900 18…
3  POLYGON ((525201.000 182512.600, 525181.500 18…
4  POLYGON ((540373.600 157530.400, 540361.200 15…
```

[25]:

```
#
fig, ax = plt.subplots(1, 1, figsize=(10, 8))
myData.plot(column='Total owed a prevention or relief duty\n', cmap='coolwarm',
 ↪linewidth=0.8, ax=ax, edgecolor='0.8', legend=True)
plt.title('London Total owed a prevention or relief duty\n')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
#    PNG
plt.savefig('London Total owed a prevention or relief duty\n', dpi=300,
 ↪bbox_inches='tight')
plt.show()
```

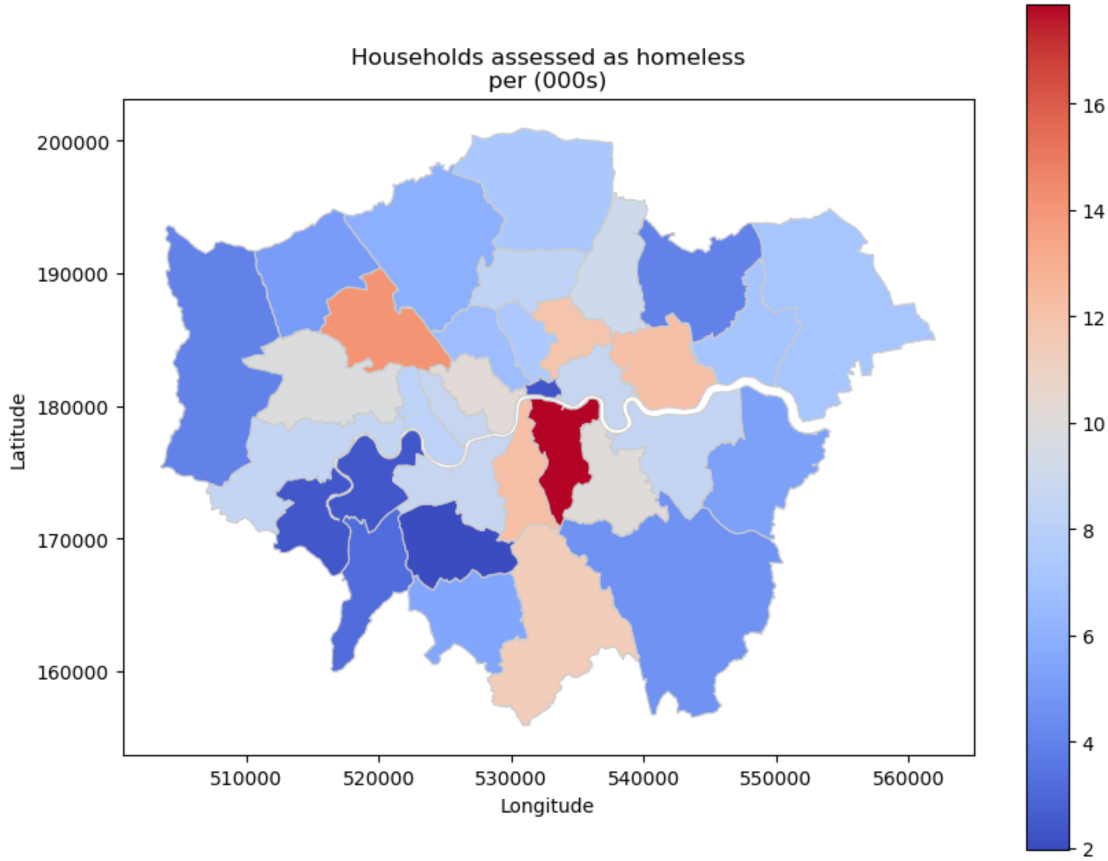


[24]:
```
#
fig, ax = plt.subplots(1, 1, figsize=(10, 8))
myData.plot(column='Crime_Rate', cmap='coolwarm', linewidth=0.8, ax=ax,␣
↪edgecolor='0.8', legend=True)
plt.title('London Crime_Rate')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
#     PNG
plt.savefig('London Crime_Rate', dpi=300, bbox_inches='tight')
plt.show()
```



[23]:
```
#
fig, ax = plt.subplots(1, 1, figsize=(10, 8))
myData.plot(column='Households assessed as homeless\nper (000s)',␣
↪cmap='coolwarm', linewidth=0.8, ax=ax, edgecolor='0.8', legend=True)
plt.title('Households assessed as homeless\nper (000s)')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
#     PNG
```

```
plt.savefig('Households assessed as homeless\nper (000s)', dpi=300,␣
↪bbox_inches='tight')
plt.show()
```



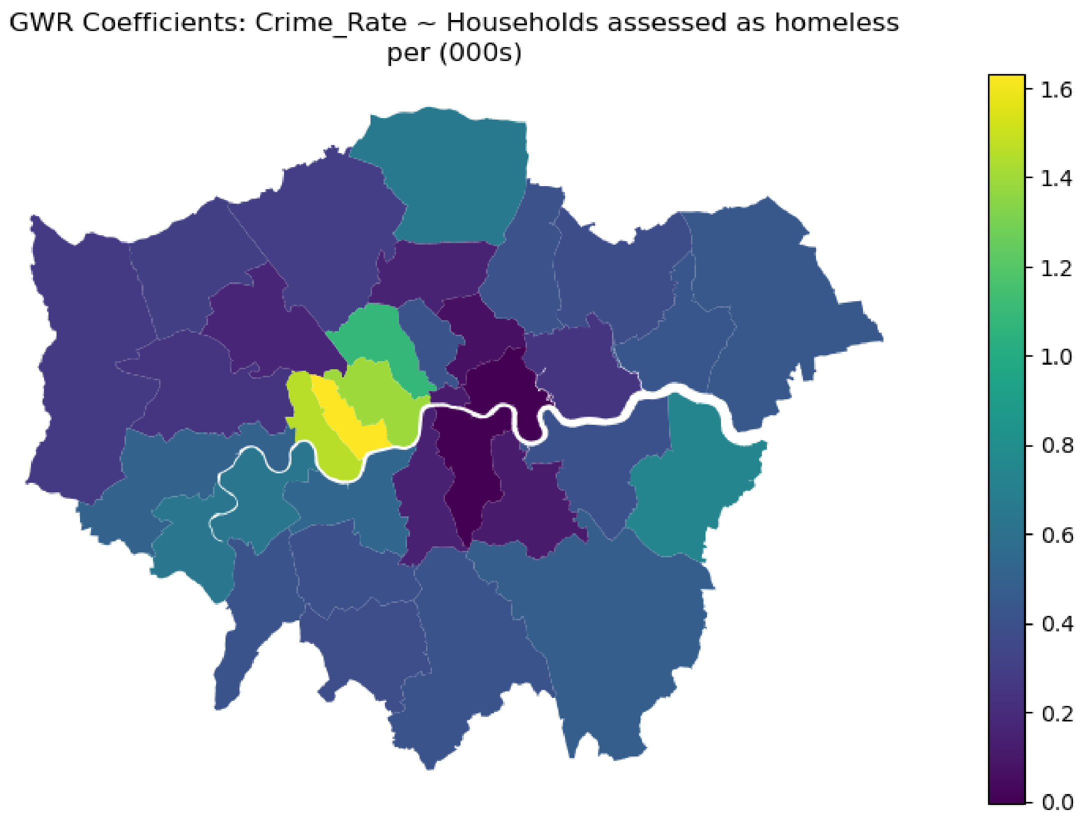
[22]:
```
import geopandas as gpd
import pandas as pd
from mgwr.gwr import GWR
from mgwr.sel_bw import Sel_BW
import geopandas as gpd
import pandas as pd
import matplotlib.pyplot as plt

def run_gwr(myData, y_column, x_column, bandwidth):
    #
    gdf = myData[[y_column, x_column, 'geometry']]
    gdf['coords'] = gdf['geometry'].apply(lambda geom: geom.centroid.coords[0])
    gdf[['X', 'Y']] = pd.DataFrame(gdf['coords'].tolist(), index=gdf.index)

    #
```

```
    y = gdf[y_column].values.reshape((-1, 1))
    y = (y - y.mean(axis=0)) / y.std(axis=0)
    X = gdf[[x_column]].values
    X = (X - X.mean(axis=0)) / X.std(axis=0)
    coords = list(zip(gdf['X'], gdf['Y']))

    #     GWR
    gwr_model = GWR(coords, y, X, bandwidth)
    gwr_results = gwr_model.fit()
    print(gwr_results.summary())

    #
    gdf['GWR_Coefficient'] = gwr_results.params[:, 1]
    fig, ax = plt.subplots(figsize=(10, 6))
    gdf.plot(column='GWR_Coefficient', legend=True, ax=ax)
    ax.set_title('GWR Coefficients: ' + y_column + ' ~ ' + x_column)
    ax.axis('off')


        #
    plt.savefig(y_column + '_' + x_column + '_GWR.png', dpi=300,␣
 ↪bbox_inches='tight')
    #
    plt.show()

#
run_gwr(myData, "Crime_Rate", "Households assessed as homeless\nper (000s)", 10)
run_gwr(myData, "Crime_Rate", "homeless didn't get accomadation", 10)
run_gwr(myData, "Crime_Rate", "Total owed a prevention or relief duty\n", 10)
```

```
/opt/conda/lib/python3.11/site-packages/geopandas/geodataframe.py:1543:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  super().__setitem__(key, value)
/opt/conda/lib/python3.11/site-packages/geopandas/geodataframe.py:1543:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  super().__setitem__(key, value)
/opt/conda/lib/python3.11/site-packages/geopandas/geodataframe.py:1543:
```

```
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  super().__setitem__(key, value)
/opt/conda/lib/python3.11/site-packages/geopandas/geodataframe.py:1543:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  super().__setitem__(key, value)
===========================================================================
Model type                                                         Gaussian
Number of observations:                                                  33
Number of covariates:                                                     2

Global Regression Results
---------------------------------------------------------------------------
Residual sum of squares:                                             27.077
Log-likelihood:                                                     -43.561
AIC:                                                                 91.122
AICc:                                                                93.950
BIC:                                                                -81.314
R2:                                                                   0.179
Adj. R2:                                                              0.153

Variable                              Est.         SE  t(Est/SE)    p-value
------------------------------- ---------- ---------- ---------- ----------
X0                                  -0.000      0.163     -0.000      1.000
X1                                   0.424      0.163      2.604      0.009

Geographically Weighted Regression (GWR) Results
---------------------------------------------------------------------------
Spatial kernel:                                           Adaptive bisquare
Bandwidth used:                                                      10.000

Diagnostic information
---------------------------------------------------------------------------
Residual sum of squares:                                              4.962
Effective number of parameters (trace(S)):                           15.245
Degree of freedom (n - trace(S)):                                    17.755
Sigma estimate:                                                       0.529
Log-likelihood:                                                     -15.564
```

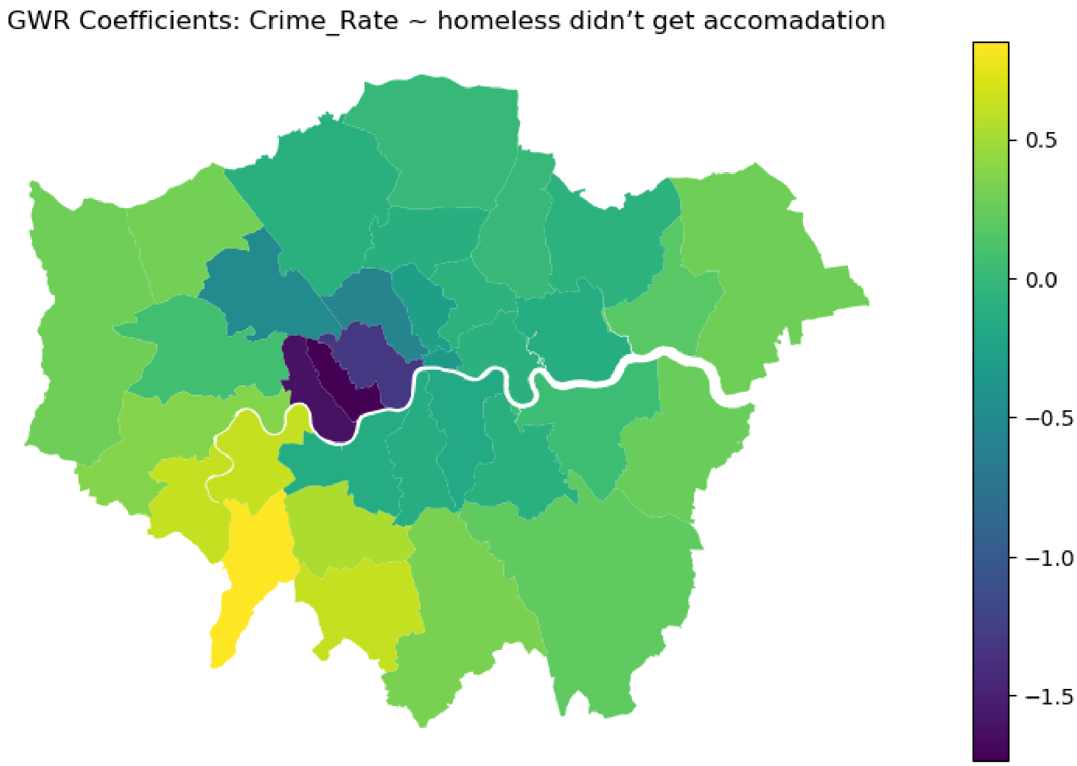
```
AIC:                                                                 63.617
AICc:                                                                99.180
BIC:                                                                 87.928
R2:                                                                   0.850
Adjusted R2:                                                          0.713
Adj. alpha (95%):                                                     0.007
Adj. critical t value (95%):                                          2.908

Summary Statistics For GWR Parameter Estimates
---------------------------------------------------------------------------
Variable                   Mean        STD         Min      Median        Max
-------------------- ---------- ---------- ---------- ---------- ----------
X0                        0.145      0.604     -0.544      0.019      1.740
X1                        0.464      0.393     -0.005      0.400      1.633
===========================================================================

None
```




```
/opt/conda/lib/python3.11/site-packages/geopandas/geodataframe.py:1543:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  super().__setitem__(key, value)
/opt/conda/lib/python3.11/site-packages/geopandas/geodataframe.py:1543:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  super().__setitem__(key, value)
/opt/conda/lib/python3.11/site-packages/geopandas/geodataframe.py:1543:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  super().__setitem__(key, value)
/opt/conda/lib/python3.11/site-packages/geopandas/geodataframe.py:1543:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  super().__setitem__(key, value)
===========================================================================
Model type                                                         Gaussian
Number of observations:                                                  33
Number of covariates:                                                     2

Global Regression Results
---------------------------------------------------------------------------
Residual sum of squares:                                             32.782
Log-likelihood:                                                     -46.716
AIC:                                                                 97.432
AICc:                                                               100.259
BIC:                                                                -75.609
R2:                                                                   0.007
Adj. R2:                                                             -0.025

Variable                              Est.         SE  t(Est/SE)    p-value
------------------------------- ---------- ---------- ---------- ----------
X0                                  -0.000      0.179     -0.000      1.000
X1                                  -0.081      0.179     -0.454      0.650
```

```
Geographically Weighted Regression (GWR) Results
---------------------------------------------------------------------------
Spatial kernel:                                           Adaptive bisquare
Bandwidth used:                                                      10.000

Diagnostic information
---------------------------------------------------------------------------
Residual sum of squares:                                              8.928
Effective number of parameters (trace(S)):                           14.614
Degree of freedom (n - trace(S)):                                    18.386
Sigma estimate:                                                       0.697
Log-likelihood:                                                     -25.254
AIC:                                                                 81.738
AICc:                                                               113.403
BIC:                                                                105.105
R2:                                                                   0.729
Adjusted R2:                                                          0.502
Adj. alpha (95%):                                                     0.007
Adj. critical t value (95%):                                          2.891

Summary Statistics For GWR Parameter Estimates
---------------------------------------------------------------------------
Variable                   Mean        STD        Min     Median        Max
-------------------- ---------- ---------- ---------- ---------- ----------
X0                        0.076      0.448     -0.640      0.072      1.181
X1                       -0.076      0.564     -1.734     -0.070      0.850
===========================================================================

None
```

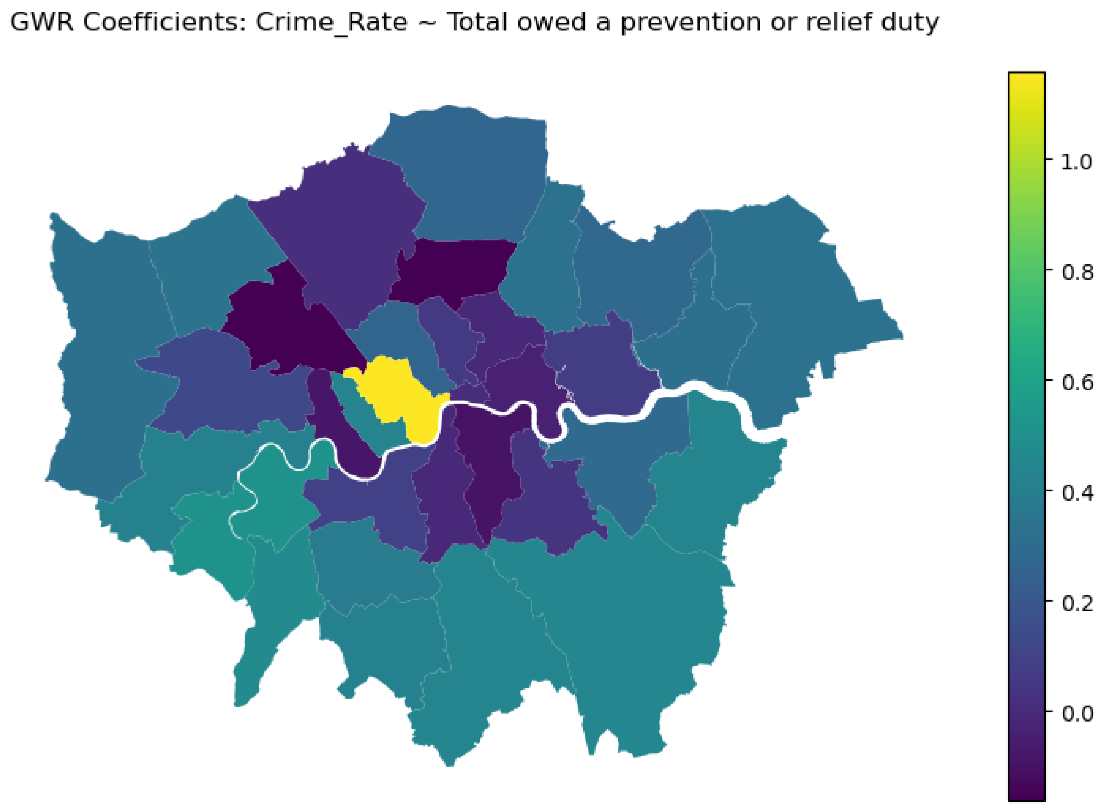



```
/opt/conda/lib/python3.11/site-packages/geopandas/geodataframe.py:1543:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  super().__setitem__(key, value)
/opt/conda/lib/python3.11/site-packages/geopandas/geodataframe.py:1543:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  super().__setitem__(key, value)
/opt/conda/lib/python3.11/site-packages/geopandas/geodataframe.py:1543:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

```
  super().__setitem__(key, value)
/opt/conda/lib/python3.11/site-packages/geopandas/geodataframe.py:1543:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  super().__setitem__(key, value)
```

```
===========================================================================
Model type                                                         Gaussian
Number of observations:                                                  33
Number of covariates:                                                     2

Global Regression Results
---------------------------------------------------------------------------
Residual sum of squares:                                             31.546
Log-likelihood:                                                     -46.081
AIC:                                                                 96.163
AICc:                                                                98.990
BIC:                                                                -76.846
R2:                                                                   0.044
Adj. R2:                                                              0.013

Variable                              Est.         SE  t(Est/SE)    p-value
------------------------------- ---------- ---------- ---------- ----------
X0                                  -0.000      0.176     -0.000      1.000
X1                                   0.210      0.176      1.195      0.232

Geographically Weighted Regression (GWR) Results
---------------------------------------------------------------------------
Spatial kernel:                                           Adaptive bisquare
Bandwidth used:                                                      10.000

Diagnostic information
---------------------------------------------------------------------------
Residual sum of squares:                                              6.851
Effective number of parameters (trace(S)):                           14.752
Degree of freedom (n - trace(S)):                                    18.248
Sigma estimate:                                                       0.613
Log-likelihood:                                                     -20.885
AIC:                                                                 73.273
AICc:                                                               105.751
BIC:                                                                 96.845
R2:                                                                   0.792
Adjusted R2:                                                          0.615
Adj. alpha (95%):                                                     0.007
```

```
Adj. critical t value (95%):                                          2.895

Summary Statistics For GWR Parameter Estimates
---------------------------------------------------------------------------
Variable                    Mean        STD         Min      Median        Max
-------------------- ---------- ---------- ---------- ---------- ----------
X0                         0.206      0.728     -0.541     -0.030      2.444
X1                         0.233      0.261     -0.163      0.277      1.158
===========================================================================

None
```



[19]:
```
df_crime_2020
```

[19]:

	Borough	Year	Crime_Count	Population	Crime_Rate
0	Barking and Dagenham	2020	19187.0	219000	87.611872
1	Barnet	2020	27388.0	396000	69.161616
2	Bexley	2020	15601.0	251000	62.155378
3	Brent	2020	28799.0	330000	87.269697
4	Bromley	2020	22013.0	335000	65.710448
5	Camden	2020	29574.0	260000	113.746154
6	Croydon	2020	33313.0	390000	85.417949

```
7                    Ealing  2020   29361.0   344000   85.351744
8                   Enfield  2020   28862.0   340000   84.888235
9                 Greenwich  2020   25752.0   289000   89.107266
10                  Hackney  2020   30718.0   285000  107.782456
11   Hammersmith and Fulham  2020   19018.0   180000  105.655556
12                 Haringey  2020   28855.0   279000  103.422939
13                   Harrow  2020   15861.0   250000      63.444
14                 Havering  2020   16831.0   261000    64.48659
15               Hillingdon  2020   24954.0   310000   80.496774
16                 Hounslow  2020   24314.0   273000   89.062271
17                Islington  2020   25456.0   240000  106.066667
18   Kensington and Chelsea  2020   17935.0   154000  116.461039
19     Kingston upon Thames  2020   11354.0   179000   63.430168
20                  Lambeth  2020   31014.0   328000   94.554878
21                 Lewisham  2020   27578.0   309000   89.249191
22                   Merton  2020   13588.0   209000   65.014354
23                   Newham  2020   33219.0   358000   92.790503
24                Redbridge  2020   23146.0   311000   74.424437
25     Richmond upon Thames  2020   12275.0   198000   61.994949
26                Southwark  2020   32246.0   321000  100.454829
27                   Sutton  2020   13059.0   206000   63.393204
28            Tower Hamlets  2020   31614.0   320000    98.79375
29          Waltham Forest  2020   23559.0   280000   84.139286
30               Wandsworth  2020   24918.0   325000   76.670769
31              Westminster  2020   48558.0   253000  191.928854
32           City of London  2020       908     9000       100.9
```

[20]:
```python
crime1.columns
```

[20]:
```
Index(['MajorText', 'MinorText', 'BoroughName', '2010', '2011', '2012', '2013',
       '2014', '2015', '2016', '2017', '2018', '2019', '2020', '2021'],
      dtype='object')
```

[14]:
```python
#  2020
col=['MajorText', 'MinorText', 'BoroughName', '2020']
crime2 = crime1[col]
crime2.head()
```

[14]:
```
                   MajorText                          MinorText  \
0  Arson and Criminal Damage                              Arson
1  Arson and Criminal Damage                    Criminal Damage
2                   Burglary  Burglary Business and Community
3                   Burglary                  Domestic Burglary
4              Drug Offences                   Drug Trafficking

            BoroughName  2020
0  Barking and Dagenham    48
```

```
1  Barking and Dagenham  1276
2  Barking and Dagenham   282
3  Barking and Dagenham   935
4  Barking and Dagenham   163
```

[15]:
```
crime20=crime2.groupby(['BoroughName','MajorText']).sum().reset_index()
crime20.columns
```

[15]:
```
Index(['BoroughName', 'MajorText', 'MinorText', '2020'], dtype='object')
```

[16]:
```
#  pivot  'MajorText'    '2020'
crime2020 = crime20.pivot(index='BoroughName', columns='MajorText',␣
↪values='2020').reset_index()

#  NaN 0  NaN
crime2020 = crime2020.fillna(0)
#   drop()
crime2020 = crime2020.drop(columns=['Historical Fraud and Forgery'])
crime2020.rename(columns={'BoroughName': 'Borough'}, inplace=True)
crime2020
```

[16]:
```
MajorText                                   Borough  \
0                              Barking and Dagenham
1                                            Barnet
2                                            Bexley
3                                             Brent
4                                           Bromley
5                                            Camden
6                                           Croydon
7                                            Ealing
8                                           Enfield
9                                         Greenwich
10                                          Hackney
11                           Hammersmith and Fulham
12                                         Haringey
13                                           Harrow
14                                         Havering
15                                       Hillingdon
16                                         Hounslow
17                                        Islington
18                           Kensington and Chelsea
19                             Kingston upon Thames
20                                          Lambeth
21                                         Lewisham
22          London Heathrow and London City Airports
23                                           Merton
24                                           Newham
```

```
25                                        Redbridge
26                             Richmond upon Thames
27                                        Southwark
28                                           Sutton
29                                    Tower Hamlets
30                                   Waltham Forest
31                                       Wandsworth
32                                      Westminster

MajorText  Arson and Criminal Damage  Burglary  Drug Offences  \
0                               1324      1217           1682
1                               1918      2816           1214
2                               1430      1028            864
3                               2060      2127           2291
4                               1728      1746           1290
5                               1483      2281           2044
6                               2440      2343           2831
7                               2054      2199           2320
8                               1834      2455           1567
9                               1986      1674           2025
10                              1758      2583           1689
11                              1136      1470           1220
12                              1784      2058           2016
13                               976      1361           1114
14                              1146      1299           1193
15                              1886      1906           1756
16                              1673      1761           1728
17                              1454      2110           1580
18                               869      1621           1149
19                               738       866           1167
20                              2003      2581           2529
21                              2088      2447           1613
22                                46         7             46
23                              1018       983           1022
24                              1982      1883           3018
25                              1416      1729           1786
26                               916      1299            626
27                              1967      2546           2411
28                               949      1027            787
29                              1991      2714           3061
30                              1579      1850           1833
31                              1556      2307           1583
32                              2023      3046           3442

MajorText  Miscellaneous Crimes Against Society  Possession of Weapons  \
0                                           367                    202
1                                           360                    173
```

2	248	124
3	421	262
4	365	186
5	277	172
6	529	381
7	516	236
8	451	239
9	459	263
10	343	334
11	250	137
12	396	277
13	256	117
14	286	143
15	458	200
16	457	203
17	295	194
18	177	143
19	158	84
20	498	386
21	361	294
22	444	16
23	239	118
24	482	344
25	319	181
26	169	51
27	367	344
28	213	116
29	470	274
30	349	226
31	360	182
32	440	321

MajorText	Public Order Offences	Robbery	Sexual Offences	Theft	\
0	1179	738	593	3148	
1	1757	919	547	5080	
2	1269	309	372	2253	
3	2068	984	599	4708	
4	1612	496	497	3994	
5	1886	1204	568	10239	
6	2272	963	1007	4661	
7	2161	875	712	4842	
8	1819	1106	686	4290	
9	1991	757	685	4312	
10	2137	1617	708	8487	
11	1478	550	428	5030	
12	1771	1570	736	5655	
13	1029	515	392	2398	

14	1152	408	440	3155
15	1828	535	501	4275
16	1643	596	515	4182
17	1935	1185	495	7508
18	1129	588	371	5363
19	797	230	416	2367
20	2364	1128	1003	6663
21	2115	962	724	4659
22	73	2	22	687
23	1005	390	316	2295
24	1925	1641	814	7940
25	1308	677	514	4408
26	842	235	281	2531
27	2111	1541	761	8417
28	923	247	329	2050
29	2206	998	780	7073
30	1390	734	504	4697
31	1716	784	623	5278
32	3001	2150	917	20490

MajorText	Vehicle Offences	Violence Against the Person
0	2343	6394
1	5155	7449
2	2393	5311
3	4371	8908
4	3679	6420
5	3033	6387
6	5050	10836
7	4611	8835
8	5854	8561
9	3232	8368
10	3202	7860
11	2316	5003
12	5044	7548
13	3017	4686
14	2481	5128
15	4085	7524
16	3781	7775
17	2626	6074
18	2576	3949
19	1141	3390
20	3019	8840
21	3923	8392
22	101	127
23	2046	4156
24	3982	9208
25	4193	6615

```
26                       2304                          3021
27                       3438                          8343
28                       2271                          4147
29                       3508                          8539
30                       3779                          6618
31                       3624                          6905
32                       3822                          8906
```

[17]:
```python
#     borough
merge = pd.merge(Data, crime2020, on=['Borough'],how='inner')


merge.head()
```

[17]:
```
                Borough  Year  Crime_Count  Population  Crime_Rate  \
0  Barking and Dagenham  2020      19187.0      219000   87.611872
1                Barnet  2020      27388.0      396000   69.161616
2                Bexley  2020      15601.0      251000   62.155378
3                 Brent  2020      28799.0      330000   87.269697
4               Bromley  2020      22013.0      335000   65.710448

   Total owed a prevention or relief duty\n  \
0                                    1434.0
1                                    2035.0
2                                     794.0
3                                    2934.0
4                                    1170.0

   Households assessed as homeless\nper (000s)  \
0                                         7.07
1                                         5.98
2                                         5.20
3                                        14.11
4                                         4.67

   homeless end prevention duty dindt get  \
0                                   500.0
1                                   429.0
2                                     NaN
3                                   426.0
4                                     NaN

   homeless with relief duty didn't get  homeless didn't get accomadation  \
0                                 320.0                               820
1                                 513.0                               942
2                                   NaN                               550
3                                 481.0                               907
```

```
4                                  NaN                              260

   …  Burglary  Drug Offences  Miscellaneous Crimes Against Society  \
0  …      1217           1682                                   367
1  …      2816           1214                                   360
2  …      1028            864                                   248
3  …      2127           2291                                   421
4  …      1746           1290                                   365

   Possession of Weapons  Public Order Offences  Robbery  Sexual Offences  \
0                    202                   1179      738              593
1                    173                   1757      919              547
2                    124                   1269      309              372
3                    262                   2068      984              599
4                    186                   1612      496              497

   Theft  Vehicle Offences  Violence Against the Person
0   3148              2343                         6394
1   5080              5155                         7449
2   2253              2393                         5311
3   4708              4371                         8908
4   3994              3679                         6420

[5 rows x 21 columns]
```

[25]:
```python
merge.columns
```

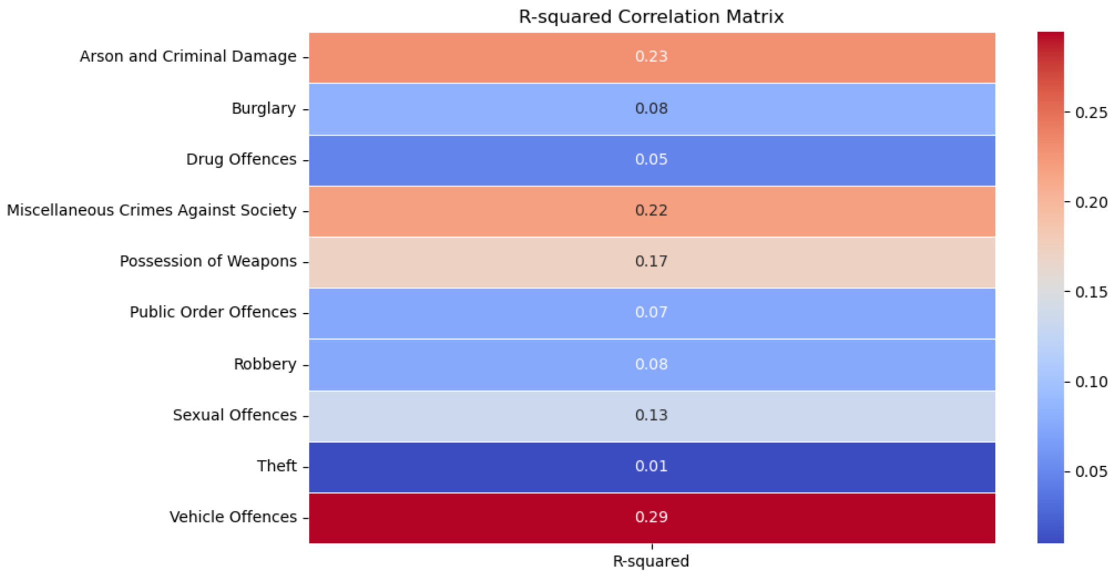
[25]:
```
Index(['Borough', 'Year', 'Crime_Count', 'Population', 'Crime_Rate',
       'Total owed a prevention or relief duty\n',
       'Households assessed as homeless\nper (000s)',
       'homeless end prevention duty dindt get',
       'homeless with relief duty didn’t get',
       'homeless didn’t get accomadation', 'Arson and Criminal Damage',
       'Burglary', 'Drug Offences', 'Miscellaneous Crimes Against Society',
       'Possession of Weapons', 'Public Order Offences', 'Robbery',
       'Sexual Offences', 'Theft', 'Vehicle Offences',
       'Violence Against the Person'],
      dtype='object')
```

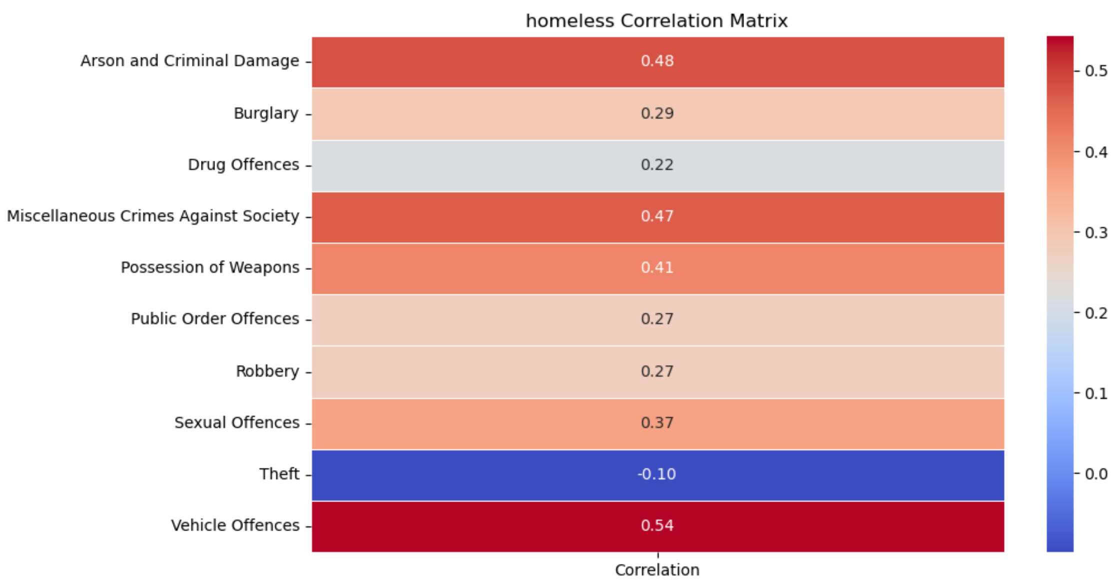
[18]:
```python
import seaborn as sns

#
homeless_crime_r_squared = {}
homeless_crime_correlations = {}
#
#       'Homeless_Count'
#
```

```
for crime_type in merge.columns[10:20]:
    correlation, _ = stats.pearsonr(merge['homeless didn’t get accomadation'], merge[crime_type])
    homeless_crime_correlations[crime_type] = correlation
    homeless_crime_r_squared[crime_type] = correlation**2  #     R-squared

#
max_correlation_crime1 = max(homeless_crime_correlations, key=homeless_crime_correlations.get)

#
print("        ", homeless_crime_correlations)
print("         ", max_correlation_crime1)

#
correlation_matrix = pd.DataFrame.from_dict(homeless_crime_correlations, orient='index', columns=['Correlation'])
#  R-squared
r_squared_matrix = pd.DataFrame.from_dict(homeless_crime_r_squared, orient='index', columns=['R-squared'])

#  R-squared
plt.figure(figsize=(10, 6))
sns.heatmap(r_squared_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('R-squared Correlation Matrix')
plt.savefig('homeless_crime R-squared Correlation Matrix.png', dpi=300, bbox_inches='tight')
plt.show()
#


#
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('homeless Correlation Matrix')
plt.savefig('homeless_crime Correlation Matrix.png', dpi=300, bbox_inches='tight')
plt.show()
```

```
        {'Arson and Criminal Damage': 0.4790147793225451, 'Burglary':
0.2875324510151715, 'Drug Offences': 0.21711377432304768, 'Miscellaneous Crimes
Against Society': 0.4671568104660664, 'Possession of Weapons':
0.4143161686250144, 'Public Order Offences': 0.2724544067850863, 'Robbery':
0.27477608099119566, 'Sexual Offences': 0.3655813704617722, 'Theft':
-0.0982104331425581, 'Vehicle Offences': 0.5424730534597486}
```

```
Vehicle Offences
```





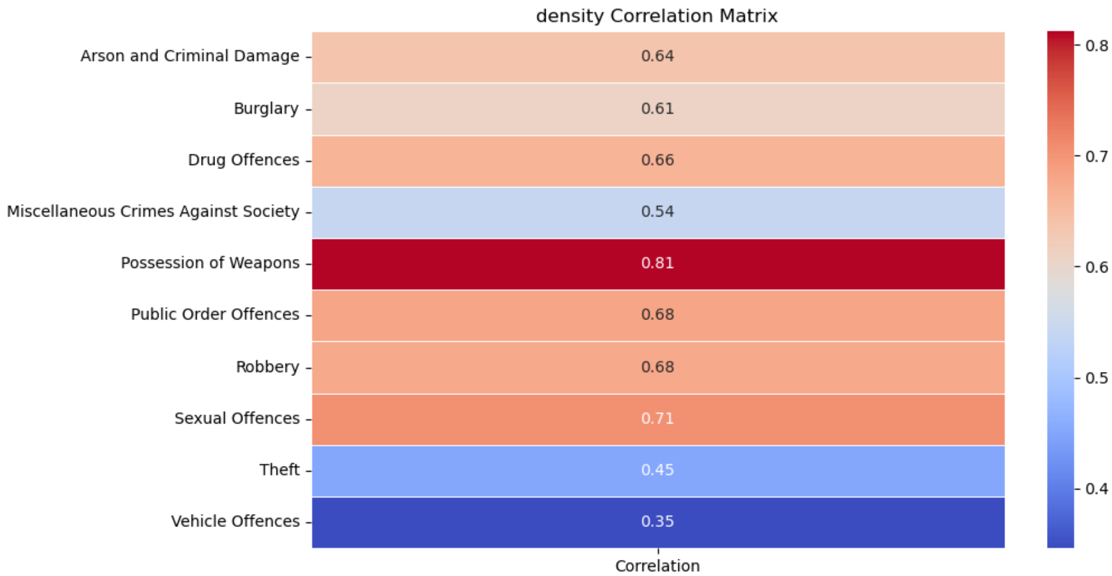
[20]:
```
density_crime_correlations = {}
density_crime_r_squared = {}  #   R-squared
#
#        'Homeless_Count'
#
for crime_type in merge.columns[10:20]:
```

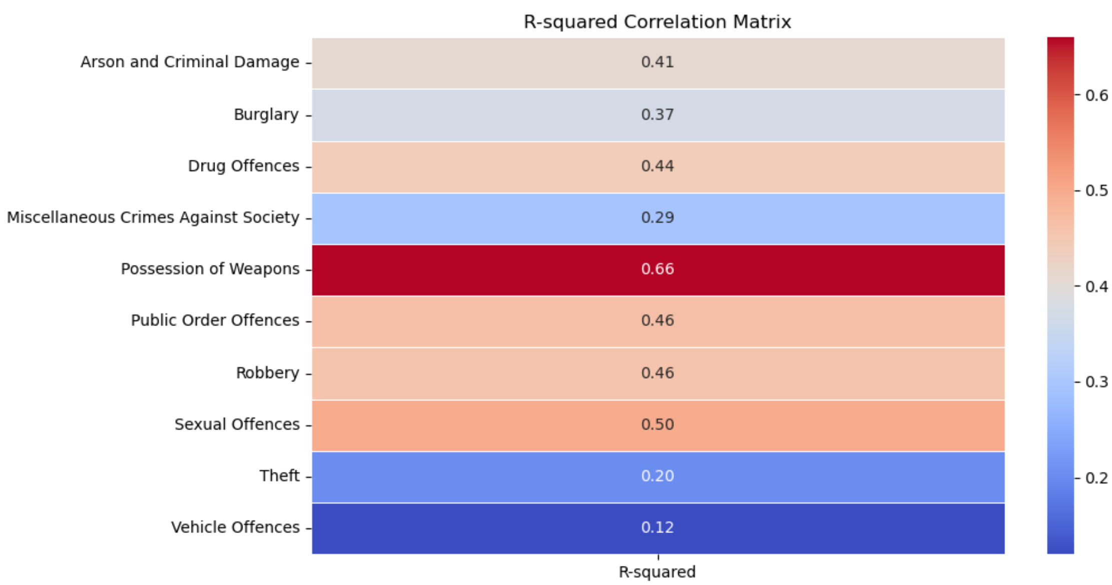
```
    correlation, _ = stats.pearsonr(merge['Households assessed as homeless\nper (000s)'], merge[crime_type])
    density_crime_correlations[crime_type] = correlation
    density_crime_r_squared[crime_type] = correlation**2  #      R-squared

#
max_correlation_crime2 = max(density_crime_correlations, key=density_crime_correlations.get)

#
print("        ", density_crime_correlations)
print("         ", max_correlation_crime2)

#
correlation_matrix = pd.DataFrame.from_dict(density_crime_correlations, orient='index', columns=['Correlation'])

#
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('density Correlation Matrix')
plt.savefig('density_crime Correlation Matrix.png', dpi=300, bbox_inches='tight')
plt.show()


#  R-squared
r_squared_matrix = pd.DataFrame.from_dict(density_crime_r_squared, orient='index', columns=['R-squared'])

#  R-squared
plt.figure(figsize=(10, 6))
sns.heatmap(r_squared_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('R-squared Correlation Matrix')
plt.savefig('density_crime R-squared Correlation Matrix.png', dpi=300, bbox_inches='tight')
plt.show()



#

#R² < 0.3
#0.3  R² < 0.5
```

```
#R²   0.5
```

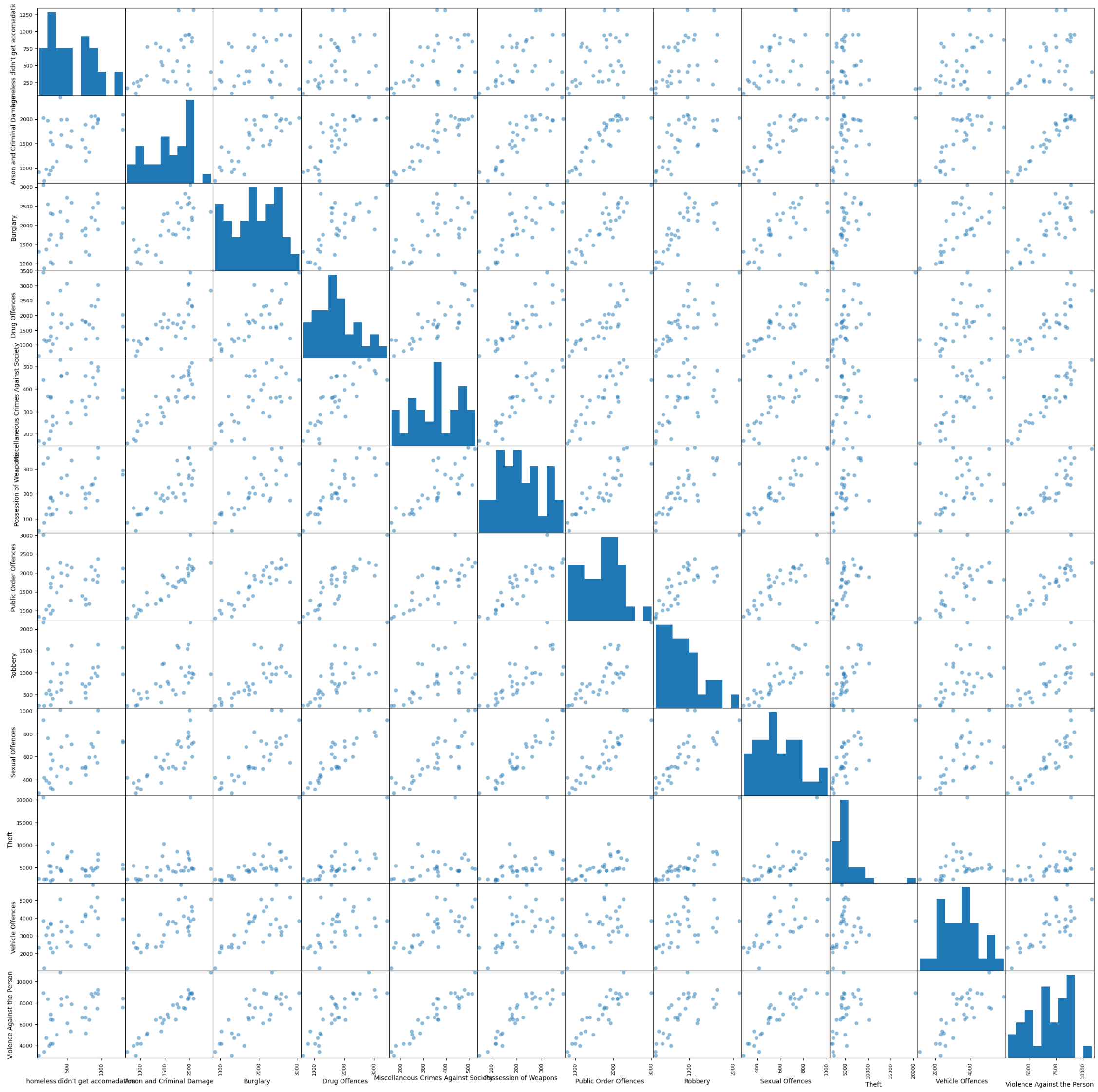
{'Arson and Criminal Damage': 0.6389059868389364, 'Burglary': 0.6095431127761631, 'Drug Offences': 0.6623713828380153, 'Miscellaneous Crimes Against Society': 0.5414508411908654, 'Possession of Weapons': 0.8119059407551821, 'Public Order Offences': 0.6812083525865842, 'Robbery': 0.6760498659123422, 'Sexual Offences': 0.7061701504069515, 'Theft': 0.4509598399119232, 'Vehicle Offences': 0.3459660315872681}

Possession of Weapons





[21]:
```
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
#
crime_df = merge[['homeless didn’t get accomadation', 'Arson and Criminal␣
 ↪Damage',
       'Burglary', 'Drug Offences', 'Miscellaneous Crimes Against Society',
       'Possession of Weapons', 'Public Order Offences', 'Robbery',
       'Sexual Offences', 'Theft', 'Vehicle Offences',
       'Violence Against the Person']]

#   scatter_matrix
scatter_matrix(crime_df, alpha=0.5, figsize=(30, 30), diagonal='hist',␣
 ↪marker='o')
plt.savefig('scatter_matrix.png', dpi=300, bbox_inches='tight')
plt.show()
```


homeless didn't get accomadation
Arson and Criminal Damage
Burglary
Drug Offences
Miscellaneous Crimes Against Society
Possession of Weapons
Public Order Offences
Robbery
Sexual Offences
Theft
Vehicle Offences
Violence Against the Person