Data Exploration - Crime and Geography

Continuing from previous exploration of crime and geography, look at how the crime is distributed across London

In [214…

```
# !pip install selenium   #not required from attempt to programatically export folium maps. Ran into known windows 10 issues, and decided for this exercise little value in proceeding.
try:
    print("Importing libraries...\n")
    from progressbar import ProgressBar
    from bs4 import BeautifulSoup as bts # library for web scraping
    import numpy as np # library to handle data in a vectorized manner
    import pandas as pd # library for data analysis
    from pandas.io.json import json_normalize
    import matplotlib.cm as cm
    import matplotlib.colors as colors
    import requests # library to handle requests
    from geopy.geocoders import Nominatim # convert an address into latitude and longitude values
    import matplotlib as mp # library for visualization
    from sklearn.cluster import KMeans # import k-means from clustering stage
    from geopy.geocoders import Nominatim # convert an address into latitude and longitude values
    import folium # map rendering library
    import lxml
    import re
    from time import sleep

    from matplotlib import pyplot as plt
    from matplotlib.pyplot import figure

    import datetime
    import dateutil


    import io
    from PIL import Image
    from selenium import webdriver

    print("All libraries imported successfully!\n")
except:
    print("ERROR: Could not import all libraries!\n")

%matplotlib inline
```

```
Importing libraries...

All libraries imported successfully!
```

For mapping visualisations get central coordinates of London

In [293…

```
address = 'London'

geolocator = Nominatim(user_agent="ldn_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of London are {}, {}.'.format(latitude, longitude))
```

```
The geograpical coordinate of London are 51.5073219, -0.1276474.
```

Borough Level Data

Retrieve the borough level data, as concuclded in previous notebook aggregate over 24 month period

In [216…

```
#get crime data
boroughCrime = pd.read_csv('https://data.london.gov.uk/download/recorded_crime_summary/d2e9ccfc-a054-41e3-89fb-53c2bc3ed87a/MPS%20Borough%20Level%20Crime%20%28most%20recent%2024%20months%29.csv
boroughCrime.rename(columns={'LookUp_BoroughName':'Borough'}, inplace=True)

#create a list of month columns by working back from current data - note the range here may need tweaking  (ie to 23,-1,01) depending on when dataset last updated
date_list=[datetime.date.today()- dateutil.relativedelta.relativedelta(months = x) for x in range(24,0,-1)]
month_list=[datetime.date.strftime(x,'%Y%m') for x in date_list]
boroughCrime['Total'] = 0

#for each month add to total then drop column
for month in month_list:
    boroughCrime['Total'] = boroughCrime['Total']+boroughCrime[month]
    boroughCrime=boroughCrime.drop(month,1)

CrimeByBoroughMinor = pd.pivot_table(boroughCrime, values='Total', index = ['Borough'], columns='MinorText',aggfunc=np.sum).reset_index()
CrimeByBoroughMajor = pd.pivot_table(boroughCrime, values='Total', index = ['Borough'], columns='MajorText',aggfunc=np.sum).reset_index()
```

In [217…

```
CrimeByBoroughMinor.head()
```

Out[217…

MinorText	Borough	Absconding from Lawful Custody	Aggravated Vehicle Taking	Aiding Suicide	Arson	Bail Offences	Bicycle Theft	Bigamy	Burglary - Business and Community	Burglary - Residential	...	Shoplifting	Soliciting for Prostitution	Theft from Person	Theft from a Motor Vehicle	Theft or Taking of a Motor Vehicle	Threat or Possession With Intent to Commit Crimina	Violence with Injury	Violence without Injury	Violent Disorder	Wildlife Crime
0	Barking and Dagenham	NaN	62.0	NaN	114.0	1.0	350.0	3.0	593.0	1979.0	...	1628.0	NaN	1014.0	2145.0	2140.0	141.0	3988.0	8897.0	5.0	2.0
1	Barnet	NaN	74.0	NaN	114.0	NaN	549.0	NaN	1155.0	4742.0	...	3081.0	NaN	1239.0	7167.0	2644.0	158.0	4472.0	10602.0	4.0	NaN
2	Bexley	NaN	61.0	NaN	123.0	NaN	185.0	NaN	518.0	1734.0	...	1551.0	NaN	279.0	2993.0	1467.0	171.0	3402.0	7120.0	2.0	1.0
3	Brent	7.0	81.0	NaN	134.0	1.0	726.0	5.0	868.0	3527.0	...	2066.0	NaN	1467.0	4931.0	2281.0	169.0	5613.0	12085.0	9.0	NaN
4	Bromley	1.0	62.0	NaN	207.0	NaN	302.0	NaN	892.0	3027.0	...	3771.0	NaN	694.0	4541.0	1925.0	205.0	4108.0	8825.0	2.0	NaN

5 rows × 55 columns

In [218…

```
CrimeByBoroughMajor.head()
```

Out[218…

MajorText	Borough	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
0	Barking and Dagenham	2702	2572	2974	662	332	2408	1569	1281	6539	4870	12888
1	Barnet	3859	5897	2153	743	304	3756	1966	1160	10722	11304	15081
2	Bexley	2958	2252	1656	497	232	2479	609	766	4664	5310	10527
3	Brent	4078	4395	4228	779	521	4097	2074	1226	9920	8381	17718

MajorText	Borough	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
4	Bromley	3705	3919	2436	677	325	3291	925	1029	8697	7771	12936

Clean the minor crime data

- set NAN to 0
- remove crimes from the data where more than 10 boroughs have had no occurance - this is has there is insufficient data to draw conclusions

In [219…
```
#check how many null values in each column
CrimeByBoroughMinor.isnull().sum()
```

Out[219…
```
MinorText
Borough                                            0
Absconding from Lawful Custody                    10
Aggravated Vehicle Taking                          1
Aiding Suicide                                    31
Arson                                              0
Bail Offences                                     10
Bicycle Theft                                      0
Bigamy                                            18
Burglary - Business and Community                  0
Burglary - Residential                             1
Criminal Damage                                    0
Dangerous Driving                                  0
Disclosure, Obstruction, False or Misleading State 3
Drug Trafficking                                   0
Exploitation of Prostitution                      12
Forgery or Use of Drug Prescription                5
Fraud or Forgery Associated with Driver Records    2
Going Equipped for Stealing                        0
Handling Stolen Goods                              0
Homicide                                           1
Interfering with a Motor Vehicle                   0
Making, Supplying or Possessing Articles for use i 0
Obscene Publications                               0
Offender Management Act                           17
Other Firearm Offences                             5
Other Forgery                                      0
Other Knife Offences                              31
Other Notifiable Offences                          0
Other Offences Against the State, or Public Order  0
Other Sexual Offences                              0
Other Theft                                        0
Perjury                                           21
Perverting Course of Justice                       0
Possession of Article with Blade or Point          0
Possession of Drugs                                0
Possession of False Documents                      0
Possession of Firearm with Intent                  1
Possession of Firearms Offences                    0
Possession of Other Weapon                         0
Profitting From or Concealing Proceeds of Crime    1
Public Fear Alarm or Distress                      0
Racially or Religiously Aggravated Public Fear, Al 0
Rape                                               0
Robbery of Business Property                       1
Robbery of Personal Property                       0
Shoplifting                                        0
Soliciting for Prostitution                       27
Theft from Person                                  0
Theft from a Motor Vehicle                         0
Theft or Taking of a Motor Vehicle                 0
```

```
Threat or Possession With Intent to Commit Crimina     0
Violence with Injury                                   0
Violence without Injury                                0
Violent Disorder                                       4
Wildlife Crime                                        28
dtype: int64
```

In [220…

```
#remove columns with 10 or more boroughs with NaN data
CrimeByBoroughMinor=CrimeByBoroughMinor[CrimeByBoroughMinor.columns[CrimeByBoroughMinor.isnull().sum()<=10]]
#replace NaN with zero as that is what it represents
CrimeByBoroughMinor=CrimeByBoroughMinor.fillna(0)
```

Remove additional spaces from the borough name so it can be used for look ups, add a total crime column to show all crimes across the borough

In [221…

```
CrimeByBoroughMinor['Borough'] = CrimeByBoroughMinor['Borough'].str.strip()
CrimeByBoroughMajor['Borough'] = CrimeByBoroughMinor['Borough'].str.strip()

CrimeByBoroughMinor.set_index(['Borough'], inplace=True)
CrimeByBoroughMajor.set_index(['Borough'], inplace=True)

CrimeByBoroughMajor['Total Crime'] = CrimeByBoroughMajor.sum(axis=1)
CrimeByBoroughMinor['Total Crime'] = CrimeByBoroughMinor.sum(axis=1)
```

In [222…

```
CrimeByBoroughMinor.head()
```

Out[222…

MinorText	Absconding from Lawful Custody	Aggravated Vehicle Taking	Arson	Bail Offences	Bicycle Theft	Burglary - Business and Community	Burglary - Residential	Criminal Damage	Dangerous Driving	Disclosure, Obstruction, False or Misleading State	…	Robbery of Personal Property	Shoplifting	Theft from Person	Theft from a Motor Vehicle	Theft or Taking of a Motor Vehicle	Threat or Possession With Intent to Commit Crimina	Violence with Injury	Violence without Injury	Violent Disorder	Total Crime
Borough																					
Barking and Dagenham	0.0	62.0	114.0	1.0	350.0	593.0	1979.0	2588.0	31.0	1.0	…	1444.0	1628.0	1014.0	2145.0	2140.0	141.0	3988.0	8897.0	5.0	38790.0
Barnet	0.0	74.0	114.0	0.0	549.0	1155.0	4742.0	3745.0	23.0	2.0	…	1763.0	3081.0	1239.0	7167.0	2644.0	158.0	4472.0	10602.0	4.0	56944.0
Bexley	0.0	61.0	123.0	0.0	185.0	518.0	1734.0	2835.0	25.0	2.0	…	517.0	1551.0	279.0	2993.0	1467.0	171.0	3402.0	7120.0	2.0	31948.0
Brent	7.0	81.0	134.0	1.0	726.0	868.0	3527.0	3944.0	37.0	2.0	…	1896.0	2066.0	1467.0	4931.0	2281.0	169.0	5613.0	12085.0	9.0	57411.0
Bromley	1.0	62.0	207.0	0.0	302.0	892.0	3027.0	3498.0	33.0	0.0	…	766.0	3771.0	694.0	4541.0	1925.0	205.0	4108.0	8825.0	2.0	45711.0

5 rows × 47 columns

In [223…

```
CrimeByBoroughMajor.head()
```

Out[223…

MajorText	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	Total Crime
Borough												
Barking and Dagenham	2702	2572	2974	662	332	2408	1569	1281	6539	4870	12888	38797
Barnet	3859	5897	2153	743	304	3756	1966	1160	10722	11304	15081	56945

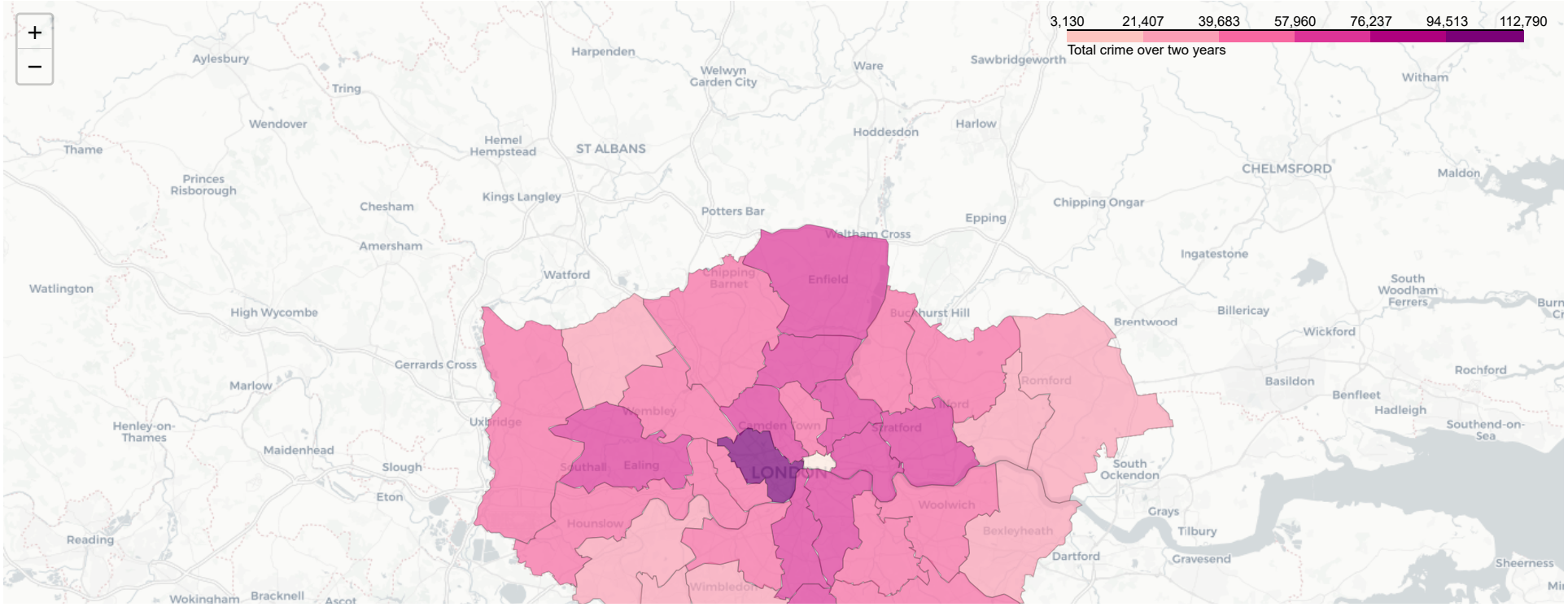
MajorText	Arson and Criminal Damage	Burglary	Drug Offences	Miscellaneous Crimes Against Society	Possession of Weapons	Public Order Offences	Robbery	Sexual Offences	Theft	Vehicle Offences	Violence Against the Person	Total Crime
Borough												
Bexley	2958	2252	1656	497	232	2479	609	766	4664	5310	10527	31950
Brent	4078	4395	4228	779	521	4097	2074	1226	9920	8381	17718	57417
Bromley	3705	3919	2436	677	325	3291	925	1029	8697	7771	12936	45711

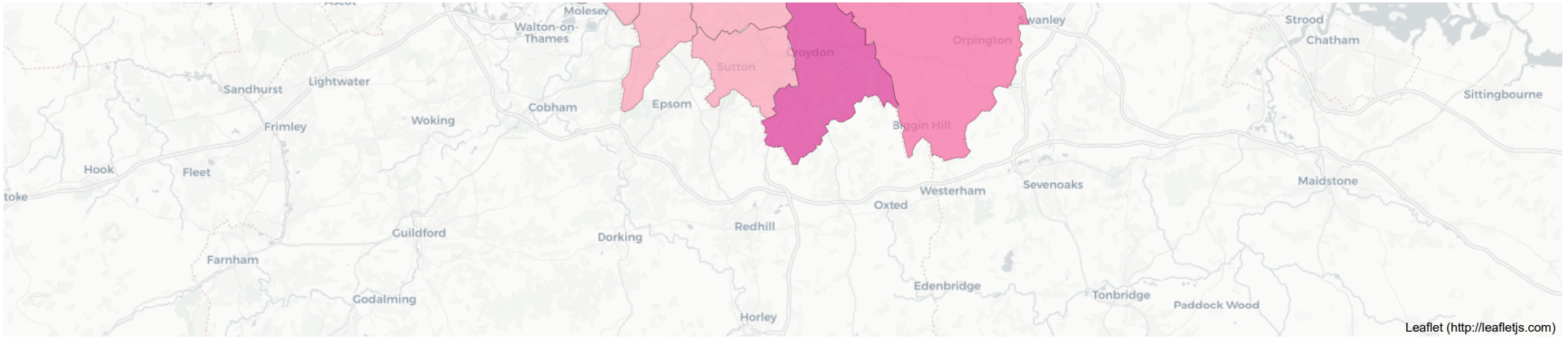
Visualise total crime accross boroughs

In [224…

```
lnd_geo = r'london_boroughs_proper.geojson'
lnd_map = folium.Map(location = [latitude, longitude], zoom_start = 10, tiles="cartodbpositron")

CrimeByBoroughMajor['Borough'] = CrimeByBoroughMajor.index

lnd_map.choropleth(
    geo_data=lnd_geo,
    data=CrimeByBoroughMajor,
    columns=['Borough','Total Crime'],
    key_on='feature.properties.name',
    fill_color='RdPu',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Total crime over two years'
)
lnd_map
```

Out[224…





In [225…

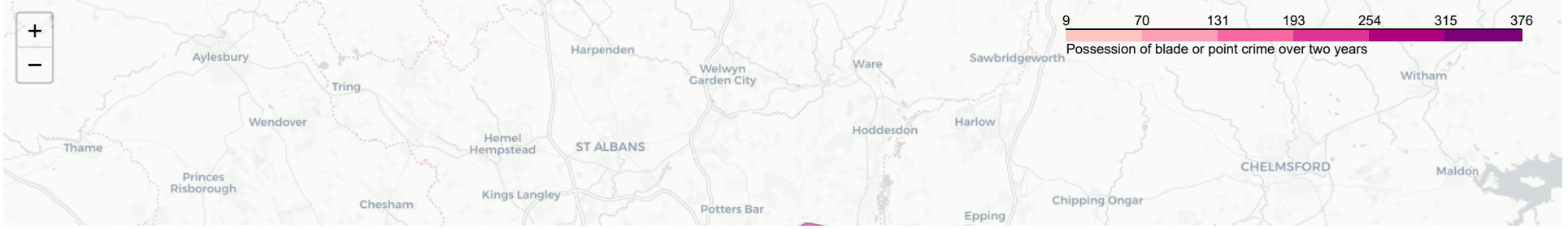
```
#save map for report
lnd_map.save('posession_blade_across_boroughs.html')
#Load the map with Selenium and save a screenshot.

#import selenium.webdriver
#driver = selenium.webdriver.PhantomJS()
#driver.set_window_size(4000, 3000)  # choose a resolution
#driver.get('total_crime_across_boroughs.html')
#time.sleep(5)
#driver.save_screenshot('posession_blade_across_boroughs.png')
```

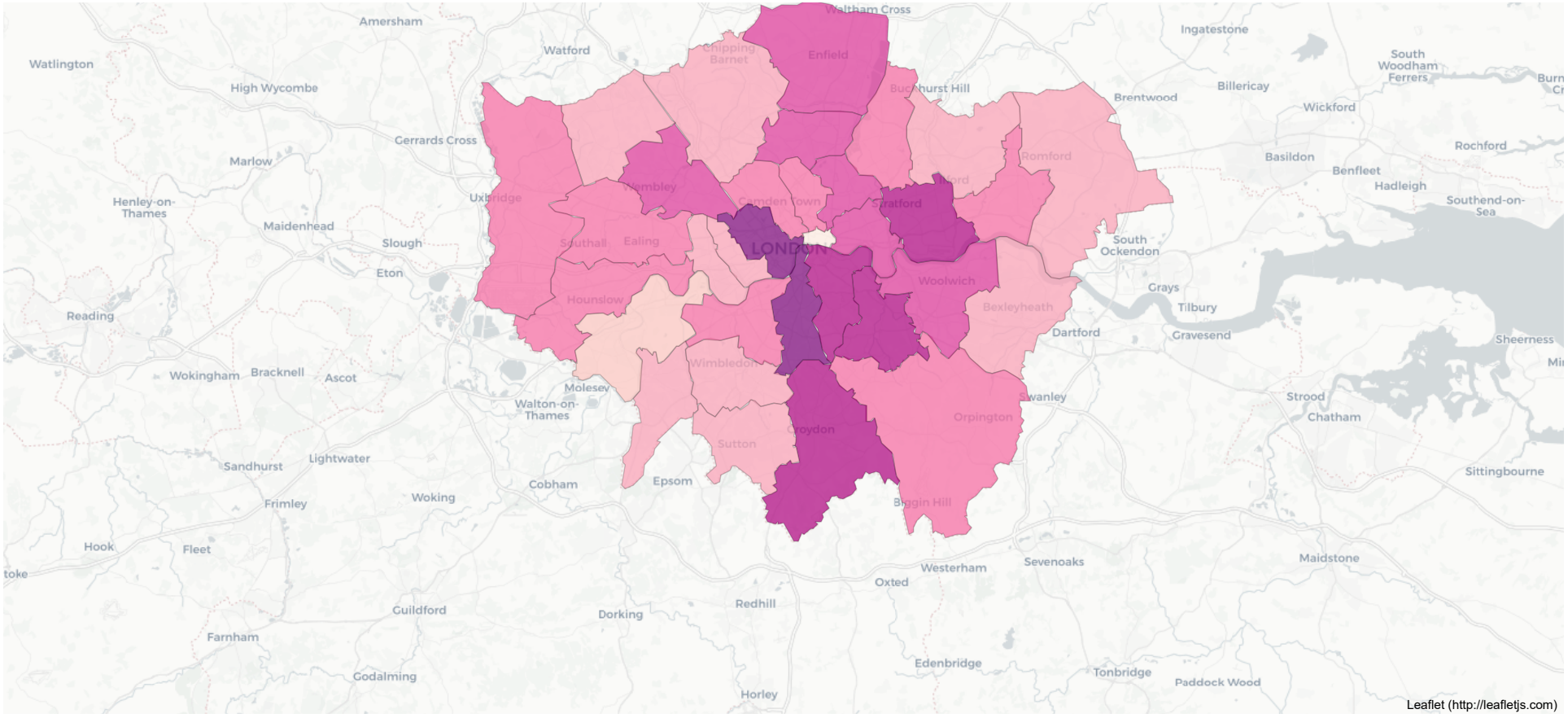
Visualise at a sub classification of crime across boroughs to see if pattern is similar

In [226…

```
lnd_geo = r'london_boroughs_proper.geojson'
lnd_map = folium.Map(location = [latitude, longitude], zoom_start = 10, tiles="cartodbpositron")

CrimeByBoroughMinor['Borough'] = CrimeByBoroughMinor.index

lnd_map.choropleth(
    geo_data=lnd_geo,
    data=CrimeByBoroughMinor,
    columns=['Borough','Possession of Article with Blade or Point'],
    key_on='feature.properties.name',
    fill_color='RdPu',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Possession  of blade or point crime over two years'
)
lnd_map
```

Out[226…





This illustrates that "Possession of Article with Blade or Point" is distributed differently to total crime, however it is not clear whether the general crime trend is because more people make more crime. To allow for this, retrieve population data.

Population data at borough level can be retrieved from the london data store. This data is a little out of data (estimate from 20017), but assuming all boroughs have had population change at the same rate, will be an appropriate proxy for up to date population data.

In [227…
```
#get population data
file ='https://data.london.gov.uk/download/london-borough-profiles/c1693b82-68b1-44ee-beb2-3decf17dc1f8/london-borough-profiles.csv'
boroughsData= pd.read_csv(file, encoding='latin1')
```

In [228…
```
boroughsData.head()
```

Out[228…

	Code	Area_name	Inner/_Outer_London	GLA_Population_Estimate_2017	GLA_Household_Estimate_2017	Inland_Area_(Hectares)	Population_density_(per_hectare)_2017	Average_Age,_2017	Proportion_of_population_aged_0-15,_2015	P
0	E09000001	City of London	Inner London	8800	5326	290	30.3	43.2	11.4	

	Code	Area_name	Inner/_Outer_London	GLA_Population_Estimate_2017	GLA_Household_Estimate_2017	Inland_Area_(Hectares)	Population_density_(per_hectare)_2017	Average_Age,_2017	Proportion_of_population_aged_0-15,_2015	P
1	E09000002	Barking and Dagenham	Outer London	209000	78188	3,611	57.9	32.9	27.2	
2	E09000003	Barnet	Outer London	389600	151423	8,675	44.9	37.3	21.1	
3	E09000004	Bexley	Outer London	244300	97736	6,058	40.3	39.0	20.6	
4	E09000005	Brent	Outer London	332100	121048	4,323	76.8	35.6	20.9	

5 rows × 84 columns

In [229…
```
#reduce to the features of intest
boroughs=pd.DataFrame(columns= ['Borough'])
boroughs['Borough']=boroughsData['Area_name']
boroughs['Population']=boroughsData['GLA_Population_Estimate_2017']
boroughs.set_index('Borough', inplace=True)
```

In [230…
```
boroughs.head()
```

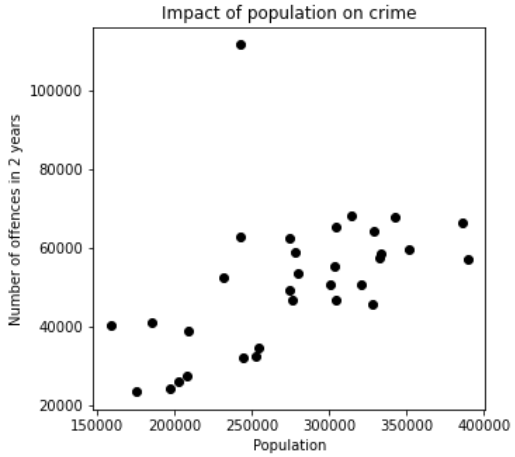
Out[230…

Borough	Population
City of London	8800
Barking and Dagenham	209000
Barnet	389600
Bexley	244300
Brent	332100

In [231…
```
#get population adjusted crime data
CrimeByBoroughMajor=CrimeByBoroughMajor.merge(boroughs, left_index= True, right_index=True)
CrimeByBoroughMinor=CrimeByBoroughMinor.merge(boroughs, left_index= True, right_index=True)
```

Plot population against total crime

In [232…
```
plt.rcParams["figure.figsize"] = (5,5)
plt.plot(CrimeByBoroughMinor['Population'],CrimeByBoroughMinor['Total Crime'],'o', color='black')

plt.title("Impact of population on crime")
plt.xlabel("Population")
plt.ylabel("Number of offences in 2 years")
plt.savefig('populationvscrime.png')
```



This indicates some correlation between crime and population, to adjust for this calculate offenses per 10,000 capita.

In [233…
```
CrimeByBoroughMajor.head()
```

Out[233…

Borough	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	Total Crime	Borough	Population
Barking and Dagenham	2702	2572	2974	662	332	2408	1569	1281	6539	4870	12888	38797	Barking and Dagenham	209000
Barnet	3859	5897	2153	743	304	3756	1966	1160	10722	11304	15081	56945	Barnet	389600
Bexley	2958	2252	1656	497	232	2479	609	766	4664	5310	10527	31950	Bexley	244300
Brent	4078	4395	4228	779	521	4097	2074	1226	9920	8381	17718	57417	Brent	332100
Bromley	3705	3919	2436	677	325	3291	925	1029	8697	7771	12936	45711	Bromley	327900

In [234…
```
CrimeByBoroughMajor.drop('Borough', axis=1, inplace=True)  #drop the Borough column which was added to replicate the index and enable the choropleth plot

adjustedCrimeByBoroughMajor = pd.DataFrame()
for col in CrimeByBoroughMajor.columns:
    CrimeByBoroughMajor[col]= pd.to_numeric(CrimeByBoroughMajor[col], downcast="float") #to conduct correlation analysis column must be numeric type
    adjustedCrimeByBoroughMajor[col] = 10000*CrimeByBoroughMajor[col]/CrimeByBoroughMajor['Population']

adjustedCrimeByBoroughMajor.drop('Population', axis=1, inplace=True)
adjustedCrimeByBoroughMajor.head()
```

Out[234…

Borough	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	Total Crime
Barking and Dagenham	129.282297	123.062201	142.296651	31.674641	15.885167	115.215311	75.071770	61.291866	312.870813	233.014354	616.650718	1856.315713
Barnet	99.050308	151.360370	55.261807	19.070842	7.802875	96.406571	50.462012	29.774127	275.205339	290.143737	387.089322	1461.627269

Borough	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	Total Crime
Bexley	121.080639	92.181744	67.785510	20.343840	9.496521	101.473598	24.928367	31.354892	190.912812	217.355710	430.904625	1307.818256
Brent	122.794339	132.339657	127.311051	23.456790	15.688046	123.366456	62.451069	36.916591	298.705209	252.363746	533.514002	1728.906908
Bromley	112.991766	119.518146	74.290942	20.646539	9.911558	100.365965	28.209820	31.381519	265.233303	236.992986	394.510522	1394.053114

In [235…

```
CrimeByBoroughMinor.head()
```

Out[235…

Borough	Absconding from Lawful Custody	Aggravated Vehicle Taking	Arson	Bail Offences	Bicycle Theft	Burglary - Business and Community	Burglary - Residential	Criminal Damage	Dangerous Driving	Disclosure, Obstruction, False or Misleading State	...	Theft from Person	Theft from a Motor Vehicle	Theft or Taking of a Motor Vehicle	Threat or Possession With Intent to Commit Crimina	Violence with Injury	Violence without Injury	Violent Disorder	Total Crime	Borough	Population
Barking and Dagenham	0.0	62.0	114.0	1.0	350.0	593.0	1979.0	2588.0	31.0	1.0	...	1014.0	2145.0	2140.0	141.0	3988.0	8897.0	5.0	38790.0	Barking and Dagenham	209000
Barnet	0.0	74.0	114.0	0.0	549.0	1155.0	4742.0	3745.0	23.0	2.0	...	1239.0	7167.0	2644.0	158.0	4472.0	10602.0	4.0	56944.0	Barnet	389600
Bexley	0.0	61.0	123.0	0.0	185.0	518.0	1734.0	2835.0	25.0	2.0	...	279.0	2993.0	1467.0	171.0	3402.0	7120.0	2.0	31948.0	Bexley	244300
Brent	7.0	81.0	134.0	1.0	726.0	868.0	3527.0	3944.0	37.0	2.0	...	1467.0	4931.0	2281.0	169.0	5613.0	12085.0	9.0	57411.0	Brent	332100
Bromley	1.0	62.0	207.0	0.0	302.0	892.0	3027.0	3498.0	33.0	0.0	...	694.0	4541.0	1925.0	205.0	4108.0	8825.0	2.0	45711.0	Bromley	327900

5 rows × 49 columns

In [236…

```
CrimeByBoroughMinor.drop('Borough', axis=1, inplace=True)  #drop the Borough column which was added to replicate the index and enable the choropleth plot

adjustedCrimeByBoroughMinor = pd.DataFrame()
for col in CrimeByBoroughMinor.columns:
    CrimeByBoroughMinor[col]= pd.to_numeric(CrimeByBoroughMinor[col], downcast="float") #to conduct correlation analysis column must be numeric type
    adjustedCrimeByBoroughMinor[col] = 10000*CrimeByBoroughMinor[col]/CrimeByBoroughMinor['Population']

adjustedCrimeByBoroughMinor.drop('Population', axis=1, inplace=True)
adjustedCrimeByBoroughMinor.head()
```

Out[236…

Borough	Absconding from Lawful Custody	Aggravated Vehicle Taking	Arson	Bail Offences	Bicycle Theft	Burglary - Business and Community	Burglary - Residential	Criminal Damage	Dangerous Driving	Disclosure, Obstruction, False or Misleading State	...	Robbery of Personal Property	Shoplifting	Theft from Person	Theft from a Motor Vehicle	Theft or Taking of a Motor Vehicle	Threat or Possession With Intent to Commit Crimina	Violence with Injury	Violenc… without Injury
Barking and Dagenham	0.000000	2.966507	5.454545	0.047847	16.746411	28.373206	94.688995	123.827751	1.483254	0.047847	...	69.090909	77.894737	48.516746	102.631579	102.392344	6.746411	190.813397	425.69378…
Barnet	0.000000	1.899384	2.926078	0.000000	14.091376	29.645791	121.714579	96.124230	0.590349	0.051335	...	45.251540	79.081109	31.801848	183.957906	67.864476	4.055441	114.784394	272.12525…

Borough	Absconding from Lawful Custody	Aggravated Vehicle Taking	Arson	Bail Offences	Bicycle Theft	Burglary - Business and Community	Burglary - Residential	Criminal Damage	Dangerous Driving	Disclosure, Obstruction, False or Misleading State	...	Robbery of Personal Property	Shoplifting	Theft from Person	Theft from a Motor Vehicle	Theft or Taking of a Motor Vehicle	Threat or Possession With Intent to Commit Crimina	Violence with Injury	Violenc withou Injur
Bexley	0.000000	2.496930	5.034793	0.000000	7.572657	21.203438	70.978305	116.045845	1.023332	0.081867	...	21.162505	63.487515	11.420385	122.513303	60.049120	6.999591	139.255014	291.44494
Brent	0.210780	2.439024	4.034929	0.030111	21.860885	26.136706	106.202951	118.759410	1.114122	0.060223	...	57.091238	62.210178	44.173442	148.479374	68.684131	5.088829	169.015357	363.89641
Bromley	0.030497	1.890820	6.312900	0.000000	9.210125	27.203416	92.314730	106.678866	1.006404	0.000000	...	23.360781	115.004575	21.164989	138.487344	58.706923	6.251906	125.282098	269.13693

5 rows × 47 columns

Save datasets for use in later analysis

In [237…

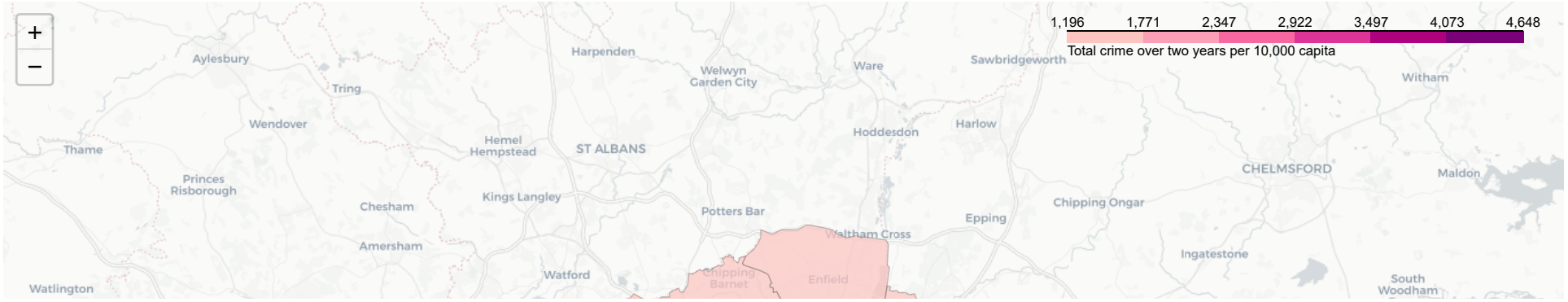
```
adjustedCrimeByBoroughMinor.to_csv('PopAdjCrimeByBoroughGranularClass.csv')
adjustedCrimeByBoroughMajor.to_csv('PopAdjCrimeByBorough.csv')
CrimeByBoroughMinor.to_csv('CrimeByBoroughGranularClass.csv')
CrimeByBoroughMajor.to_csv('CrimeByBorough.csv')
```

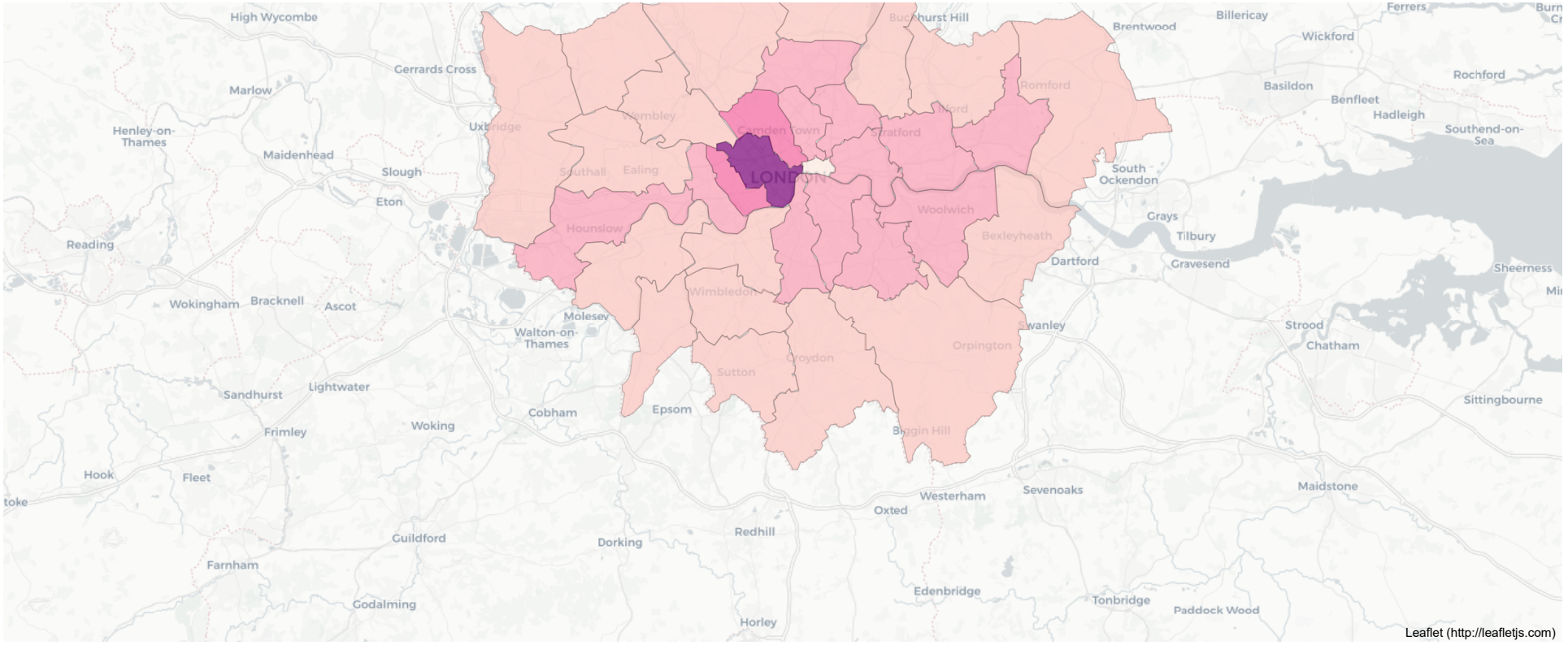
Repeat the earlier visualisations to see distribution of population adjuested crimes

In [238…

```
lnd_geo = r'london_boroughs_proper.geojson'
lnd_map = folium.Map(location = [latitude, longitude], zoom_start = 10, tiles="cartodbpositron")

adjustedCrimeByBoroughMajor['Borough'] = adjustedCrimeByBoroughMajor.index

lnd_map.choropleth(
    geo_data=lnd_geo,
    data=adjustedCrimeByBoroughMajor,
    columns=['Borough','Total Crime'],
    key_on='feature.properties.name',
    fill_color='RdPu',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Total crime over two years per 10,000 capita'
)
lnd_map.save('popadjtotalcrime.html')
lnd_map
```

Out[238…

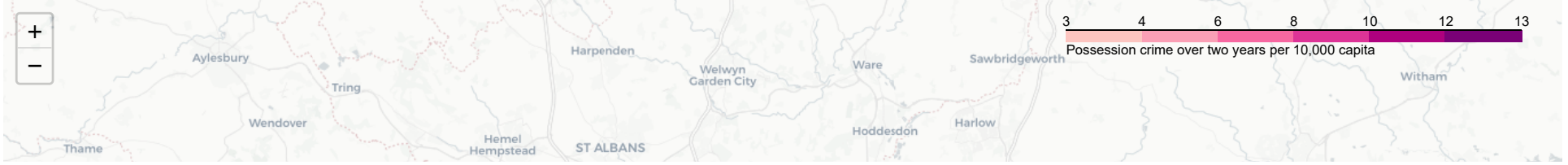


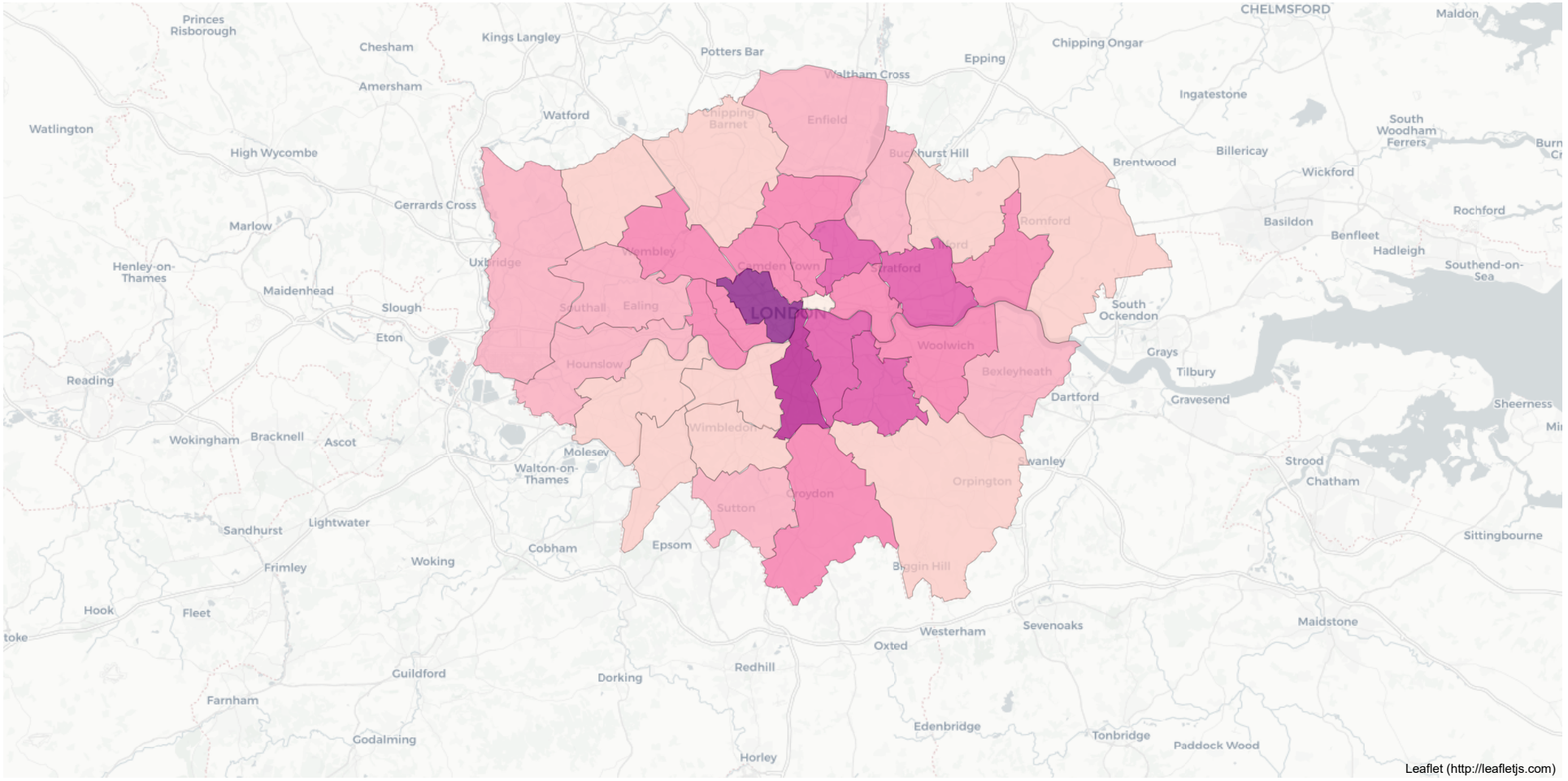


In [239…

```
lnd_geo = r'london_boroughs_proper.geojson'
lnd_map = folium.Map(location = [latitude, longitude], zoom_start = 10, tiles="cartodbpositron")

adjustedCrimeByBoroughMinor['Borough'] = adjustedCrimeByBoroughMinor.index

lnd_map.choropleth(
    geo_data=lnd_geo,
    data=adjustedCrimeByBoroughMinor,
    columns=['Borough','Possession of Article with Blade or Point'],
    key_on='feature.properties.name',
    fill_color='RdPu',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Possession crime over two years per 10,000 capita'
)
lnd_map.save('popadjbladecrime.html')
lnd_map
```

Out[239…





This emphasis the differential distribution of knife possession to overall crime. This confirms there is value in considering different categorisations of crime in the final analysis.

Ward Level Data

Repeat the same transformations to the crime data provided at ward level

Get ward population data

Retrive population data from the London data store, note this is even staler than borough level estimates and is from 2015. As ward name is not consistently recorded use ward code to key on.

In [262…

```
#London ward list
file ='https://data.london.gov.uk/download/ward-profiles-and-atlas/772d2d64-e8c6-46cb-86f9-e52b4c7851bc/ward-profiles-excel-version.csv'
wardsData= pd.read_csv(file, encoding='latin1')
#Lots of interesting data, but for this just want list of london wards
wards=pd.DataFrame(columns= ['Ward','Ward Name'])
wards['Ward Name']=wardsData['Ward name']
```

```
wards['Ward']=wardsData['New code']
wards['Population']=wardsData['Population - 2015']
wards.set_index('Ward', inplace=True)
wards.head()

#wardsData.columns
```

Out[262…

Ward	Ward Name	Population
E09000001	City of London	8100.0
E05000026	Barking and Dagenham - Abbey	14750.0
E05000027	Barking and Dagenham - Alibon	10600.0
E05000028	Barking and Dagenham - Becontree	12700.0
E05000029	Barking and Dagenham - Chadwell Heath	10400.0

In [ ]:

Get ward level crime data

Obtain the recorded crime summary by ward (most recent 24 months) from the London data store

In [263…

```
#get crime data
wardCrime = pd.read_csv('https://data.london.gov.uk/download/recorded_crime_summary/866c05de-c5cd-454b-8fe5-9e7c77ea2313/MPS%20Ward%20Level%20Crime%20%28most%20recent%2024%20months%29.csv')
wardCrime.rename(columns={'LookUp_BoroughName':'Borough'}, inplace=True)
wardCrime.rename(columns={'WardCode':'Ward'}, inplace=True)
wardCrime.head()
```

Out[263…

	MajorText	MinorText	WardName	Ward	Borough	201906	201907	201908	201909	201910	…	202008	202009	202010	202011	202012	202101	202102	202103	202104	202105
0	Arson and Criminal Damage	Arson	Abbey	E05000026	Barking and Dagenham	0	0	0	0	2	…	0	0	0	0	0	1	0	0	3	0
1	Arson and Criminal Damage	Criminal Damage	Abbey	E05000026	Barking and Dagenham	11	11	13	10	12	…	14	17	15	17	12	14	4	10	5	25
2	Burglary	Burglary - Business and Community	Abbey	E05000026	Barking and Dagenham	7	7	10	2	4	…	0	1	2	1	5	3	1	3	2	4
3	Burglary	Burglary - Residential	Abbey	E05000026	Barking and Dagenham	6	0	5	6	7	…	7	4	4	7	9	3	4	3	7	2
4	Drug Offences	Drug Trafficking	Abbey	E05000026	Barking and Dagenham	0	3	0	0	1	…	1	0	0	0	0	0	2	1	1	0

5 rows × 29 columns

Aggregate the montly data into a single Total.

In [264…

```
#create a list of month columns by working back from current data - note the range here may need tweaking  (ie to 23,-1,-1) depending on when dataset last updated
date_list=[datetime.date.today()- dateutil.relativedelta.relativedelta(months = x) for x in range(24,0,-1)]
month_list=[datetime.date.strftime(x,'%Y%m') for x in date_list]
wardCrime['Total'] = 0
```

```
for month in month_list:
    wardCrime['Total'] = wardCrime['Total']+wardCrime[month]
    wardCrime=wardCrime.drop(month,1)
wardCrime.head()
```

Out[264…

	MajorText	MinorText	WardName	Ward	Borough	Total
0	Arson and Criminal Damage	Arson	Abbey	E05000026	Barking and Dagenham	11
1	Arson and Criminal Damage	Criminal Damage	Abbey	E05000026	Barking and Dagenham	288
2	Burglary	Burglary - Business and Community	Abbey	E05000026	Barking and Dagenham	97
3	Burglary	Burglary - Residential	Abbey	E05000026	Barking and Dagenham	113
4	Drug Offences	Drug Trafficking	Abbey	E05000026	Barking and Dagenham	16

In [265…

```
wardCrime[wardCrime['Borough']=='Croydon']
```

Out[265…

	MajorText	MinorText	WardName	Ward	Borough	Total
4117	Arson and Criminal Damage	Arson	Addiscombe East	E05011462	Croydon	2
4118	Arson and Criminal Damage	Criminal Damage	Addiscombe East	E05011462	Croydon	113
4119	Burglary	Burglary - Business and Community	Addiscombe East	E05011462	Croydon	36
4120	Burglary	Burglary - Residential	Addiscombe East	E05011462	Croydon	119
4121	Drug Offences	Drug Trafficking	Addiscombe East	E05011462	Croydon	3
…	…	…	…	…	…	…
5089	Vehicle Offences	Interfering with a Motor Vehicle	Woodside	E05011489	Croydon	54
5090	Vehicle Offences	Theft from a Motor Vehicle	Woodside	E05011489	Croydon	271
5091	Vehicle Offences	Theft or Taking of a Motor Vehicle	Woodside	E05011489	Croydon	115
5092	Violence Against the Person	Violence with Injury	Woodside	E05011489	Croydon	271
5093	Violence Against the Person	Violence without Injury	Woodside	E05011489	Croydon	617

977 rows × 6 columns

In [266…

```
wardCrime[wardCrime['Ward']=='E05011489']
```

Out[266…

	MajorText	MinorText	WardName	Ward	Borough	Total
5059	Arson and Criminal Damage	Arson	Woodside	E05011489	Croydon	11
5060	Arson and Criminal Damage	Criminal Damage	Woodside	E05011489	Croydon	212
5061	Burglary	Burglary - Business and Community	Woodside	E05011489	Croydon	49
5062	Burglary	Burglary - Residential	Woodside	E05011489	Croydon	123
5063	Drug Offences	Drug Trafficking	Woodside	E05011489	Croydon	10
5064	Drug Offences	Possession of Drugs	Woodside	E05011489	Croydon	157
5065	Miscellaneous Crimes Against Society	Going Equipped for Stealing	Woodside	E05011489	Croydon	1

	MajorText	MinorText	WardName	Ward	Borough	Total
5066	Miscellaneous Crimes Against Society	Handling Stolen Goods	Woodside	E05011489	Croydon	3
5067	Miscellaneous Crimes Against Society	Obscene Publications	Woodside	E05011489	Croydon	12
5068	Miscellaneous Crimes Against Society	Other Forgery	Woodside	E05011489	Croydon	1
5069	Miscellaneous Crimes Against Society	Other Notifiable Offences	Woodside	E05011489	Croydon	4
5070	Miscellaneous Crimes Against Society	Perverting Course of Justice	Woodside	E05011489	Croydon	3
5071	Miscellaneous Crimes Against Society	Profitting From or Concealing Proceeds of Crime	Woodside	E05011489	Croydon	3
5072	Miscellaneous Crimes Against Society	Threat or Possession With Intent to Commit Cri...	Woodside	E05011489	Croydon	10
5073	Possession of Weapons	Possession of Article with Blade or Point	Woodside	E05011489	Croydon	8
5074	Possession of Weapons	Possession of Firearm with Intent	Woodside	E05011489	Croydon	2
5075	Possession of Weapons	Possession of Firearms Offences	Woodside	E05011489	Croydon	2
5076	Possession of Weapons	Possession of Other Weapon	Woodside	E05011489	Croydon	12
5077	Public Order Offences	Other Offences Against the State, or Public Order	Woodside	E05011489	Croydon	34
5078	Public Order Offences	Public Fear Alarm or Distress	Woodside	E05011489	Croydon	131
5079	Public Order Offences	Racially or Religiously Aggravated Public Fear...	Woodside	E05011489	Croydon	21
5080	Robbery	Robbery of Business Property	Woodside	E05011489	Croydon	8
5081	Robbery	Robbery of Personal Property	Woodside	E05011489	Croydon	58
5082	Sexual Offences	Other Sexual Offences	Woodside	E05011489	Croydon	30
5083	Sexual Offences	Rape	Woodside	E05011489	Croydon	29
5084	Theft	Bicycle Theft	Woodside	E05011489	Croydon	15
5085	Theft	Other Theft	Woodside	E05011489	Croydon	231
5086	Theft	Shoplifting	Woodside	E05011489	Croydon	52
5087	Theft	Theft from Person	Woodside	E05011489	Croydon	15
5088	Vehicle Offences	Aggravated Vehicle Taking	Woodside	E05011489	Croydon	2
5089	Vehicle Offences	Interfering with a Motor Vehicle	Woodside	E05011489	Croydon	54
5090	Vehicle Offences	Theft from a Motor Vehicle	Woodside	E05011489	Croydon	271
5091	Vehicle Offences	Theft or Taking of a Motor Vehicle	Woodside	E05011489	Croydon	115
5092	Violence Against the Person	Violence with Injury	Woodside	E05011489	Croydon	271
5093	Violence Against the Person	Violence without Injury	Woodside	E05011489	Croydon	617

To map to the geojson use ward code, rather than ward name. Retain a mapping from ward code to ward name for display purposes as this data is more meaningful.

In [267…
```
wardMapping = wardCrime.groupby(['Ward', 'WardName']).count()
wardMapping=wardMapping.reset_index()
wardMapping.set_index('Ward' ,inplace=True)
wardMapping['WardName'].head()
```

Out[267…
```
Ward
E05000026            Abbey
E05000027           Alibon
```

```
E05000028         Becontree
E05000029    Chadwell Heath
E05000030         Eastbrook
Name: WardName, dtype: object
```

In [268…

```
#write to csv for use in later analysis
wardMapping.to_csv('WardMapping.csv')
```

Pivot data to get a column per crime category and add a total column

In [269…

```
#using ward name not code for ease of understanding data

#drugCrime = crime[crime['MajorText']=='Drug Offences']
CrimeByWardMinor = pd.pivot_table(wardCrime, values='Total', index = ['Ward'], columns='MinorText',aggfunc=np.sum).reset_index()
CrimeByWardMajor = pd.pivot_table(wardCrime, values='Total', index = ['Ward'], columns='MajorText',aggfunc=np.sum).reset_index()
#drugCrimeByBorough.set_index(['Borough'], inplace=True)

CrimeByWardMinor['Ward'] = CrimeByWardMinor['Ward'].str.strip()
CrimeByWardMajor['Ward'] = CrimeByWardMajor['Ward'].str.strip()

CrimeByWardMinor.set_index(['Ward'], inplace=True)
CrimeByWardMajor.set_index(['Ward'], inplace=True)

CrimeByWardMajor['Total Crime'] = CrimeByWardMajor.sum(axis=1)
CrimeByWardMinor['Total Crime'] = CrimeByWardMinor.sum(axis=1)
```

In [270…

```
CrimeByWardMinor.head()
```

Out[270…

MinorText	Absconding from Lawful Custody	Aggravated Vehicle Taking	Aiding Suicide	Arson	Bail Offences	Bicycle Theft	Bigamy	Burglary - Business and Community	Burglary - Residential	Criminal Damage	…	Soliciting for Prostitution	Theft from Person	Theft from a Motor Vehicle	Theft or Taking of a Motor Vehicle	Threat or Possession With Intent to Commit Crimina	Violence with Injury	Violence without Injury	Violent Disorder	Wildlife Crime	Total Crime
Ward																					
E05000026	NaN	6.0	NaN	11.0	NaN	46.0	NaN	97.0	113.0	288.0	…	NaN	444.0	231.0	122.0	7.0	508.0	914.0	1.0	NaN	5646.0
E05000027	NaN	4.0	NaN	7.0	NaN	16.0	NaN	23.0	117.0	145.0	…	NaN	41.0	58.0	91.0	7.0	190.0	415.0	NaN	NaN	1662.0
E05000028	NaN	5.0	NaN	9.0	NaN	16.0	NaN	23.0	128.0	109.0	…	NaN	41.0	125.0	129.0	6.0	231.0	522.0	NaN	NaN	2002.0
E05000029	NaN	3.0	NaN	9.0	NaN	22.0	NaN	33.0	120.0	163.0	…	NaN	22.0	128.0	117.0	10.0	220.0	457.0	NaN	NaN	1941.0
E05000030	NaN	3.0	NaN	4.0	NaN	22.0	NaN	18.0	86.0	120.0	…	NaN	25.0	61.0	111.0	9.0	190.0	377.0	NaN	NaN	1597.0

5 rows × 55 columns

In [271…

```
CrimeByWardMinor.loc['E05011489']
```

Out[271…

```
MinorText
Absconding from Lawful Custody                  NaN
Aggravated Vehicle Taking                       2.0
Aiding Suicide                                  NaN
Arson                                          11.0
Bail Offences                                   NaN
Bicycle Theft                                  15.0
Bigamy                                          NaN
Burglary - Business and Community              49.0
Burglary - Residential                        123.0
```

```
Criminal Damage                                    212.0
Dangerous Driving                                    NaN
Disclosure, Obstruction, False or Misleading State   NaN
Drug Trafficking                                    10.0
Exploitation of Prostitution                         NaN
Forgery or Use of Drug Prescription                  NaN
Fraud or Forgery Associated with Driver Records      NaN
Going Equipped for Stealing                          1.0
Handling Stolen Goods                                3.0
Homicide                                             NaN
Interfering with a Motor Vehicle                    54.0
Making, Supplying or Possessing Articles for use i   NaN
Obscene Publications                                12.0
Offender Management Act                              NaN
Other Firearm Offences                               NaN
Other Forgery                                        1.0
Other Knife Offences                                 NaN
Other Notifiable Offences                            4.0
Other Offences Against the State, or Public Order   34.0
Other Sexual Offences                               30.0
Other Theft                                        231.0
Perjury                                              NaN
Perverting Course of Justice                         3.0
Possession of Article with Blade or Point            8.0
Possession of Drugs                                157.0
Possession of False Documents                        NaN
Possession of Firearm with Intent                    2.0
Possession of Firearms Offences                      2.0
Possession of Other Weapon                          12.0
Profitting From or Concealing Proceeds of Crime      3.0
Public Fear Alarm or Distress                      131.0
Racially or Religiously Aggravated Public Fear, Al  21.0
Rape                                                29.0
Robbery of Business Property                         8.0
Robbery of Personal Property                        58.0
Shoplifting                                         52.0
Soliciting for Prostitution                          NaN
Theft from Person                                   15.0
Theft from a Motor Vehicle                         271.0
Theft or Taking of a Motor Vehicle                 115.0
Threat or Possession With Intent to Commit Crimina  10.0
Violence with Injury                               271.0
Violence without Injury                            617.0
Violent Disorder                                     NaN
Wildlife Crime                                       NaN
Total Crime                                       2577.0
Name: E05011489, dtype: float64
```

Given the granularity of offence and geographic breakdown there are a lot of gaps in the data. ie where no offences of a type commited in a ward in two years. Remove columns for crimes with no crimes recorded for more than 200 wards

In [284…

```
CrimeByWardMinor[CrimeByWardMinor.columns[CrimeByWardMinor.isnull().sum()<=200]].loc['E05011489']
```

Out[284…

```
MinorText
Aggravated Vehicle Taking      2.0
Arson                         11.0
Bicycle Theft                 15.0
Burglary - Business and Community  49.0
Burglary - Residential       123.0
Criminal Damage              212.0
Drug Trafficking              10.0
Handling Stolen Goods          3.0
Interfering with a Motor Vehicle  54.0
Obscene Publications          12.0
Other Forgery                  1.0
Other Notifiable Offences      4.0
```

```
Other Offences Against the State, or Public Order      34.0
Other Sexual Offences                                  30.0
Other Theft                                           231.0
Perverting Course of Justice                            3.0
Possession of Article with Blade or Point               8.0
Possession of Drugs                                   157.0
Possession of Firearms Offences                         2.0
Possession of Other Weapon                             12.0
Public Fear Alarm or Distress                         131.0
Racially or Religiously Aggravated Public Fear, Al     21.0
Rape                                                   29.0
Robbery of Business Property                            8.0
Robbery of Personal Property                           58.0
Shoplifting                                            52.0
Theft from Person                                      15.0
Theft from a Motor Vehicle                            271.0
Theft or Taking of a Motor Vehicle                    115.0
Threat or Possession With Intent to Commit Crimina     10.0
Violence with Injury                                  271.0
Violence without Injury                               617.0
Total Crime                                          2577.0
Name: E05011489, dtype: float64
```

In [285…

```
#clean minor categorisation data
#remove columns with 50 or more ward with NaN data
CrimeByWardMinor=CrimeByWardMinor[CrimeByWardMinor.columns[CrimeByWardMinor.isnull().sum()<=200]]
CrimeByWardMinor=CrimeByWardMinor.fillna(0)
```

Add population to enable population adjusted crime rate to be calculated

In [286…

```
#get population adjusted crime data
CrimeByWardMajor=CrimeByWardMajor.merge(wards['Population'], left_index= True, right_index=True)
CrimeByWardMinor=CrimeByWardMinor.merge(wards['Population'], left_index= True, right_index=True)
```

In [287…

```
adjCrimeByWardMajor=pd.DataFrame()

for col in CrimeByWardMajor.columns:
    CrimeByWardMajor[col] = pd.to_numeric(CrimeByWardMajor[col], downcast="float") #numberic data type required to produce correlation matix
    adjCrimeByWardMajor[col] = 10000 *CrimeByWardMajor[col]/CrimeByWardMajor['Population'] #crimes/venues per 10,000 capita
adjCrimeByWardMajor.drop('Population', axis=1, inplace=True)
adjCrimeByWardMajor.head()
```

Out[287…

Ward	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	Total Crime
E05000026	202.711864	142.372881	448.813559	45.423729	40.677966	235.932203	229.830508	94.237288	1155.254237	268.474576	964.067797	3827.796610
E05000027	143.396226	132.075472	102.830189	22.641509	6.603774	95.283019	66.037736	43.396226	220.754717	164.150943	570.754717	1567.924528
E05000028	92.913386	118.897638	94.488189	34.645669	10.236220	91.338583	52.755906	40.944882	220.472441	226.771654	592.913386	1576.377953
E05000029	165.384615	147.115385	77.884615	36.538462	18.269231	131.730769	70.192308	50.000000	237.500000	280.769231	650.961538	1866.346154
E05000030	115.348837	96.744186	102.325581	34.418605	20.465116	97.674419	49.302326	63.255814	198.139535	180.465116	527.441860	1485.581395

In [288…

```
adjCrimeByWardMinor=pd.DataFrame()

for col in CrimeByWardMinor.columns:
    CrimeByWardMinor[col] = pd.to_numeric(CrimeByWardMinor[col], downcast="float") #numberic data type required to produce correlation matix
```

```
    adjCrimeByWardMinor[col] = 10000 *CrimeByWardMinor[col]/CrimeByWardMinor['Population'] #crimes/venues per 10,000 capita
adjCrimeByWardMinor.drop('Population', axis=1, inplace=True)
adjCrimeByWardMinor.head()
```

Out[288…

Ward	Aggravated Vehicle Taking	Arson	Bicycle Theft	Burglary - Business and Community	Burglary - Residential	Criminal Damage	Drug Trafficking	Handling Stolen Goods	Interfering with a Motor Vehicle	Obscene Publications	...	Robbery of Business Property	Robbery of Personal Property	Shoplifting	Theft from Person	Theft from a Motor Vehicle	Theft or Taking of a Motor Vehicle	Threat or Possession With Intent to Commit Crimina	Violenc with Injur
E05000026	4.067797	7.457627	31.186441	65.762712	76.610169	195.254237	10.847458	5.423729	25.084746	18.983051	...	15.593220	214.237288	411.525424	301.016949	156.610169	82.711864	4.745763	344.40678
E05000027	3.773585	6.603774	15.094340	21.698113	110.377358	136.792453	8.490566	0.000000	19.811321	8.490566	...	6.603774	59.433962	50.000000	38.679245	54.716981	85.849057	6.603774	179.24528
E05000028	3.937008	7.086614	12.598425	18.110236	100.787402	85.826772	8.661417	0.787402	22.834646	16.535433	...	3.937008	48.818898	25.984252	32.283465	98.425197	101.574803	4.724409	181.88976
E05000029	2.884615	8.653846	21.153846	31.730769	115.384615	156.730769	8.653846	3.846154	42.307692	10.576923	...	4.807692	65.384615	51.923077	21.153846	123.076923	112.500000	9.615385	211.53846
E05000030	2.790698	3.720930	20.465116	16.744186	80.000000	111.627907	3.720930	3.720930	17.674419	13.953488	...	3.720930	45.581395	24.186047	23.255814	56.744186	103.255814	8.372093	176.74418

5 rows × 33 columns

In [188…

```
adjCrimeByWardMajor=adjCrimeByWardMajor.merge(wardMapping['WardName'], left_index= True, right_index=True)
adjCrimeByWardMajor.head()
```

Out[188…

Ward	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	Total Crime	WardName
E05000026	202.711864	142.372881	448.813559	45.423729	40.677966	235.932203	229.830508	94.237288	1155.254237	268.474576	964.067797	3827.796610	Abbey
E05000027	143.396226	132.075472	102.830189	22.641509	6.603774	95.283019	66.037736	43.396226	220.754717	164.150943	570.754717	1567.924528	Alibon
E05000028	92.913386	118.897638	94.488189	34.645669	10.236220	91.338583	52.755906	40.944882	220.472441	226.771654	592.913386	1576.377953	Becontree
E05000029	165.384615	147.115385	77.884615	36.538462	18.269231	131.730769	70.192308	50.000000	237.500000	280.769231	650.961538	1866.346154	Chadwell Heath
E05000030	115.348837	96.744186	102.325581	34.418605	20.465116	97.674419	49.302326	63.255814	198.139535	180.465116	527.441860	1485.581395	Eastbrook

In [189…

```
adjCrimeByWardMinor=adjCrimeByWardMinor.merge(wardMapping['WardName'], left_index= True, right_index=True)
adjCrimeByWardMinor.head()
```

Out[189…

Ward	Arson	Bicycle Theft	Burglary - Business and Community	Burglary - Residential	Criminal Damage	Drug Trafficking	Interfering with a Motor Vehicle	Obscene Publications	Other Offences Against the State, or Public Order	Other Sexual Offences	...	Robbery of Personal Property	Shoplifting	Theft from Person	Theft from a Motor Vehicle	Theft or Taking of a Motor Vehicle	Threat or Possession With Intent to Commit Crimina	Violence with Injury	Violence without Injury
E05000026	7.457627	31.186441	65.762712	76.610169	195.254237	10.847458	25.084746	18.983051	31.186441	50.847458	...	214.237288	411.525424	301.016949	156.610169	82.711864	4.745763	344.406780	619.661017
E05000027	6.603774	15.094340	21.698113	110.377358	136.792453	8.490566	19.811321	8.490566	18.867925	25.471698	...	59.433962	50.000000	38.679245	54.716981	85.849057	6.603774	179.245283	391.509434

Ward	Arson	Bicycle Theft	Burglary - Business and Community	Burglary - Residential	Criminal Damage	Drug Trafficking	Interfering with a Motor Vehicle	Obscene Publications	Other Offences Against the State, or Public Order	Other Sexual Offences	...	Robbery of Personal Property	Shoplifting	Theft from Person	Theft from a Motor Vehicle	Theft or Taking of a Motor Vehicle	Threat or Possession With Intent to Commit Crimina	Violence with Injury	Violence without Injury
E05000028	7.086614	12.598425	18.110236	100.787402	85.826772	8.661417	22.834646	16.535433	14.960630	28.346457	...	48.818898	25.984252	32.283465	98.425197	101.574803	4.724409	181.889764	411.023622
E05000029	8.653846	21.153846	31.730769	115.384615	156.730769	8.653846	42.307692	10.576923	6.730769	27.884615	...	65.384615	51.923077	21.153846	123.076923	112.500000	9.615385	211.538462	439.423077
E05000030	3.720930	20.465116	16.744186	80.000000	111.627907	3.720930	17.674419	13.953488	19.534884	40.000000	...	45.581395	24.186047	23.255814	56.744186	103.255814	8.372093	176.744186	350.697674

5 rows × 28 columns

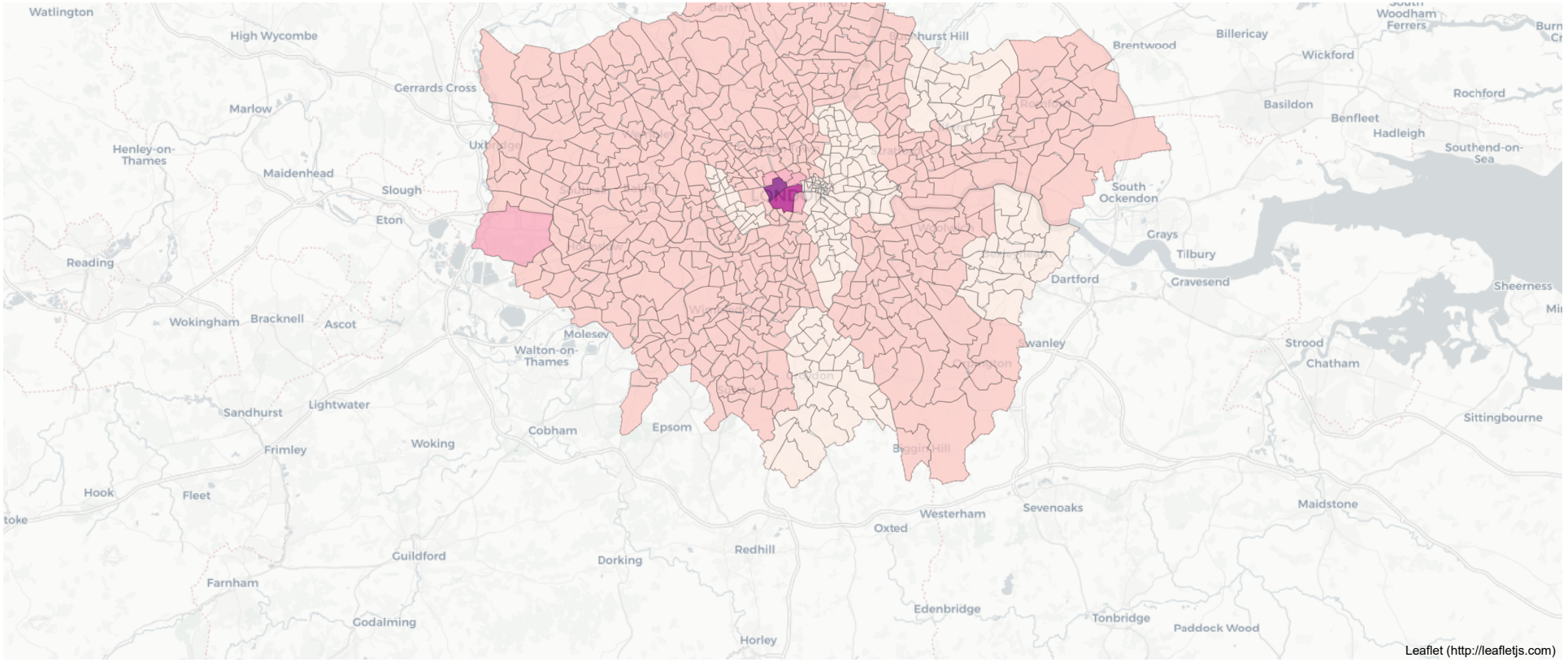
Save transformed data to file for use in later analysis

In [190…
```
adjCrimeByWardMinor.to_csv('PopAdjCrimeByWardGranularClass.csv')
adjCrimeByWardMajor.to_csv('PopAdjCrimeByWard.csv')
```

Visualise ward level crime to consider how it compares to borough level visualisation

In [289…
```
adjCrimeByWardMajor['WardCode']=adjCrimeByWardMajor.index
# create map
lnd_geo =  r'london-wards-2014.geojson'
lnd_map = folium.Map(location = [latitude, longitude], zoom_start = 10, tiles="cartodbpositron")

lnd_map.choropleth(
    geo_data=lnd_geo,
    data=adjCrimeByWardMajor,
    columns=['WardCode','Total Crime'],
    key_on='feature.properties.gss_code_ward',
    fill_color='RdPu',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Total crime per 10,000 capita'
)

lnd_map.save('popadjtotalcrimeward.html')

lnd_map
```

Out[289…



In [297…
```
CrimeByWardMajor.head()
```

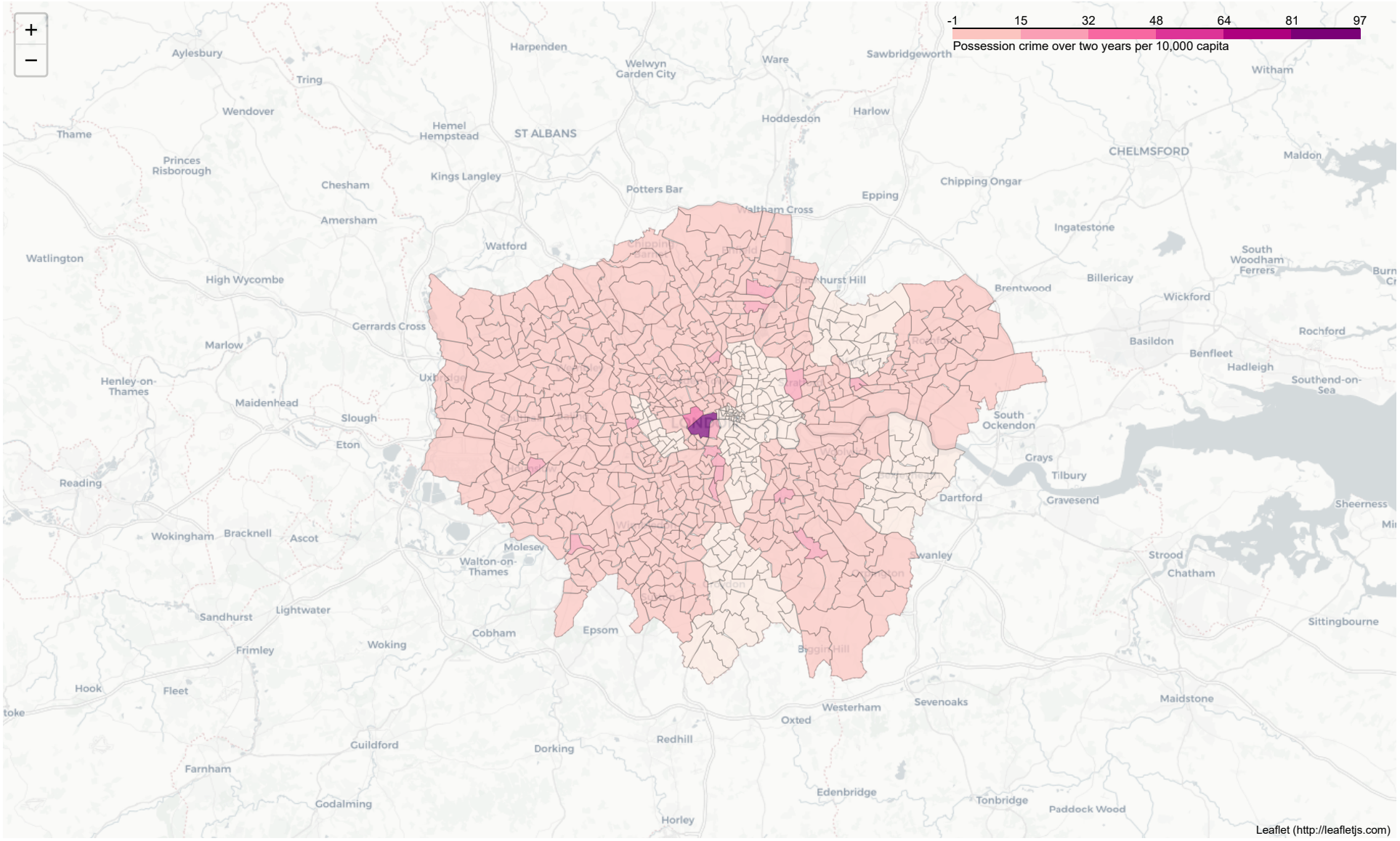
Out[297…

Ward	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	Total Crime	Population	WardCode
E05000026	299.0	210.0	662.0	67.0	60.0	348.0	339.0	139.0	1704.0	396.0	1422.0	5646.0	14750.0	E05000026
E05000027	152.0	140.0	109.0	24.0	7.0	101.0	70.0	46.0	234.0	174.0	605.0	1662.0	10600.0	E05000027
E05000028	118.0	151.0	120.0	44.0	13.0	116.0	67.0	52.0	280.0	288.0	753.0	2002.0	12700.0	E05000028
E05000029	172.0	153.0	81.0	38.0	19.0	137.0	73.0	52.0	247.0	292.0	677.0	1941.0	10400.0	E05000029
E05000030	124.0	104.0	110.0	37.0	22.0	105.0	53.0	68.0	213.0	194.0	567.0	1597.0	10750.0	E05000030

In [192…
```
lnd_geo =  r'london-wards-2014.geojson'
lnd_map = folium.Map(location = [latitude, longitude], zoom_start = 10, tiles="cartodbpositron")

adjCrimeByWardMinor['WardCode'] = adjCrimeByWardMinor.index

lnd_map.choropleth(
    geo_data=lnd_geo,
    data=adjCrimeByWardMinor,
```

```
    columns=['WardCode','Possession of Article with Blade or Point'],
    key_on='feature.properties.gss_code_ward',
    fill_color='RdPu',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Possession crime over two years per 10,000 capita'
)
lnd_map.save('popadjbladecrimeward.html')
lnd_map
```

Out[192…



Two wards have much higher crime than all others. Identify these.

In [195…
```
adjCrimeByWardMinor[['WardName', 'Total Crime']].sort_values('Total Crime', ascending=False)
```

Out[195…

Ward	WardName	Total Crime
E05000649	West End	27377.777778
E05000644	St James's	20419.917012
E05000129	Bloomsbury	7796.581197
E05000138	Holborn and Covent Garden	6796.350365
E05000641	Marylebone High Street	5978.475336
...	...	...
E05000298	Pinner South	695.774648
E05000311	Hacton	655.555556
E05000126	Shortlands	608.866995
E05000407	Coombe Vale	594.029851
E05000307	Cranham	591.304348

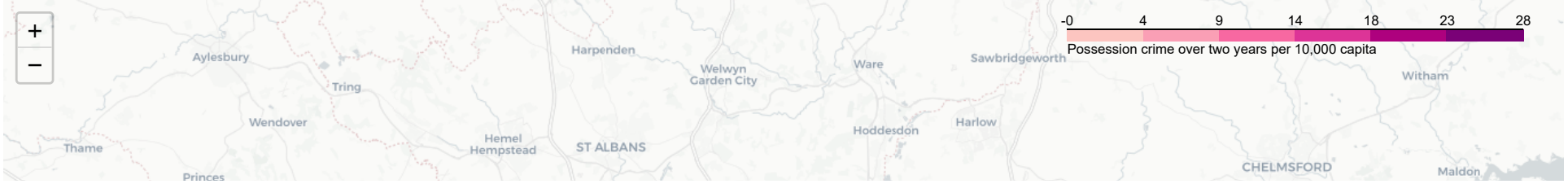
483 rows × 2 columns

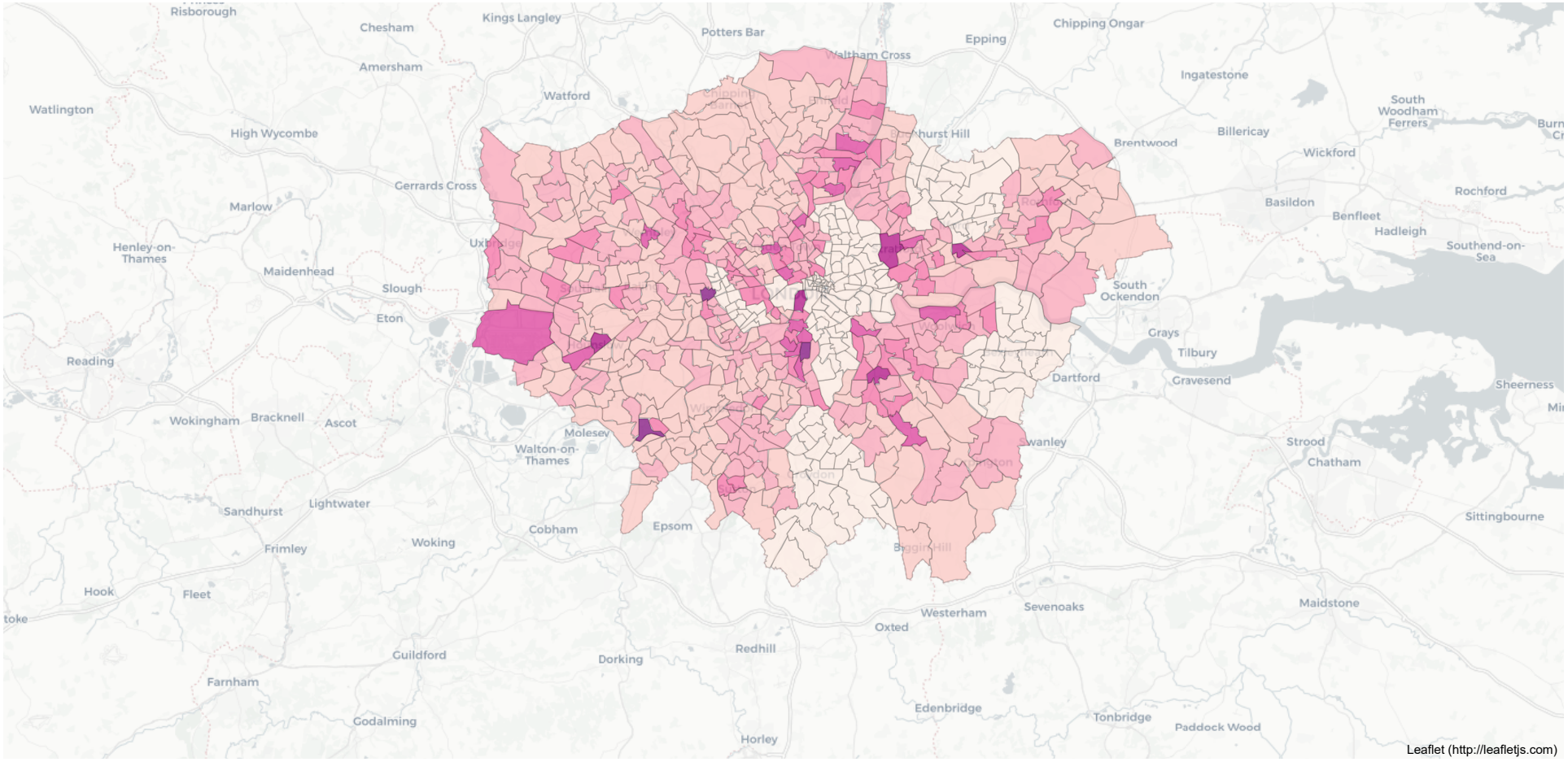
Replot data with these hotspots removed to see if more nuance is shown

In [196…
```
removehotspot = adjCrimeByWardMinor.drop(['E05000649', 'E05000644'], axis=0)
```

In [197…
```
lnd_geo =  r'london-wards-2014.geojson'
lnd_map = folium.Map(location = [latitude, longitude], zoom_start = 10, tiles="cartodbpositron")

removehotspot['WardCode'] = removehotspot.index

lnd_map.choropleth(
    geo_data=lnd_geo,
    data=removehotspot,
    columns=['WardCode','Possession of Article with Blade or Point'],
    key_on='feature.properties.gss_code_ward',
    fill_color='RdPu',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Possession crime over two years per 10,000 capita'
)
lnd_map.save('popadjbladecrimewardnohotspot.html')
lnd_map
```

Out[197…





But looking at those areas of low crime, it is clear there is an issue. Some wards are not mapping.

Wards not plotting, investigation shows this is because code has changed, reflecting redefined boundaries.

- Crime data is current boundaries
- Map is 2014
- Ward profile (population data) historic (pre 2018) boundaries.

Therefore it is not possible to meaningfully plot ward data with the datasources available

Example missing ward

adjCrimeByWardMinor.loc['E05011489']

wards.loc['E05011489']

One last attempt to plot non population adjusted data, on ward name rather than ward code

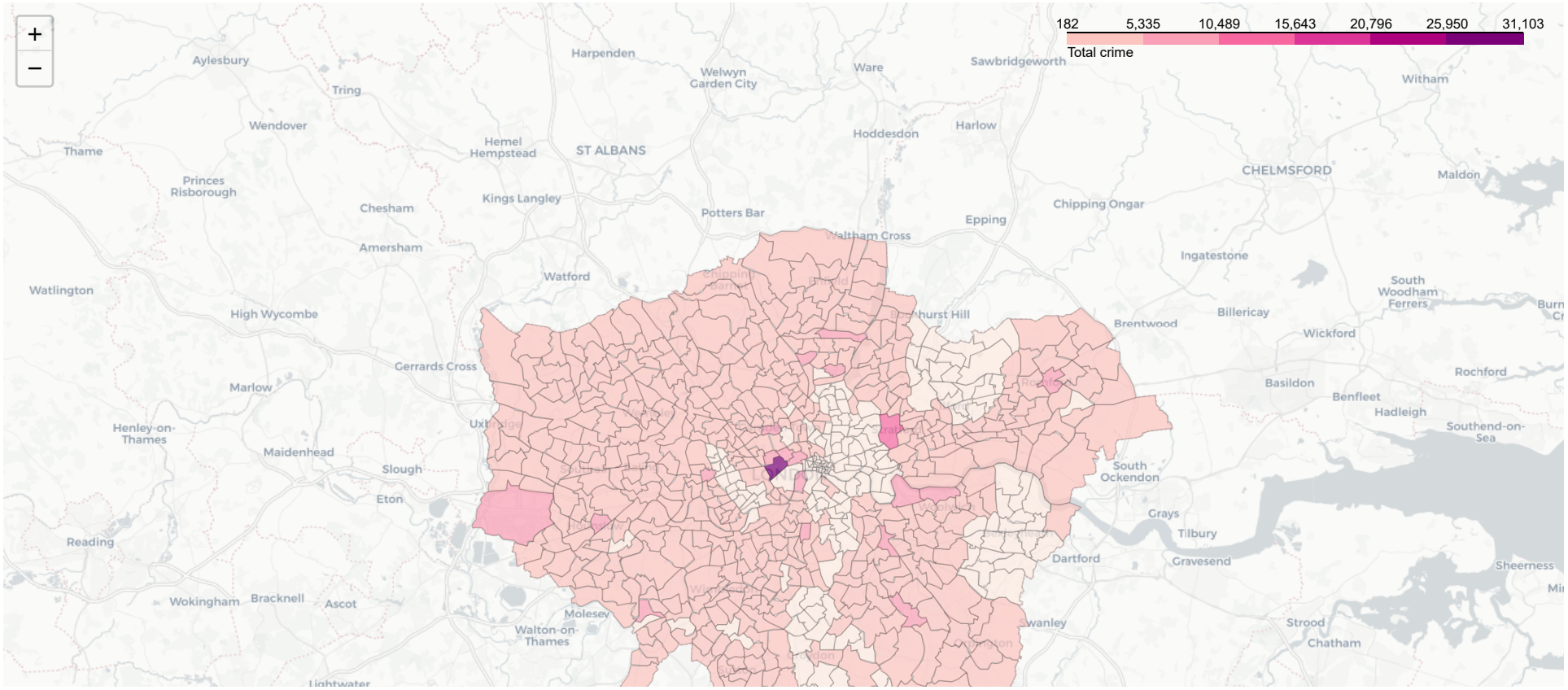
In [298…

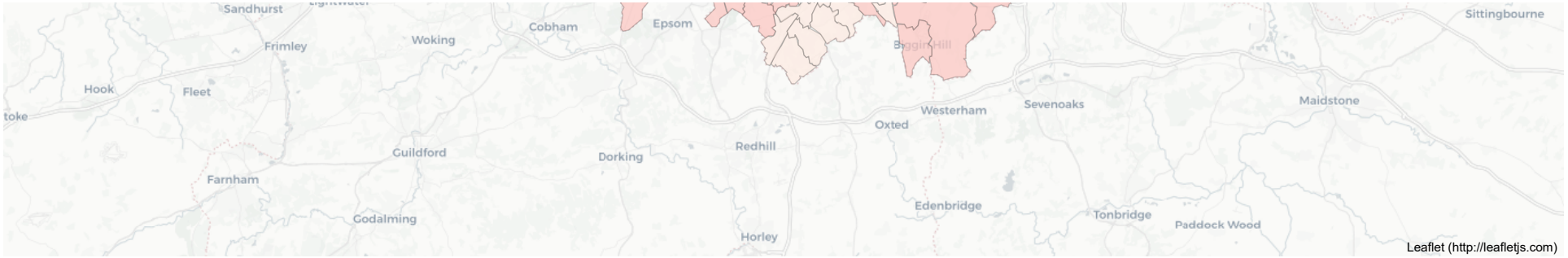
```
CrimeByWardMajor=CrimeByWardMajor.merge(wardMapping['WardName'], left_index= True, right_index=True)
```

In [301…

```
#CrimeByWardMajor['WardCode']=CrimeByWardMajor.index
# create map
lnd_geo =  r'london-wards-2014.geojson'
lnd_map = folium.Map(location = [latitude, longitude], zoom_start = 10, tiles="cartodbpositron")

lnd_map.choropleth(
    geo_data=lnd_geo,
    data=CrimeByWardMajor,
    columns=['WardName','Total Crime'],
    key_on='feature.properties.ward',
    fill_color='RdPu',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Total crime'
)



lnd_map
```

Out[301…





Conclusion: Ward level data enables more detailed breakdwon of where crime is occuring, however visualisations can be distorted by the presence of two hotspots (West End and St James) and in sizeable portions of London crime data cannot be visualised due to boundary changes

Historic Crime Data

As the 24 hour licence data is from 2018, the 2018 crime data will be used in comparison. This will be retrieved and processed in the same way as current data. Included here for completeness. See above for explanation of transformations

In [200…

```
historicCrime = pd.read_csv('https://data.london.gov.uk/download/recorded_crime_summary/2bbd58c7-6be6-40ac-99ed-38c0ee411c8e/MPS_Borough_Level_Crime_Historic.csv')
```

In [201…

```
#get crime data
#amended to use 2018 crime data only - this matches licencing data and will not contain pandemic bias
#crime = pd.read_csv('https://data.london.gov.uk/download/recorded_crime_summary/d2e9ccfc-a054-41e3-89fb-53c2bc3ed87a/MPS%20Borough%20Level%20Crime%20%28most%20recent%2024%20months%29.csv')
#crime.rename(columns={'LookUp_BoroughName':'Borough'}, inplace=True)

#date_list=[datetime.date.today()- dateutil.relativedelta.relativedelta(months = x) for x in range(25,1,-1)]
#month_list=[datetime.date.strftime(x,'%Y%m') for x in date_list]

date_list=[datetime.date.fromisoformat('2018-12-01') - dateutil.relativedelta.relativedelta(months = x) for x in range(11,-1,-1)]
month_list=[datetime.date.strftime(x,'%Y%m') for x in date_list]
historiCrime['Total'] = 0

for month in month_list:
    historiCrime['Total'] = historiCrime['Total']+historiCrime[month]
    historiCrime=historiCrime.drop(month,1)

#pivot to get column per crime
historicCrimeByBoroughMinor = pd.pivot_table(historiCrime, values='Total', index = ['Borough'], columns='Minor Category',aggfunc=np.sum).reset_index()
historicCrimeByBoroughMajor = pd.pivot_table(historiCrime, values='Total', index = ['Borough'], columns='Major Category',aggfunc=np.sum).reset_index()

#clean borough so it can be mapped to other data
historicCrimeByBoroughMinor['Borough'] = historicCrimeByBoroughMinor['Borough'].str.strip()
historicCrimeByBoroughMajor['Borough'] = historicCrimeByBoroughMajor['Borough'].str.strip()

#set borough as index
historicCrimeByBoroughMinor.set_index(['Borough'], inplace=True)
historicCrimeByBoroughMajor.set_index(['Borough'], inplace=True)

#add a total crime column
historicCrimeByBoroughMajor['Total Crime 2018'] = historicCrimeByBoroughMajor.sum(axis=1)
historicCrimeByBoroughMinor['Total Crime 2018'] = historicCrimeByBoroughMinor.sum(axis=1)
```

In [202…
```
historicCrimeByBoroughMinor.columns[historicCrimeByBoroughMinor.isnull().sum()<=8]
```

Out[202…
```
Index(['Assault with Injury', 'Burglary in Other Buildings',
       'Burglary in a Dwelling', 'Business Property', 'Common Assault',
       'Counted per Victim', 'Criminal Damage To Dwelling',
       'Criminal Damage To Motor Vehicle', 'Criminal Damage To Other Building',
       'Drug Trafficking', 'Going Equipped', 'Grievous Bodily Harm',
       'Handling Stolen Goods', 'Harassment',
       'Motor Vehicle Interference & Tampering', 'Murder', 'Offensive Weapon',
       'Other Criminal Damage', 'Other Drugs', 'Other Fraud & Forgery',
       'Other Notifiable', 'Other Sexual', 'Other Theft', 'Other Theft Person',
       'Other violence', 'Personal Property', 'Possession Of Drugs', 'Rape',
       'Theft From Motor Vehicle', 'Theft From Shops',
       'Theft/Taking Of Motor Vehicle', 'Theft/Taking of Pedal Cycle',
       'Wounding/GBH', 'Total Crime 2018'],
      dtype='object', name='Minor Category')
```

In [203…
```
#for minor classifications not all crimes have data
#clean remove crimes with low data, and set nan to 0
#tolerance of only 8 boroughs not reporting data
historicCrimeByBoroughMinor=historicCrimeByBoroughMinor[historicCrimeByBoroughMinor.columns[historicCrimeByBoroughMinor.isnull().sum()<=8]]
historicCrimeByBoroughMinor=historicCrimeByBoroughMinor.fillna(0)
```

In [ ]:
```
```

In [204…
```
#let's adjust for population, ie do more facilities represent more people and so more crime?
historicCrimeByBoroughMajor=historicCrimeByBoroughMajor.merge(boroughs, left_index= True, right_index=True)
historicCrimeByBoroughMajor['Population'].head()
```

Out[204…
```
Borough
Barking and Dagenham    209000
Barnet                  389600
Bexley                  244300
Brent                   332100
Bromley                 327900
Name: Population, dtype: int64
```

In [206…
```
adjHistoricCrimeByBoroughMajor=historicCrimeByBoroughMajor #.merge(boroughs, left_index= True, right_index=True)

for col in adjHistoricCrimeByBoroughMajor.columns:
    adjHistoricCrimeByBoroughMajor[col] = pd.to_numeric(adjHistoricCrimeByBoroughMajor[col], downcast="float") #numberic data type required to produce correlation matix
    adjHistoricCrimeByBoroughMajor[col] = 10000 *adjHistoricCrimeByBoroughMajor[col]/adjHistoricCrimeByBoroughMajor['Population'] #crimes/venues per 10,000 capita

adjHistoricCrimeByBoroughMajor.drop('Population', axis=1, inplace=True)
adjHistoricCrimeByBoroughMajor.head()
```

Out[206…

Borough	Burglary	Criminal Damage	Drugs	Fraud or Forgery	Other Notifiable Offences	Robbery	Sexual Offences	Theft and Handling	Violence Against the Person	Total Crime 2018
Barking and Dagenham	91.531097	73.110046	43.636364	1.100478	18.086124	39.473682	24.401915	269.043060	310.287079	870.669922
Barnet	113.757698	51.463039	17.736139	1.309035	13.809036	19.378851	16.221766	281.314178	210.061600	725.051331
Bexley	80.515762	64.879250	24.273434	0.450266	15.963978	13.221449	15.882113	200.368393	211.870651	627.425293
Brent	112.014450	65.943993	53.477867	1.415236	19.211081	43.992775	20.867208	291.629028	316.350494	924.902222
Bromley	96.431839	67.429092	21.439463	0.670936	12.351327	13.815187	17.749313	249.374832	228.514771	707.776733

In [ ]:

In [207…
```
#let's adjust for population, ie do more facilities represent more people and so more crime?
historicCrimeByBoroughMinor=historicCrimeByBoroughMinor.merge(boroughs, left_index= True, right_index=True)
historicCrimeByBoroughMinor['Population'].head()
```

Out[207…
```
Borough
Barking and Dagenham    209000
Barnet                  389600
Bexley                  244300
Brent                   332100
Bromley                 327900
Name: Population, dtype: int64
```

In [211…
```
adjHistoricCrimeByBoroughMinor=historicCrimeByBoroughMinor #.merge(boroughs, left_index= True, right_index=True)

for col in adjHistoricCrimeByBoroughMinor.columns:
    adjHistoricCrimeByBoroughMinor[col] = pd.to_numeric(adjHistoricCrimeByBoroughMinor[col], downcast="float") #numberic data type required to produce correlation matix
    adjHistoricCrimeByBoroughMinor[col] = 10000 *adjHistoricCrimeByBoroughMinor[col]/adjHistoricCrimeByBoroughMinor['Population'] #crimes/venues per 10,000 capita
adjHistoricCrimeByBoroughMinor.drop('Population', axis=1, inplace=True)
adjHistoricCrimeByBoroughMinor.head()
```

Out[211…

Borough	Assault with Injury	Burglary in Other Buildings	Burglary in a Dwelling	Business Property	Common Assault	Counted per Victim	Criminal Damage To Dwelling	Criminal Damage To Motor Vehicle	Criminal Damage To Other Building	Drug Trafficking	...	Other violence	Personal Property	Possession Of Drugs	Rape	Theft From Motor Vehicle	Theft From Shops	Theft/Taking Of Motor Vehicle	Theft/Taking of Pedal Cycle	Woundin
Barking and Dagenham	68.086121	30.095694	61.435406	2.727273	80.239235	0.0	15.358851	30.143541	4.162679	2.966507	...	21.100479	36.746410	40.622009	10.095694	51.913876	37.799042	60.669857	9.712918	30.
Barnet	39.810062	39.014374	74.743324	1.745380	56.160164	0.0	10.087269	21.201233	4.132443	2.464066	...	9.009240	17.633471	15.118071	7.032854	77.900414	40.066734	39.373718	4.491786	16.
Bexley	46.090874	24.109701	56.406055	2.251330	54.441261	0.0	13.589849	28.039293	5.075727	1.146132	...	10.929186	10.970119	23.045437	6.385592	53.172333	41.301678	31.764225	4.748260	18.
Brent	61.276726	36.826256	75.188194	3.432701	98.614876	0.0	14.754592	22.613670	6.564288	4.125264	...	16.440830	40.560074	48.539597	7.347184	64.227646	40.981632	44.534779	10.237880	33.
Bromley	46.477585	29.673681	66.758156	2.104300	58.401951	0.0	13.815187	23.940226	5.672461	1.158890	...	9.667582	11.710888	20.189081	6.221409	61.878620	63.250996	27.447393	6.556877	18.

5 rows × 34 columns

Save transformed data to file for use in later analysis

In [212…
```
adjHistoricCrimeByBoroughMinor.to_csv('PopAdjCrimeHistoricGranularClass.csv')
adjHistoricCrimeByBoroughMajor.to_csv('PopAdjCrimeHistoric.csv')
```

In [213…
```
historicCrimeByBoroughMinor.to_csv('CrimeHistoricGranularClass.csv')
historicCrimeByBoroughMajor.to_csv('CrimeHistoric.csv')
```

In [ ]: