Q9 ANSWER

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import folium
import os, re
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import normalize
from IPython.display import IFrame
from sklearn.cluster import AgglomerativeClustering
import scipy.cluster.hierarchy as shc

%matplotlib inline

import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
path_to_data = './crime'
cd = os.path.dirname(os.path.abspath(path_to_data))

i = 0
columns = range(1,100)
dfList = []

for root, dirs, files in os.walk(cd):
    for fname in files:
        if re.match("^.*.csv$", fname):
            frame = pd.read_csv(os.path.join(root, fname))
            frame['key'] = "file{}".format(i)
            dfList.append(frame)
            i += 1

dataset = pd.concat(dfList)
```

In [3]:
```
dataset.head()
```

Out[3]:

	Crime ID	Month	Reported by	Falls within	Longitude	Latitude	Location	LSOA code	LSOA name	Crime type
0	NaN	2021-06	Essex Police	Essex Police	0.864094	51.971811	On or near Bear Street	E01029906	Babergh 009D	Anti-social behaviour
1	NaN	2021-06	Essex Police	Essex Police	0.436057	51.639952	On or near Coach Mews	E01021237	Basildon 001A	Anti-social behaviour
2	d91fddaaae8b0664cf330fc1a85bfdcddc57256d0bd2b3...	2021-06	Essex Police	Essex Police	0.437217	51.642455	On or near Bridleway	E01021238	Basildon 001B	Vehicle crime
3	f5104dc9cd4aaa31f162b0bed7b7f7714f0bdf266fa388...	2021-06	Essex Police	Essex Police	0.435880	51.643391	On or near Penwood Close	E01021238	Basildon 001B	Violence and sexual offences
4	faa6b0a7146e1e2816512d2f2505d98c384451518f3935...	2021-06	Essex Police	Essex Police	0.435880	51.643391	On or near Penwood Close	E01021238	Basildon 001B	Violence and sexual offences

In [4]:
```
print(dataset.shape)
```

```
(4231520, 13)
```

In [5]:
```
name_number = 'PreciousAdaugoReginald2-2325671.csv'
dataset.to_csv(name_number, index=False)
```

In [6]:
```
data = pd.read_csv(name_number)
```

In [7]:
```
data['town'] = data['LSOA name'].str.split(' ').str[0]
```

In [8]:
```
data.head()
```

Out[8]:

	Crime ID	Month	Reported by	Falls within	Longitude	Latitude	Location	LSOA code	LSOA name	Crime type
0	NaN	2021-06	Essex Police	Essex Police	0.864094	51.971811	On or near Bear Street	E01029906	Babergh 009D	Anti-social behaviour
1	NaN	2021-06	Essex Police	Essex Police	0.436057	51.639952	On or near Coach Mews	E01021237	Basildon 001A	Anti-social behaviour
2	d91fddaaae8b0664cf330fc1a85bfdcddc57256d0bd2b3...	2021-06	Essex Police	Essex Police	0.437217	51.642455	On or near Bridleway	E01021238	Basildon 001B	Vehicle crime
3	f5104dc9cd4aaa31f162b0bed7b7f7714f0bdf266fa388...	2021-06	Essex Police	Essex Police	0.435880	51.643391	On or near Penwood Close	E01021238	Basildon 001B	Violence and sexual offences
4	faa6b0a7146e1e2816512d2f2505d98c384451518f3935...	2021-06	Essex Police	Essex Police	0.435880	51.643391	On or near Penwood Close	E01021238	Basildon 001B	Violence and sexual offences

In [9]:
```
towns = ['Basildon']
filtered_data = data[data.town.str.contains('|'.join(towns), na=False)]
filtered_data.head()
```

Out[9]:

	Crime ID	Month	Reported by	Falls within	Longitude	Latitude	Location	LSOA code	LSOA name	Crime type
1	NaN	2021-06	Essex Police	Essex Police	0.436057	51.639952	On or near Coach Mews	E01021237	Basildon 001A	Anti-social behaviour
2	d91fddaaae8b0664cf330fc1a85bfdcddc57256d0bd2b3...	2021-06	Essex Police	Essex Police	0.437217	51.642455	On or near Bridleway	E01021238	Basildon 001B	Vehicle crime
3	f5104dc9cd4aaa31f162b0bed7b7f7714f0bdf266fa388...	2021-06	Essex Police	Essex Police	0.435880	51.643391	On or near Penwood Close	E01021238	Basildon 001B	Violence and sexual offences
4	faa6b0a7146e1e2816512d2f2505d98c384451518f3935...	2021-06	Essex Police	Essex Police	0.435880	51.643391	On or near Penwood Close	E01021238	Basildon 001B	Violence and sexual offences
5	009729cbc836771f2f96b686541993ef683a04928b6c23...	2021-06	Essex Police	Essex Police	0.432812	51.642519	On or near Robin Close	E01021238	Basildon 001B	Violence and sexual offences

Preparation of data for clustering¶¶

Column selection

In [10]:
```
filtered_important_data = filtered_data[['LSOA code','Crime type']]
filtered_important_data = pd.get_dummies(filtered_important_data, columns=['Crime type'])
clustering_data = filtered_important_data.groupby(['LSOA code']).agg(
     {'Crime type_Anti-social behaviour':'sum',
 'Crime type_Bicycle theft':'sum',
 'Crime type_Burglary':'sum',
 'Crime type_Criminal damage and arson':'sum',
 'Crime type_Drugs':'sum',
 'Crime type_Other crime':'sum',
 'Crime type_Other theft':'sum',
 'Crime type_Possession of weapons':'sum',
 'Crime type_Public order':'sum',
 'Crime type_Robbery':'sum',
 'Crime type_Shoplifting':'sum',
 'Crime type_Theft from the person':'sum',
 'Crime type_Vehicle crime':'sum',
 'Crime type_Violence and sexual offences':'sum'
 }
 ).reset_index()
```

In [11]:
```
clustering_data[:5]
```

Out[11]:

	LSOA code	Crime type_Anti-social behaviour	Crime type_Bicycle theft	Crime type_Burglary	Crime type_Criminal damage and arson	Crime type_Drugs	Crime type_Other crime	Crime type_Other theft	Crime type_Possession of weapons	Crime type_Public order	type_Ro
0	E01021237	20.0	0.0	20.0	0.0	20.0	0.0	60.0	40.0	160.0	
1	E01021238	40.0	0.0	20.0	60.0	0.0	0.0	0.0	0.0	80.0	
2	E01021239	80.0	0.0	40.0	80.0	80.0	20.0	80.0	20.0	40.0	
3	E01021240	400.0	40.0	40.0	200.0	60.0	0.0	140.0	20.0	200.0	
4	E01021241	200.0	0.0	60.0	260.0	60.0	20.0	40.0	0.0	220.0	

In [12]:
```
clustering_data_original = clustering_data.copy()
clustering_data_original.head()
```

Out[12]:

	LSOA code	Crime type_Anti-social behaviour	Crime type_Bicycle theft	Crime type_Burglary	Crime type_Criminal damage and arson	Crime type_Drugs	Crime type_Other crime	Crime type_Other theft	Crime type_Possession of weapons	Crime type_Public order	type_Ro
0	E01021237	20.0	0.0	20.0	0.0	20.0	0.0	60.0	40.0	160.0	
1	E01021238	40.0	0.0	20.0	60.0	0.0	0.0	0.0	0.0	80.0	
2	E01021239	80.0	0.0	40.0	80.0	80.0	20.0	80.0	20.0	40.0	
3	E01021240	400.0	40.0	40.0	200.0	60.0	0.0	140.0	20.0	200.0	
4	E01021241	200.0	0.0	60.0	260.0	60.0	20.0	40.0	0.0	220.0	

In [13]:
```
clustering_data.drop(['LSOA code'], axis = 1, inplace = True, errors = 'ignore')
clustering_data.head()
```

Out[13]:

	Crime type_Anti-social behaviour	Crime type_Bicycle theft	Crime type_Burglary	Crime type_Criminal damage and arson	Crime type_Drugs	Crime type_Other crime	Crime type_Other theft	Crime type_Possession of weapons	Crime type_Public order	Crime type_Robbery	type_
0	20.0	0.0	20.0	0.0	20.0	0.0	60.0	40.0	160.0	20.0	
1	40.0	0.0	20.0	60.0	0.0	0.0	0.0	0.0	80.0	0.0	
2	80.0	0.0	40.0	80.0	80.0	20.0	80.0	20.0	40.0	0.0	
3	400.0	40.0	40.0	200.0	60.0	0.0	140.0	20.0	200.0	20.0	
4	200.0	0.0	60.0	260.0	60.0	20.0	40.0	0.0	220.0	0.0	

Normalization step

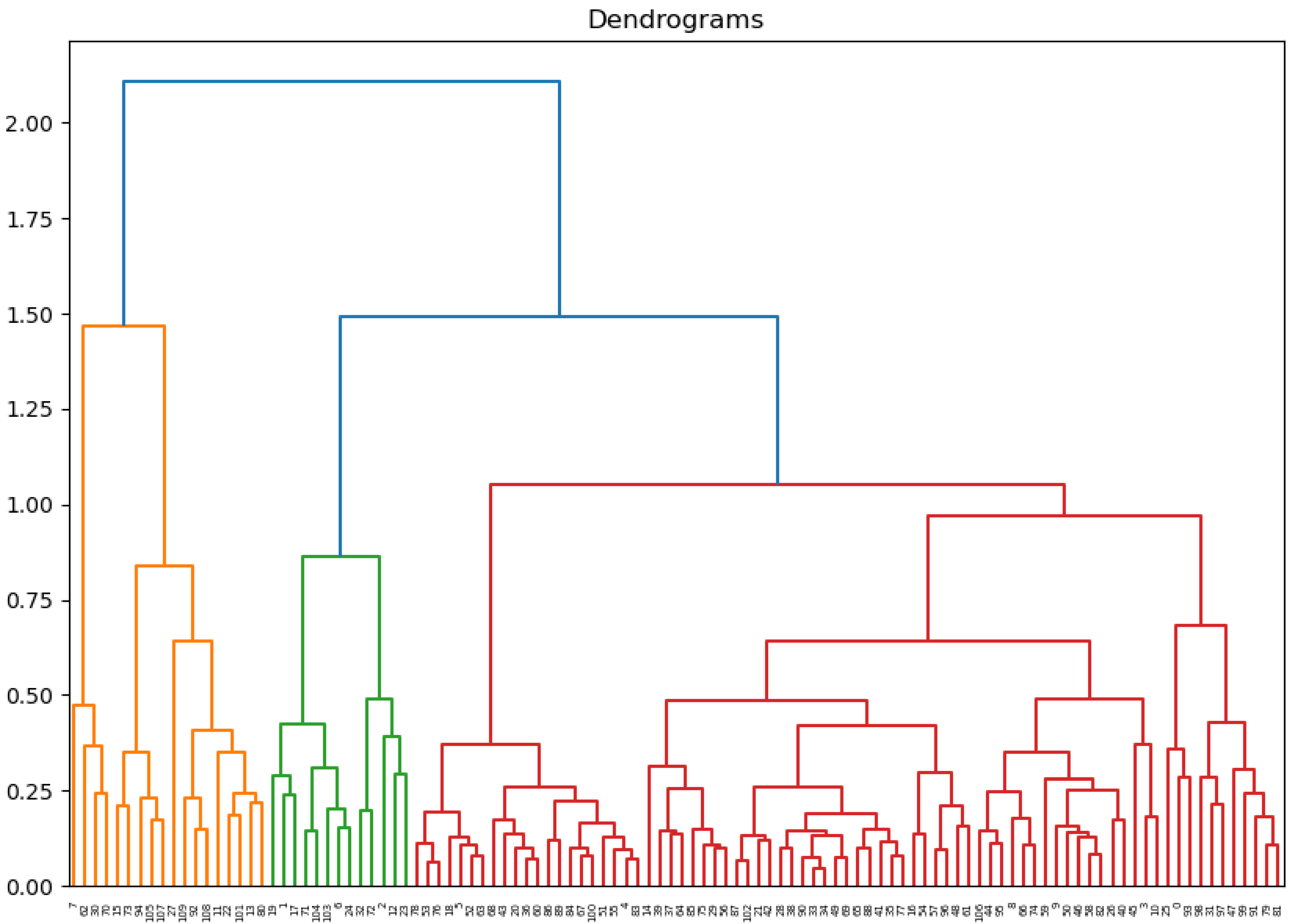
In [14]:
```
data_scaled = normalize(clustering_data)
data_scaled = pd.DataFrame(data_scaled, columns=clustering_data.columns)
data_scaled.head()
```

Out[14]:

	Crime type_Anti-social behaviour	Crime type_Bicycle theft	Crime type_Burglary	Crime type_Criminal damage and arson	Crime type_Drugs	Crime type_Other crime	Crime type_Other theft	Crime type_Possession of weapons	Crime type_Public order	Crime type_Robbery	type_
0	0.057544	0.000000	0.057544	0.000000	0.057544	0.000000	0.172631	0.115087	0.460348	0.057544	
1	0.081992	0.000000	0.040996	0.122988	0.000000	0.000000	0.000000	0.000000	0.163984	0.000000	
2	0.317221	0.000000	0.158610	0.317221	0.317221	0.079305	0.317221	0.079305	0.158610	0.000000	
3	0.427765	0.042776	0.042776	0.213882	0.064165	0.000000	0.149718	0.021388	0.213882	0.021388	
4	0.159516	0.000000	0.047855	0.207371	0.047855	0.015952	0.031903	0.000000	0.175467	0.000000	

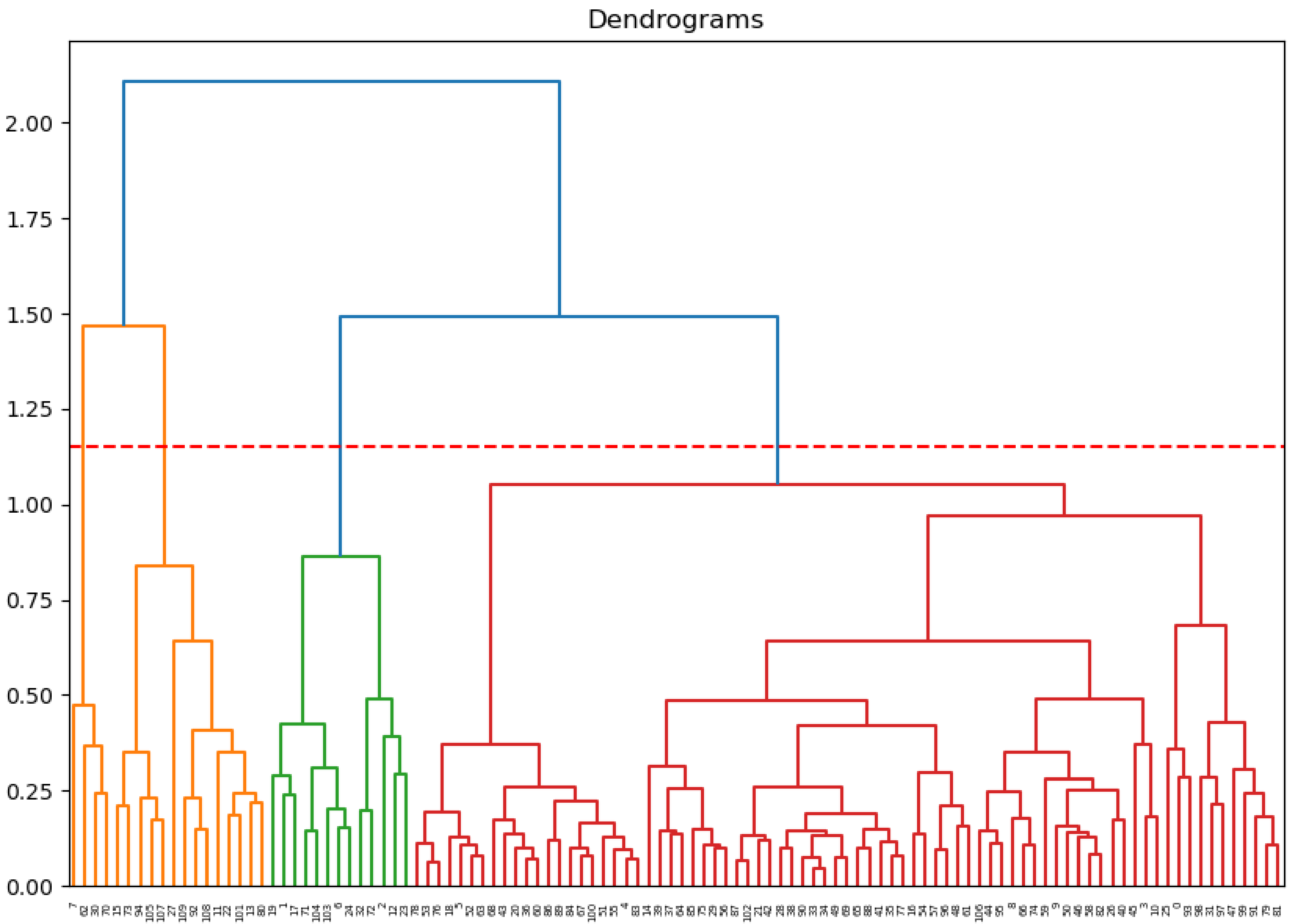
Determination of number of clusters with the use of the Dendogram

In [15]:
```
plt.figure(figsize=(10, 7))
plt.title("Dendrograms")
dend = shc.dendrogram(shc.linkage(data_scaled, method='ward'))
```


Dendrograms
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00

In [16]:
```
plt.figure(figsize=(10, 7))
plt.title("Dendrograms")
dend = shc.dendrogram(shc.linkage(data_scaled, method='ward'))
plt.axhline(y=1.150, color='r', linestyle='--')
```

Out[16]:
```
<matplotlib.lines.Line2D at 0x27acd7e0be0>
```


Dendrograms
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00

Agglomerative clustering

In [17]:
```
cluster = AgglomerativeClustering(n_clusters=4, affinity='euclidean', linkage='ward')
cluster_ids = cluster.fit_predict(data_scaled)
```

In [18]:
```
clustering_data['cluster'] = cluster_ids
clustering_data.head()
```

Out[18]:

	Crime type_Anti-social behaviour	Crime type_Bicycle theft	Crime type_Burglary	Crime type_Criminal damage and arson	Crime type_Drugs	Crime type_Other crime	Crime type_Other theft	Crime type_Possession of weapons	Crime type_Public order	Crime type_Robbery	type_
0	20.0	0.0	20.0	0.0	20.0	0.0	60.0	40.0	160.0	20.0	
1	40.0	0.0	20.0	60.0	0.0	0.0	0.0	0.0	80.0	0.0	
2	80.0	0.0	40.0	80.0	80.0	20.0	80.0	20.0	40.0	0.0	
3	400.0	40.0	40.0	200.0	60.0	0.0	140.0	20.0	200.0	20.0	
4	200.0	0.0	60.0	260.0	60.0	20.0	40.0	0.0	220.0	0.0	

In [19]:
```
hierarchical_cluster = pd.DataFrame(round(clustering_data.groupby('cluster').mean(),1))
hierarchical_cluster
```

Out[19]:

cluster	Crime type_Anti-social behaviour	Crime type_Bicycle theft	Crime type_Burglary	Crime type_Criminal damage and arson	Crime type_Drugs	Crime type_Other crime	Crime type_Other theft	Crime type_Possession of weapons	Crime type_Public order	Crime type_Robbery
0	683.8	33.4	133.4	398.5	164.8	92.2	241.0	46.6	515.7	57.7
1	1044.3	8.6	108.6	374.3	81.4	51.4	212.9	35.7	352.9	37.1
2	220.0	3.1	107.7	178.5	76.9	49.2	106.2	23.1	121.5	16.9
3	840.0	15.0	205.0	380.0	190.0	130.0	455.0	35.0	495.0	55.0

Visualization of clusters

In [20]:
```
clustering_data_original['cluster'] = cluster_ids
clusters = clustering_data_original[['LSOA code', 'cluster']]
```

In [21]:
```
clusters.head()
```

Out[21]:

	LSOA code	cluster
0	E01021237	0
1	E01021238	2
2	E01021239	2
3	E01021240	0
4	E01021241	0

In [22]:
```
clusters.shape
```

Out[22]: (110, 2)

In [23]:
```
clustered_full = pd.merge(filtered_data, clusters, on='LSOA code')
clustered_full.head()
```

Out[23]:

	Crime ID	Month	Reported by	Falls within	Longitude	Latitude	Location	LSOA code	LSOA name	Crime type
0	NaN	2021-06	Essex Police	Essex Police	0.436057	51.639952	On or near Coach Mews	E01021237	Basildon 001A	Anti-social behaviour
1	6525a41d8af97ebaf31695775731dbc9ccc618d224453e...	2021-07	Essex Police	Essex Police	0.440629	51.639767	On or near Martingale Close	E01021237	Basildon 001A	Public order
2	a653ddc246de60337163026c70c4866e3de3eec1d4a8e5...	2021-07	Essex Police	Essex Police	0.432440	51.638525	On or near Parking Area	E01021237	Basildon 001A	Violence and sexual offences
3	b33a5c1572b6cd9d3a1d2572a670e0832b28f4828ec357...	2021-09	Essex Police	Essex Police	0.439140	51.641641	On or near Derby Close	E01021237	Basildon 001A	Burglary
4	dc164a3c2be1a571b597eb38d9e15f64b325657d95098d...	2021-09	Essex Police	Essex Police	0.432440	51.638525	On or near Parking Area	E01021237	Basildon 001A	Drugs

In [24]:
```
def get_color(cluster_id):
    if cluster_id == 1:
        return 'darkred'
    if cluster_id == 2:
        return 'green'
    if cluster_id == 0:
        return 'amber'
    if cluster_id == 3:
        return 'blue'
```

Creation of map with clusters

In [25]:
```
#create a map
this_map = folium.Map(location =[clustered_full["Latitude"].mean(),
                                 clustered_full["Longitude"].mean()], zoom_start=5)

def plot_dot(point):
    '''input: series that contains a numeric named latitude and a numeric named longitude
    this function creates a CircleMarker and adds it to your this_map'''
    folium.CircleMarker(location=[point.Latitude, point.Longitude],
                        radius=2,
                        color=point.color,
                        weight=1).add_to(this_map)


clustered_full["color"] = clustered_full["cluster"].apply(lambda x: get_color(x))

#use df.apply(,axis=1) to iterate through every row in your dataframe
clustered_full.apply(plot_dot, axis = 1)

#Set the zoom to the maximum possible
this_map.fit_bounds(this_map.get_bounds())

#Save the map to an HTML file
this_map.save(os.path.join('Crime_map2.html'))
#IFrame(src='Crime_map.html', width=1000, height=600)
```

This website below shows evidence of crime rates in the town of Basildon beetwen June 2021 and June 2022

[https://www.varbes.com/crime/basildon-crime (https://www.varbes.com/crime/basildon-crime)](https://www.varbes.com/crime/basildon-crime)

Brief discussion of results

For this task a clustering was performed for the town named Basildon which is an area covered by the Essex Police.

After having created the 4 clusters using hierarchical clustering algorithm it can be seen from the table in cell 20 that ClusterID3 contains they LSOA codes that have the highest number of crimes which means that those are very high risk areas.

ClusterID 2 has the LSOA ecodes the contain the lowest number of crimes therefore those are defined as low risk areas. ClusterID 0 has LSOA codes with a moderate number of crimes whereas ClusterID 1 has LSOA codes where the crimes are mild.

Then at the end a map file was created that contains a map full of clusters which therefore contains LSOA codes for the named Basildon.