Import Libraries

In [12]:

```
import pandas as pd
from sklearn.preprocessing import normalize
from matplotlib import pyplot as plt
import seaborn as sns
import scipy.cluster.hierarchy as sch
from sklearn.cluster import AgglomerativeClustering
```

In [13]:

```
crime_data = pd.read_csv('crime_data.csv')
crime_data
```

Out[13]:

	Unnamed: 0	Murder	Assault	UrbanPop	Rape
0	Alabama	13.2	236	58	21.2
1	Alaska	10.0	263	48	44.5
2	Arizona	8.1	294	80	31.0
3	Arkansas	8.8	190	50	19.5
4	California	9.0	276	91	40.6
5	Colorado	7.9	204	78	38.7
6	Connecticut	3.3	110	77	11.1
7	Delaware	5.9	238	72	15.8
8	Florida	15.4	335	80	31.9
9	Georgia	17.4	211	60	25.8
10	Hawaii	5.3	46	83	20.2
11	Idaho	2.6	120	54	14.2
12	Illinois	10.4	249	83	24.0
13	Indiana	7.2	113	65	21.0
14	Iowa	2.2	56	57	11.3
15	Kansas	6.0	115	66	18.0
16	Kentucky	9.7	109	52	16.3
17	Louisiana	15.4	249	66	22.2
18	Maine	2.1	83	51	7.8
19	Maryland	11.3	300	67	27.8
20	Massachusetts	4.4	149	85	16.3
21	Michigan	12.1	255	74	35.1
22	Minnesota	2.7	72	66	14.9
23	Mississippi	16.1	259	44	17.1
24	Missouri	9.0	178	70	28.2
25	Montana	6.0	109	53	16.4
26	Nebraska	4.3	102	62	16.5
27	Nevada	12.2	252	81	46.0
28	New Hampshire	2.1	57	56	9.5
29	New Jersey	7.4	159	89	18.8
30	New Mexico	11.4	285	70	32.1
31	New York	11.1	254	86	26.1
32	North Carolina	13.0	337	45	16.1
33	North Dakota	0.8	45	44	7.3

	Unnamed: 0	Murder	Assault	UrbanPop	Rape
34	Ohio	7.3	120	75	21.4
35	Oklahoma	6.6	151	68	20.0
36	Oregon	4.9	159	67	29.3
37	Pennsylvania	6.3	106	72	14.9
38	Rhode Island	3.4	174	87	8.3
39	South Carolina	14.4	279	48	22.5
40	South Dakota	3.8	86	45	12.8
41	Tennessee	13.2	188	59	26.9
42	Texas	12.7	201	80	25.5
43	Utah	3.2	120	80	22.9
44	Vermont	2.2	48	32	11.2
45	Virginia	8.5	156	63	20.7
46	Washington	4.0	145	73	26.2
47	West Virginia	5.7	81	39	9.3
48	Wisconsin	2.6	53	66	10.8
49	Wyoming	6.8	161	60	15.6

In [14]:

```
crime_data.shape
```

Out[14]:

```
(50, 5)
```

In [15]:

```
crime_data.isna().sum()
```

Out[15]:

```
Unnamed: 0    0
Murder        0
Assault       0
UrbanPop      0
Rape          0
dtype: int64
```

In [16]:

```
crime_data.dtypes
```

Out[16]:

```
Unnamed: 0     object
Murder        float64
Assault         int64
UrbanPop        int64
Rape          float64
dtype: object
```

In [28]:

```
crime_data2=crime_data.iloc[:,1:]
crime_data2
```

Out[28]:

	Murder	Assault	UrbanPop	Rape
0	13.2	236	58	21.2
1	10.0	263	48	44.5
2	8.1	294	80	31.0
3	8.8	190	50	19.5
4	9.0	276	91	40.6
5	7.9	204	78	38.7
6	3.3	110	77	11.1
7	5.9	238	72	15.8
8	15.4	335	80	31.9
9	17.4	211	60	25.8
10	5.3	46	83	20.2
11	2.6	120	54	14.2
12	10.4	249	83	24.0
13	7.2	113	65	21.0
14	2.2	56	57	11.3
15	6.0	115	66	18.0
16	9.7	109	52	16.3
17	15.4	249	66	22.2
18	2.1	83	51	7.8
19	11.3	300	67	27.8
20	4.4	149	85	16.3
21	12.1	255	74	35.1
22	2.7	72	66	14.9
23	16.1	259	44	17.1
24	9.0	178	70	28.2
25	6.0	109	53	16.4
26	4.3	102	62	16.5
27	12.2	252	81	46.0
28	2.1	57	56	9.5
29	7.4	159	89	18.8
30	11.4	285	70	32.1
31	11.1	254	86	26.1
32	13.0	337	45	16.1

	Murder	Assault	UrbanPop	Rape
33	0.8	45	44	7.3
34	7.3	120	75	21.4
35	6.6	151	68	20.0
36	4.9	159	67	29.3
37	6.3	106	72	14.9
38	3.4	174	87	8.3
39	14.4	279	48	22.5
40	3.8	86	45	12.8
41	13.2	188	59	26.9
42	12.7	201	80	25.5
43	3.2	120	80	22.9
44	2.2	48	32	11.2
45	8.5	156	63	20.7
46	4.0	145	73	26.2
47	5.7	81	39	9.3
48	2.6	53	66	10.8
49	6.8	161	60	15.6

In [29]:

```
crime_data_norm = pd.DataFrame(normalize(crime_data2),columns=crime_data2.columns)
crime_data_norm
```

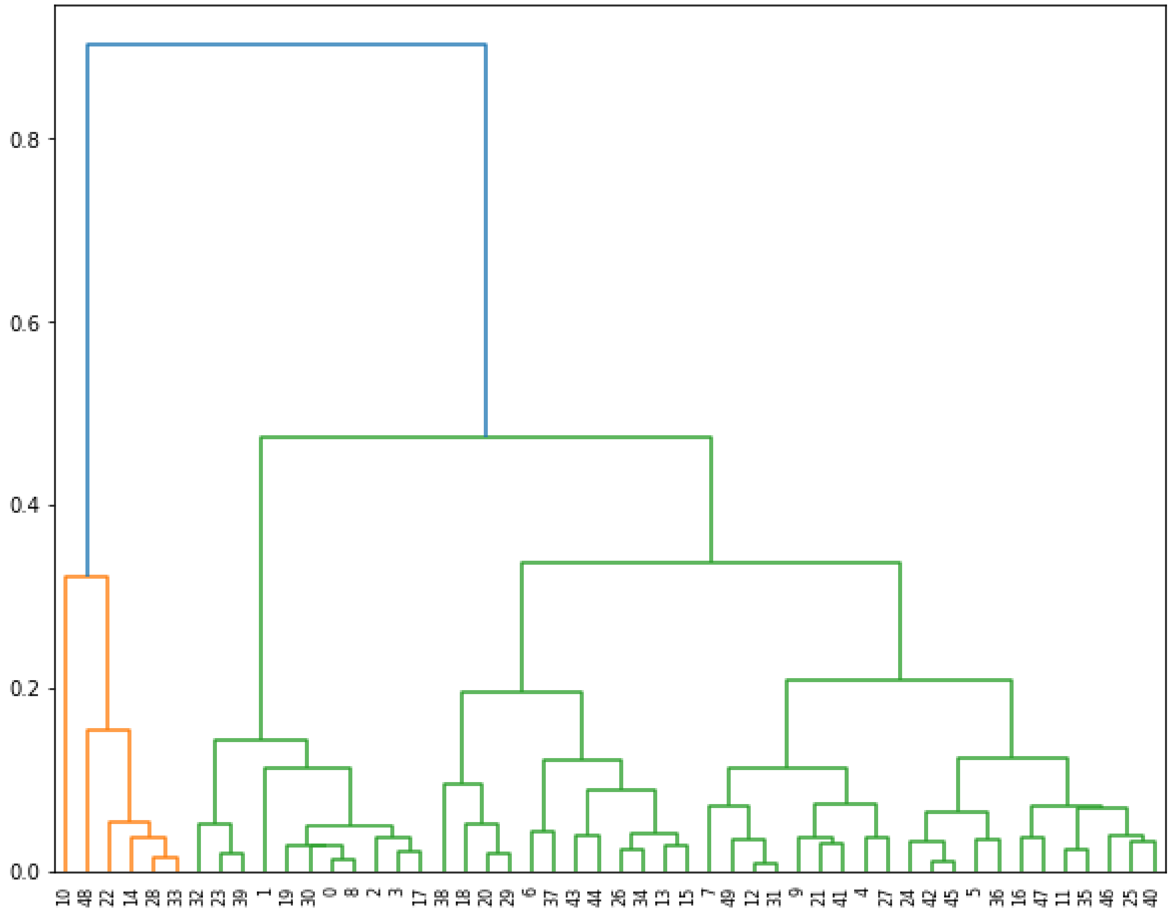
Out[29]:

	Murder	Assault	UrbanPop	Rape
0	0.054031	0.966016	0.237411	0.086778
1	0.036872	0.969739	0.176987	0.164081
2	0.026439	0.959624	0.261122	0.101185
3	0.044528	0.961392	0.252998	0.098669
4	0.030657	0.940134	0.309972	0.138295
5	0.035594	0.919142	0.351437	0.174367
6	0.024486	0.816202	0.571341	0.082362
7	0.023674	0.954965	0.288897	0.063397
8	0.044478	0.967547	0.231056	0.092134
9	0.078534	0.952332	0.270805	0.116446
10	0.054546	0.473419	0.854213	0.207893
11	0.019640	0.906483	0.407917	0.107267
12	0.039428	0.944007	0.314669	0.090989
13	0.054447	0.854521	0.491539	0.158805
14	0.027251	0.693660	0.706047	0.139971
15	0.044795	0.858568	0.492743	0.134385
16	0.079346	0.891624	0.425362	0.133335
17	0.059457	0.961347	0.254815	0.085710
18	0.021483	0.849097	0.521734	0.079795
19	0.036587	0.971339	0.216932	0.090011
20	0.025527	0.864425	0.493128	0.094565
21	0.045132	0.951126	0.276013	0.130920
22	0.027317	0.728452	0.667747	0.150749
23	0.061041	0.981958	0.166819	0.064832
24	0.046500	0.919676	0.361670	0.145701
25	0.048998	0.890131	0.432816	0.133928
26	0.035662	0.845935	0.514196	0.136842
27	0.045363	0.937005	0.301180	0.171041
28	0.026088	0.708107	0.695684	0.118018
29	0.040364	0.867284	0.485461	0.102547
30	0.038586	0.964660	0.236934	0.108651
31	0.041163	0.941927	0.318920	0.096789
32	0.038166	0.989371	0.132112	0.047267
33	0.012626	0.710188	0.694406	0.115208

	Murder	Assault	UrbanPop	Rape
34	0.050940	0.837376	0.523360	0.149332
35	0.039535	0.904523	0.407335	0.119804
36	0.027987	0.908164	0.382685	0.167353
37	0.048778	0.820702	0.557458	0.115363
38	0.017459	0.893478	0.446739	0.042620
39	0.050641	0.981163	0.168802	0.079126
40	0.038785	0.877767	0.459297	0.130644
41	0.066230	0.943274	0.296027	0.134968
42	0.058203	0.921161	0.366631	0.116864
43	0.021908	0.821558	0.547706	0.156781
44	0.037410	0.816227	0.544152	0.190453
45	0.050082	0.919147	0.371194	0.121964
46	0.024318	0.881521	0.443800	0.159282
47	0.062942	0.894442	0.430657	0.102695
48	0.030455	0.620812	0.773086	0.126505
49	0.039384	0.932482	0.347509	0.090352

In [30]:

```
plt.figure(figsize=(10,8))
dendrogram = sch.dendrogram(sch.linkage(crime_data_norm,'complete'))
```



In [31]:

```
hc = AgglomerativeClustering(n_clusters=5,affinity='euclidean',linkage='ward')
hc
```

Out[31]:

```
AgglomerativeClustering(n_clusters=5)
```

In [32]:

```
pred = hc.fit_predict(crime_data_norm)
pred
```

Out[32]:

```
array([0, 0, 0, 0, 0, 3, 1, 0, 0, 0, 4, 3, 0, 1, 2, 1, 3, 0, 1, 0, 1, 0,
       2, 0, 3, 3, 1, 0, 2, 1, 0, 0, 0, 2, 1, 3, 3, 1, 1, 0, 3, 0, 3, 1,
       1, 3, 3, 3, 2, 0], dtype=int64)
```

In [33]:

```
clusters = pd.DataFrame(pred,columns=['clusters'])
clusters
```

Out[33]:

	clusters
0	0
1	0
2	0
3	0
4	0
5	3
6	1
7	0
8	0
9	0
10	4
11	3
12	0
13	1
14	2
15	1
16	3
17	0
18	1
19	0
20	1
21	0
22	2
23	0
24	3
25	3
26	1
27	0
28	2
29	1
30	0
31	0
32	0
33	2

	clusters
34	1
35	3
36	3
37	1
38	1
39	0
40	3
41	0
42	3
43	1
44	1
45	3
46	3
47	3
48	2
49	0

In [34]:

```
crime_data['clusters']=clusters
```

In [35]:

```
crime_data
```

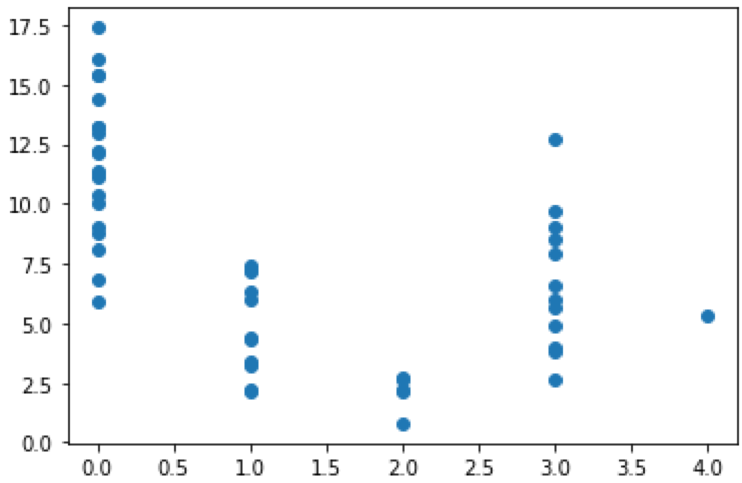
Out[35]:

	Unnamed: 0	Murder	Assault	UrbanPop	Rape	clusters
0	Alabama	13.2	236	58	21.2	0
1	Alaska	10.0	263	48	44.5	0
2	Arizona	8.1	294	80	31.0	0
3	Arkansas	8.8	190	50	19.5	0
4	California	9.0	276	91	40.6	0
5	Colorado	7.9	204	78	38.7	3
6	Connecticut	3.3	110	77	11.1	1
7	Delaware	5.9	238	72	15.8	0
8	Florida	15.4	335	80	31.9	0
9	Georgia	17.4	211	60	25.8	0
10	Hawaii	5.3	46	83	20.2	4
11	Idaho	2.6	120	54	14.2	3
12	Illinois	10.4	249	83	24.0	0
13	Indiana	7.2	113	65	21.0	1
14	Iowa	2.2	56	57	11.3	2
15	Kansas	6.0	115	66	18.0	1
16	Kentucky	9.7	109	52	16.3	3
17	Louisiana	15.4	249	66	22.2	0
18	Maine	2.1	83	51	7.8	1
19	Maryland	11.3	300	67	27.8	0
20	Massachusetts	4.4	149	85	16.3	1
21	Michigan	12.1	255	74	35.1	0
22	Minnesota	2.7	72	66	14.9	2
23	Mississippi	16.1	259	44	17.1	0
24	Missouri	9.0	178	70	28.2	3
25	Montana	6.0	109	53	16.4	3
26	Nebraska	4.3	102	62	16.5	1
27	Nevada	12.2	252	81	46.0	0
28	New Hampshire	2.1	57	56	9.5	2
29	New Jersey	7.4	159	89	18.8	1
30	New Mexico	11.4	285	70	32.1	0
31	New York	11.1	254	86	26.1	0
32	North Carolina	13.0	337	45	16.1	0
33	North Dakota	0.8	45	44	7.3	2

	Unnamed: 0	Murder	Assault	UrbanPop	Rape	clusters
34	Ohio	7.3	120	75	21.4	1
35	Oklahoma	6.6	151	68	20.0	3
36	Oregon	4.9	159	67	29.3	3
37	Pennsylvania	6.3	106	72	14.9	1
38	Rhode Island	3.4	174	87	8.3	1
39	South Carolina	14.4	279	48	22.5	0
40	South Dakota	3.8	86	45	12.8	3
41	Tennessee	13.2	188	59	26.9	0
42	Texas	12.7	201	80	25.5	3
43	Utah	3.2	120	80	22.9	1
44	Vermont	2.2	48	32	11.2	1
45	Virginia	8.5	156	63	20.7	3
46	Washington	4.0	145	73	26.2	3
47	West Virginia	5.7	81	39	9.3	3
48	Wisconsin	2.6	53	66	10.8	2
49	Wyoming	6.8	161	60	15.6	0

In [36]:

```
plt.scatter(crime_data['clusters'],crime_data['Murder'])
plt.show()
```



K-Means

In [37]:

```
from sklearn.cluster import KMeans
```

In [38]:

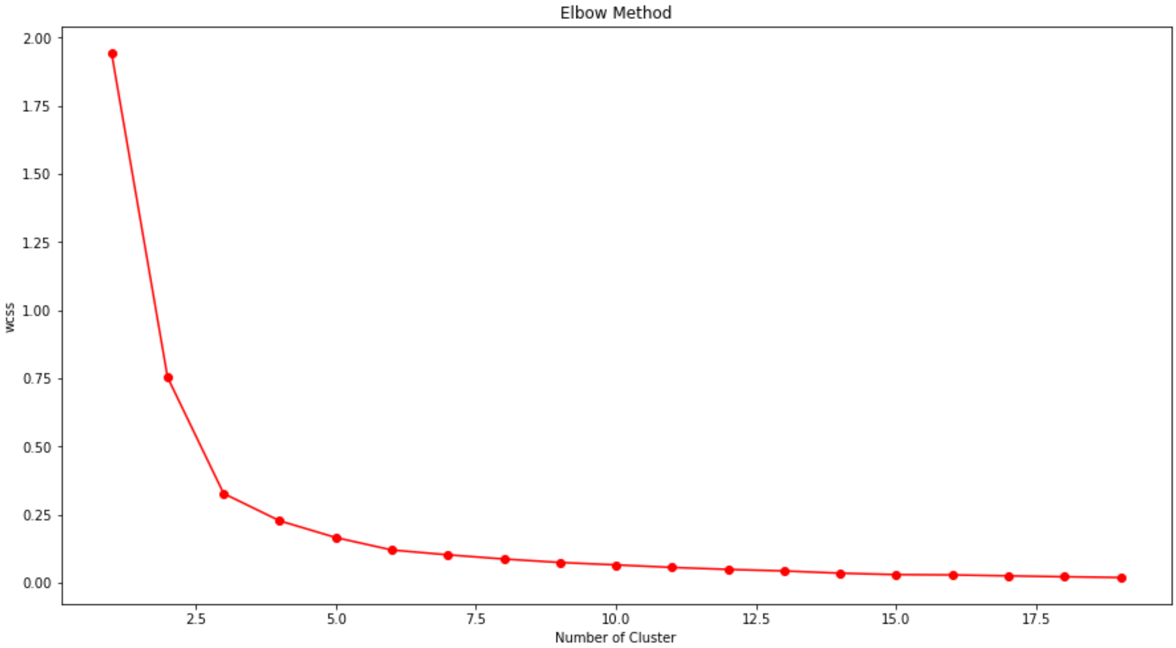
```
wcss=[]
for i in range(1,20):
    kmeans=KMeans(n_clusters=i)
    kmeans.fit(crime_data_norm)
    wcss.append(kmeans.inertia_)
    print(i,wcss)

plt.figure(figsize=(15,8))
plt.plot(range(1,20),wcss,'ro-')
plt.title('Elbow Method')
plt.xlabel('Number of Cluster')
plt.ylabel('wcss')
plt.show()
```

```
C:\Users\Asus\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:881: Us
erWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(

1 [1.9452076233849003]
2 [1.9452076233849003, 0.7540963759591796]
3 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132]
4 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.22760245765
174864]
5 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.22760245765
174864, 0.16626184516362305]
6 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.22760245765
174864, 0.16626184516362305, 0.12059896172575112]
7 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.22760245765
174864, 0.16626184516362305, 0.12059896172575112, 0.10285038064179203]
8 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.22760245765
174864, 0.16626184516362305, 0.12059896172575112, 0.10285038064179203, 0.087
50294302084376]
9 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.22760245765
174864, 0.16626184516362305, 0.12059896172575112, 0.10285038064179203, 0.087
50294302084376, 0.07476658315018699]
10 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.2276024576
5174864, 0.16626184516362305, 0.12059896172575112, 0.10285038064179203, 0.08
750294302084376, 0.07476658315018699, 0.06586485040975154]
11 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.2276024576
5174864, 0.16626184516362305, 0.12059896172575112, 0.10285038064179203, 0.08
750294302084376, 0.07476658315018699, 0.06586485040975154, 0.056615763339214
03]
12 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.2276024576
5174864, 0.16626184516362305, 0.12059896172575112, 0.10285038064179203, 0.08
750294302084376, 0.07476658315018699, 0.06586485040975154, 0.056615763339214
03, 0.04926903709664399]
13 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.2276024576
5174864, 0.16626184516362305, 0.12059896172575112, 0.10285038064179203, 0.08
750294302084376, 0.07476658315018699, 0.06586485040975154, 0.056615763339214
03, 0.04926903709664399, 0.04366649849224775]
14 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.2276024576
5174864, 0.16626184516362305, 0.12059896172575112, 0.10285038064179203, 0.08
750294302084376, 0.07476658315018699, 0.06586485040975154, 0.056615763339214
03, 0.04926903709664399, 0.04366649849224775, 0.035428075586089194]
15 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.2276024576
5174864, 0.16626184516362305, 0.12059896172575112, 0.10285038064179203, 0.08
```

```
750294302084376, 0.07476658315018699, 0.06586485040975154, 0.056615763339214
03, 0.04926903709664399, 0.04366649849224775, 0.035428075586089194, 0.030188
089870473975]
16 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.2276024576
5174864, 0.16626184516362305, 0.12059896172575112, 0.10285038064179203, 0.08
750294302084376, 0.07476658315018699, 0.06586485040975154, 0.056615763339214
03, 0.04926903709664399, 0.04366649849224775, 0.035428075586089194, 0.030188
089870473975, 0.029076212286584257]
17 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.2276024576
5174864, 0.16626184516362305, 0.12059896172575112, 0.10285038064179203, 0.08
750294302084376, 0.07476658315018699, 0.06586485040975154, 0.056615763339214
03, 0.04926903709664399, 0.04366649849224775, 0.035428075586089194, 0.030188
089870473975, 0.029076212286584257, 0.02565656054691864]
18 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.2276024576
5174864, 0.16626184516362305, 0.12059896172575112, 0.10285038064179203, 0.08
750294302084376, 0.07476658315018699, 0.06586485040975154, 0.056615763339214
03, 0.04926903709664399, 0.04366649849224775, 0.035428075586089194, 0.030188
089870473975, 0.029076212286584257, 0.02565656054691864, 0.0223421125062760
5]
19 [1.9452076233849003, 0.7540963759591796, 0.3278478050693132, 0.2276024576
5174864, 0.16626184516362305, 0.12059896172575112, 0.10285038064179203, 0.08
750294302084376, 0.07476658315018699, 0.06586485040975154, 0.056615763339214
03, 0.04926903709664399, 0.04366649849224775, 0.035428075586089194, 0.030188
089870473975, 0.029076212286584257, 0.02565656054691864, 0.0223421125062760
5, 0.01950539001009962]
```



In [39]:

```
kmeans = KMeans(n_clusters=3)
y_knn = kmeans.fit_predict(crime_data_norm)
```

In [40]:

```
clusters = pd.DataFrame(y_knn,columns=['clusters'])
clusters
```

Out[40]:

	clusters
0	1
1	1
2	1
3	1
4	1
5	1
6	2
7	1
8	1
9	1
10	0
11	2
12	1
13	2
14	0
15	2
16	2
17	1
18	2
19	1
20	2
21	1
22	0
23	1
24	1
25	2
26	2
27	1
28	0
29	2
30	1
31	1
32	1

	clusters
33	0
34	2
35	2
36	2
37	2
38	2
39	1
40	2
41	1
42	1
43	2
44	2
45	1
46	2
47	2
48	0
49	1

In [41]:

```
crime_data['clusters']=clusters
```

In [42]:

```
crime_data
```

Out[42]:

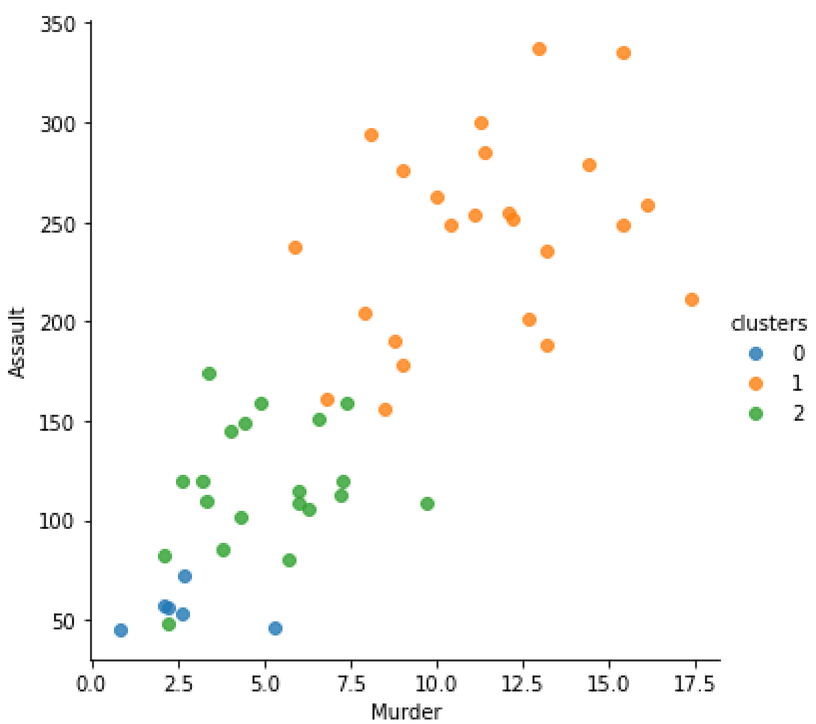
	Unnamed: 0	Murder	Assault	UrbanPop	Rape	clusters
0	Alabama	13.2	236	58	21.2	1
1	Alaska	10.0	263	48	44.5	1
2	Arizona	8.1	294	80	31.0	1
3	Arkansas	8.8	190	50	19.5	1
4	California	9.0	276	91	40.6	1
5	Colorado	7.9	204	78	38.7	1
6	Connecticut	3.3	110	77	11.1	2
7	Delaware	5.9	238	72	15.8	1
8	Florida	15.4	335	80	31.9	1
9	Georgia	17.4	211	60	25.8	1
10	Hawaii	5.3	46	83	20.2	0
11	Idaho	2.6	120	54	14.2	2
12	Illinois	10.4	249	83	24.0	1
13	Indiana	7.2	113	65	21.0	2
14	Iowa	2.2	56	57	11.3	0
15	Kansas	6.0	115	66	18.0	2
16	Kentucky	9.7	109	52	16.3	2
17	Louisiana	15.4	249	66	22.2	1
18	Maine	2.1	83	51	7.8	2
19	Maryland	11.3	300	67	27.8	1
20	Massachusetts	4.4	149	85	16.3	2
21	Michigan	12.1	255	74	35.1	1
22	Minnesota	2.7	72	66	14.9	0
23	Mississippi	16.1	259	44	17.1	1
24	Missouri	9.0	178	70	28.2	1
25	Montana	6.0	109	53	16.4	2
26	Nebraska	4.3	102	62	16.5	2
27	Nevada	12.2	252	81	46.0	1
28	New Hampshire	2.1	57	56	9.5	0
29	New Jersey	7.4	159	89	18.8	2
30	New Mexico	11.4	285	70	32.1	1
31	New York	11.1	254	86	26.1	1
32	North Carolina	13.0	337	45	16.1	1
33	North Dakota	0.8	45	44	7.3	0

	Unnamed: 0	Murder	Assault	UrbanPop	Rape	clusters
34	Ohio	7.3	120	75	21.4	2
35	Oklahoma	6.6	151	68	20.0	2
36	Oregon	4.9	159	67	29.3	2
37	Pennsylvania	6.3	106	72	14.9	2
38	Rhode Island	3.4	174	87	8.3	2
39	South Carolina	14.4	279	48	22.5	1
40	South Dakota	3.8	86	45	12.8	2
41	Tennessee	13.2	188	59	26.9	1
42	Texas	12.7	201	80	25.5	1
43	Utah	3.2	120	80	22.9	2
44	Vermont	2.2	48	32	11.2	2
45	Virginia	8.5	156	63	20.7	1
46	Washington	4.0	145	73	26.2	2
47	West Virginia	5.7	81	39	9.3	2
48	Wisconsin	2.6	53	66	10.8	0
49	Wyoming	6.8	161	60	15.6	1

In [43]:

```
sns.lmplot('Murder','Assault',data=crime_data,hue='clusters',fit_reg=False,)
plt.show()
```

```
C:\Users\Asus\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variables as keyword args: x, y. From version 0.1
2, the only valid positional argument will be `data`, and passing other argu
ments without an explicit keyword will result in an error or misinterpretati
on.
  warnings.warn(
```



DBSCAN

In [44]:

```
crime_data.head()
```

Out[44]:

	Unnamed: 0	Murder	Assault	UrbanPop	Rape	clusters
0	Alabama	13.2	236	58	21.2	1
1	Alaska	10.0	263	48	44.5	1
2	Arizona	8.1	294	80	31.0	1
3	Arkansas	8.8	190	50	19.5	1
4	California	9.0	276	91	40.6	1

In [45]:

```
x=crime_data.iloc[:,[2,3]].values
```

In [46]:

```
x.shape
```

Out[46]:

```
(50, 2)
```

In [47]:

```
from sklearn.cluster import DBSCAN
db=DBSCAN(eps=3, min_samples=5,metric='euclidean')
```

In [48]:

```
model=db.fit(x)
```

In [49]:

```
model
```

Out[49]:

```
DBSCAN(eps=3)
```

In [50]:

```
label = model.labels_
```

In [51]:

```
label
```

Out[51]:

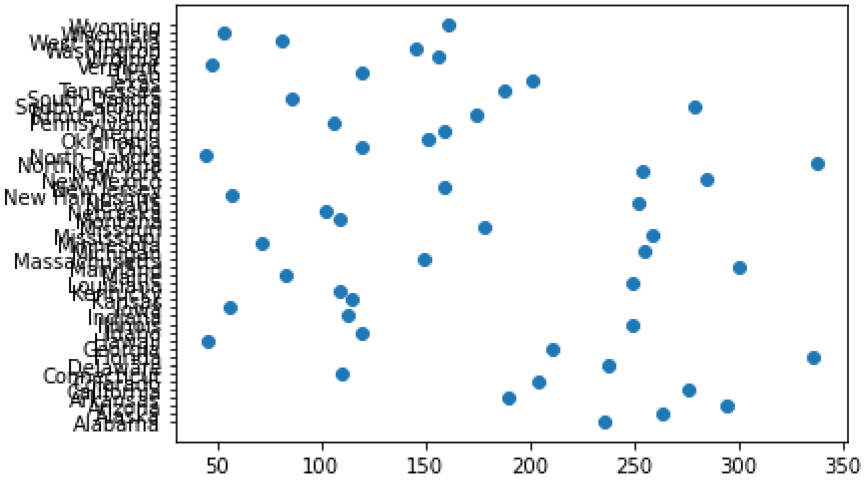
```
array([-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
       -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
       -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],
      dtype=int64)
```

In [52]:

```
plt.scatter(y=crime_data['Unnamed: 0'], x=crime_data['Assault'])
```

Out[52]:

```
<matplotlib.collections.PathCollection at 0x156ca98cdc0>
```



In [ ]: