In [1]:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sn
from sklearn.cluster import KMeans
import scipy.cluster.hierarchy as sch
from sklearn.cluster import AgglomerativeClustering
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

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

Out[2]:

	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
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

	Unnamed: 0	Murder	Assault	UrbanPop	Rape
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 [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  50 non-null     object 
 1   Murder      50 non-null     float64
 2   Assault     50 non-null     int64  
 3   UrbanPop    50 non-null     int64  
 4   Rape        50 non-null     float64
dtypes: float64(2), int64(2), object(1)
memory usage: 2.1+ KB
```

In [4]:
```
crime=data.drop("Unnamed: 0",axis=1)
crime
```

Out[4]:

	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
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

	Murder	Assault	UrbanPop	Rape
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 [5]:

```
def norm_func(i):
    x = (i-i.min())/(i.max()-i.min())
    return (x)
```

In [6]:

```
df_norm = norm_func(crime.iloc[:,:])
df_norm
```

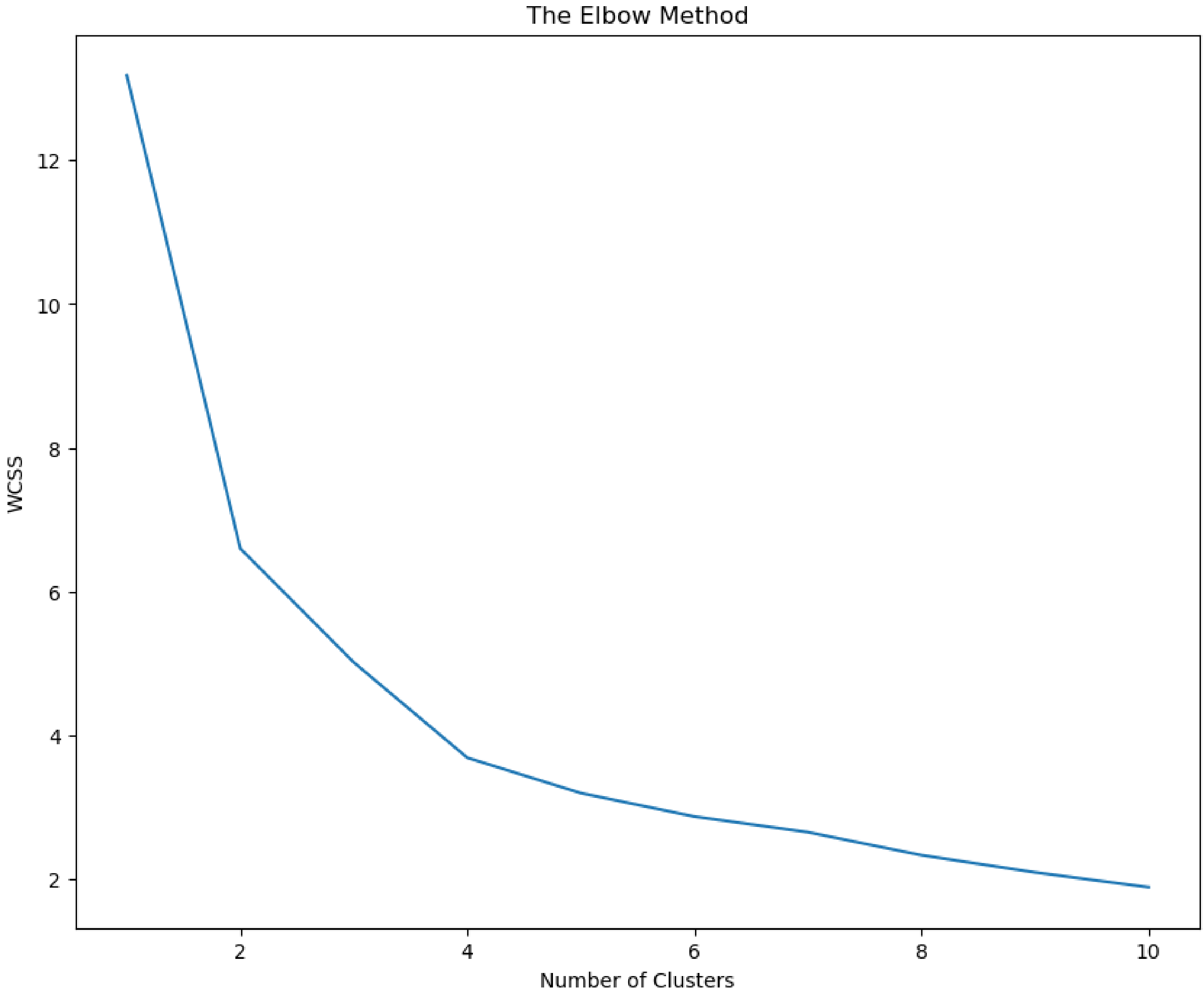
Out[6]:

	Murder	Assault	UrbanPop	Rape
0	0.746988	0.654110	0.440678	0.359173
1	0.554217	0.746575	0.271186	0.961240
2	0.439759	0.852740	0.813559	0.612403
3	0.481928	0.496575	0.305085	0.315245
4	0.493976	0.791096	1.000000	0.860465
5	0.427711	0.544521	0.779661	0.811370
6	0.150602	0.222603	0.762712	0.098191
7	0.307229	0.660959	0.677966	0.219638
8	0.879518	0.993151	0.813559	0.635659
9	1.000000	0.568493	0.474576	0.478036
10	0.271084	0.003425	0.864407	0.333333
11	0.108434	0.256849	0.372881	0.178295
12	0.578313	0.698630	0.864407	0.431525
13	0.385542	0.232877	0.559322	0.354005
14	0.084337	0.037671	0.423729	0.103359
15	0.313253	0.239726	0.576271	0.276486
16	0.536145	0.219178	0.338983	0.232558
17	0.879518	0.698630	0.576271	0.385013
18	0.078313	0.130137	0.322034	0.012920
19	0.632530	0.873288	0.593220	0.529716
20	0.216867	0.356164	0.898305	0.232558
21	0.680723	0.719178	0.711864	0.718346
22	0.114458	0.092466	0.576271	0.196382
23	0.921687	0.732877	0.203390	0.253230
24	0.493976	0.455479	0.644068	0.540052
25	0.313253	0.219178	0.355932	0.235142
26	0.210843	0.195205	0.508475	0.237726
27	0.686747	0.708904	0.830508	1.000000
28	0.078313	0.041096	0.406780	0.056848
29	0.397590	0.390411	0.966102	0.297158
30	0.638554	0.821918	0.644068	0.640827
31	0.620482	0.715753	0.915254	0.485788
32	0.734940	1.000000	0.220339	0.227390
33	0.000000	0.000000	0.203390	0.000000
34	0.391566	0.256849	0.728814	0.364341
35	0.349398	0.363014	0.610169	0.328165
36	0.246988	0.390411	0.593220	0.568475
37	0.331325	0.208904	0.677966	0.196382
38	0.156627	0.441781	0.932203	0.025840

	Murder	Assault	UrbanPop	Rape
39	0.819277	0.801370	0.271186	0.392765
40	0.180723	0.140411	0.220339	0.142119
41	0.746988	0.489726	0.457627	0.506460
42	0.716867	0.534247	0.813559	0.470284
43	0.144578	0.256849	0.813559	0.403101
44	0.084337	0.010274	0.000000	0.100775
45	0.463855	0.380137	0.525424	0.346253
46	0.192771	0.342466	0.694915	0.488372
47	0.295181	0.123288	0.118644	0.051680
48	0.108434	0.027397	0.576271	0.090439
49	0.361446	0.397260	0.474576	0.214470

In [7]:

```
fig = plt.figure(figsize=(10, 8))
WCSS = []
for i in range(1, 11):
    clf = KMeans(n_clusters=i)
    clf.fit(df_norm)
    WCSS.append(clf.inertia_)
plt.plot(range(1, 11), WCSS)
plt.title('The Elbow Method')
plt.ylabel('WCSS')
plt.xlabel('Number of Clusters')
plt.show()
```




```
In [8]:  clf = KMeans(n_clusters=5)
          y_kmeans = clf.fit_predict(df_norm)

In [11]: y_kmeans
          clf.labels_

Out[11]: array([1, 1, 2, 3, 2, 2, 4, 3, 2, 1, 4, 0, 2, 3, 0, 3, 3, 1, 0, 2, 4, 2,
                 0, 1, 3, 0, 0, 2, 0, 4, 2, 2, 1, 0, 3, 3, 3, 4, 4, 1, 0, 1, 2, 4,
                 0, 3, 3, 0, 0, 3])

In [12]: y_kmeans

Out[12]: array([1, 1, 2, 3, 2, 2, 4, 3, 2, 1, 4, 0, 2, 3, 0, 3, 3, 1, 0, 2, 4, 2,
                 0, 1, 3, 0, 0, 2, 0, 4, 2, 2, 1, 0, 3, 3, 3, 4, 4, 1, 0, 1, 2, 4,
                 0, 3, 3, 0, 0, 3])

In [13]: clf.cluster_centers_

Out[13]: array([[0.13805221, 0.10616438, 0.34039548, 0.1171404 ],
                 [0.80045181, 0.7114726 , 0.36440678, 0.44541344],
                 [0.6177437 , 0.75031133, 0.798151  , 0.65421658],
                 [0.37700803, 0.36957763, 0.56073446, 0.35400517],
                 [0.2383821 , 0.268591  , 0.84503632, 0.2266519 ]])

In [14]: clf.inertia_

Out[14]: 3.2250722560447906
```

In [15]:

```
md=pd.Series(y_kmeans)  # converting numpy array into pandas series object
crime['clust']=md # creating a  new column and assigning it to new column
crime
```

Loading [MathJax]/extensions/Safe.js

Out[15]:

	Murder	Assault	UrbanPop	Rape	clust
0	13.2	236	58	21.2	1
1	10.0	263	48	44.5	1
2	8.1	294	80	31.0	2
3	8.8	190	50	19.5	3
4	9.0	276	91	40.6	2
5	7.9	204	78	38.7	2
6	3.3	110	77	11.1	4
7	5.9	238	72	15.8	3
8	15.4	335	80	31.9	2
9	17.4	211	60	25.8	1
10	5.3	46	83	20.2	4
11	2.6	120	54	14.2	0
12	10.4	249	83	24.0	2
13	7.2	113	65	21.0	3
14	2.2	56	57	11.3	0
15	6.0	115	66	18.0	3
16	9.7	109	52	16.3	3
17	15.4	249	66	22.2	1
18	2.1	83	51	7.8	0
19	11.3	300	67	27.8	2
20	4.4	149	85	16.3	4
21	12.1	255	74	35.1	2
22	2.7	72	66	14.9	0
23	16.1	259	44	17.1	1
24	9.0	178	70	28.2	3
25	6.0	109	53	16.4	0
26	4.3	102	62	16.5	0
27	12.2	252	81	46.0	2
28	2.1	57	56	9.5	0
29	7.4	159	89	18.8	4
30	11.4	285	70	32.1	2
31	11.1	254	86	26.1	2
32	13.0	337	45	16.1	1
33	0.8	45	44	7.3	0
34	7.3	120	75	21.4	3
35	6.6	151	68	20.0	3
36	4.9	159	67	29.3	3
37	6.3	106	72	14.9	4
38	3.4	174	87	8.3	4

	Murder	Assault	UrbanPop	Rape	clust
39	14.4	279	48	22.5	1
40	3.8	86	45	12.8	0
41	13.2	188	59	26.9	1
42	12.7	201	80	25.5	2
43	3.2	120	80	22.9	4
44	2.2	48	32	11.2	0
45	8.5	156	63	20.7	3
46	4.0	145	73	26.2	3
47	5.7	81	39	9.3	0
48	2.6	53	66	10.8	0
49	6.8	161	60	15.6	3

In [16]:
```
crime.groupby(crime.clust).mean()
```

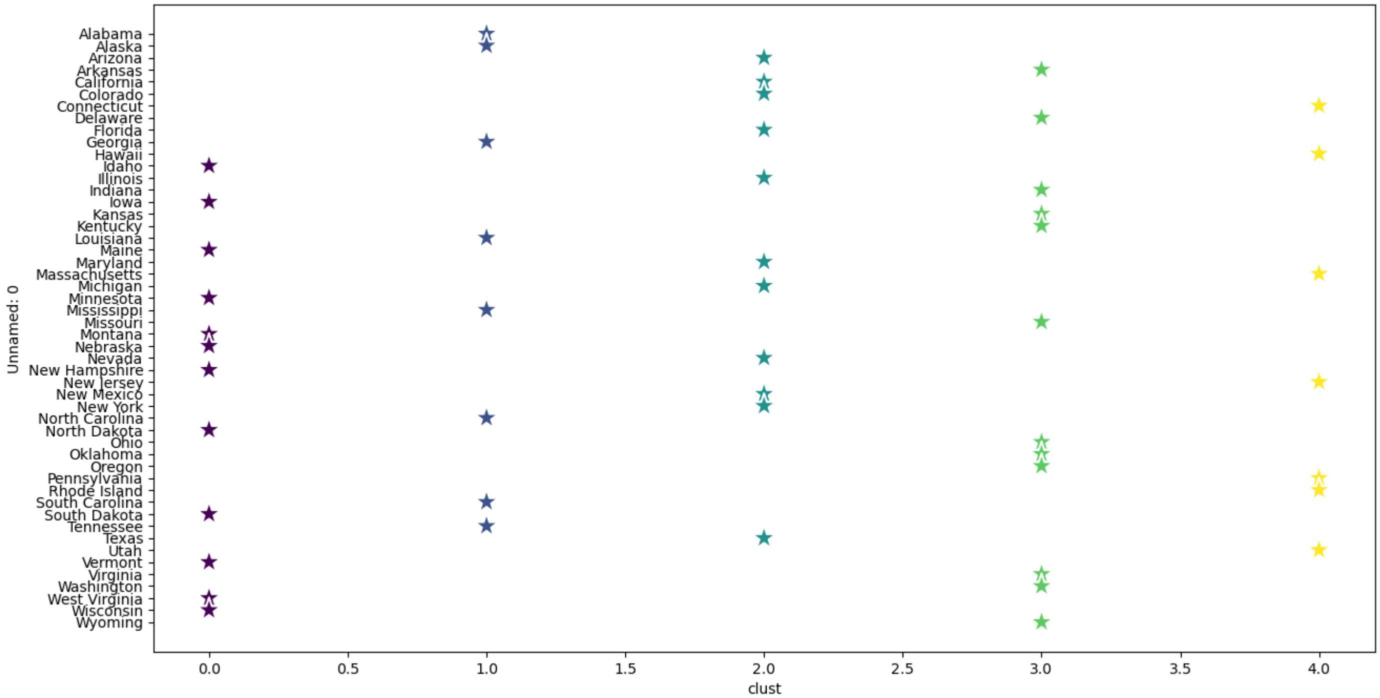
Out[16]:

	Murder	Assault	UrbanPop	Rape
clust				
0	3.091667	76.000000	52.083333	11.833333
1	14.087500	252.750000	53.500000	24.537500
2	11.054545	264.090909	79.090909	32.618182
3	7.058333	152.916667	65.083333	21.000000
4	4.757143	123.428571	81.857143	16.071429

In [17]:
```
WCSS
```

Out[17]:
```
[13.184122550256443,
 6.596893867946198,
 5.010878493006418,
 3.6834561535859134,
 3.1911357068589448,
 2.8653579551726827,
 2.647067978181021,
 2.3268788028371215,
 2.087607764352812,
 1.880754331689618]
```

In [18]:
```
plt.figure(figsize=(15,8))
sn.scatterplot(crime['clust'],data['Unnamed: 0'],c=clf.labels_,s=300,marker='*')
plt.show();
```



In [19]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import DBSCAN
```

In [21]:
```
crime
```

Out[21]:

	Murder	Assault	UrbanPop	Rape	clust
0	13.2	236	58	21.2	1
1	10.0	263	48	44.5	1
2	8.1	294	80	31.0	2
3	8.8	190	50	19.5	3
4	9.0	276	91	40.6	2
5	7.9	204	78	38.7	2
6	3.3	110	77	11.1	4
7	5.9	238	72	15.8	3
8	15.4	335	80	31.9	2
9	17.4	211	60	25.8	1
10	5.3	46	83	20.2	4
11	2.6	120	54	14.2	0
12	10.4	249	83	24.0	2
13	7.2	113	65	21.0	3
14	2.2	56	57	11.3	0
15	6.0	115	66	18.0	3
16	9.7	109	52	16.3	3
17	15.4	249	66	22.2	1
18	2.1	83	51	7.8	0
19	11.3	300	67	27.8	2
20	4.4	149	85	16.3	4
21	12.1	255	74	35.1	2
22	2.7	72	66	14.9	0
23	16.1	259	44	17.1	1
24	9.0	178	70	28.2	3
25	6.0	109	53	16.4	0
26	4.3	102	62	16.5	0
27	12.2	252	81	46.0	2
28	2.1	57	56	9.5	0
29	7.4	159	89	18.8	4
30	11.4	285	70	32.1	2
31	11.1	254	86	26.1	2
32	13.0	337	45	16.1	1
33	0.8	45	44	7.3	0
34	7.3	120	75	21.4	3
35	6.6	151	68	20.0	3
36	4.9	159	67	29.3	3
37	6.3	106	72	14.9	4
38	3.4	174	87	8.3	4

	Murder	Assault	UrbanPop	Rape	clust
39	14.4	279	48	22.5	1
40	3.8	86	45	12.8	0
41	13.2	188	59	26.9	1
42	12.7	201	80	25.5	2
43	3.2	120	80	22.9	4
44	2.2	48	32	11.2	0
45	8.5	156	63	20.7	3
46	4.0	145	73	26.2	3
47	5.7	81	39	9.3	0
48	2.6	53	66	10.8	0
49	6.8	161	60	15.6	3

In [22]:

```
array=crime.values
array
```

Out[22]:
```
array([[ 13.2, 236. ,  58. ,  21.2,   1. ],
       [ 10. , 263. ,  48. ,  44.5,   1. ],
       [  8.1, 294. ,  80. ,  31. ,   2. ],
       [  8.8, 190. ,  50. ,  19.5,   3. ],
       [  9. , 276. ,  91. ,  40.6,   2. ],
       [  7.9, 204. ,  78. ,  38.7,   2. ],
       [  3.3, 110. ,  77. ,  11.1,   4. ],
       [  5.9, 238. ,  72. ,  15.8,   3. ],
       [ 15.4, 335. ,  80. ,  31.9,   2. ],
       [ 17.4, 211. ,  60. ,  25.8,   1. ],
       [  5.3,  46. ,  83. ,  20.2,   4. ],
       [  2.6, 120. ,  54. ,  14.2,   0. ],
       [ 10.4, 249. ,  83. ,  24. ,   2. ],
       [  7.2, 113. ,  65. ,  21. ,   3. ],
       [  2.2,  56. ,  57. ,  11.3,   0. ],
       [  6. , 115. ,  66. ,  18. ,   3. ],
       [  9.7, 109. ,  52. ,  16.3,   3. ],
       [ 15.4, 249. ,  66. ,  22.2,   1. ],
       [  2.1,  83. ,  51. ,   7.8,   0. ],
       [ 11.3, 300. ,  67. ,  27.8,   2. ],
       [  4.4, 149. ,  85. ,  16.3,   4. ],
       [ 12.1, 255. ,  74. ,  35.1,   2. ],
       [  2.7,  72. ,  66. ,  14.9,   0. ],
       [ 16.1, 259. ,  44. ,  17.1,   1. ],
       [  9. , 178. ,  70. ,  28.2,   3. ],
       [  6. , 109. ,  53. ,  16.4,   0. ],
       [  4.3, 102. ,  62. ,  16.5,   0. ],
       [ 12.2, 252. ,  81. ,  46. ,   2. ],
       [  2.1,  57. ,  56. ,   9.5,   0. ],
       [  7.4, 159. ,  89. ,  18.8,   4. ],
       [ 11.4, 285. ,  70. ,  32.1,   2. ],
       [ 11.1, 254. ,  86. ,  26.1,   2. ],
       [ 13. , 337. ,  45. ,  16.1,   1. ],
       [  0.8,  45. ,  44. ,   7.3,   0. ],
       [  7.3, 120. ,  75. ,  21.4,   3. ],
       [  6.6, 151. ,  68. ,  20. ,   3. ],
       [  4.9, 159. ,  67. ,  29.3,   3. ],
       [  6.3, 106. ,  72. ,  14.9,   4. ],
       [  3.4, 174. ,  87. ,   8.3,   4. ],
       [ 14.4, 279. ,  48. ,  22.5,   1. ],
       [  3.8,  86. ,  45. ,  12.8,   0. ],
       [ 13.2, 188. ,  59. ,  26.9,   1. ],
       [ 12.7, 201. ,  80. ,  25.5,   2. ],
       [  3.2, 120. ,  80. ,  22.9,   4. ],
       [  2.2,  48. ,  32. ,  11.2,   0. ],
       [  8.5, 156. ,  63. ,  20.7,   3. ],
       [  4. , 145. ,  73. ,  26.2,   3. ],
       [  5.7,  81. ,  39. ,   9.3,   0. ],
       [  2.6,  53. ,  66. ,  10.8,   0. ],
       [  6.8, 161. ,  60. ,  15.6,   3. ]])
```

In [23]:
```
stscaler = StandardScaler().fit(array)
X = stscaler.transform(array)
X
```

Out[23]:
```
array([[ 1.25517927,  0.79078716, -0.52619514, -0.00345116, -0.63748035],
       [ 0.51301858,  1.11805959, -1.22406668,  2.50942392, -0.63748035],
       [ 0.07236067,  1.49381682,  1.00912225,  1.05346626,  0.08692914],
       [ 0.23470832,  0.23321191, -1.08449238, -0.18679398,  0.81133862],
       [ 0.28109336,  1.2756352 ,  1.77678094,  2.08881393,  0.08692914],
       [ 0.02597562,  0.40290872,  0.86954794,  1.88390137,  0.08692914],
       [-1.04088037, -0.73648418,  0.79976079, -1.09272319,  1.53574811],
       [-0.43787481,  0.81502956,  0.45082502, -0.58583422,  0.81133862],
       [ 1.76541475,  1.99078607,  1.00912225,  1.1505301 ,  0.08692914],
       [ 2.22926518,  0.48775713, -0.38662083,  0.49265293, -0.63748035],
       [-0.57702994, -1.51224105,  1.21848371, -0.11129987,  1.53574811],
       [-1.20322802, -0.61527217, -0.80534376, -0.75839217, -1.36188983],
       [ 0.60578867,  0.94836277,  1.21848371,  0.29852525,  0.08692914],
       [-0.13637203, -0.70012057, -0.03768506, -0.0250209 ,  0.81133862],
       [-1.29599811, -1.39102904, -0.5959823 , -1.07115345, -1.36188983],
       [-0.41468229, -0.67587817,  0.03210209, -0.34856705,  0.81133862],
       [ 0.44344101, -0.74860538, -0.94491807, -0.53190987,  0.81133862],
       [ 1.76541475,  0.94836277,  0.03210209,  0.10439756, -0.63748035],
       [-1.31919063, -1.06375661, -1.01470522, -1.44862395, -1.36188983],
       [ 0.81452136,  1.56654403,  0.10188925,  0.70835037,  0.08692914],
       [-0.78576263, -0.26375734,  1.35805802, -0.53190987,  1.53574811],
       [ 1.00006153,  1.02108998,  0.59039932,  1.49564599,  0.08692914],
       [-1.1800355 , -1.19708982,  0.03210209, -0.68289807, -1.36188983],
       [ 1.9277624 ,  1.06957478, -1.5032153 , -0.44563089, -0.63748035],
       [ 0.28109336,  0.0877575 ,  0.31125071,  0.75148985,  0.81133862],
       [-0.41468229, -0.74860538, -0.87513091, -0.521125  , -1.36188983],
       [-0.80895515, -0.83345379, -0.24704653, -0.51034012, -1.36188983],
       [ 1.02325405,  0.98472638,  1.0789094 ,  2.671197  ,  0.08692914],
       [-1.31919063, -1.37890783, -0.66576945, -1.26528114, -1.36188983],
       [-0.08998698, -0.14254532,  1.63720664, -0.26228808,  1.53574811],
       [ 0.83771388,  1.38472601,  0.31125071,  1.17209984,  0.08692914],
       [ 0.76813632,  1.00896878,  1.42784517,  0.52500755,  0.08692914],
       [ 1.20879423,  2.01502847, -1.43342815, -0.55347961, -0.63748035],
       [-1.62069341, -1.52436225, -1.5032153 , -1.50254831, -1.36188983],
       [-0.11317951, -0.61527217,  0.66018648,  0.01811858,  0.81133862],
       [-0.27552716, -0.23951493,  0.1716764 , -0.13286962,  0.81133862],
       [-0.66980002, -0.14254532,  0.10188925,  0.87012344,  0.81133862],
       [-0.34510472, -0.78496898,  0.45082502, -0.68289807,  1.53574811],
       [-1.01768785,  0.03927269,  1.49763233, -1.39469959,  1.53574811],
       [ 1.53348953,  1.3119988 , -1.22406668,  0.13675217, -0.63748035],
       [-0.92491776, -1.027393  , -1.43342815, -0.90938037, -1.36188983],
       [ 1.25517927,  0.20896951, -0.45640799,  0.61128652, -0.63748035],
       [ 1.13921666,  0.36654512,  1.00912225,  0.46029832,  0.08692914],
       [-1.06407289, -0.61527217,  1.00912225,  0.17989166,  1.53574811],
       [-1.29599811, -1.48799864, -2.34066115, -1.08193832, -1.36188983],
       [ 0.16513075, -0.17890893, -0.17725937, -0.05737552,  0.81133862],
       [-0.87853272, -0.31224214,  0.52061217,  0.53579242,  0.81133862],
       [-0.48425985, -1.08799901, -1.85215107, -1.28685088, -1.36188983],
       [-1.20322802, -1.42739264,  0.03210209, -1.1250778 , -1.36188983],
       [-0.22914211, -0.11830292, -0.38662083, -0.60740397,  0.81133862]])
```

In [24]:
```
dbscan = DBSCAN(eps=1.25, min_samples=5)
dbscan.fit(X)
```

Out[24]:
```
DBSCAN(eps=1.25)
```

In [30]:
```
dbscan.labels_
```

Out[30]:
```
array([ 0, -1,  1,  3, -1,  1,  3,  3, -1,  0,  3,  2,  1,  3,  2,  3,  3,
        0,  2,  1,  3,  1,  2, -1,  3,  2,  2, -1,  2,  3,  1,  1, -1,  2,
        3,  3,  3,  3,  3,  0,  2,  0, -1,  3,  2,  3,  3,  2,  2,  3],
      dtype=int64)
```

In [29]:
```
c=pd.DataFrame(dbscan.labels_,columns=['cluster'])
```

In [31]:
```
c
pd.set_option("display.max_rows", None)
```

In [32]:
```
c
```

Out[32]:

	cluster
0	0
1	-1
2	1
3	3
4	-1
5	1
6	3
7	3
8	-1
9	0
10	3
11	2
12	1
13	3
14	2
15	3
16	3
17	0
18	2
19	1
20	3
21	1
22	2
23	-1
24	3
25	2
26	2
27	-1
28	2
29	3
30	1
31	1
32	-1
33	2
34	3
35	3
36	3
37	3
38	3

	cluster
39	0
40	2
41	0
42	-1
43	3
44	2
45	3
46	3
47	2
48	2
49	3

In [33]:
```
df = pd.concat([data,c],axis=1)
df
```

Out[33]:

	Unnamed: 0	Murder	Assault	UrbanPop	Rape	cluster
0	Alabama	13.2	236	58	21.2	0
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	3
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	3
7	Delaware	5.9	238	72	15.8	3
8	Florida	15.4	335	80	31.9	-1
9	Georgia	17.4	211	60	25.8	0
10	Hawaii	5.3	46	83	20.2	3
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	3
14	Iowa	2.2	56	57	11.3	2
15	Kansas	6.0	115	66	18.0	3
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	2
19	Maryland	11.3	300	67	27.8	1
20	Massachusetts	4.4	149	85	16.3	3
21	Michigan	12.1	255	74	35.1	1
22	Minnesota	2.7	72	66	14.9	2
23	Mississippi	16.1	259	44	17.1	-1
24	Missouri	9.0	178	70	28.2	3
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	2
29	New Jersey	7.4	159	89	18.8	3
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	2
34	Ohio	7.3	120	75	21.4	3
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	3
38	Rhode Island	3.4	174	87	8.3	3

	Unnamed: 0	Murder	Assault	UrbanPop	Rape	cluster
39	South Carolina	14.4	279	48	22.5	0
40	South Dakota	3.8	86	45	12.8	2
41	Tennessee	13.2	188	59	26.9	0
42	Texas	12.7	201	80	25.5	-1
43	Utah	3.2	120	80	22.9	3
44	Vermont	2.2	48	32	11.2	2
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	2
48	Wisconsin	2.6	53	66	10.8	2
49	Wyoming	6.8	161	60	15.6	3

In [34]:
```
d1=dbscan.labels_
d1
```

Out[34]:
```
array([ 0, -1,  1,  3, -1,  1,  3,  3, -1,  0,  3,  2,  1,  3,  2,  3,  3,
        0,  2,  1,  3,  1,  2, -1,  3,  2,  2, -1,  2,  3,  1,  1, -1,  2,
        3,  3,  3,  3,  3,  0,  2,  0, -1,  3,  2,  3,  3,  2,  2,  3],
      dtype=int64)
```

In [35]:
```
import sklearn
sklearn.metrics.silhouette_score(X, d1)
```

Out[35]:
```
0.34840326387500054
```

In [36]:
```
from sklearn.cluster import KMeans
clf = KMeans(n_clusters=5)
y_kmeans = clf.fit_predict(X)
```

In [37]:
```
y_kmeans
```

Out[37]:
```
array([4, 0, 0, 1, 0, 0, 3, 1, 0, 4, 3, 2, 0, 1, 2, 1, 1, 4, 2, 0, 3, 0,
       2, 4, 1, 2, 2, 0, 2, 3, 0, 0, 4, 2, 1, 1, 1, 3, 3, 4, 2, 4, 0, 3,
       2, 1, 1, 2, 2, 1])
```

In [38]:
```
cl1=pd.DataFrame(y_kmeans,columns=['Kcluster'])
cl1
```

Out[38]:

	Kcluster
0	4
1	0
2	0
3	1
4	0
5	0
6	3
7	1
8	0
9	4
10	3
11	2
12	0
13	1
14	2
15	1
16	1
17	4
18	2
19	0
20	3
21	0
22	2
23	4
24	1
25	2
26	2
27	0
28	2
29	3
30	0
31	0
32	4
33	2
34	1
35	1
36	1
37	3
38	3

	Kcluster
39	4
40	2
41	4
42	0
43	3
44	2
45	1
46	1
47	2
48	2
49	1

In [39]:
```
df1 = pd.concat([df,cl1],axis=1)
df1
```

Out[39]:

	Unnamed: 0	Murder	Assault	UrbanPop	Rape	cluster	Kcluster
0	Alabama	13.2	236	58	21.2	0	4
1	Alaska	10.0	263	48	44.5	-1	0
2	Arizona	8.1	294	80	31.0	1	0
3	Arkansas	8.8	190	50	19.5	3	1
4	California	9.0	276	91	40.6	-1	0
5	Colorado	7.9	204	78	38.7	1	0
6	Connecticut	3.3	110	77	11.1	3	3
7	Delaware	5.9	238	72	15.8	3	1
8	Florida	15.4	335	80	31.9	-1	0
9	Georgia	17.4	211	60	25.8	0	4
10	Hawaii	5.3	46	83	20.2	3	3
11	Idaho	2.6	120	54	14.2	2	2
12	Illinois	10.4	249	83	24.0	1	0
13	Indiana	7.2	113	65	21.0	3	1
14	Iowa	2.2	56	57	11.3	2	2
15	Kansas	6.0	115	66	18.0	3	1
16	Kentucky	9.7	109	52	16.3	3	1
17	Louisiana	15.4	249	66	22.2	0	4
18	Maine	2.1	83	51	7.8	2	2
19	Maryland	11.3	300	67	27.8	1	0
20	Massachusetts	4.4	149	85	16.3	3	3
21	Michigan	12.1	255	74	35.1	1	0
22	Minnesota	2.7	72	66	14.9	2	2
23	Mississippi	16.1	259	44	17.1	-1	4
24	Missouri	9.0	178	70	28.2	3	1
25	Montana	6.0	109	53	16.4	2	2
26	Nebraska	4.3	102	62	16.5	2	2
27	Nevada	12.2	252	81	46.0	-1	0
28	New Hampshire	2.1	57	56	9.5	2	2
29	New Jersey	7.4	159	89	18.8	3	3
30	New Mexico	11.4	285	70	32.1	1	0
31	New York	11.1	254	86	26.1	1	0
32	North Carolina	13.0	337	45	16.1	-1	4
33	North Dakota	0.8	45	44	7.3	2	2
34	Ohio	7.3	120	75	21.4	3	1
35	Oklahoma	6.6	151	68	20.0	3	1
36	Oregon	4.9	159	67	29.3	3	1
37	Pennsylvania	6.3	106	72	14.9	3	3
38	Rhode Island	3.4	174	87	8.3	3	3

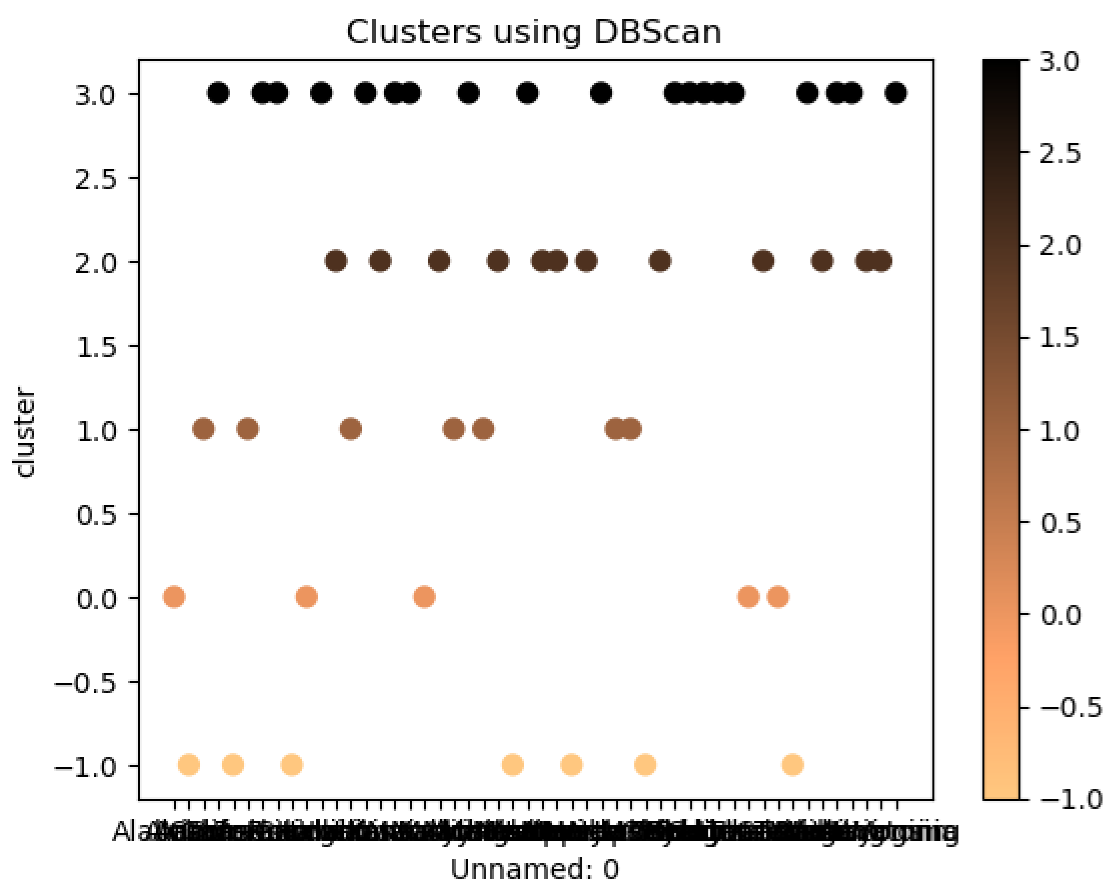
	Unnamed: 0	Murder	Assault	UrbanPop	Rape	cluster	Kcluster
39	South Carolina	14.4	279	48	22.5	0	4
40	South Dakota	3.8	86	45	12.8	2	2
41	Tennessee	13.2	188	59	26.9	0	4
42	Texas	12.7	201	80	25.5	-1	0
43	Utah	3.2	120	80	22.9	3	3
44	Vermont	2.2	48	32	11.2	2	2
45	Virginia	8.5	156	63	20.7	3	1
46	Washington	4.0	145	73	26.2	3	1
47	West Virginia	5.7	81	39	9.3	2	2
48	Wisconsin	2.6	53	66	10.8	2	2
49	Wyoming	6.8	161	60	15.6	3	1

In [40]:
```
sklearn.metrics.silhouette_score(X, y_kmeans)
```

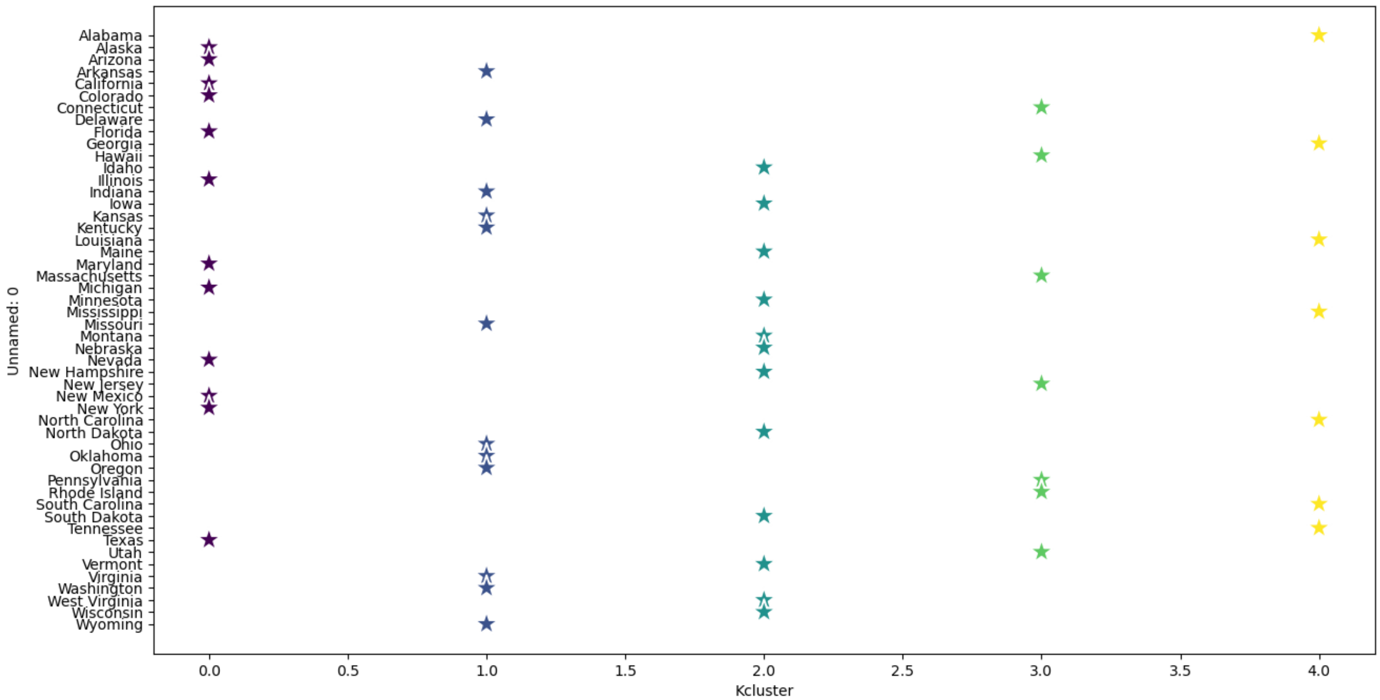
Out[40]:
```
0.41020559566840903
```

In [41]:
```
df.plot(x="Unnamed: 0",y ="cluster",c=dbscan.labels_ ,kind="scatter",s=50 ,cmap=plt.cm.c
plt.title('Clusters using DBScan')
```

Out[41]:
```
Text(0.5, 1.0, 'Clusters using DBScan')
```



In [42]:
```
plt.figure(figsize=(15,8))
sn.scatterplot(df1['Kcluster'],df1['Unnamed: 0'],c=clf.labels_,s=300,marker='*')
plt.show();
```

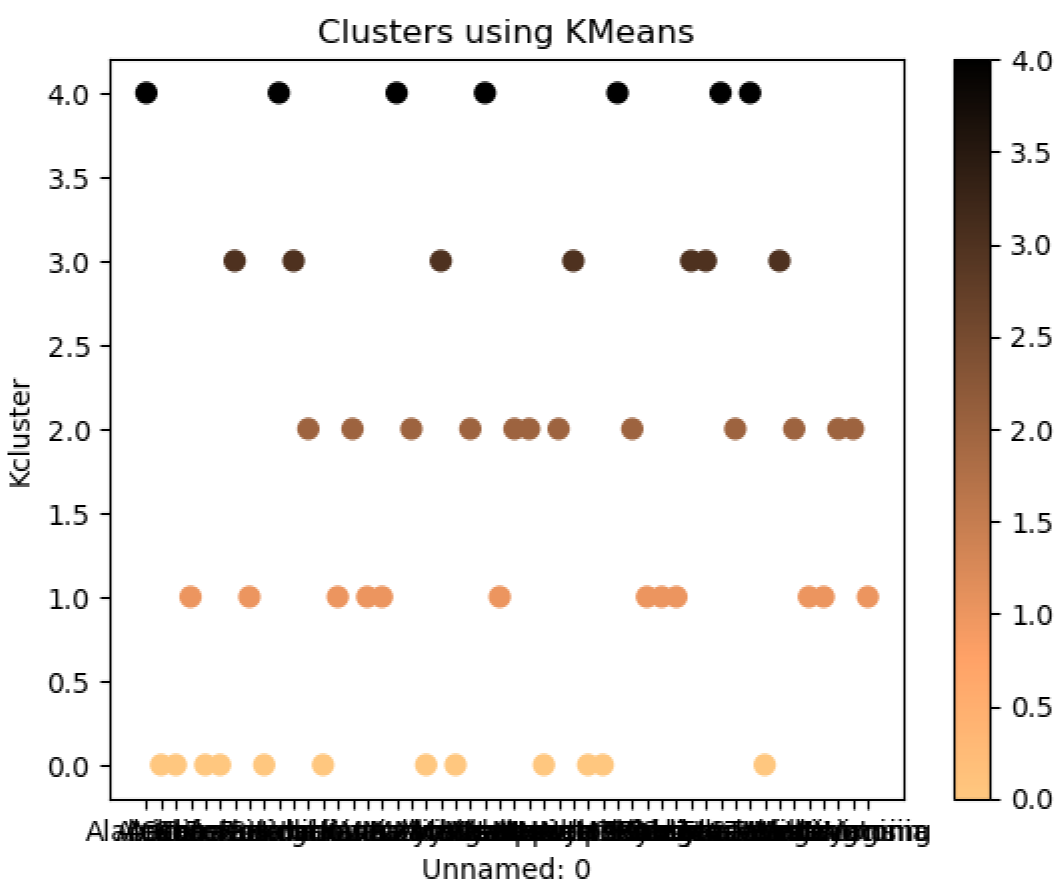


In [43]:

```
df1.plot(x="Unnamed: 0",y ="Kcluster",c=y_kmeans ,kind="scatter",s=50 ,cmap=plt.cm.coppe
plt.title('Clusters using KMeans')
```

Out[43]:

```
Text(0.5, 1.0, 'Clusters using KMeans')
```



In [44]:

```
data
```

Out[44]:

	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
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

	Unnamed: 0	Murder	Assault	UrbanPop	Rape
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 [45]:
```
crime
```

Out[45]:

	Murder	Assault	UrbanPop	Rape	clust
0	13.2	236	58	21.2	1
1	10.0	263	48	44.5	1
2	8.1	294	80	31.0	2
3	8.8	190	50	19.5	3
4	9.0	276	91	40.6	2
5	7.9	204	78	38.7	2
6	3.3	110	77	11.1	4
7	5.9	238	72	15.8	3
8	15.4	335	80	31.9	2
9	17.4	211	60	25.8	1
10	5.3	46	83	20.2	4
11	2.6	120	54	14.2	0
12	10.4	249	83	24.0	2
13	7.2	113	65	21.0	3
14	2.2	56	57	11.3	0
15	6.0	115	66	18.0	3
16	9.7	109	52	16.3	3
17	15.4	249	66	22.2	1
18	2.1	83	51	7.8	0
19	11.3	300	67	27.8	2
20	4.4	149	85	16.3	4
21	12.1	255	74	35.1	2
22	2.7	72	66	14.9	0
23	16.1	259	44	17.1	1
24	9.0	178	70	28.2	3
25	6.0	109	53	16.4	0
26	4.3	102	62	16.5	0
27	12.2	252	81	46.0	2
28	2.1	57	56	9.5	0
29	7.4	159	89	18.8	4
30	11.4	285	70	32.1	2
31	11.1	254	86	26.1	2
32	13.0	337	45	16.1	1
33	0.8	45	44	7.3	0
34	7.3	120	75	21.4	3
35	6.6	151	68	20.0	3
36	4.9	159	67	29.3	3
37	6.3	106	72	14.9	4
38	3.4	174	87	8.3	4

	Murder	Assault	UrbanPop	Rape	clust
39	14.4	279	48	22.5	1
40	3.8	86	45	12.8	0
41	13.2	188	59	26.9	1
42	12.7	201	80	25.5	2
43	3.2	120	80	22.9	4
44	2.2	48	32	11.2	0
45	8.5	156	63	20.7	3
46	4.0	145	73	26.2	3
47	5.7	81	39	9.3	0
48	2.6	53	66	10.8	0
49	6.8	161	60	15.6	3

In [46]:

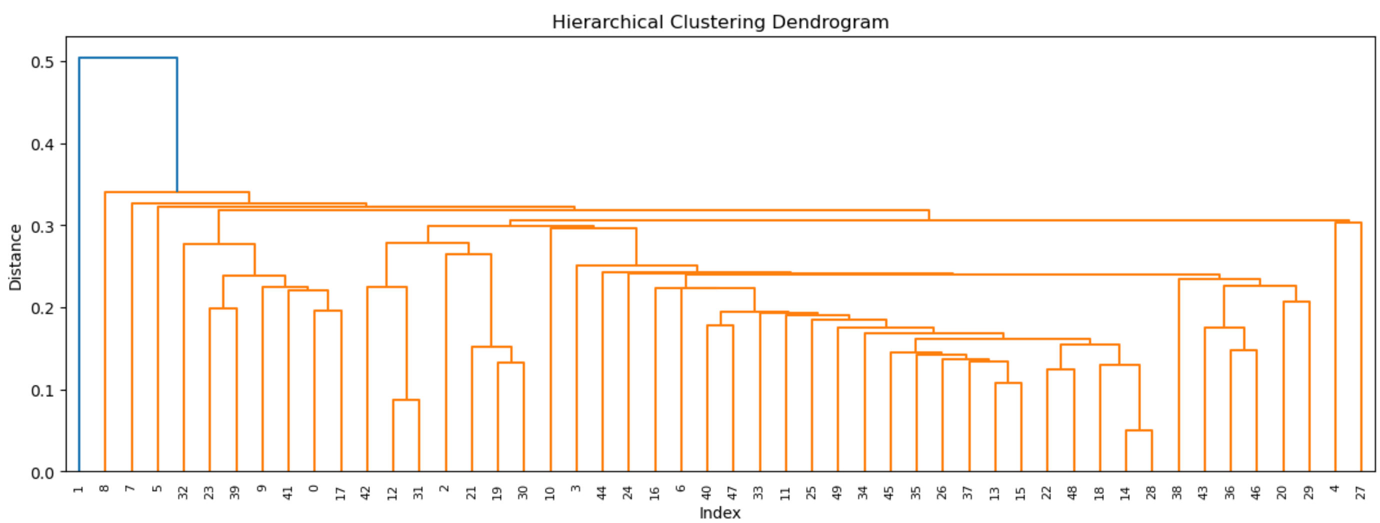
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
crime_subset = pd.DataFrame(scaler.fit_transform(crime.iloc[:,1:7]))
crime_subset
```

Out[46]:

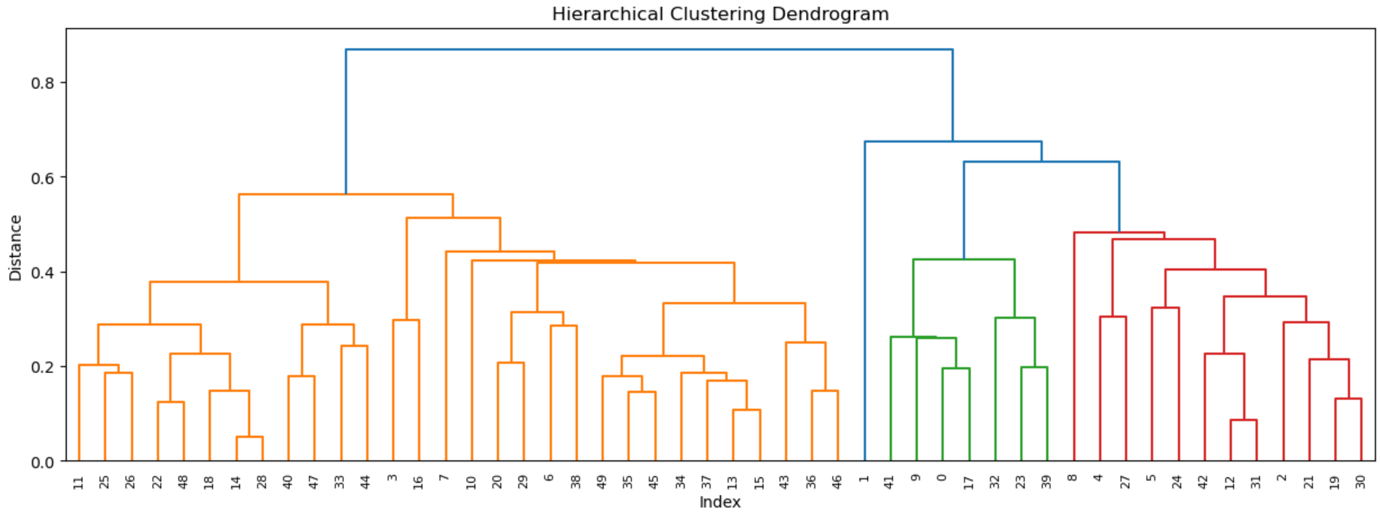
	0	1	2	3
0	0.790787	-0.526195	-0.003451	-0.637480
1	1.118060	-1.224067	2.509424	-0.637480
2	1.493817	1.009122	1.053466	0.086929
3	0.233212	-1.084492	-0.186794	0.811339
4	1.275635	1.776781	2.088814	0.086929
5	0.402909	0.869548	1.883901	0.086929
6	-0.736484	0.799761	-1.092723	1.535748
7	0.815030	0.450825	-0.585834	0.811339
8	1.990786	1.009122	1.150530	0.086929
9	0.487757	-0.386621	0.492653	-0.637480
10	-1.512241	1.218484	-0.111300	1.535748
11	-0.615272	-0.805344	-0.758392	-1.361890
12	0.948363	1.218484	0.298525	0.086929
13	-0.700121	-0.037685	-0.025021	0.811339
14	-1.391029	-0.595982	-1.071153	-1.361890
15	-0.675878	0.032102	-0.348567	0.811339
16	-0.748605	-0.944918	-0.531910	0.811339
17	0.948363	0.032102	0.104398	-0.637480
18	-1.063757	-1.014705	-1.448624	-1.361890
19	1.566544	0.101889	0.708350	0.086929
20	-0.263757	1.358058	-0.531910	1.535748
21	1.021090	0.590399	1.495646	0.086929
22	-1.197090	0.032102	-0.682898	-1.361890
23	1.069575	-1.503215	-0.445631	-0.637480
24	0.087757	0.311251	0.751490	0.811339
25	-0.748605	-0.875131	-0.521125	-1.361890
26	-0.833454	-0.247047	-0.510340	-1.361890
27	0.984726	1.078909	2.671197	0.086929
28	-1.378908	-0.665769	-1.265281	-1.361890
29	-0.142545	1.637207	-0.262288	1.535748
30	1.384726	0.311251	1.172100	0.086929
31	1.008969	1.427845	0.525008	0.086929
32	2.015028	-1.433428	-0.553480	-0.637480
33	-1.524362	-1.503215	-1.502548	-1.361890
34	-0.615272	0.660186	0.018119	0.811339
35	-0.239515	0.171676	-0.132870	0.811339
36	-0.142545	0.101889	0.870123	0.811339
37	-0.784969	0.450825	-0.682898	1.535748
38	0.039273	1.497632	-1.394700	1.535748

	0	1	2	3
39	1.311999	-1.224067	0.136752	-0.637480
40	-1.027393	-1.433428	-0.909380	-1.361890
41	0.208970	-0.456408	0.611287	-0.637480
42	0.366545	1.009122	0.460298	0.086929
43	-0.615272	1.009122	0.179892	1.535748
44	-1.487999	-2.340661	-1.081938	-1.361890
45	-0.178909	-0.177259	-0.057376	0.811339
46	-0.312242	0.520612	0.535792	0.811339
47	-1.087999	-1.852151	-1.286851	-1.361890
48	-1.427393	0.032102	-1.125078	-1.361890
49	-0.118303	-0.386621	-0.607404	0.811339

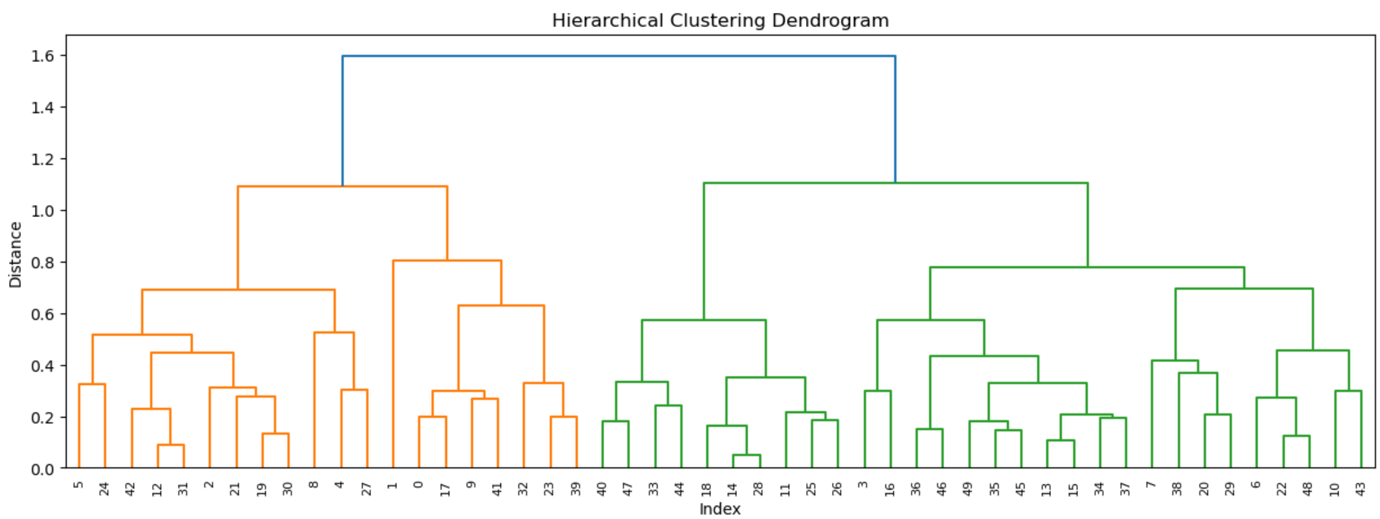
In [47]:
```python
from scipy.cluster.hierarchy import linkage
import scipy.cluster.hierarchy as sch
p = np.array(df_norm)
z = linkage(df_norm, method="single",metric="euclidean")
plt.figure(figsize=(15, 5))
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Index')
plt.ylabel('Distance')
sch.dendrogram(z,)
plt.show()
```



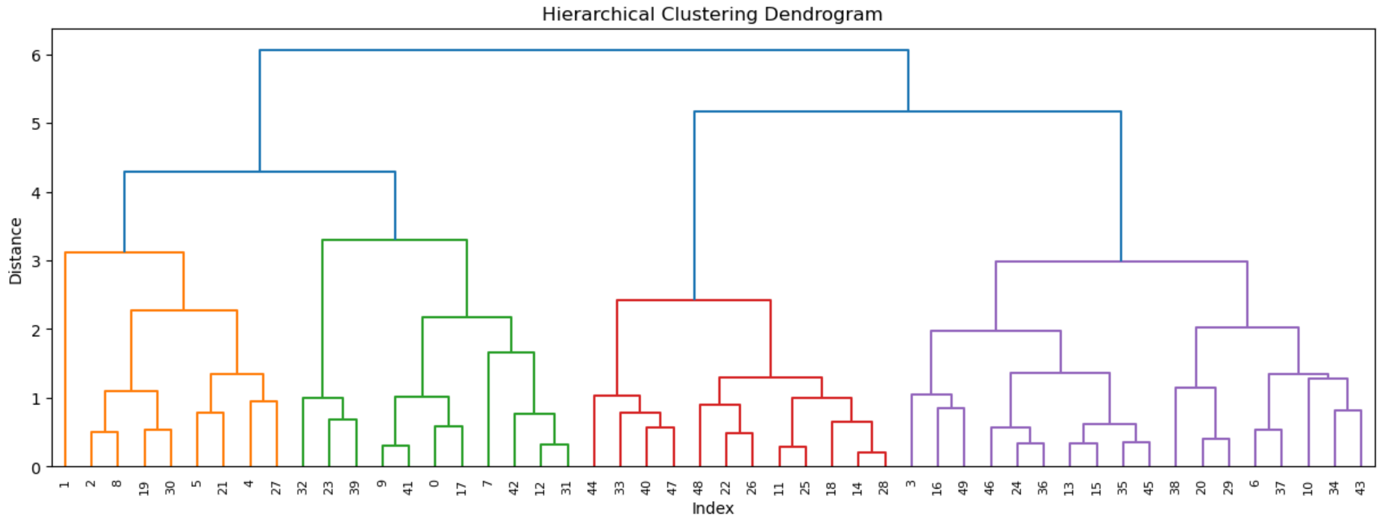
In [48]:
```python
p = np.array(df_norm)
z = linkage(df_norm, method="average",metric="euclidean")
plt.figure(figsize=(15, 5))
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Index')
plt.ylabel('Distance')
sch.dendrogram(z,)
plt.show()
```



In [49]:
```python
p = np.array(df_norm)
z = linkage(df_norm, method="complete",metric="euclidean")
plt.figure(figsize=(15, 5))
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Index')
plt.ylabel('Distance')
sch.dendrogram(z,)
plt.show()
```



In [50]:
```python
p = np.array(crime_subset)
z = linkage(crime_subset, method="complete",metric="euclidean")
plt.figure(figsize=(15, 5))
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Index')
plt.ylabel('Distance')
sch.dendrogram(z,)
plt.show()
```



In [51]:

```
from sklearn.cluster import AgglomerativeClustering
h_complete = AgglomerativeClustering(n_clusters=5, linkage='complete',affinity = "euclid

cluster_labels=pd.Series(h_complete.labels_)
cluster_labels
crime['clust']=cluster_labels
crime
```

Loading [MathJax]/extensions/Safe.js

Out[51]:

	Murder	Assault	UrbanPop	Rape	clust
0	13.2	236	58	21.2	3
1	10.0	263	48	44.5	4
2	8.1	294	80	31.0	1
3	8.8	190	50	19.5	0
4	9.0	276	91	40.6	1
5	7.9	204	78	38.7	1
6	3.3	110	77	11.1	0
7	5.9	238	72	15.8	0
8	15.4	335	80	31.9	1
9	17.4	211	60	25.8	3
10	5.3	46	83	20.2	0
11	2.6	120	54	14.2	2
12	10.4	249	83	24.0	1
13	7.2	113	65	21.0	0
14	2.2	56	57	11.3	2
15	6.0	115	66	18.0	0
16	9.7	109	52	16.3	0
17	15.4	249	66	22.2	3
18	2.1	83	51	7.8	2
19	11.3	300	67	27.8	1
20	4.4	149	85	16.3	0
21	12.1	255	74	35.1	1
22	2.7	72	66	14.9	0
23	16.1	259	44	17.1	3
24	9.0	178	70	28.2	1
25	6.0	109	53	16.4	2
26	4.3	102	62	16.5	2
27	12.2	252	81	46.0	1
28	2.1	57	56	9.5	2
29	7.4	159	89	18.8	0
30	11.4	285	70	32.1	1
31	11.1	254	86	26.1	1
32	13.0	337	45	16.1	3
33	0.8	45	44	7.3	2
34	7.3	120	75	21.4	0
35	6.6	151	68	20.0	0
36	4.9	159	67	29.3	0
37	6.3	106	72	14.9	0
38	3.4	174	87	8.3	0

	Murder	Assault	UrbanPop	Rape	clust
39	14.4	279	48	22.5	3
40	3.8	86	45	12.8	2
41	13.2	188	59	26.9	3
42	12.7	201	80	25.5	1
43	3.2	120	80	22.9	0
44	2.2	48	32	11.2	2
45	8.5	156	63	20.7	0
46	4.0	145	73	26.2	0
47	5.7	81	39	9.3	2
48	2.6	53	66	10.8	0
49	6.8	161	60	15.6	0

In [53]:
```
crime.iloc[:,1:].groupby(crime.clust).mean()
```

Out[53]:

	Assault	UrbanPop	Rape	clust
clust				
0	132.300000	70.800000	18.100000	0.0
1	256.916667	78.333333	32.250000	1.0
2	78.700000	49.300000	11.630000	2.0
3	251.285714	54.285714	21.685714	3.0
4	263.000000	48.000000	44.500000	4.0

In [54]:
```
data = crime[(crime.clust==0)]
data
```

Out[54]:

	Murder	Assault	UrbanPop	Rape	clust
3	8.8	190	50	19.5	0
6	3.3	110	77	11.1	0
7	5.9	238	72	15.8	0
10	5.3	46	83	20.2	0
13	7.2	113	65	21.0	0
15	6.0	115	66	18.0	0
16	9.7	109	52	16.3	0
20	4.4	149	85	16.3	0
22	2.7	72	66	14.9	0
29	7.4	159	89	18.8	0
34	7.3	120	75	21.4	0
35	6.6	151	68	20.0	0
36	4.9	159	67	29.3	0
37	6.3	106	72	14.9	0
38	3.4	174	87	8.3	0
43	3.2	120	80	22.9	0
45	8.5	156	63	20.7	0
46	4.0	145	73	26.2	0
48	2.6	53	66	10.8	0
49	6.8	161	60	15.6	0

In [55]:

```
data = crime[(crime.clust==1)]
data
```

Out[55]:

	Murder	Assault	UrbanPop	Rape	clust
2	8.1	294	80	31.0	1
4	9.0	276	91	40.6	1
5	7.9	204	78	38.7	1
8	15.4	335	80	31.9	1
12	10.4	249	83	24.0	1
19	11.3	300	67	27.8	1
21	12.1	255	74	35.1	1
24	9.0	178	70	28.2	1
27	12.2	252	81	46.0	1
30	11.4	285	70	32.1	1
31	11.1	254	86	26.1	1
42	12.7	201	80	25.5	1

In [56]:

```
data = crime[(crime.clust==2)]
data
```

Out[56]:

	Murder	Assault	UrbanPop	Rape	clust
11	2.6	120	54	14.2	2
14	2.2	56	57	11.3	2
18	2.1	83	51	7.8	2
25	6.0	109	53	16.4	2
26	4.3	102	62	16.5	2
28	2.1	57	56	9.5	2
33	0.8	45	44	7.3	2
40	3.8	86	45	12.8	2
44	2.2	48	32	11.2	2
47	5.7	81	39	9.3	2

In [57]:

```
data = crime[(crime.clust==3)]
data
```

Out[57]:

	Murder	Assault	UrbanPop	Rape	clust
0	13.2	236	58	21.2	3
9	17.4	211	60	25.8	3
17	15.4	249	66	22.2	3
23	16.1	259	44	17.1	3
32	13.0	337	45	16.1	3
39	14.4	279	48	22.5	3
41	13.2	188	59	26.9	3

In [59]:

```
data = crime[(crime.clust==4)]
data
```

Out[59]:

	Murder	Assault	UrbanPop	Rape	clust
1	10.0	263	48	44.5	4

In [ ]: