Assignment-07-Clustering_Crime_data

In [1]:
```
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.cluster.hierarchy as sch
from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.preprocessing import normalize
```

In [3]:
```
# Import Dataset
crime=pd.read_csv('crime_data.csv')
crime
```

Out[3]:

	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

	Unnamed: 0	Murder	Assault	UrbanPop	Rape
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 [4]:
```
crime.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 [5]:
```
crime.drop(['Unnamed: 0'],axis=1,inplace=True)
crime
```

Out[5]:

	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

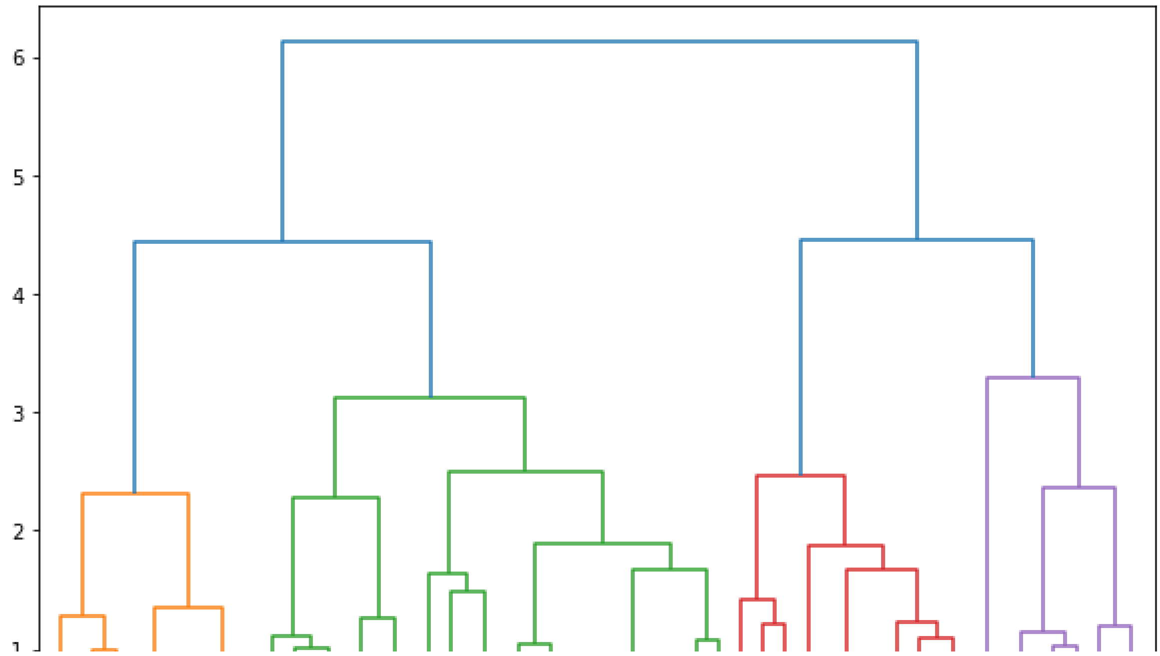
	Murder	Assault	UrbanPop	Rape
39	14.4	279	48	22.5
40	3.8	86	45	12.8
41	13.2	188	59	26.9
42	12.7	201	80	25.5
43	3.2	120	80	22.9
44	2.2	48	32	11.2
45	8.5	156	63	20.7
46	4.0	145	73	26.2
47	5.7	81	39	9.3
48	2.6	53	66	10.8
49	6.8	161	60	15.6

In [6]:
```
# Normalize heterogenous numerical data using standard scalar fit transform to dataset
crime_norm=StandardScaler().fit_transform(crime)
crime_norm
```

Out[6]:
```
array([[ 1.25517927,  0.79078716, -0.52619514, -0.00345116],
       [ 0.51301858,  1.11805959, -1.22406668,  2.50942392],
       [ 0.07236067,  1.49381682,  1.00912225,  1.05346626],
       [ 0.23470832,  0.23321191, -1.08449238, -0.18679398],
       [ 0.28109336,  1.2756352 ,  1.77678094,  2.08881393],
       [ 0.02597562,  0.40290872,  0.86954794,  1.88390137],
       [-1.04088037, -0.73648418,  0.79976079, -1.09272319],
       [-0.43787481,  0.81502956,  0.45082502, -0.58583422],
       [ 1.76541475,  1.99078607,  1.00912225,  1.1505301 ],
       [ 2.22926518,  0.48775713, -0.38662083,  0.49265293],
       [-0.57702994, -1.51224105,  1.21848371, -0.11129987],
       [-1.20322802, -0.61527217, -0.80534376, -0.75839217],
       [ 0.60578867,  0.94836277,  1.21848371,  0.29852525],
       [-0.13637203, -0.70012057, -0.03768506, -0.0250209 ],
       [-1.29599811, -1.39102904, -0.5959823 , -1.07115345],
       [-0.41468229, -0.67587817,  0.03210209, -0.34856705],
       [ 0.44344101, -0.74860538, -0.94491807, -0.53190987],
       [ 1.76541475,  0.94836277,  0.03210209,  0.10439756],
       [-1.31919063, -1.06375661, -1.01470522, -1.44862395],
       [ 0.81452136,  1.56654403,  0.10188925,  0.70835037],
       [-0.78576263, -0.26375734,  1.35805802, -0.53190987],
       [ 1.00006153,  1.02108998,  0.59039932,  1.49564599],
       [-1.1800355 , -1.19708982,  0.03210209, -0.68289807],
       [ 1.9277624 ,  1.06957478, -1.5032153 , -0.44563089],
       [ 0.28109336,  0.0877575 ,  0.31125071,  0.75148985],
       [-0.41468229, -0.74860538, -0.87513091, -0.521125  ],
       [-0.80895515, -0.83345379, -0.24704653, -0.51034012],
       [ 1.02325405,  0.98472638,  1.0789094 ,  2.671197  ],
       [-1.31919063, -1.37890783, -0.66576945, -1.26528114],
       [-0.08998698, -0.14254532,  1.63720664, -0.26228808],
       [ 0.83771388,  1.38472601,  0.31125071,  1.17209984],
       [ 0.76813632,  1.00896878,  1.42784517,  0.52500755],
       [ 1.20879423,  2.01502847, -1.43342815, -0.55347961],
       [-1.62069341, -1.52436225, -1.5032153 , -1.50254831],
       [-0.11317951, -0.61527217,  0.66018648,  0.01811858],
       [-0.27552716, -0.23951493,  0.1716764 , -0.13286962],
       [-0.66980002, -0.14254532,  0.10188925,  0.87012344],
       [-0.34510472, -0.78496898,  0.45082502, -0.68289807],
       [-1.01768785,  0.03927269,  1.49763233, -1.39469959],
       [ 1.53348953,  1.3119988 , -1.22406668,  0.13675217],
       [-0.92491776, -1.027393  , -1.43342815, -0.90938037],
       [ 1.25517927,  0.20896951, -0.45640799,  0.61128652],
       [ 1.13921666,  0.36654512,  1.00912225,  0.46029832],
       [-1.06407289, -0.61527217,  1.00912225,  0.17989166],
       [-1.29599811, -1.48799864, -2.34066115, -1.08193832],
       [ 0.16513075, -0.17890893, -0.17725937, -0.05737552],
       [-0.87853272, -0.31224214,  0.52061217,  0.53579242],
       [-0.48425985, -1.08799901, -1.85215107, -1.28685088],
       [-1.20322802, -1.42739264,  0.03210209, -1.1250778 ],
       [-0.22914211, -0.11830292, -0.38662083, -0.60740397]])
```

Hierarchical Clustering

In [7]:
```python
# Create Dendrograms
plt.figure(figsize=(10, 7))
dendograms=sch.dendrogram(sch.linkage(crime_norm,'complete'))
```



In [8]:
```python
# Create Clusters (y)
hclusters=AgglomerativeClustering(n_clusters=5,affinity='euclidean',linkage='ward')
hclusters
```

Out[8]:
```
AgglomerativeClustering(n_clusters=5)
```

In [9]:
```python
y=pd.DataFrame(hclusters.fit_predict(crime_norm),columns=['clustersid'])
y['clustersid'].value_counts()
```

Out[9]:
```
1    15
0    12
2    12
3     7
4     4
Name: clustersid, dtype: int64
```

In [12]:
```
# Adding clusters to dataset
crime['clustersid']=hclusters.labels_
crime
```

Out[12]:

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

	Murder	Assault	UrbanPop	Rape	clustersid
38	3.4	174	87	8.3	1
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	0
43	3.2	120	80	22.9	1
44	2.2	48	32	11.2	2
45	8.5	156	63	20.7	4
46	4.0	145	73	26.2	1
47	5.7	81	39	9.3	2
48	2.6	53	66	10.8	2
49	6.8	161	60	15.6	4

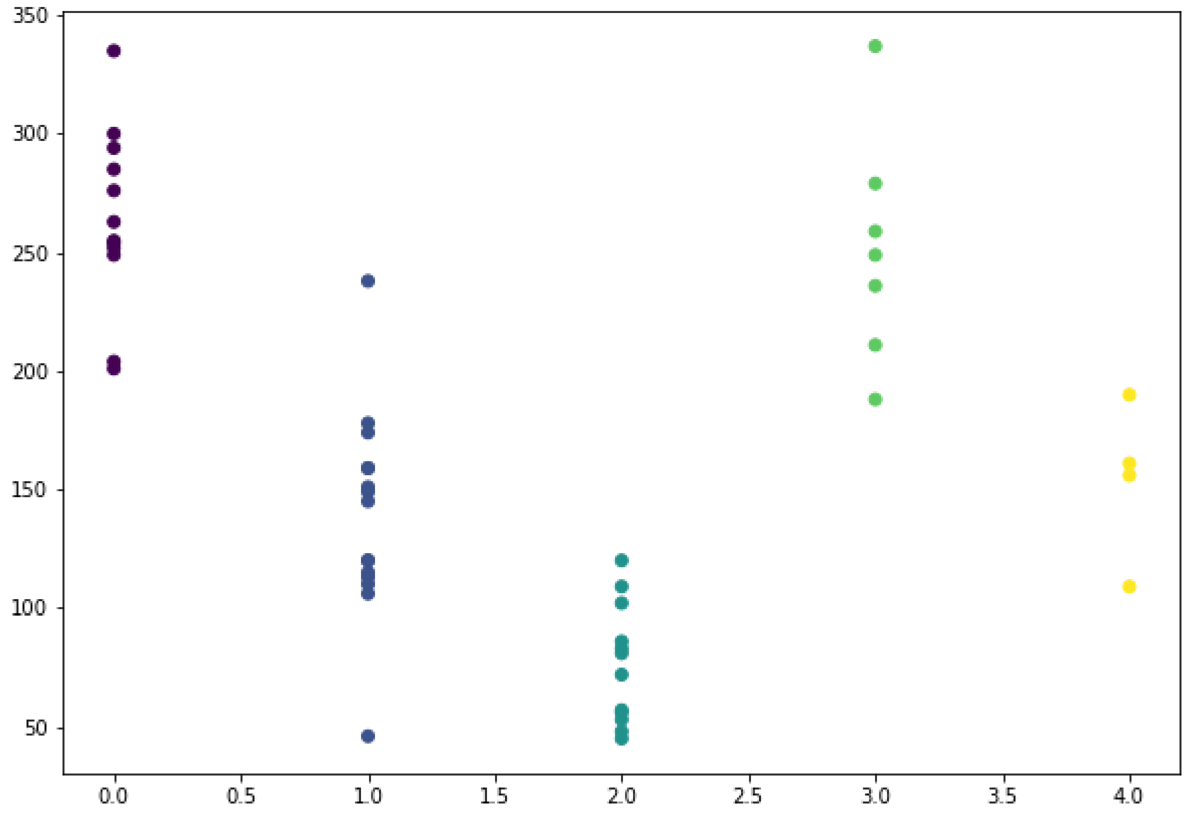
In [13]:
```
crime.groupby('clustersid').agg(['mean']).reset_index()
```

Out[13]:

	clustersid	Murder	Assault	UrbanPop	Rape
		mean	mean	mean	mean
0	0	10.966667	264.000000	76.500000	33.608333
1	1	5.613333	138.866667	75.266667	19.493333
2	2	3.091667	76.000000	52.083333	11.833333
3	3	14.671429	251.285714	54.285714	21.685714
4	4	8.450000	154.000000	56.250000	18.025000

In [15]:
```python
# Plot Clusters
plt.figure(figsize=(10, 7))
plt.scatter(crime['clustersid'],crime['Assault'], c=hclusters.labels_)
```

Out[15]: <matplotlib.collections.PathCollection at 0x1a2294aa400>

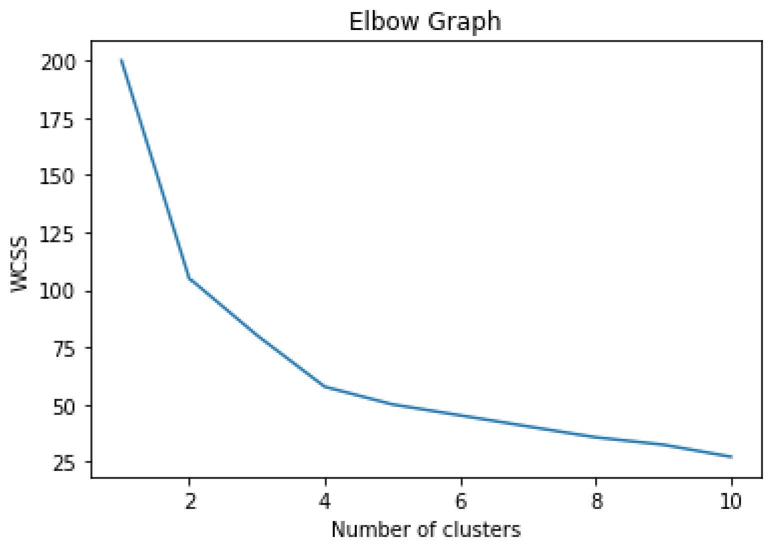


K Means Clustering

In [19]:
```python
# within-cluster sum-of-squares criterion
wcss=[]
for i in range (1,11):
    kmeans=KMeans(n_clusters=i,random_state=2)
    kmeans.fit(crime_norm)
    wcss.append(kmeans.inertia_)

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

In [20]:
```
# Plot K values range vs WCSS to get Elbow graph for choosing K (no. of clusters)
plt.plot(range(1,11),wcss)
plt.title('Elbow Graph')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



Build Cluster algorithm using K=4

In [21]:
```
# Cluster algorithm using K=4
clusters4=KMeans(4,random_state=30).fit(crime_norm)
clusters4
```

Out[21]:
```
KMeans(n_clusters=4, random_state=30)
```

In [22]:
```
clusters4.labels_
```

Out[22]:
```
array([0, 3, 3, 0, 3, 3, 1, 1, 3, 0, 1, 2, 3, 1, 2, 1, 2, 0, 2, 3, 1, 3,
       2, 0, 3, 2, 2, 3, 2, 1, 3, 3, 0, 2, 1, 1, 1, 1, 1, 0, 2, 0, 3, 1,
       2, 1, 1, 2, 2, 1])
```

In [29]:
```python
# Assign clusters to the data set
crime2=crime.copy()
crime2.drop(['clustersid'],axis=1,inplace=True)
crime2['clusters_id']=clusters4.labels_
crime2
```

Out[29]:

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

	Murder	Assault	UrbanPop	Rape	clusters_id
37	6.3	106	72	14.9	1
38	3.4	174	87	8.3	1
39	14.4	279	48	22.5	0
40	3.8	86	45	12.8	2
41	13.2	188	59	26.9	0
42	12.7	201	80	25.5	3
43	3.2	120	80	22.9	1
44	2.2	48	32	11.2	2
45	8.5	156	63	20.7	1
46	4.0	145	73	26.2	1
47	5.7	81	39	9.3	2
48	2.6	53	66	10.8	2
49	6.8	161	60	15.6	1

In [30]:
```
# Compute the centroids for K=4 clusters with 11 variables
clusters4.cluster_centers_
```

Out[30]:
```
array([[ 1.42622412,  0.88321132, -0.82279055,  0.01946669],
       [-0.49440658, -0.3864845 ,  0.58167593, -0.26431024],
       [-0.97130281, -1.11783581, -0.93954982, -0.97657842],
       [ 0.70212683,  1.04999438,  0.72997363,  1.28990383]])
```

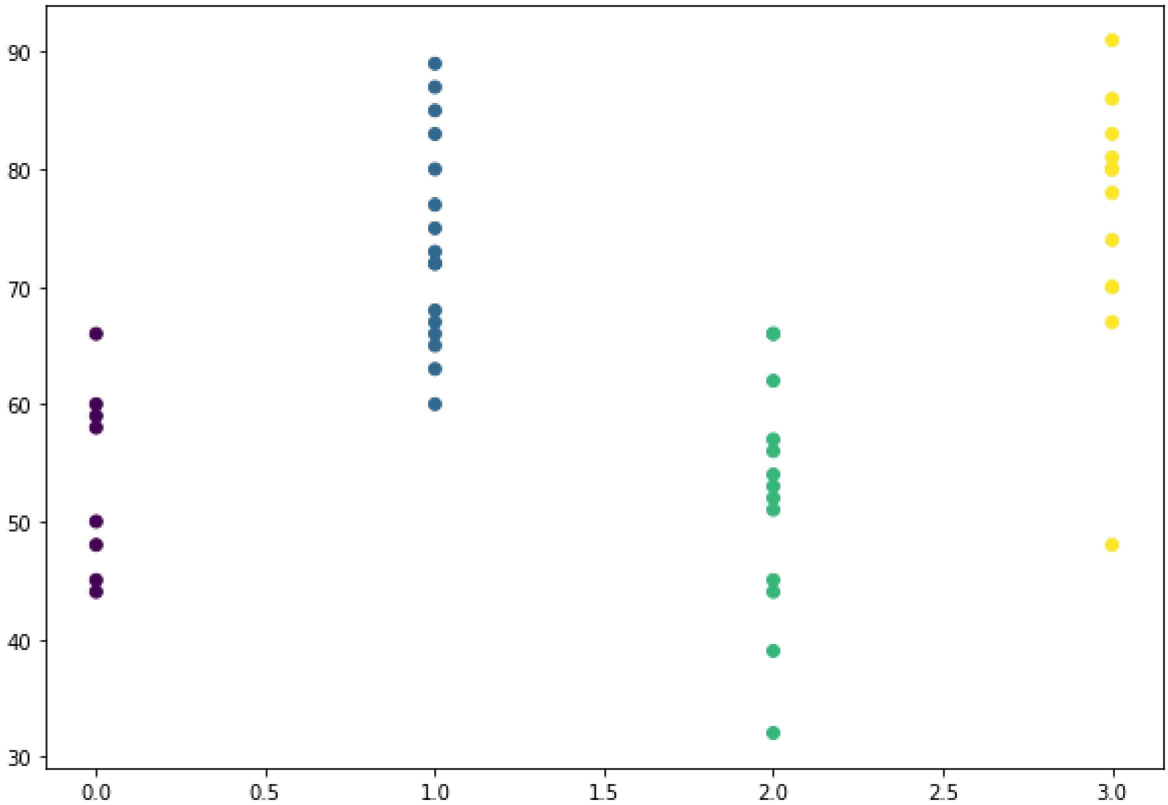
In [31]:
```
# Group data by Clusters (K=4)
crime2.groupby('clusters_id').agg(['mean']).reset_index()
```

Out[31]:

	clusters_id	Murder	Assault	UrbanPop	Rape
		mean	mean	mean	mean
0	0	13.937500	243.625000	53.750000	21.412500
1	1	5.656250	138.875000	73.875000	18.781250
2	2	3.600000	78.538462	52.076923	12.176923
3	3	10.815385	257.384615	76.000000	33.192308

In [50]:
```python
# Plot Clusters
plt.figure(figsize=(10, 7))
plt.scatter(crime2['clusters_id'],crime2['UrbanPop'], c=clusters4.labels_)
```

Out[50]: <matplotlib.collections.PathCollection at 0x1a22d27e0a0>



Build Cluster algorithm using K=5

In [35]:
```python
# Cluster algorithm using K=5
clusters5=KMeans(5,random_state=30).fit(crime_norm)
clusters5
```

Out[35]: KMeans(n_clusters=5, random_state=30)

In [36]:
```
clusters5.labels_
```

Out[36]:
```
array([1, 2, 2, 0, 2, 2, 4, 0, 2, 1, 4, 3, 2, 0, 3, 0, 0, 1, 3, 2, 4, 2,
       3, 1, 0, 0, 0, 2, 3, 4, 2, 2, 1, 3, 0, 0, 0, 0, 4, 1, 3, 1, 2, 4,
       3, 0, 0, 3, 3, 0])
```

In [37]:
```
# Assign clusters to the data set
crime3=crime.copy()
crime3.drop(['clustersid'],axis=1,inplace=True)
crime3['clusters_id']=clusters5.labels_
crime3
```

Out[37]:

	Murder	Assault	UrbanPop	Rape	clusters_id
0	13.2	236	58	21.2	1
1	10.0	263	48	44.5	2
2	8.1	294	80	31.0	2
3	8.8	190	50	19.5	0
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	0
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	3
12	10.4	249	83	24.0	2
13	7.2	113	65	21.0	0
14	2.2	56	57	11.3	3
15	6.0	115	66	18.0	0
16	9.7	109	52	16.3	0
17	15.4	249	66	22.2	1
18	2.1	83	51	7.8	3
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	3
23	16.1	259	44	17.1	1
24	9.0	178	70	28.2	0
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	3
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	3
34	7.3	120	75	21.4	0
35	6.6	151	68	20.0	0
36	4.9	159	67	29.3	0

	Murder	Assault	UrbanPop	Rape	clusters_id
37	6.3	106	72	14.9	0
38	3.4	174	87	8.3	4
39	14.4	279	48	22.5	1
40	3.8	86	45	12.8	3
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	3
45	8.5	156	63	20.7	0
46	4.0	145	73	26.2	0
47	5.7	81	39	9.3	3
48	2.6	53	66	10.8	3
49	6.8	161	60	15.6	0

In [38]:
```
# Compute the centroids for K=5 clusters with 11 variables
clusters5.cluster_centers_
```

Out[38]:
```
array([[-0.23996529, -0.33082798, -0.0702524 , -0.1343076 ],
       [ 1.59644066,  0.97606837, -0.78540457,  0.0489325 ],
       [ 0.73721295,  1.13018079,  0.76486721,  1.33477166],
       [-1.18467401, -1.2201201 , -1.01470522, -1.11321445],
       [-0.76257011, -0.53850456,  1.25337729, -0.53550482]])
```

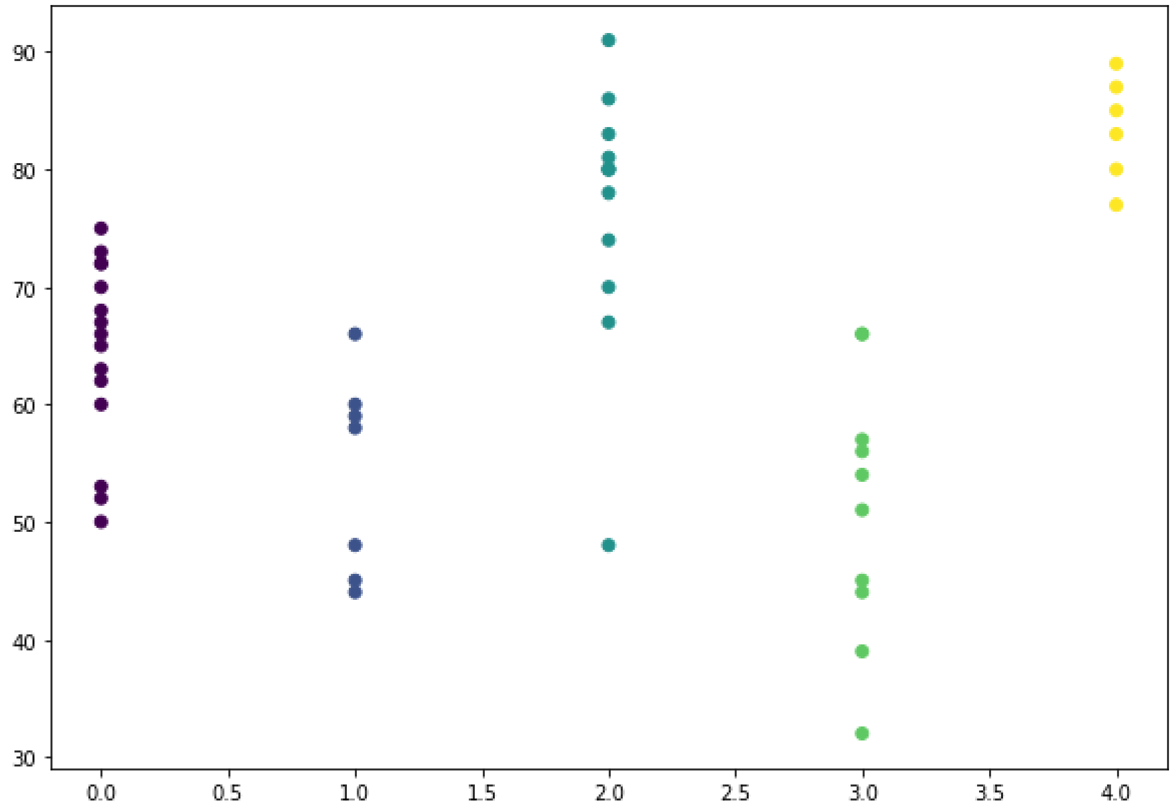
In [39]:
```
# Group data by Clusters (K=5)
crime3.groupby('clusters_id').agg(['mean']).reset_index()
```

Out[39]:

	clusters_id	Murder	Assault	UrbanPop	Rape
		mean	mean	mean	mean
0	0	6.753333	143.466667	64.533333	19.986667
1	1	14.671429	251.285714	54.285714	21.685714
2	2	10.966667	264.000000	76.500000	33.608333
3	3	2.680000	70.100000	51.000000	10.910000
4	4	4.500000	126.333333	83.500000	16.266667

In [49]:
```
# Plot Clusters
plt.figure(figsize=(10, 7))
plt.scatter(crime3['clusters_id'],crime3['UrbanPop'], c=clusters5.labels_)
```

Out[49]: <matplotlib.collections.PathCollection at 0x1a22d2303d0>



DBSCAN Clustering

In [42]:
```
# DBSCAN Clustering
dbscan=DBSCAN(eps=1,min_samples=4)
dbscan.fit(crime_norm)
```

Out[42]: DBSCAN(eps=1, min_samples=4)

In [43]:

```
#Noisy samples are given the label -1.
dbscan.labels_
```

Out[43]:

```
array([ 0, -1, -1, -1, -1, -1,  1, -1, -1, -1, -1,  1, -1,  1,  1,  1,  1,
        0,  1, -1,  1, -1,  1, -1,  1,  1,  1, -1,  1,  1, -1, -1, -1,  1,
        1,  1,  1,  1,  1,  0,  1,  0, -1,  1,  1,  1,  1,  1,  1,  1],
      dtype=int64)
```

In [44]:
```
# Adding clusters to dataset
crime['clusters']=dbscan.labels_
crime
```

Out[44]:

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

	Murder	Assault	UrbanPop	Rape	clustersid	clusters
38	3.4	174	87	8.3	1	1
39	14.4	279	48	22.5	3	0
40	3.8	86	45	12.8	2	1
41	13.2	188	59	26.9	3	0
42	12.7	201	80	25.5	0	-1
43	3.2	120	80	22.9	1	1
44	2.2	48	32	11.2	2	1
45	8.5	156	63	20.7	4	1
46	4.0	145	73	26.2	1	1
47	5.7	81	39	9.3	2	1
48	2.6	53	66	10.8	2	1
49	6.8	161	60	15.6	4	1

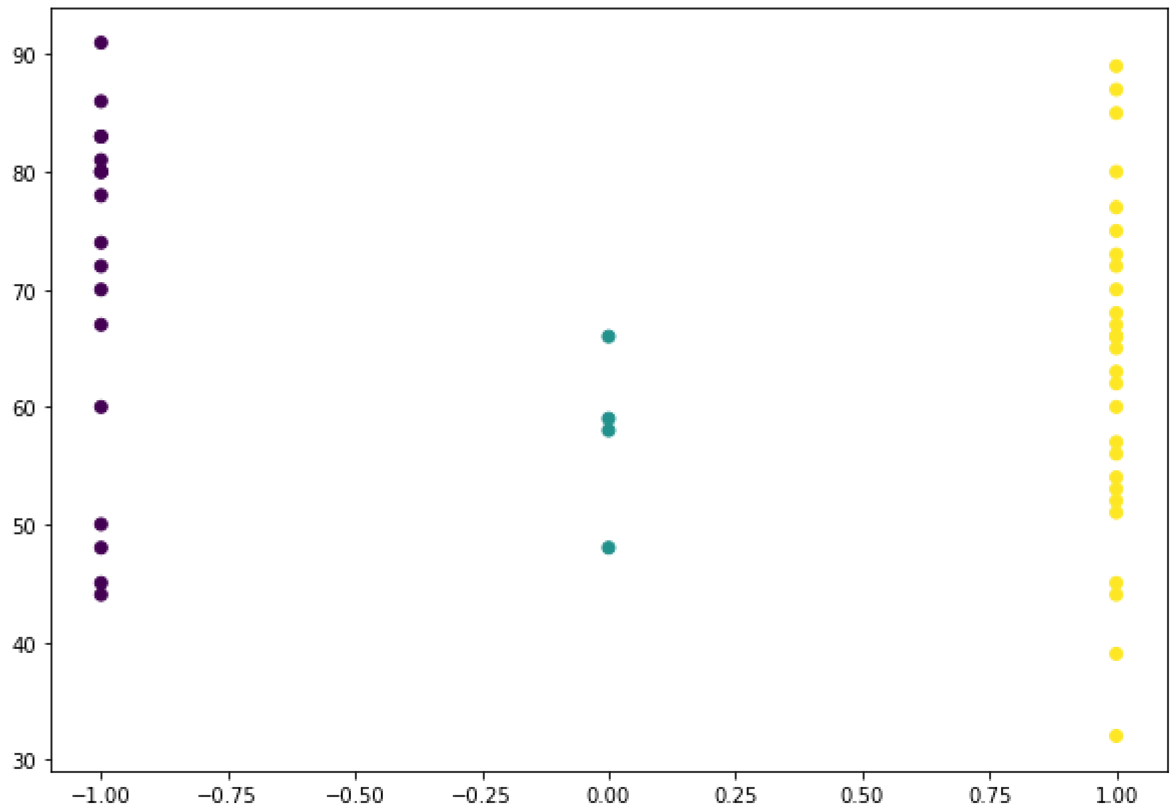
In [45]:
```
crime.groupby('clusters').agg(['mean']).reset_index()
```

Out[45]:

	clusters	Murder	Assault	UrbanPop	Rape	clustersid
		mean	mean	mean	mean	mean
0	-1	11.005556	247.166667	70.666667	28.766667	0.833333
1	0	14.050000	238.000000	57.750000	23.200000	3.000000
2	1	4.825000	112.035714	63.357143	16.107143	1.750000

In [48]:
```
# Plot Clusters
plt.figure(figsize=(10, 7))
plt.scatter(crime['clusters'],crime['UrbanPop'], c=dbscan.labels_)
```

Out[48]: <matplotlib.collections.PathCollection at 0x1a22d1f66a0>



In [ ]: