Assignment-07-DBSCAN Clustering (Crimes)

In [33]:
```
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
```

In [34]:
```
crim=pd.read_csv('crime_data.csv')
crim
```

Out[34]:

	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 [36]:

```
#Normalized data fuction
def norm_func(i):
    x=(i-i.min())/(i.max()-i.min())
    return(x)
```

In [37]:

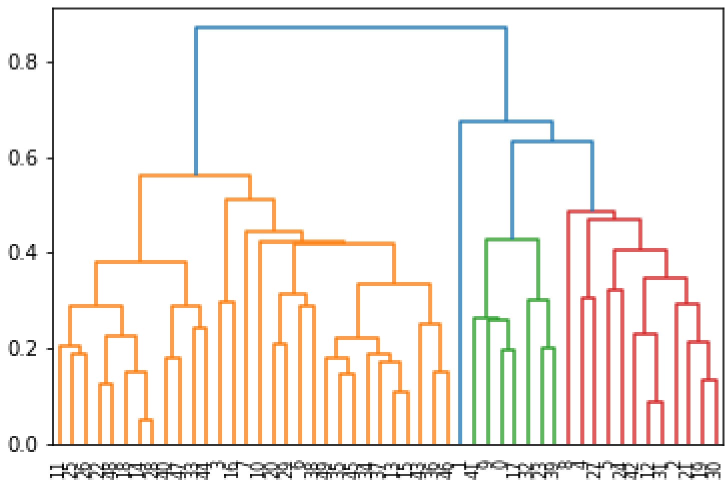
```
df_norm=norm_func(crim.iloc[:,1:])
df_norm
```

Out[37]:

	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 [38]:
```
dendrogram=sch.dendrogram(sch.linkage(df_norm,method='average'))
```



In [39]:
```
# create clusters
hc=AgglomerativeClustering(n_clusters=3,affinity='euclidean',linkage='complete')
hc
```

Out[39]:
```
AgglomerativeClustering(linkage='complete', n_clusters=3)
```

In [40]:
```
y_hc=hc.fit_predict(df_norm)
y_hc
```

Out[40]:
```
array([0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 2, 0, 1, 2, 1, 1, 0, 2, 0, 1, 0,
       1, 0, 0, 2, 2, 0, 2, 1, 0, 0, 0, 2, 1, 1, 1, 1, 1, 0, 2, 0, 0, 1,
       2, 1, 1, 2, 1, 1], dtype=int64)
```

In [41]:
```
crim['h_clusterid']=hc.labels_
crim
```

Out[41]:

	Unnamed: 0	Murder	Assault	UrbanPop	Rape	h_clusterid
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	1
4	California	9.0	276	91	40.6	0
5	Colorado	7.9	204	78	38.7	0
6	Connecticut	3.3	110	77	11.1	1
7	Delaware	5.9	238	72	15.8	1
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	1
11	Idaho	2.6	120	54	14.2	2
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	1
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	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	1
23	Mississippi	16.1	259	44	17.1	0
24	Missouri	9.0	178	70	28.2	0
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	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
34	Ohio	7.3	120	75	21.4	1
35	Oklahoma	6.6	151	68	20.0	1
36	Oregon	4.9	159	67	29.3	1
37	Pennsylvania	6.3	106	72	14.9	1
38	Rhode Island	3.4	174	87	8.3	1

	Unnamed: 0	Murder	Assault	UrbanPop	Rape	h_clusterid
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	0
43	Utah	3.2	120	80	22.9	1
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	1
47	West Virginia	5.7	81	39	9.3	2
48	Wisconsin	2.6	53	66	10.8	1
49	Wyoming	6.8	161	60	15.6	1

Kmeans

In [42]:

```
from sklearn.cluster import KMeans
from scipy.spatial.distance import cdist
```

In [43]:

```
crim1=pd.read_csv('crime_data.csv')
```

In [44]:

```
#Normalized data fuction
def norm_func(i):
    x=(i-i.min())/(i.max()-i.min())
    return(x)
```

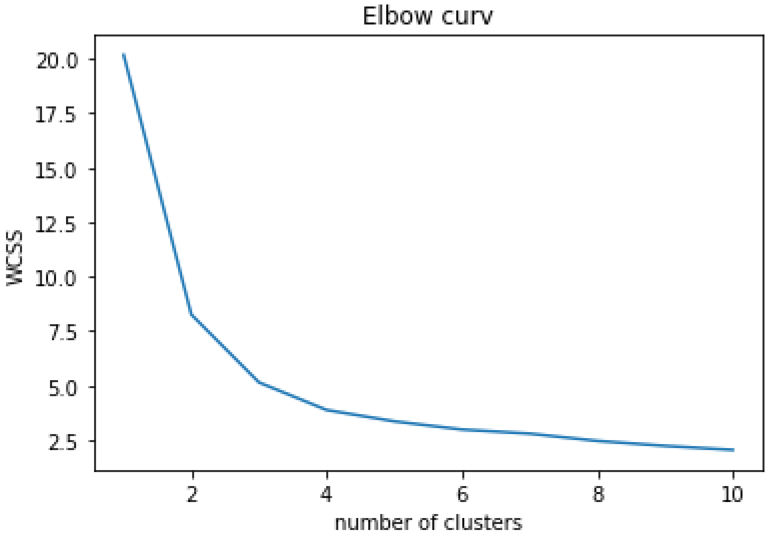
In [45]:

```
df_norm=norm_func(crim.iloc[:,1:])
```

In [46]:

```
# Elbow curv
wcss=[]
for i in range(1,11):
    kmeans=KMeans(n_clusters=i)
    kmeans.fit(df_norm)
    wcss.append(kmeans.inertia_)
plt.plot(range(1,11),wcss)
plt.title('Elbow curv')
plt.xlabel('number of clusters')
plt.ylabel('WCSS')
plt.show()
```

```
C:\Users\HP\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1036: UserWarning: KM
eans is known to have a memory leak on Windows with MKL, when there are less chunks than
available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=
1.
  warnings.warn(
```



In [47]:
```
# selecting 4 clusters from above scree plot
model=KMeans(n_clusters=4)
model.fit(df_norm)
model.labels_
```

Out[47]:
```
array([3, 0, 0, 2, 0, 0, 2, 2, 0, 3, 2, 1, 0, 2, 1, 2, 2, 3, 1, 0, 2, 0,
       2, 3, 0, 1, 1, 0, 1, 2, 0, 0, 3, 1, 2, 2, 2, 2, 2, 3, 1, 3, 0, 2,
       1, 2, 2, 1, 2, 2])
```

In [48]:
```
x=pd.Series(model.labels_)
crim1['Clust']=x
crim1
```

Loading [MathJax]/extensions/Safe.js

Out[48]:

	Unnamed: 0	Murder	Assault	UrbanPop	Rape	Clust
0	Alabama	13.2	236	58	21.2	3
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	2
4	California	9.0	276	91	40.6	0
5	Colorado	7.9	204	78	38.7	0
6	Connecticut	3.3	110	77	11.1	2
7	Delaware	5.9	238	72	15.8	2
8	Florida	15.4	335	80	31.9	0
9	Georgia	17.4	211	60	25.8	3
10	Hawaii	5.3	46	83	20.2	2
11	Idaho	2.6	120	54	14.2	1
12	Illinois	10.4	249	83	24.0	0
13	Indiana	7.2	113	65	21.0	2
14	Iowa	2.2	56	57	11.3	1
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	3
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	2
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	3
24	Missouri	9.0	178	70	28.2	0
25	Montana	6.0	109	53	16.4	1
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	1
29	New Jersey	7.4	159	89	18.8	2
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	3
33	North Dakota	0.8	45	44	7.3	1
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

Loading [MathJax]/extensions/Safe.js

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

In [49]:
```
crim1.iloc[:,1:5].groupby(crim1.Clust).mean()
```

Out[49]:

Clust	Murder	Assault	UrbanPop	Rape
0	10.815385	257.384615	76.000000	33.192308
1	3.180000	78.700000	49.300000	11.630000
2	5.715000	132.300000	70.800000	18.100000
3	14.671429	251.285714	54.285714	21.685714

DBSCAN

In [50]:
```
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
```

In [51]:
```
crim3=pd.read_csv('crime_data.csv')
crim3.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 [52]:
```
df=crim3.iloc[:,1:5]
df.values
```

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

In [53]:
```
stscaler=StandardScaler().fit(df.values)
x=stscaler.transform(df.values)
x
```

Out[53]:
```
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]])
```

In [54]:
```
dbscan=DBSCAN(eps=2,min_samples=5)
dbscan.fit(x)
```

Out[54]:
```
DBSCAN(eps=2)
```

In [55]:
```
dbscan.labels_
```

Out[55]:
```
array([ 0, -1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0],
      dtype=int64)
```

In [56]:
```
cl=pd.DataFrame(dbscan.labels_,columns=['cluster'])
```

```
cl
```

Out[56]:

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

	cluster
39	0
40	0
41	0
42	0
43	0
44	0
45	0
46	0
47	0
48	0
49	0

In [57]:
```
pd.concat([crim3,cl],axis=1)
```

Out[57]:

	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	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	0
6	Connecticut	3.3	110	77	11.1	0
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	0
11	Idaho	2.6	120	54	14.2	0
12	Illinois	10.4	249	83	24.0	0
13	Indiana	7.2	113	65	21.0	0
14	Iowa	2.2	56	57	11.3	0
15	Kansas	6.0	115	66	18.0	0
16	Kentucky	9.7	109	52	16.3	0
17	Louisiana	15.4	249	66	22.2	0
18	Maine	2.1	83	51	7.8	0
19	Maryland	11.3	300	67	27.8	0
20	Massachusetts	4.4	149	85	16.3	0
21	Michigan	12.1	255	74	35.1	0
22	Minnesota	2.7	72	66	14.9	0
23	Mississippi	16.1	259	44	17.1	0
24	Missouri	9.0	178	70	28.2	0
25	Montana	6.0	109	53	16.4	0
26	Nebraska	4.3	102	62	16.5	0
27	Nevada	12.2	252	81	46.0	0
28	New Hampshire	2.1	57	56	9.5	0
29	New Jersey	7.4	159	89	18.8	0
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	0
34	Ohio	7.3	120	75	21.4	0
35	Oklahoma	6.6	151	68	20.0	0
36	Oregon	4.9	159	67	29.3	0
37	Pennsylvania	6.3	106	72	14.9	0
38	Rhode Island	3.4	174	87	8.3	0

Loading [MathJax]/extensions/Safe.js

	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	0
41	Tennessee	13.2	188	59	26.9	0
42	Texas	12.7	201	80	25.5	0
43	Utah	3.2	120	80	22.9	0
44	Vermont	2.2	48	32	11.2	0
45	Virginia	8.5	156	63	20.7	0
46	Washington	4.0	145	73	26.2	0
47	West Virginia	5.7	81	39	9.3	0
48	Wisconsin	2.6	53	66	10.8	0
49	Wyoming	6.8	161	60	15.6	0

In [58]:

```
# Adding clusters to dataset
crim3['clusters']=dbscan.labels_
crim3
```

Out[58]:

	Unnamed: 0	Murder	Assault	UrbanPop	Rape	clusters
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	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	0
6	Connecticut	3.3	110	77	11.1	0
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	0
11	Idaho	2.6	120	54	14.2	0
12	Illinois	10.4	249	83	24.0	0
13	Indiana	7.2	113	65	21.0	0
14	Iowa	2.2	56	57	11.3	0
15	Kansas	6.0	115	66	18.0	0
16	Kentucky	9.7	109	52	16.3	0
17	Louisiana	15.4	249	66	22.2	0
18	Maine	2.1	83	51	7.8	0
19	Maryland	11.3	300	67	27.8	0
20	Massachusetts	4.4	149	85	16.3	0
21	Michigan	12.1	255	74	35.1	0
22	Minnesota	2.7	72	66	14.9	0
23	Mississippi	16.1	259	44	17.1	0
24	Missouri	9.0	178	70	28.2	0
25	Montana	6.0	109	53	16.4	0
26	Nebraska	4.3	102	62	16.5	0
27	Nevada	12.2	252	81	46.0	0
28	New Hampshire	2.1	57	56	9.5	0
29	New Jersey	7.4	159	89	18.8	0
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	0
34	Ohio	7.3	120	75	21.4	0
35	Oklahoma	6.6	151	68	20.0	0
36	Oregon	4.9	159	67	29.3	0
37	Pennsylvania	6.3	106	72	14.9	0
38	Rhode Island	3.4	174	87	8.3	0

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

In [59]:
```
crim3.groupby('clusters').agg(['mean']).reset_index()
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_22220\3983941485.py:1: FutureWarning: ['Unname
d: 0'] did not aggregate successfully. If any error is raised this will raise in a futur
e version of pandas. Drop these columns/ops to avoid this warning.
  crim3.groupby('clusters').agg(['mean']).reset_index()
```

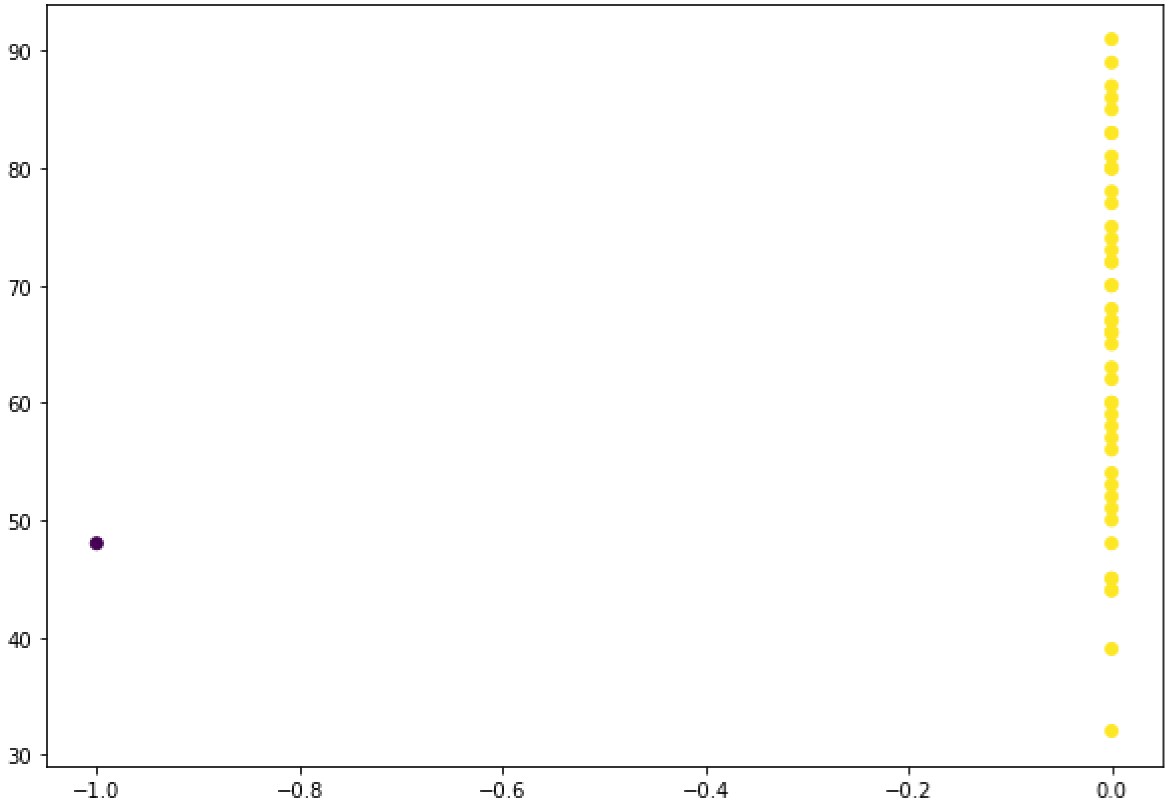
Out[59]:

	clusters	Murder	Assault	UrbanPop	Rape
		mean	mean	mean	mean
0	-1	10.000000	263.000000	48.000000	44.500000
1	0	7.742857	168.877551	65.897959	20.757143

In [60]:
```
# Plot Clusters
plt.figure(figsize=(10, 7))
plt.scatter(crim3['clusters'],crim3['UrbanPop'], c=dbscan.labels_)
```

Out[60]:

```
<matplotlib.collections.PathCollection at 0x2b3274a2d00>
```



Assignment-07-Clustering-Hierarchical (Airlines)

Using Normalize Function

In [77]:
```
# 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.preprocessing import normalize
```

In [78]:
```
# Import Dataset
airline=pd.read_csv('Airlines.csv')
airline
```

Out[78]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12
0	1	28143	0	1	1	1	174	1	
1	2	19244	0	1	1	1	215	2	
2	3	41354	0	1	1	1	4123	4	
3	4	14776	0	1	1	1	500	1	
4	5	97752	0	4	1	1	43300	26	20
...	...	...	...	...	...	...	...	...	
3994	4017	18476	0	1	1	1	8525	4	2
3995	4018	64385	0	1	1	1	981	5	
3996	4019	73597	0	3	1	1	25447	8	
3997	4020	54899	0	1	1	1	500	1	5
3998	4021	3016	0	1	1	1	0	0	

3999 rows × 12 columns

In [79]:

```
airline.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3999 entries, 0 to 3998
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   ID#                3999 non-null   int64
 1   Balance            3999 non-null   int64
 2   Qual_miles         3999 non-null   int64
 3   cc1_miles          3999 non-null   int64
 4   cc2_miles          3999 non-null   int64
 5   cc3_miles          3999 non-null   int64
 6   Bonus_miles        3999 non-null   int64
 7   Bonus_trans        3999 non-null   int64
 8   Flight_miles_12mo  3999 non-null   int64
 9   Flight_trans_12    3999 non-null   int64
 10  Days_since_enroll  3999 non-null   int64
 11  Award?             3999 non-null   int64
dtypes: int64(12)
memory usage: 375.0 KB
```

In [80]:

```
airline2=airline.drop(['ID#'],axis=1)
airline2
```

Out[80]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Fl
0	28143	0	1	1	1	174	1	0	
1	19244	0	1	1	1	215	2	0	
2	41354	0	1	1	1	4123	4	0	
3	14776	0	1	1	1	500	1	0	
4	97752	0	4	1	1	43300	26	2077	
...	...	...	...	...	...	...	...	...	
3994	18476	0	1	1	1	8525	4	200	
3995	64385	0	1	1	1	981	5	0	
3996	73597	0	3	1	1	25447	8	0	
3997	54899	0	1	1	1	500	1	500	
3998	3016	0	1	1	1	0	0	0	

3999 rows × 11 columns

In [81]:

```
# Normalize heterogenous numerical data
airline2_norm=pd.DataFrame(normalize(airline2),columns=airline2.columns)
airline2_norm
```

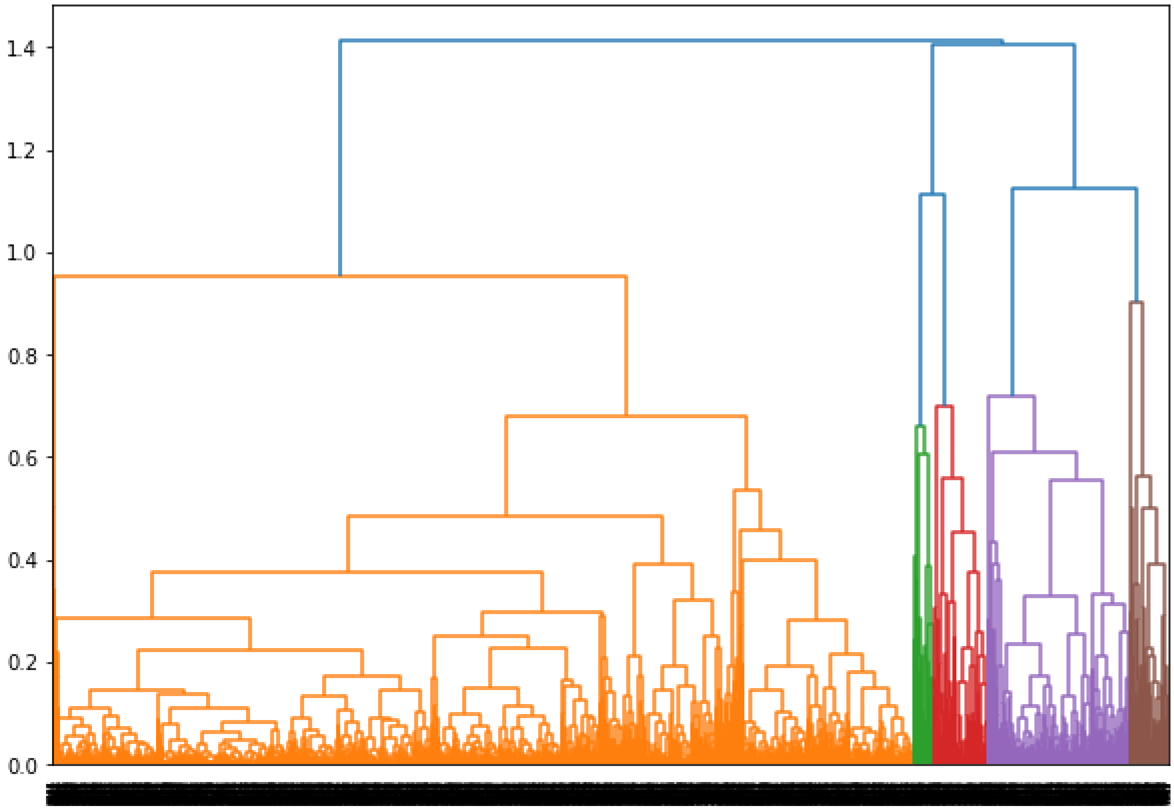
Out[81]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	F
0	0.970414	0.0	0.000034	0.000034	0.000034	0.006000	0.000034	0.000000	
1	0.940209	0.0	0.000049	0.000049	0.000049	0.010504	0.000098	0.000000	
2	0.981113	0.0	0.000024	0.000024	0.000024	0.097817	0.000095	0.000000	
3	0.904428	0.0	0.000061	0.000061	0.000061	0.030605	0.000061	0.000000	
4	0.912226	0.0	0.000037	0.000009	0.000009	0.404078	0.000243	0.019383	
...	...	...	...	...	...	...	...	...	
3994	0.905810	0.0	0.000049	0.000049	0.000049	0.417949	0.000196	0.009805	
3995	0.999649	0.0	0.000016	0.000016	0.000016	0.015231	0.000078	0.000000	
3996	0.944948	0.0	0.000039	0.000013	0.000013	0.326726	0.000103	0.000000	
3997	0.999592	0.0	0.000018	0.000018	0.000018	0.009104	0.000018	0.009104	
3998	0.907271	0.0	0.000301	0.000301	0.000301	0.000000	0.000000	0.000000	

3999 rows × 11 columns

In [89]:

```
# Create Dendrograms
plt.figure(figsize=(10, 7))
dendograms=sch.dendrogram(sch.linkage(airline2_norm,'complete'))
```



In [86]:
```
# Create Clusters (y)
hclusters=AgglomerativeClustering(n_clusters=5,affinity='euclidean',linkage='ward')
hclusters
```

Out[86]: AgglomerativeClustering(n_clusters=5)

In [87]:
```
y=pd.DataFrame(hclusters.fit_predict(airline2_norm),columns=['clustersid'])
y['clustersid'].value_counts()
```

Out[87]:
```
2    1547
4    1191
3     579
1     453
0     229
Name: clustersid, dtype: int64
```

In [93]:
```
# Adding clusters to dataset
airline2['clustersid']=hclusters.labels_
airline2
```

Out[93]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Fl
0	28143	0	1	1	1	174	1	0	
1	19244	0	1	1	1	215	2	0	
2	41354	0	1	1	1	4123	4	0	
3	14776	0	1	1	1	500	1	0	
4	97752	0	4	1	1	43300	26	2077	
...	...	...	...	...	...	...	...	...	
3994	18476	0	1	1	1	8525	4	200	
3995	64385	0	1	1	1	981	5	0	
3996	73597	0	3	1	1	25447	8	0	
3997	54899	0	1	1	1	500	1	500	
3998	3016	0	1	1	1	0	0	0	

3999 rows × 12 columns

In [94]:

```
airline2.groupby('clustersid').agg(['mean']).reset_index()
```

Out[94]:

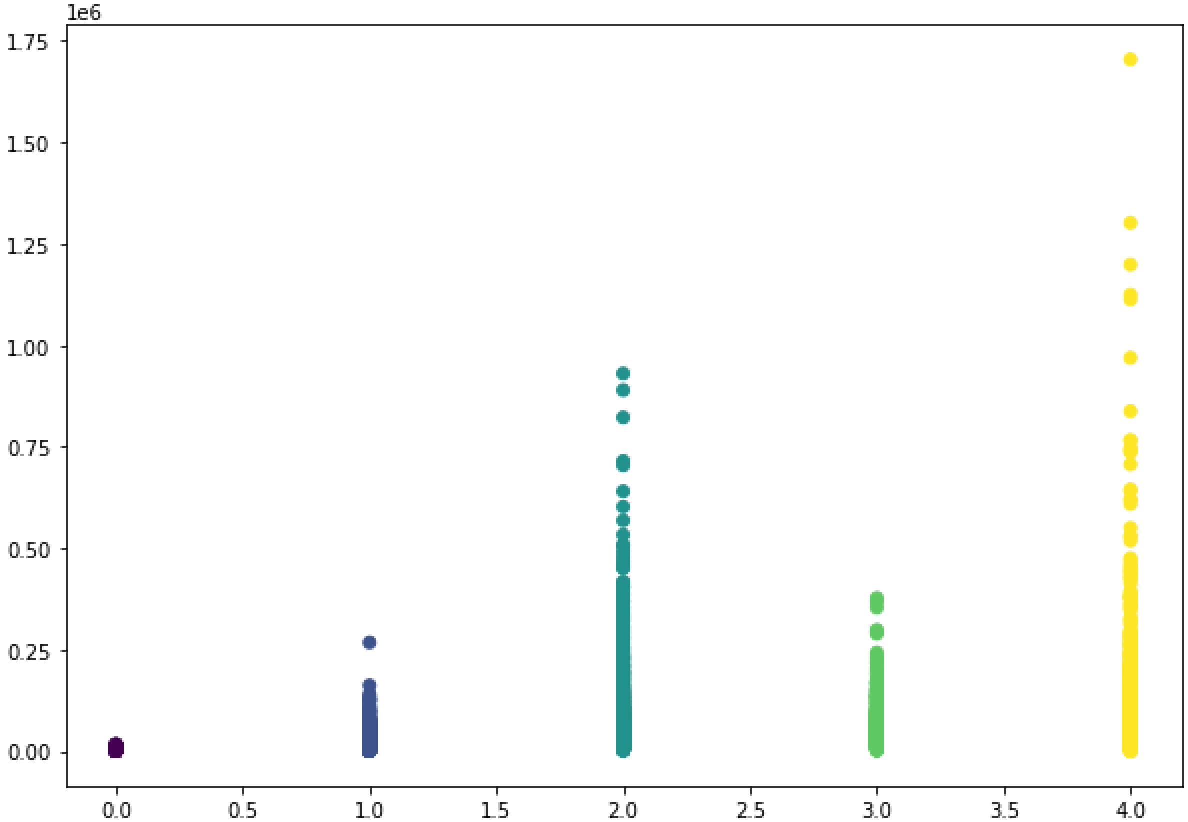
	clustersid	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_m
		mean	mean	mean	mean	mean	mean	mean	
0	0	5524.222707	8.755459	1.000000	1.000000	1.000000	584.532751	2.401747	
1	1	31066.514349	111.415011	3.200883	1.026490	1.070640	40266.935982	17.289183	6
2	2	81201.080802	136.521008	2.115061	1.013575	1.000646	16350.149968	13.574014	4
3	3	69569.894646	97.257340	3.326425	1.032815	1.022453	35743.675302	17.784111	4
4	4	94957.590260	215.220823	1.141058	1.005038	1.002519	3524.928631	5.640638	4

In [95]:

```
# Plot Clusters
plt.figure(figsize=(10, 7))
plt.scatter(airline2['clustersid'],airline2['Balance'], c=hclusters.labels_)
```

Out[95]:

```
<matplotlib.collections.PathCollection at 0x2b329d06400>
```




```
In [ ]:
```