In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
```

In [2]:
```python
# Import Dataset
crime = pd.read_csv('crime_data.csv')
crime
```

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

	Unnamed: 0	Murder	Assault	UrbanPop	Rape
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
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]:
```
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

	Murder	Assault	UrbanPop	Rape
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 [6]:
```
# Normalize heterogenous numerical data using standard scalar fit transform to dc
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]])
```

In [7]:
```
# DBSCAN Clustering
dbscan=DBSCAN(eps=1,min_samples=4)
dbscan.fit(crime_norm)
```

Out[7]:
```
DBSCAN(eps=1, min_samples=4)
```

In [8]:
```
#Noisy samples are given the label -1.
dbscan.labels_
```

Out[8]:
```
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 [9]:
```python
# Adding clusters to dataset
crime['clusters']=dbscan.labels_
crime
```

Out[9]:

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

	Murder	Assault	UrbanPop	Rape	clusters
33	0.8	45	44	7.3	1
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
38	3.4	174	87	8.3	1
39	14.4	279	48	22.5	0
40	3.8	86	45	12.8	1
41	13.2	188	59	26.9	0
42	12.7	201	80	25.5	-1
43	3.2	120	80	22.9	1
44	2.2	48	32	11.2	1
45	8.5	156	63	20.7	1
46	4.0	145	73	26.2	1
47	5.7	81	39	9.3	1
48	2.6	53	66	10.8	1
49	6.8	161	60	15.6	1

In [10]:

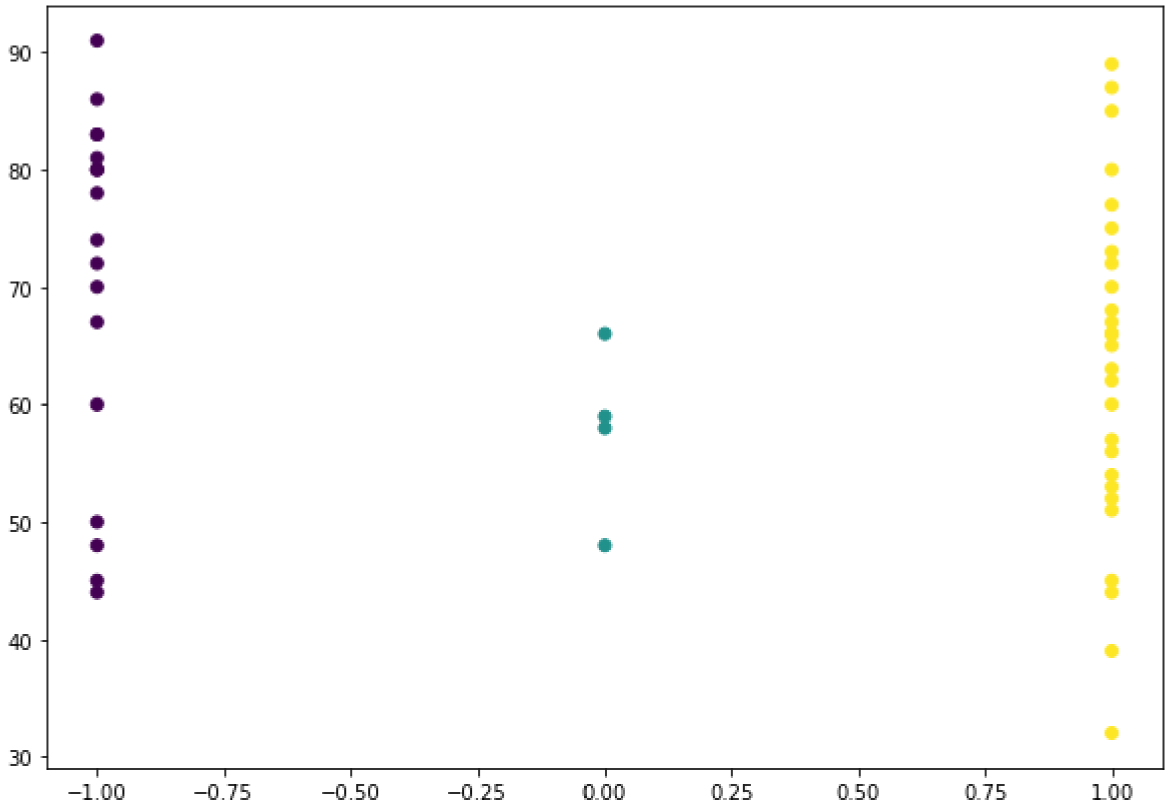
```
crime.groupby('clusters').agg(['mean']).reset_index()
```

Out[10]:

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

In [11]:
```python
# Plot Clusters
plt.figure(figsize=(10, 7))
plt.scatter(crime['clusters'],crime['UrbanPop'], c=dbscan.labels_)
```

Out[11]: <matplotlib.collections.PathCollection at 0x184845acf10>



In [ ]: