Perform Clustering(Hierarchical, Kmeans & DBSCAN) for the crime data and identify the number of clusters formed and draw inferences.

Data Description: Murder -- Muder rates in different places of United States
Assualt- Assualt rate in different places of United States
UrbanPop - urban population in different places of United States
Rape - Rape rate in different places of United States

In [1]:
```
import pandas as pd
import numpy as np
import pandas as pd
from pandas import plotting

# for visualizations
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('fivethirtyeight')

# for interactive visualizations
import plotly.offline as py
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go
```

In [2]:
```
data_ = pd.read_csv("crime_data.csv").rename({"Unnamed: 0":"State"}, axis =1)
data_
```

Out[2]:

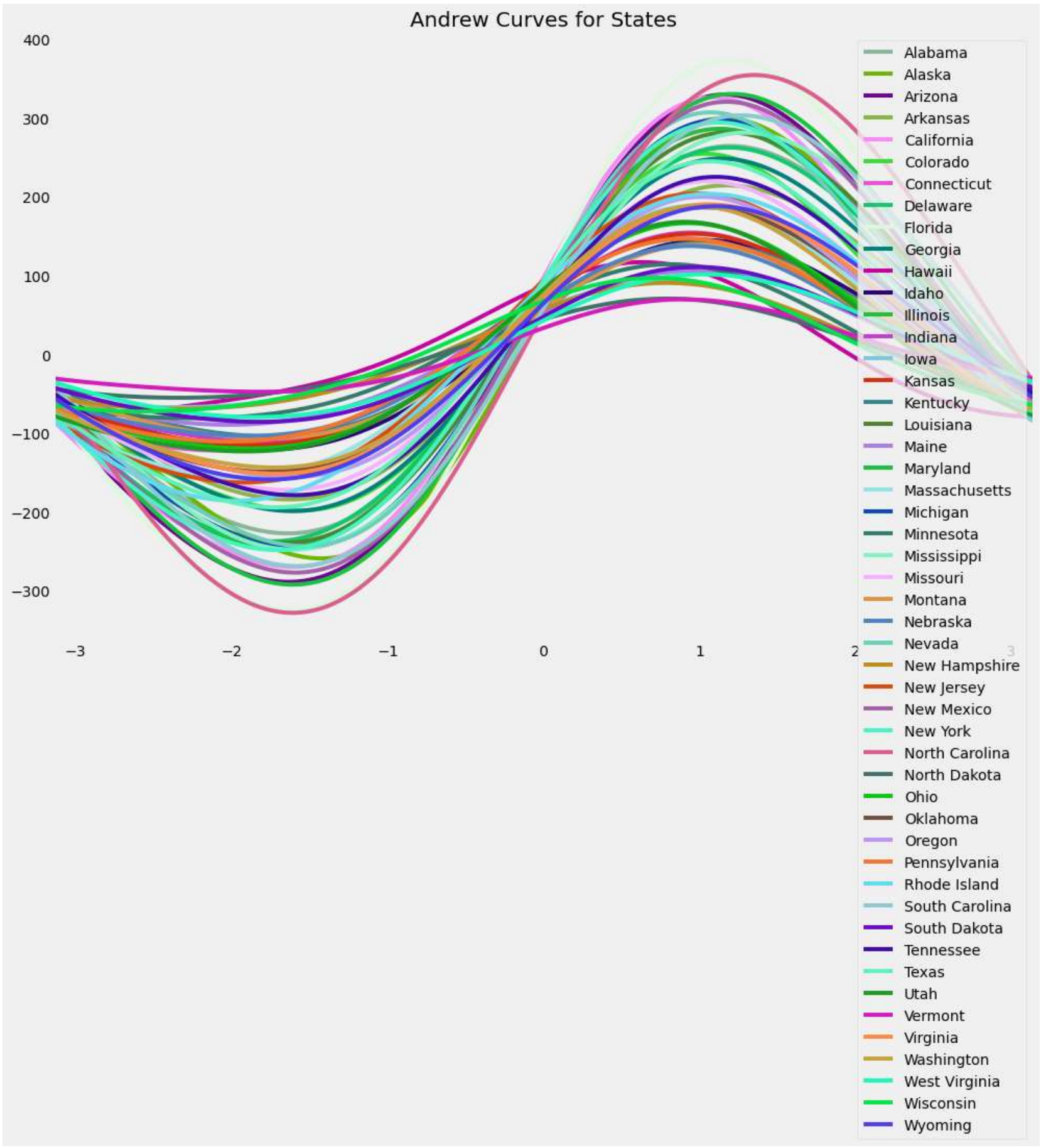
	State	Murder	Assault	UrbanPop	Rape
0	Alabama	13.2	236	58	21.2
1	Alaska	10.0	263	48	44.5
2	Arizona	8.1	294	80	31.0
3	Arkansas	8.8	190	50	19.5
4	California	9.0	276	91	40.6
5	Colorado	7.9	204	78	38.7
6	Connecticut	3.3	110	77	11.1
7	Delaware	5.9	238	72	15.8
8	Florida	15.4	335	80	31.9
9	Georgia	17.4	211	60	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
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_.isnull().any().any()
```

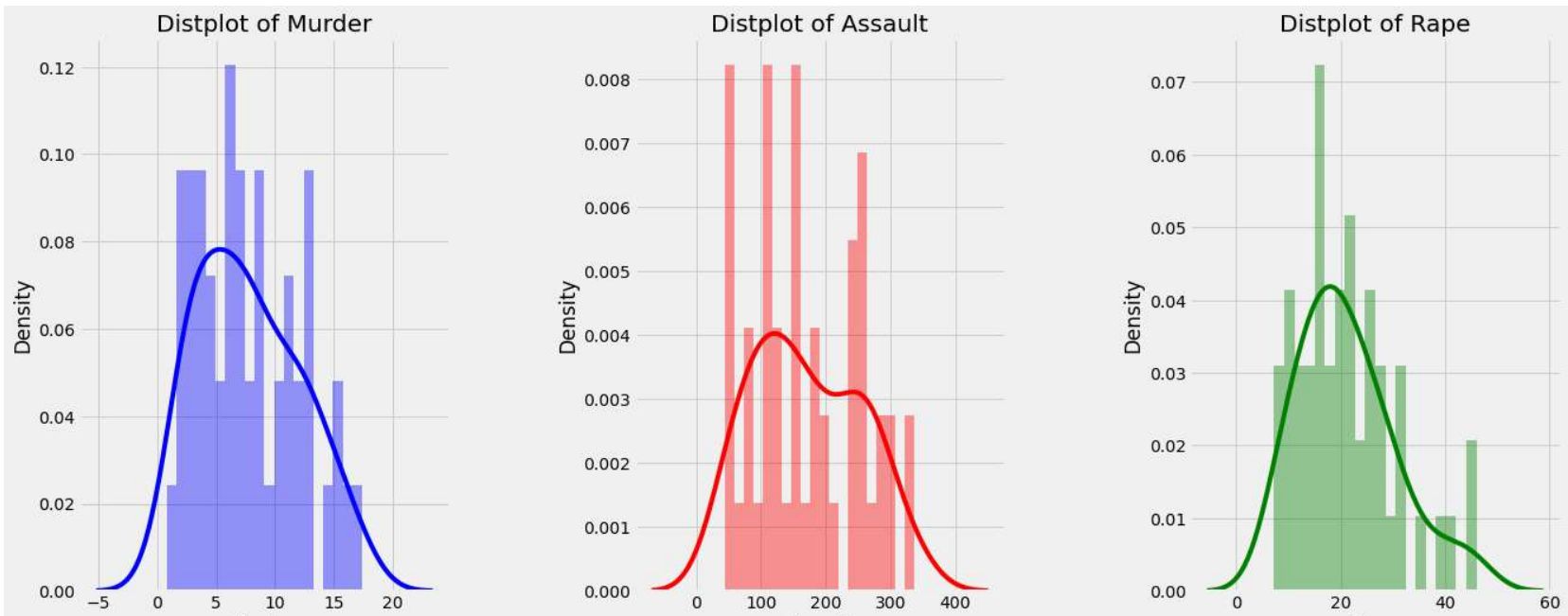
Out[3]: False

Let's see if we can find any pattern in our multivariate data using Andrew's curve.
Note: Andrews curves are a method for visualizing multidimensional data by mapping each observation onto a function.
Andrews curves that are represented by functions close together suggest that the corresponding data points will also be close together.

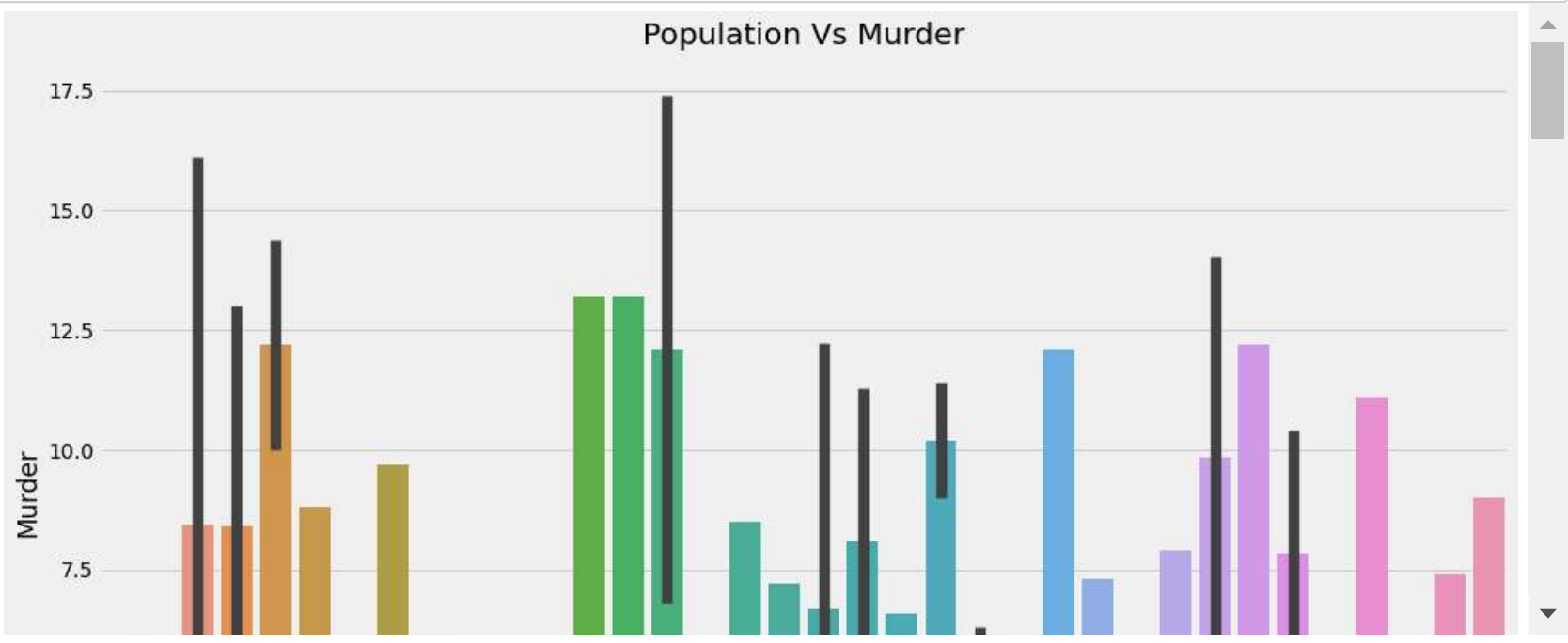
In [4]:
```
plt.rcParams['figure.figsize'] = (15, 10)


plotting.andrews_curves(data_, "State")
plt.title('Andrew Curves for States', fontsize = 20)
plt.show()
```



In [5]:
```
import warnings
warnings.filterwarnings('ignore')


plt.figure(1 , figsize = (20 , 8))
n = 0
for x , i in zip(['Murder' , 'Assault' , 'Rape'], ['blue', 'red', 'green']):
    n += 1
    plt.subplot(1 , 3 , n)
    plt.subplots_adjust(hspace =0.5 , wspace = 0.5)
    sns.distplot(data_[x] , bins = 20, color = i)
    plt.title('Distplot of {}'.format(x))
plt.show()
```

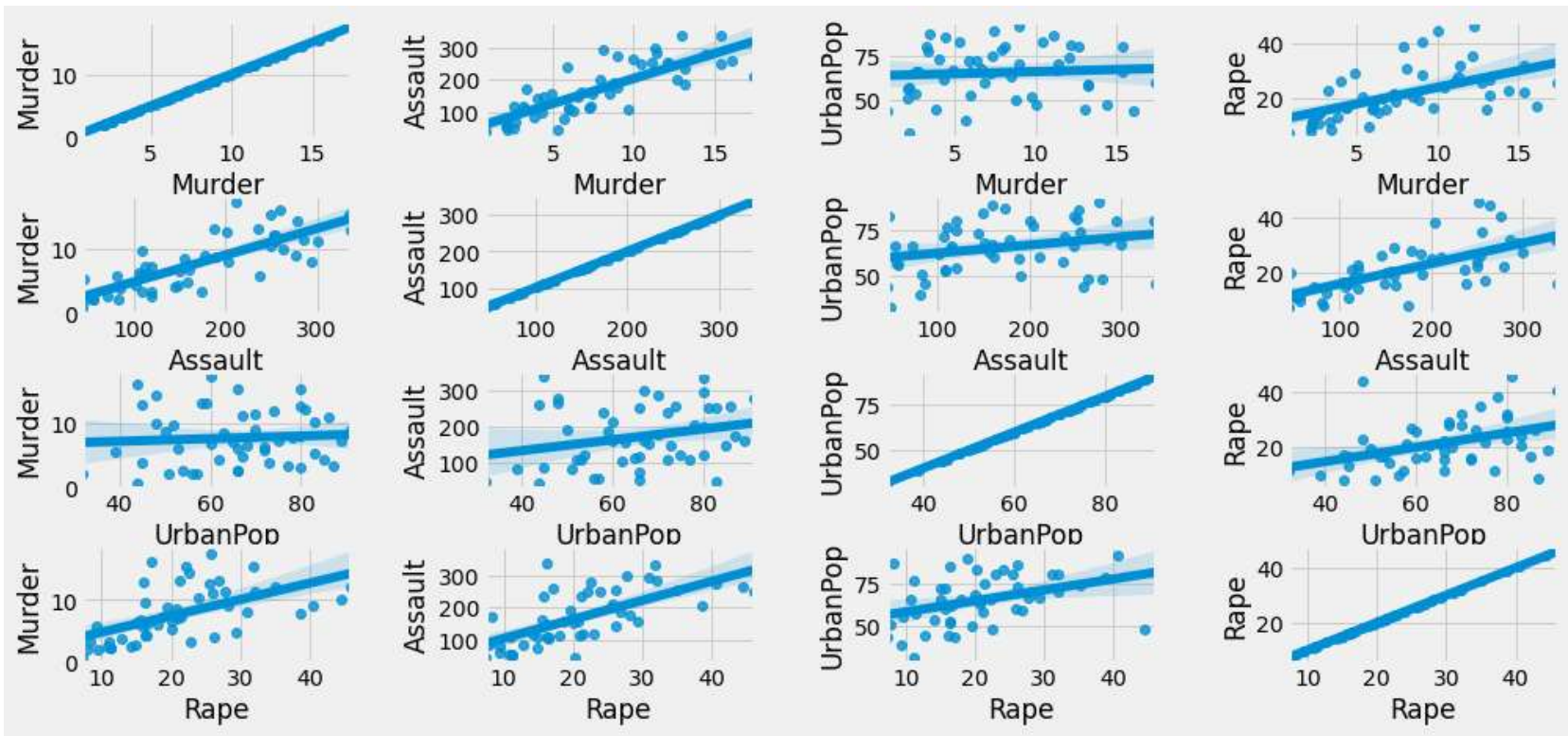


In [6]:
```
for i in ["Murder", "Assault", "Rape"]:
    sns.barplot(x="UrbanPop", y=i, data=data_)
    plt.title('Population Vs {}'.format(i))
    plt.show()
```



Relation between all the variables

In [7]:
```
plt.figure(1 , figsize = (15 , 7))
n = 0
for x in ['Murder', 'Assault', 'UrbanPop', 'Rape']:
    for y in ['Murder', 'Assault', 'UrbanPop', 'Rape']:
        n += 1
        plt.subplot(4 , 4 , n)
        plt.subplots_adjust(hspace = 0.5 , wspace = 0.5)
        sns.regplot(x = x , y = y , data = data_)
        plt.ylabel(y.split()[0]+' '+y.split()[1] if len(y.split()) > 1 else y )
plt.show()
```



In [8]:
```
from sklearn import preprocessing
crime_rates_standardized = preprocessing.scale(data_[['Murder', 'Assault', 'UrbanPop', 'Rape']])
print(crime_rates_standardized)
crime_data = pd.DataFrame(crime_rates_standardized)
```

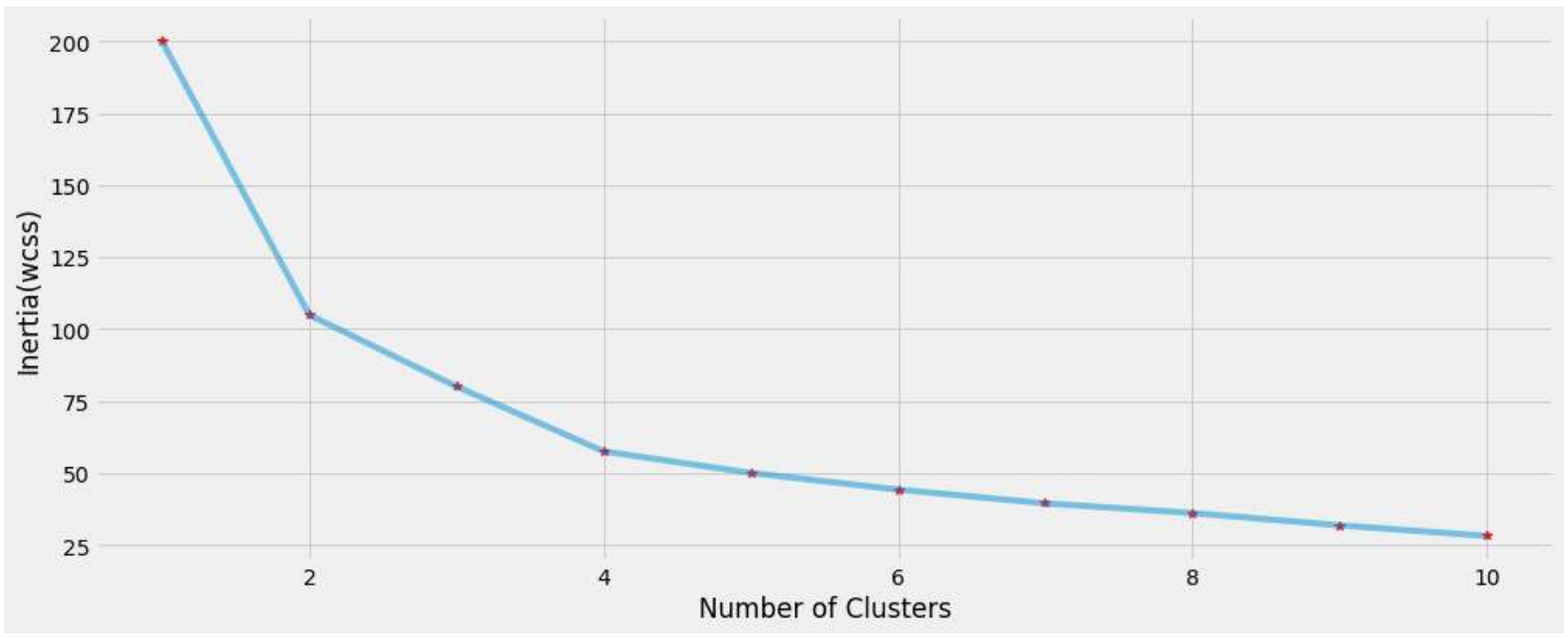
```
[[ 1.25517927  0.79078716 -0.52619514 -0.00345116]
 [ 0.51301858  1.11805959 -1.22406668  2.50942392]
 [ 0.07236067  1.49381682  1.00912225  1.05346626]
 [ 0.23470832  0.23321191 -1.08449238 -0.18679398]
 [ 0.28109336  1.2756352   1.77678094  2.08881393]
 [ 0.02597562  0.40290872  0.86954794  1.88390137]
 [-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 [9]:
```
# K-Means clustering
X1 = crime_data.iloc[:,:].values
X = crime_data.iloc[:,:].values
X2 = data_.iloc[:,1:].values
```

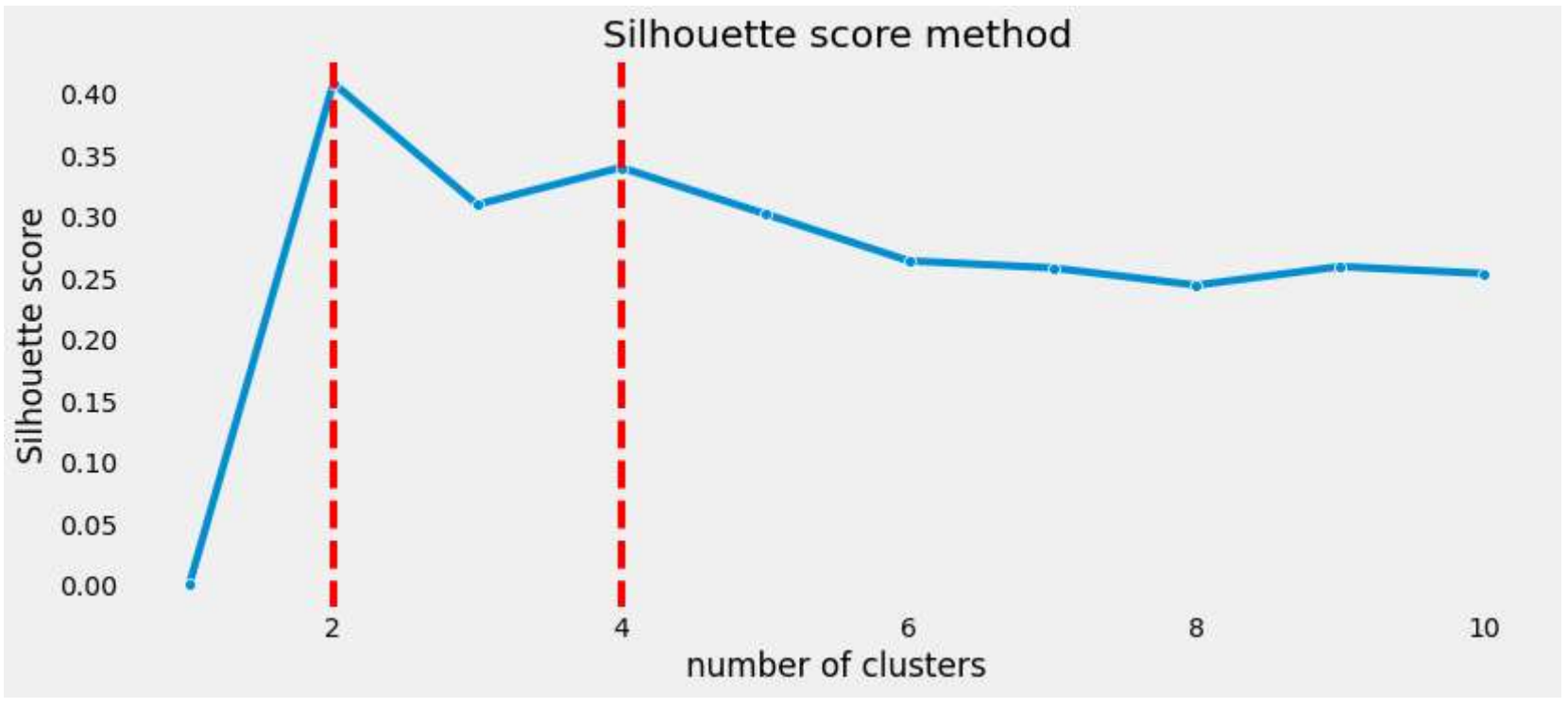
In [10]:
```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

wcss =[]
s_scores = []
for i in range(1,11): # Silhoutte score reqy=uires more than one cluster otherwise it will throw an error
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=111, algorithm = 'elkan')
    kmeans.fit(X1)
    #print(np.unique(kmeans.labels_))
    if i!=1:
        silhouette_avg = silhouette_score(X1, kmeans.labels_)
    else:
        silhouette_avg=0

    print("K-Means: for n_clusters =", i, ", the average silhouette_score is", silhouette_avg)

    wcss.append(kmeans.inertia_)
    s_scores.append(silhouette_avg)
```

```
K-Means: for n_clusters = 1 , the average silhouette_score is 0
K-Means: for n_clusters = 2 , the average silhouette_score is 0.4084890326217641
K-Means: for n_clusters = 3 , the average silhouette_score is 0.3095091512791688
K-Means: for n_clusters = 4 , the average silhouette_score is 0.33968891433344395
K-Means: for n_clusters = 5 , the average silhouette_score is 0.30197451297119854
K-Means: for n_clusters = 6 , the average silhouette_score is 0.26358689876767544
K-Means: for n_clusters = 7 , the average silhouette_score is 0.25783033161129554
K-Means: for n_clusters = 8 , the average silhouette_score is 0.24376794652232758
K-Means: for n_clusters = 9 , the average silhouette_score is 0.25899868946628024
K-Means: for n_clusters = 10 , the average silhouette_score is 0.2532530967149398
```

In [11]:
```
plt.figure(1 , figsize = (15 ,6))
plt.plot(np.arange(1 , 11) , wcss , '*', color ='red')
plt.plot(np.arange(1 , 11) , wcss , '-' , alpha = 0.5)
plt.xlabel('Number of Clusters') , plt.ylabel('Inertia(wcss)')
plt.show()
```



In [12]:
```
fig, ax = plt.subplots(figsize=(12,5))
ax = sns.lineplot(np.arange(1 , 11), s_scores, marker='o', ax=ax)
ax.set_title("Silhouette score method")
ax.set_xlabel("number of clusters")
ax.set_ylabel("Silhouette score")
ax.axvline(2, ls="--", c="red")
ax.axvline(4, ls="--", c="red")
plt.grid()
plt.show()
```



In [13]:
```
kmeans = KMeans(n_clusters=2, init='k-means++', random_state=111, algorithm = 'elkan')
y_kmeans = kmeans.fit_predict(X1)
labels = kmeans.labels_
centroids = kmeans.cluster_centers_
```

In [14]:
```
print(y_kmeans)
```

```
[1 1 1 0 1 1 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 1 0 1 1 0 0 1 0 0 1 1 1 0 0 0 0
 0 0 1 0 1 1 0 0 0 0 0 0 0]
```

In [15]:
```
data_
```

Out[15]:

	State	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
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

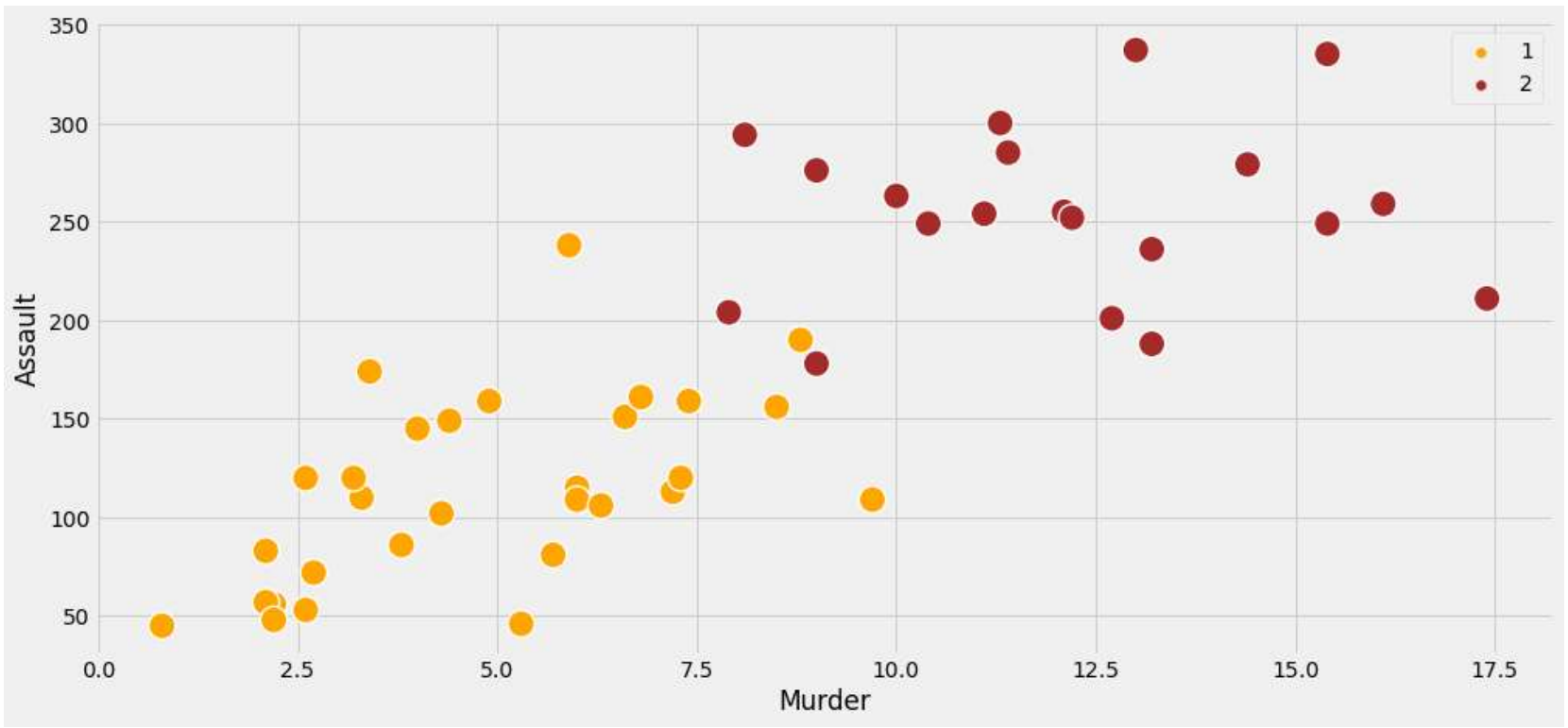
In [16]:
```
y_kmeans_1 = y_kmeans +1
clusters = list(y_kmeans_1)
data_["Cluster"] = clusters
data_
```

Out[16]:

	State	Murder	Assault	UrbanPop	Rape	Cluster
0	Alabama	13.2	236	58	21.2	2
1	Alaska	10.0	263	48	44.5	2
2	Arizona	8.1	294	80	31.0	2
3	Arkansas	8.8	190	50	19.5	1
4	California	9.0	276	91	40.6	2
5	Colorado	7.9	204	78	38.7	2
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	2
9	Georgia	17.4	211	60	25.8	2
10	Hawaii	5.3	46	83	20.2	1
11	Idaho	2.6	120	54	14.2	1
12	Illinois	10.4	249	83	24.0	2
13	Indiana	7.2	113	65	21.0	1
14	Iowa	2.2	56	57	11.3	1
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	2
18	Maine	2.1	83	51	7.8	1
19	Maryland	11.3	300	67	27.8	2
20	Massachusetts	4.4	149	85	16.3	1
21	Michigan	12.1	255	74	35.1	2
22	Minnesota	2.7	72	66	14.9	1
23	Mississippi	16.1	259	44	17.1	2
24	Missouri	9.0	178	70	28.2	2
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	2
28	New Hampshire	2.1	57	56	9.5	1
29	New Jersey	7.4	159	89	18.8	1
30	New Mexico	11.4	285	70	32.1	2
31	New York	11.1	254	86	26.1	2
32	North Carolina	13.0	337	45	16.1	2
33	North Dakota	0.8	45	44	7.3	1
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
39	South Carolina	14.4	279	48	22.5	2
40	South Dakota	3.8	86	45	12.8	1
41	Tennessee	13.2	188	59	26.9	2
42	Texas	12.7	201	80	25.5	2
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	1
46	Washington	4.0	145	73	26.2	1
47	West Virginia	5.7	81	39	9.3	1
48	Wisconsin	2.6	53	66	10.8	1
49	Wyoming	6.8	161	60	15.6	1

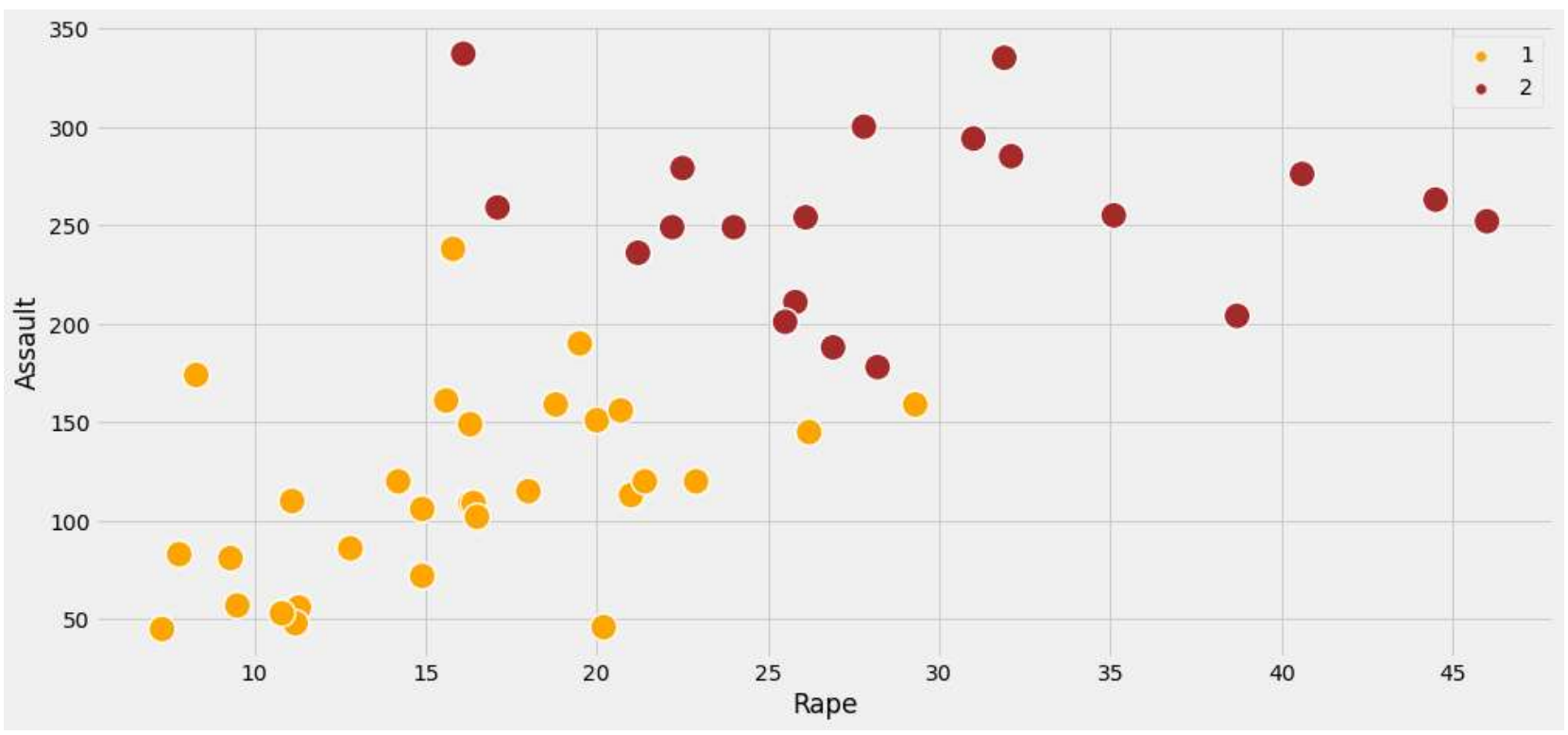
In [17]:
```
import seaborn as sns

plt.figure(figsize=(15,7))
sns.scatterplot(x=data_['Murder'], y = data_['Assault'],hue=y_kmeans_1, s=300, palette=['orange','brown'], l
```

Out[17]: <AxesSubplot:xlabel='Murder', ylabel='Assault'>



In [18]:
```
import seaborn as sns

plt.figure(figsize=(15,7))
sns.scatterplot(x=data_['Rape'], y = data_['Assault'],hue=y_kmeans_1, s=300, palette=['orange','brown'], leg
```

Out[18]: <AxesSubplot:xlabel='Rape', ylabel='Assault'>



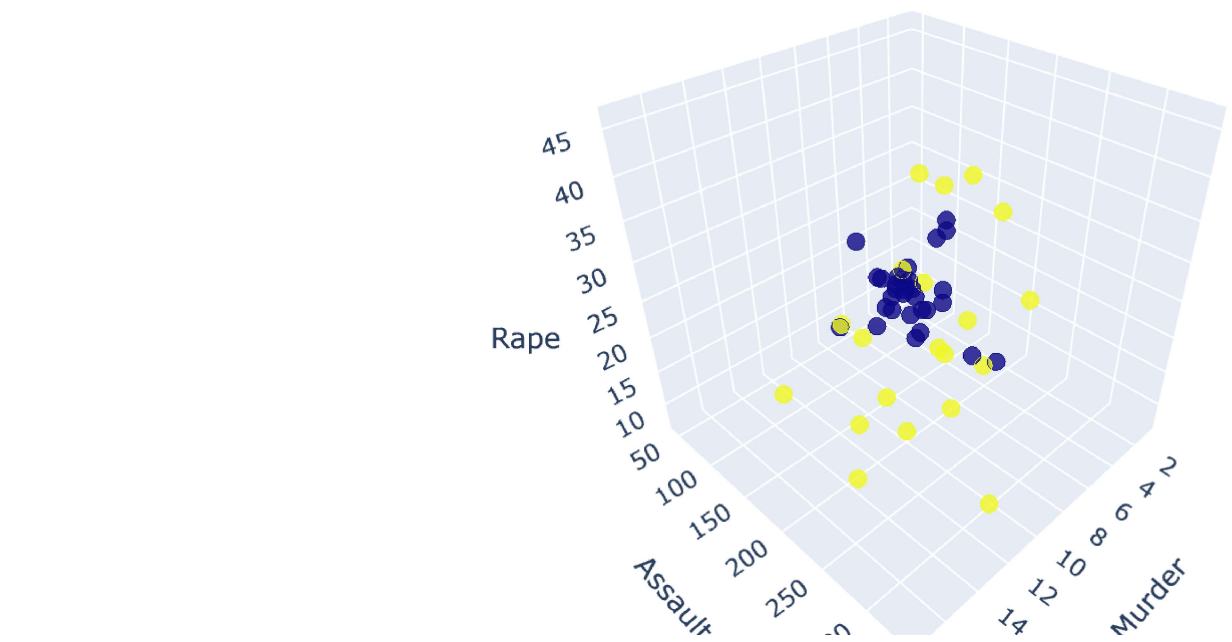
In [19]:
```
data_
```

Out[19]:

	State	Murder	Assault	UrbanPop	Rape	Cluster
0	Alabama	13.2	236	58	21.2	2
1	Alaska	10.0	263	48	44.5	2
2	Arizona	8.1	294	80	31.0	2
3	Arkansas	8.8	190	50	19.5	1
4	California	9.0	276	91	40.6	2
5	Colorado	7.9	204	78	38.7	2
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	2
9	Georgia	17.4	211	60	25.8	2
10	Hawaii	5.3	46	83	20.2	1
11	Idaho	2.6	120	54	14.2	1
12	Illinois	10.4	249	83	24.0	2
13	Indiana	7.2	113	65	21.0	1
14	Iowa	2.2	56	57	11.3	1
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	2
18	Maine	2.1	83	51	7.8	1
19	Maryland	11.3	300	67	27.8	2
20	Massachusetts	4.4	149	85	16.3	1
21	Michigan	12.1	255	74	35.1	2
22	Minnesota	2.7	72	66	14.9	1
23	Mississippi	16.1	259	44	17.1	2
24	Missouri	9.0	178	70	28.2	2
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	2
28	New Hampshire	2.1	57	56	9.5	1
29	New Jersey	7.4	159	89	18.8	1
30	New Mexico	11.4	285	70	32.1	2
31	New York	11.1	254	86	26.1	2
32	North Carolina	13.0	337	45	16.1	2
33	North Dakota	0.8	45	44	7.3	1
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
39	South Carolina	14.4	279	48	22.5	2
40	South Dakota	3.8	86	45	12.8	1
41	Tennessee	13.2	188	59	26.9	2
42	Texas	12.7	201	80	25.5	2
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	1
46	Washington	4.0	145	73	26.2	1
47	West Virginia	5.7	81	39	9.3	1
48	Wisconsin	2.6	53	66	10.8	1
49	Wyoming	6.8	161	60	15.6	1

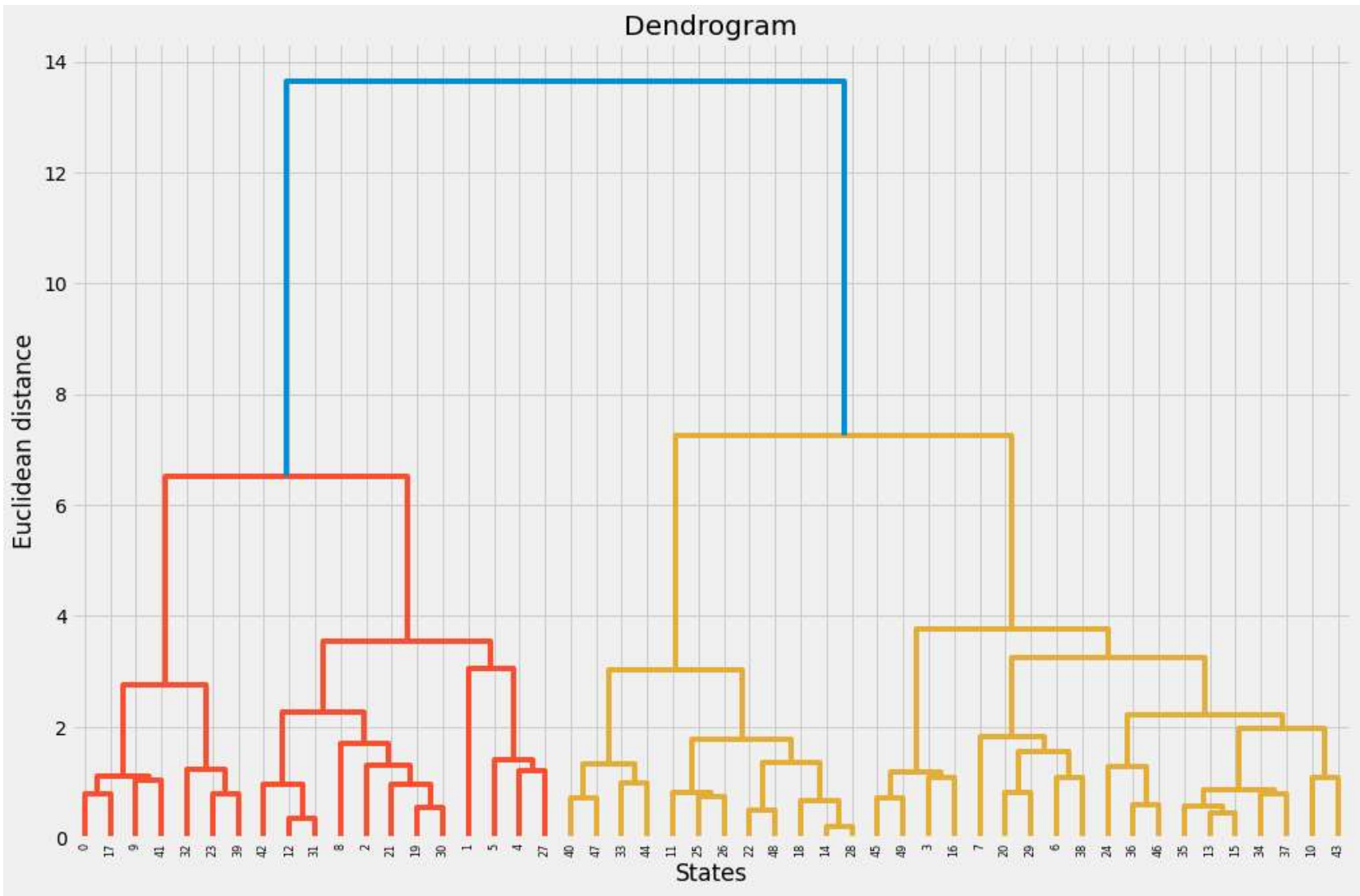
In [20]:

```
data_['label'] =  labels
trace1 = go.Scatter3d(
    x= data_['Murder'],
    y= data_['Assault'],
    z= data_['Rape'],
   # z1= crime_data['UrbanPop'],
    mode='markers',
     marker=dict(
        color = data_['label'],
        size= 5,
        line=dict(
            color= data_['label'],
            width= 12
        ),
        opacity=0.8
     )
)
data = [trace1]
layout = go.Layout(
#     margin=dict(
#         l=0,
#         r=0,
#         b=0,
#         t=0
#     )
    title= 'Clusters',
    scene = dict(
            xaxis = dict(title  = 'Murder'),
            yaxis = dict(title  = 'Assault'),
            zaxis = dict(title  = 'Rape')
        )
)
fig = go.Figure(data=data, layout=layout)
py.offline.iplot(fig)
```



Hierarchical clustering

In [21]:
```
import scipy.cluster.hierarchy as sch
dendrogram = sch.dendrogram(sch.linkage(X1, method = 'ward'))
plt.title('Dendrogram')
plt.xlabel('States')
plt.ylabel('Euclidean distance')
plt.show()
```



In [22]:
```
from sklearn.cluster import AgglomerativeClustering
HC = AgglomerativeClustering(n_clusters=2, affinity='euclidean', linkage = 'ward')
y_HC = HC.fit_predict(X1)
```

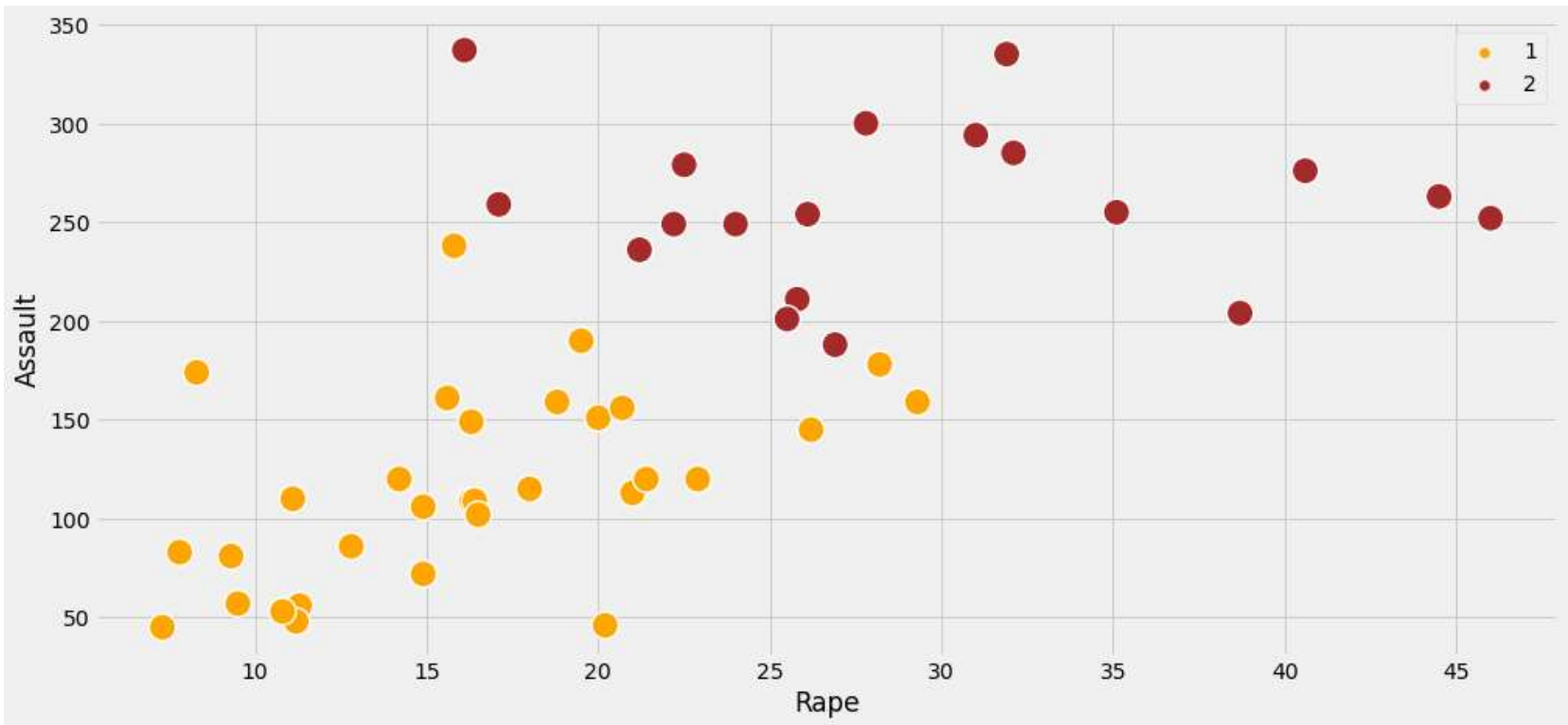
In [23]:
```
y_HC_1 = y_HC +1
clusters2 = list(y_HC_1)
data_["HierarchicalCluster"] = clusters2
data_.head()
```

Out[23]:

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

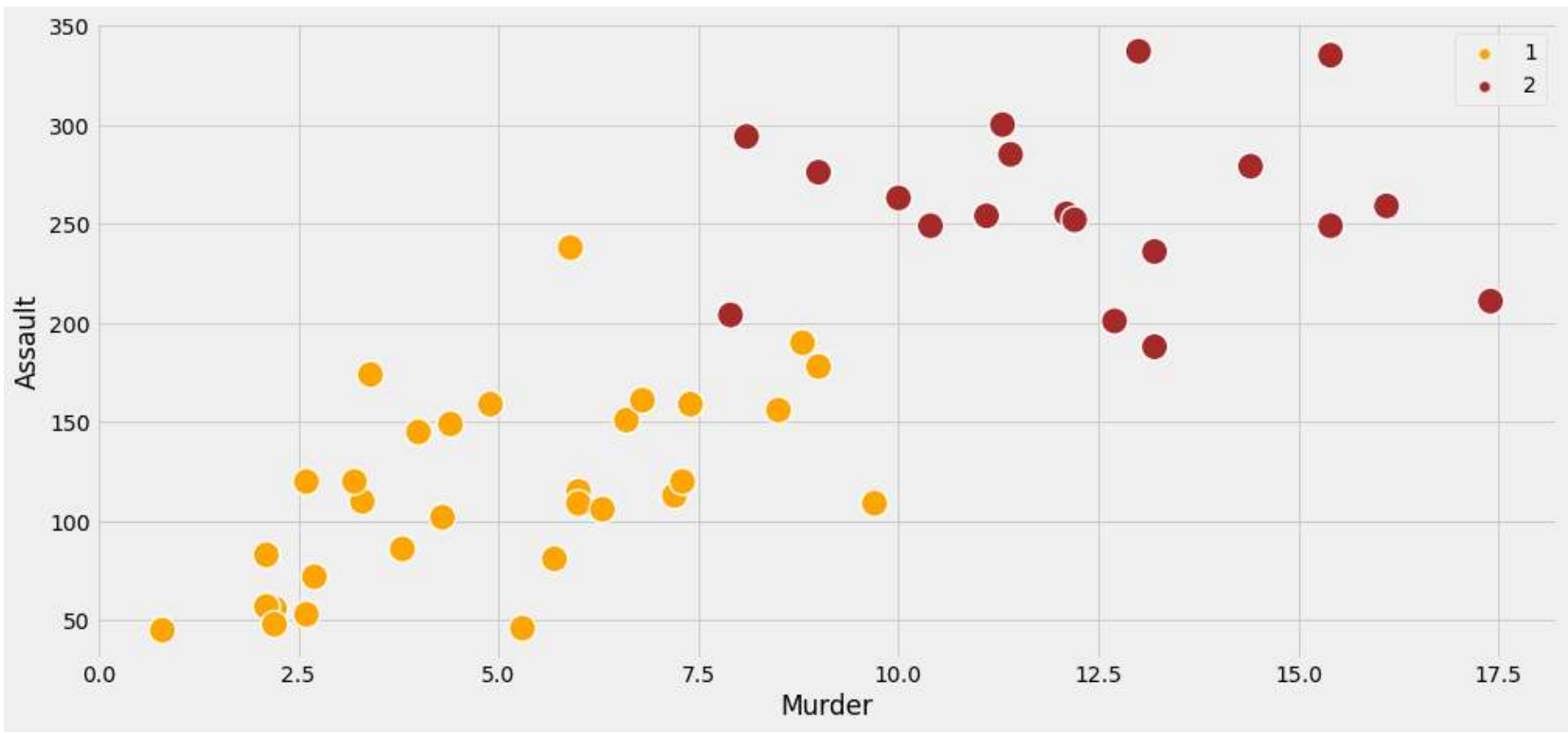
In [24]:
```
import seaborn as sns

plt.figure(figsize=(15,7))
sns.scatterplot(x=data_['Rape'], y = data_['Assault'],hue=y_HC_1, s=300, palette=['orange','brown'], legend=
```

Out[24]: <AxesSubplot:xlabel='Rape', ylabel='Assault'>

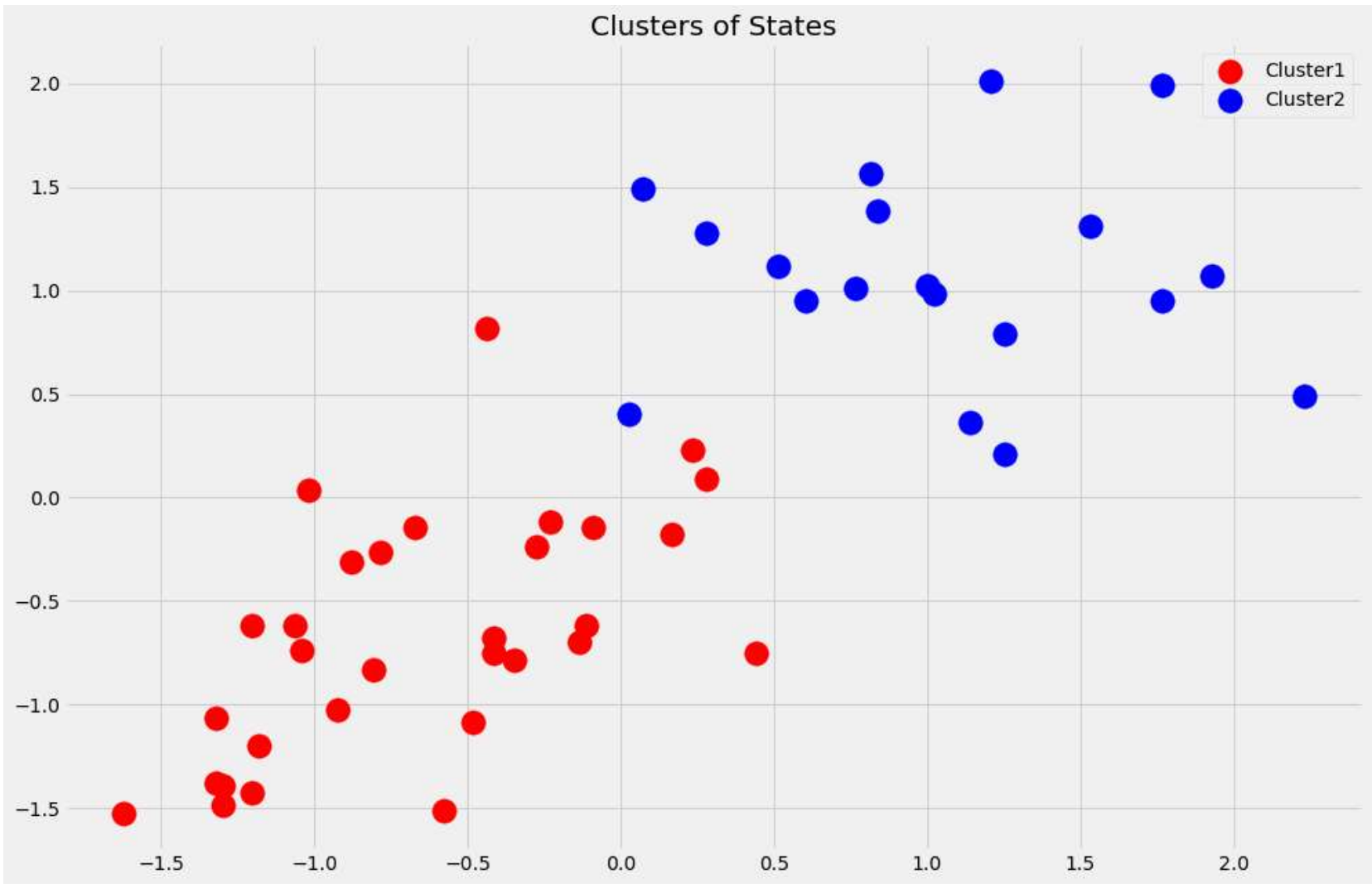


In [25]:
```
import seaborn as sns

plt.figure(figsize=(15,7))
sns.scatterplot(x=data_['Murder'], y = data_['Assault'],hue=y_HC_1, s=300, palette=['orange','brown'], legen
```

Out[25]: <AxesSubplot:xlabel='Murder', ylabel='Assault'>



In [26]:
```
#alternate way
plt.scatter(X1[y_HC == 0,0], X1[y_HC == 0,1], s=300, c='red', label = 'Cluster1')
plt.scatter(X1[y_HC == 1,0], X1[y_HC == 1,1], s=300, c='blue', label = 'Cluster2')
#plt.scatter(X1[y_HC == 2,0], X1[y_HC == 2,1], s=300, c='green', label = 'Cluster3')
#plt.scatter(X1[y_HC == 3,0], X1[y_HC == 3,1], s=300, c='orange', label = 'Cluster4')
plt.title("Clusters of States")
plt.legend()
plt.show()
```



DBSCAN

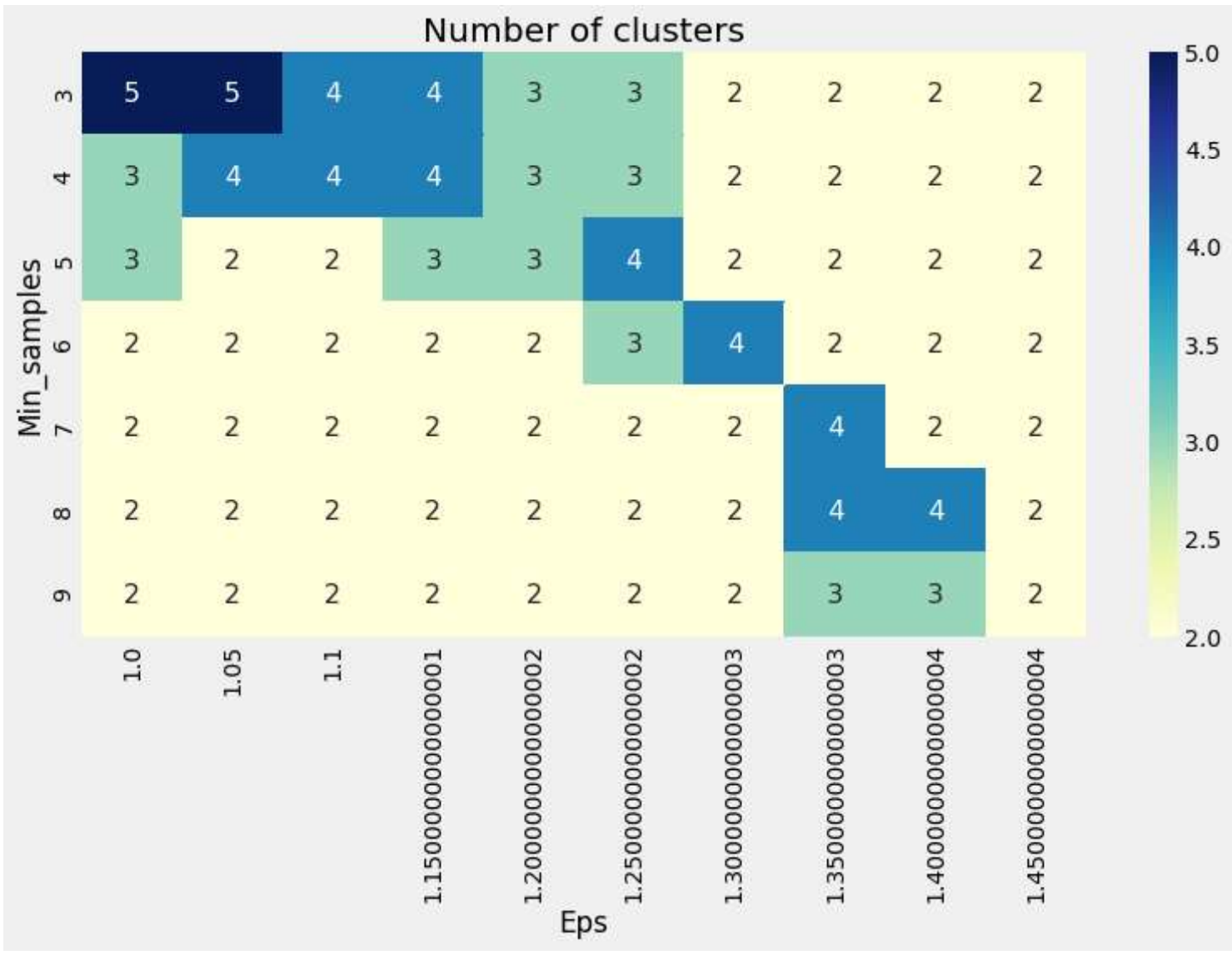
In [27]:
```
from sklearn.cluster import DBSCAN
```

In [28]:
```
from itertools import product
eps_values = np.arange(1,1.5,0.05)
min_samples = np.arange(3,10)
DBSCAN_params = list(product(eps_values, min_samples))
#DBSCAN_params
```

In [29]:
```
no_clusters = []
si_score = []

for i in DBSCAN_params:
    DBS_clustering = DBSCAN(eps=i[0], min_samples=i[1]).fit(X1)
    no_clusters.append(len(np.unique(DBS_clustering.labels_)))
    #print(len(np.unique(DBS_clustering.labels_)))
    if len(np.unique(DBS_clustering.labels_)) >1:
        si_score.append(silhouette_score(X1,DBS_clustering.labels_))
    else:
        si_score.append(0)
```

In [30]:
```
plot_data = pd.DataFrame.from_records(DBSCAN_params, columns =['Eps', 'Min_samples'])
plot_data['No_of_clusters'] = no_clusters

pivot_1 = pd.pivot_table(plot_data, values='No_of_clusters', index='Min_samples', columns='Eps')
```

In [31]:
```
fig, ax = plt.subplots(figsize=(12,6))
sns.heatmap(pivot_1, annot=True,annot_kws={"size": 16}, cmap="YlGnBu", ax=ax)
ax.set_title('Number of clusters')
plt.show()
```



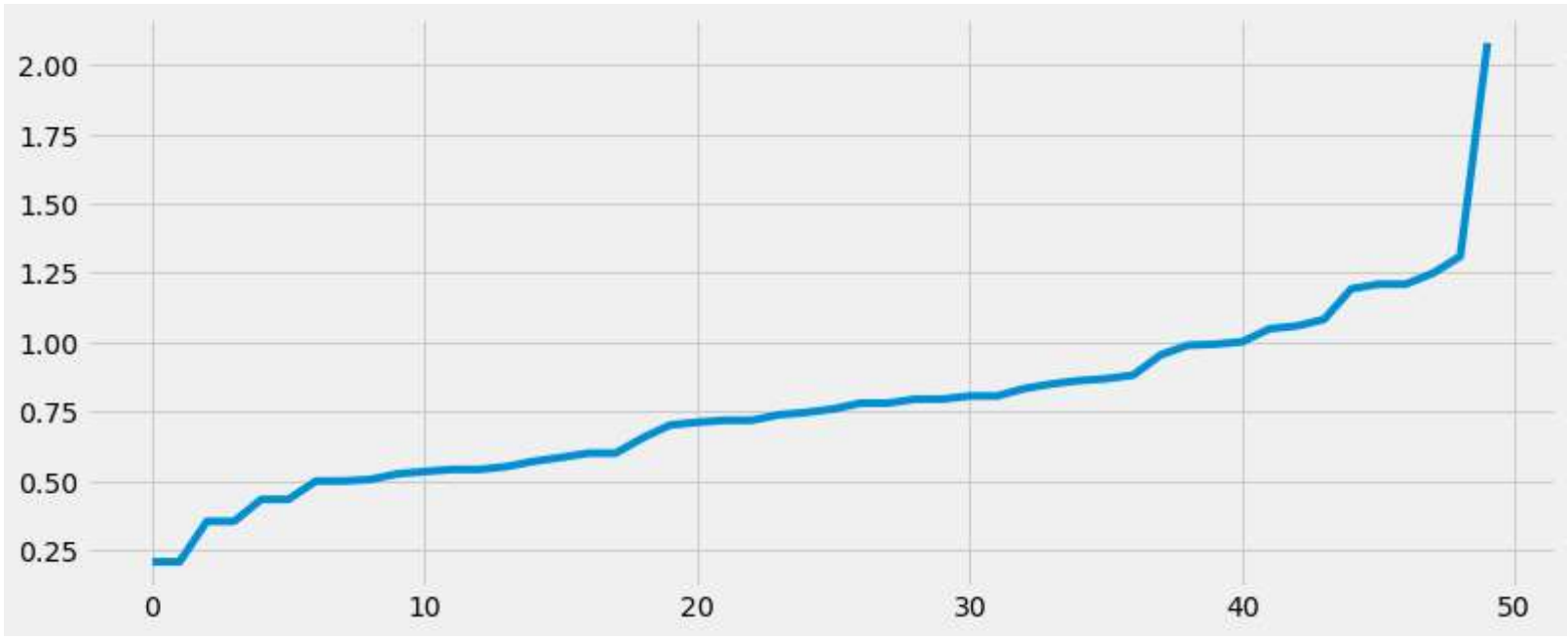
In [32]:
```
tmp = pd.DataFrame.from_records(DBSCAN_params, columns =['Eps', 'Min_samples'])
tmp['Sil_score'] = si_score

pivot_1 = pd.pivot_table(tmp, values='Sil_score', index='Min_samples', columns='Eps')
```

In [33]:
```
fig, ax = plt.subplots(figsize=(15,6))
sns.heatmap(pivot_1, annot=True, annot_kws={"size": 12}, cmap="YlGnBu", ax=ax)
plt.show()
```

Alternate way

In [34]:
```
from sklearn.neighbors import NearestNeighbors
neigh = NearestNeighbors(n_neighbors=2)
nbrs = neigh.fit(X1)
distances, indices = nbrs.kneighbors(X1)
distances = np.sort(distances, axis = 0)
distances = distances[:,1]

fig, ax = plt.subplots(figsize=(12,5))
plt.plot(distances)
#ax.axvline(46, ls="--", c="red")

#ax.axhline(24, ls="--", c="red")
```

Out[34]: [<matplotlib.lines.Line2D at 0x2b7a7ea4370>]



Add comments similar to [https://www.kaggle.com/datark1/customers-clustering-k-means-dbscan-and-ap (https://www.kaggle.com/datark1/customers-clustering-k-means-dbscan-and-ap)](https://www.kaggle.com/datark1/customers-clustering-k-means-dbscan-and-ap)

In [35]:
```
DBS_clustering = DBSCAN(eps=1.25, min_samples=5).fit(X1)

DBSCAN_clustered = data_.copy()
DBSCAN_clustered_ = data_.copy()
DBSCAN_clustered.loc[:,'Cluster2'] = DBS_clustering.labels_
DBSCAN_clustered_.loc[:,'Cluster2'] = DBS_clustering.labels_+1
DBSCAN_clustered
```

Out[35]:

	State	Murder	Assault	UrbanPop	Rape	Cluster	label	HierarchicalCluster	Cluster2
0	Alabama	13.2	236	58	21.2	2	1	2	0
1	Alaska	10.0	263	48	44.5	2	1	2	-1
2	Arizona	8.1	294	80	31.0	2	1	2	1
3	Arkansas	8.8	190	50	19.5	1	0	1	2
4	California	9.0	276	91	40.6	2	1	2	-1
5	Colorado	7.9	204	78	38.7	2	1	2	1
6	Connecticut	3.3	110	77	11.1	1	0	1	2
7	Delaware	5.9	238	72	15.8	1	0	1	2
8	Florida	15.4	335	80	31.9	2	1	2	-1
9	Georgia	17.4	211	60	25.8	2	1	2	0
10	Hawaii	5.3	46	83	20.2	1	0	1	2
11	Idaho	2.6	120	54	14.2	1	0	1	2
12	Illinois	10.4	249	83	24.0	2	1	2	1
13	Indiana	7.2	113	65	21.0	1	0	1	2
14	Iowa	2.2	56	57	11.3	1	0	1	2
15	Kansas	6.0	115	66	18.0	1	0	1	2
16	Kentucky	9.7	109	52	16.3	1	0	1	2
17	Louisiana	15.4	249	66	22.2	2	1	2	0
18	Maine	2.1	83	51	7.8	1	0	1	2
19	Maryland	11.3	300	67	27.8	2	1	2	1
20	Massachusetts	4.4	149	85	16.3	1	0	1	2
21	Michigan	12.1	255	74	35.1	2	1	2	1
22	Minnesota	2.7	72	66	14.9	1	0	1	2
23	Mississippi	16.1	259	44	17.1	2	1	2	-1
24	Missouri	9.0	178	70	28.2	2	1	1	2
25	Montana	6.0	109	53	16.4	1	0	1	2
26	Nebraska	4.3	102	62	16.5	1	0	1	2
27	Nevada	12.2	252	81	46.0	2	1	2	-1
28	New Hampshire	2.1	57	56	9.5	1	0	1	2
29	New Jersey	7.4	159	89	18.8	1	0	1	2
30	New Mexico	11.4	285	70	32.1	2	1	2	1
31	New York	11.1	254	86	26.1	2	1	2	1
32	North Carolina	13.0	337	45	16.1	2	1	2	-1
33	North Dakota	0.8	45	44	7.3	1	0	1	2
34	Ohio	7.3	120	75	21.4	1	0	1	2
35	Oklahoma	6.6	151	68	20.0	1	0	1	2
36	Oregon	4.9	159	67	29.3	1	0	1	2
37	Pennsylvania	6.3	106	72	14.9	1	0	1	2
38	Rhode Island	3.4	174	87	8.3	1	0	1	2
39	South Carolina	14.4	279	48	22.5	2	1	2	0
40	South Dakota	3.8	86	45	12.8	1	0	1	2
41	Tennessee	13.2	188	59	26.9	2	1	2	0
42	Texas	12.7	201	80	25.5	2	1	2	2
43	Utah	3.2	120	80	22.9	1	0	1	2
44	Vermont	2.2	48	32	11.2	1	0	1	2
45	Virginia	8.5	156	63	20.7	1	0	1	2
46	Washington	4.0	145	73	26.2	1	0	1	2
47	West Virginia	5.7	81	39	9.3	1	0	1	2
48	Wisconsin	2.6	53	66	10.8	1	0	1	2
49	Wyoming	6.8	161	60	15.6	1	0	1	2

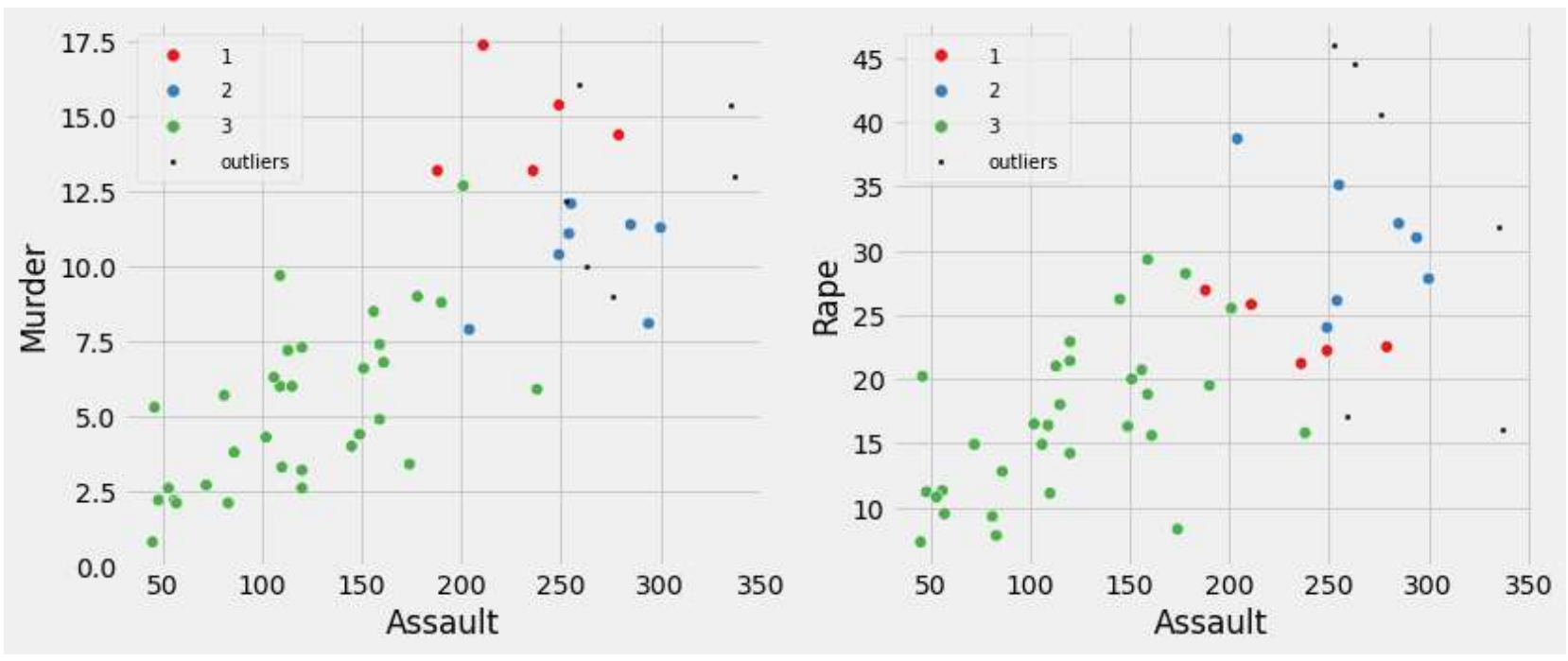
In [36]:
```
# Cluster sizes
DBSCAN_clust_sizes = DBSCAN_clustered_.groupby('Cluster2').size().to_frame()
DBSCAN_clust_sizes.columns = ["DBSCAN_size"]
DBSCAN_clust_sizes
#0th cluster here indicates Outliers
```

Out[36]:

Cluster2	DBSCAN_size
0	6
1	5
2	7
3	32

In [37]:
```
outliers = DBSCAN_clustered[DBSCAN_clustered["Cluster2"] == -1]
```

In [38]:
```
fig2, (axes) = plt.subplots(1,2,figsize=(12,5))


sns.scatterplot('Assault', 'Murder',
                data=DBSCAN_clustered_[DBSCAN_clustered_['Cluster2']!=0],
                hue='Cluster2', ax=axes[0], palette='Set1', legend='full', s=45)

sns.scatterplot('Assault', 'Rape',
                data=DBSCAN_clustered_[DBSCAN_clustered_['Cluster2']!=0],
                hue='Cluster2', palette='Set1', ax=axes[1], legend='full', s=45)

axes[0].scatter(outliers['Assault'], outliers['Murder'], s=5, label='outliers', c="k")
axes[1].scatter(outliers['Assault'], outliers['Rape'], s=5, label='outliers', c="k")
axes[0].legend()
axes[1].legend()

plt.setp(axes[0].get_legend().get_texts(), fontsize='10')
plt.setp(axes[1].get_legend().get_texts(), fontsize='10')

plt.show()
```

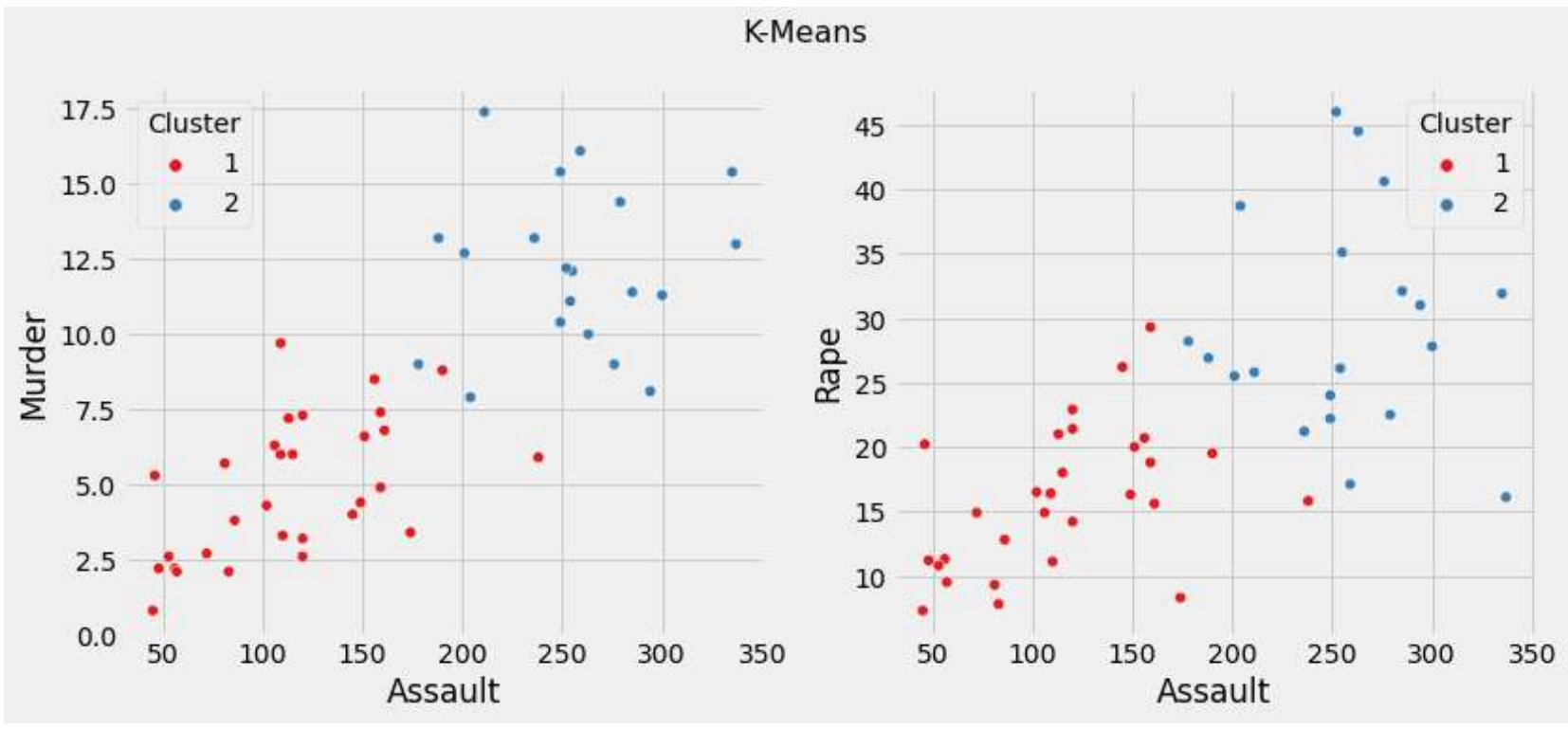


In [39]:
```
#TODO: AFFINITY algorithm
data_
```

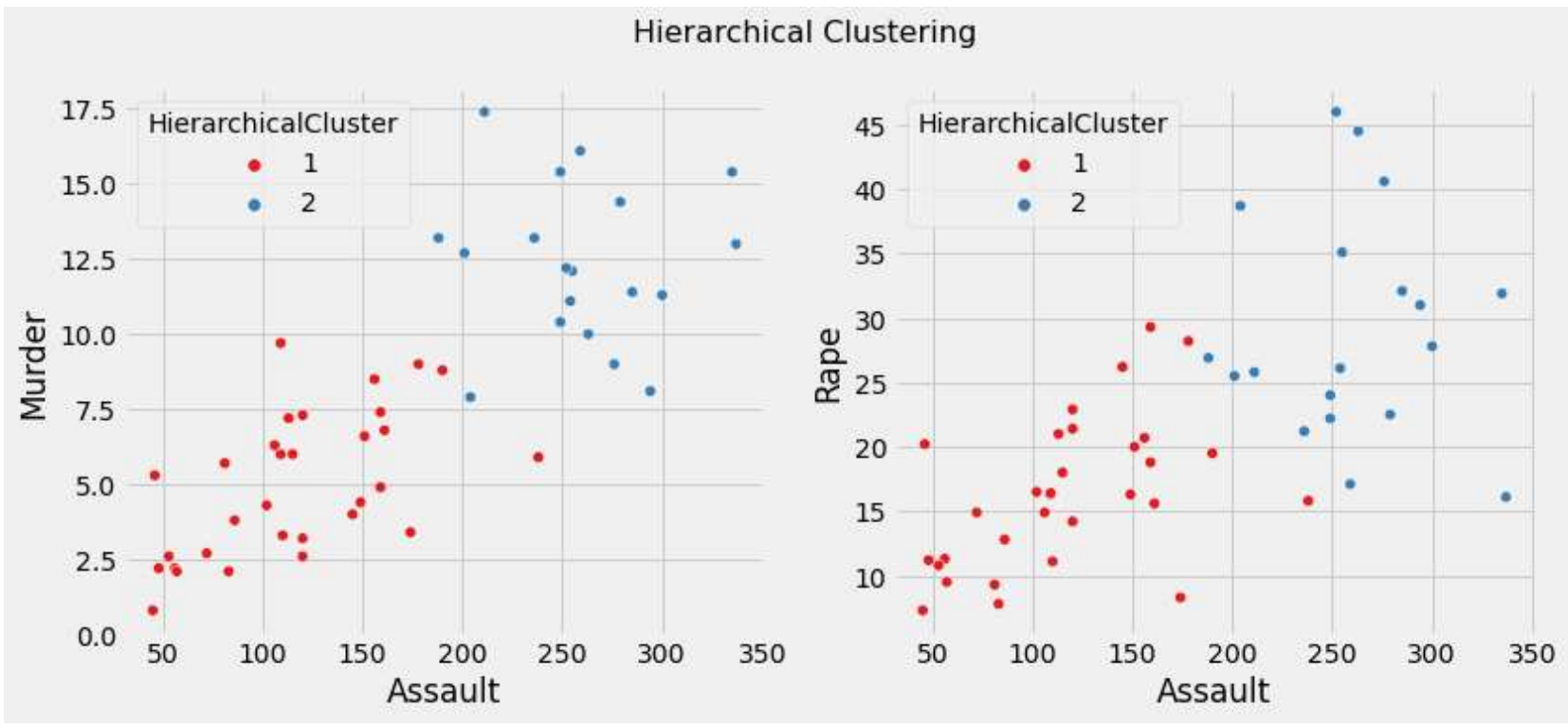
Out[39]:

	State	Murder	Assault	UrbanPop	Rape	Cluster	label	HierarchicalCluster
0	Alabama	13.2	236	58	21.2	2	1	2
1	Alaska	10.0	263	48	44.5	2	1	2
2	Arizona	8.1	294	80	31.0	2	1	2
3	Arkansas	8.8	190	50	19.5	1	0	1
4	California	9.0	276	91	40.6	2	1	2
5	Colorado	7.9	204	78	38.7	2	1	2
6	Connecticut	3.3	110	77	11.1	1	0	1
7	Delaware	5.9	238	72	15.8	1	0	1
8	Florida	15.4	335	80	31.9	2	1	2
9	Georgia	17.4	211	60	25.8	2	1	2
10	Hawaii	5.3	46	83	20.2	1	0	1
11	Idaho	2.6	120	54	14.2	1	0	1
12	Illinois	10.4	249	83	24.0	2	1	2
13	Indiana	7.2	113	65	21.0	1	0	1
14	Iowa	2.2	56	57	11.3	1	0	1
15	Kansas	6.0	115	66	18.0	1	0	1
16	Kentucky	9.7	109	52	16.3	1	0	1
17	Louisiana	15.4	249	66	22.2	2	1	2
18	Maine	2.1	83	51	7.8	1	0	1
19	Maryland	11.3	300	67	27.8	2	1	2
20	Massachusetts	4.4	149	85	16.3	1	0	1
21	Michigan	12.1	255	74	35.1	2	1	2
22	Minnesota	2.7	72	66	14.9	1	0	1
23	Mississippi	16.1	259	44	17.1	2	1	2
24	Missouri	9.0	178	70	28.2	2	1	1
25	Montana	6.0	109	53	16.4	1	0	1
26	Nebraska	4.3	102	62	16.5	1	0	1
27	Nevada	12.2	252	81	46.0	2	1	2
28	New Hampshire	2.1	57	56	9.5	1	0	1
29	New Jersey	7.4	159	89	18.8	1	0	1
30	New Mexico	11.4	285	70	32.1	2	1	2
31	New York	11.1	254	86	26.1	2	1	2
32	North Carolina	13.0	337	45	16.1	2	1	2
33	North Dakota	0.8	45	44	7.3	1	0	1
34	Ohio	7.3	120	75	21.4	1	0	1
35	Oklahoma	6.6	151	68	20.0	1	0	1
36	Oregon	4.9	159	67	29.3	1	0	1
37	Pennsylvania	6.3	106	72	14.9	1	0	1
38	Rhode Island	3.4	174	87	8.3	1	0	1
39	South Carolina	14.4	279	48	22.5	2	1	2
40	South Dakota	3.8	86	45	12.8	1	0	1
41	Tennessee	13.2	188	59	26.9	2	1	2
42	Texas	12.7	201	80	25.5	2	1	2
43	Utah	3.2	120	80	22.9	1	0	1
44	Vermont	2.2	48	32	11.2	1	0	1
45	Virginia	8.5	156	63	20.7	1	0	1
46	Washington	4.0	145	73	26.2	1	0	1
47	West Virginia	5.7	81	39	9.3	1	0	1
48	Wisconsin	2.6	53	66	10.8	1	0	1
49	Wyoming	6.8	161	60	15.6	1	0	1

In [40]:
```
fig11, (axes) = plt.subplots(1,2,figsize=(12,5))

sns.scatterplot('Assault', 'Murder', data=data_,
                hue='Cluster', ax=axes[0], palette='Set1', legend='full')

sns.scatterplot('Assault', 'Rape', data=data_,
                hue='Cluster', palette='Set1', ax=axes[1], legend='full')

# plotting centroids
fig11.suptitle('K-Means', fontsize=16)
plt.show()
```

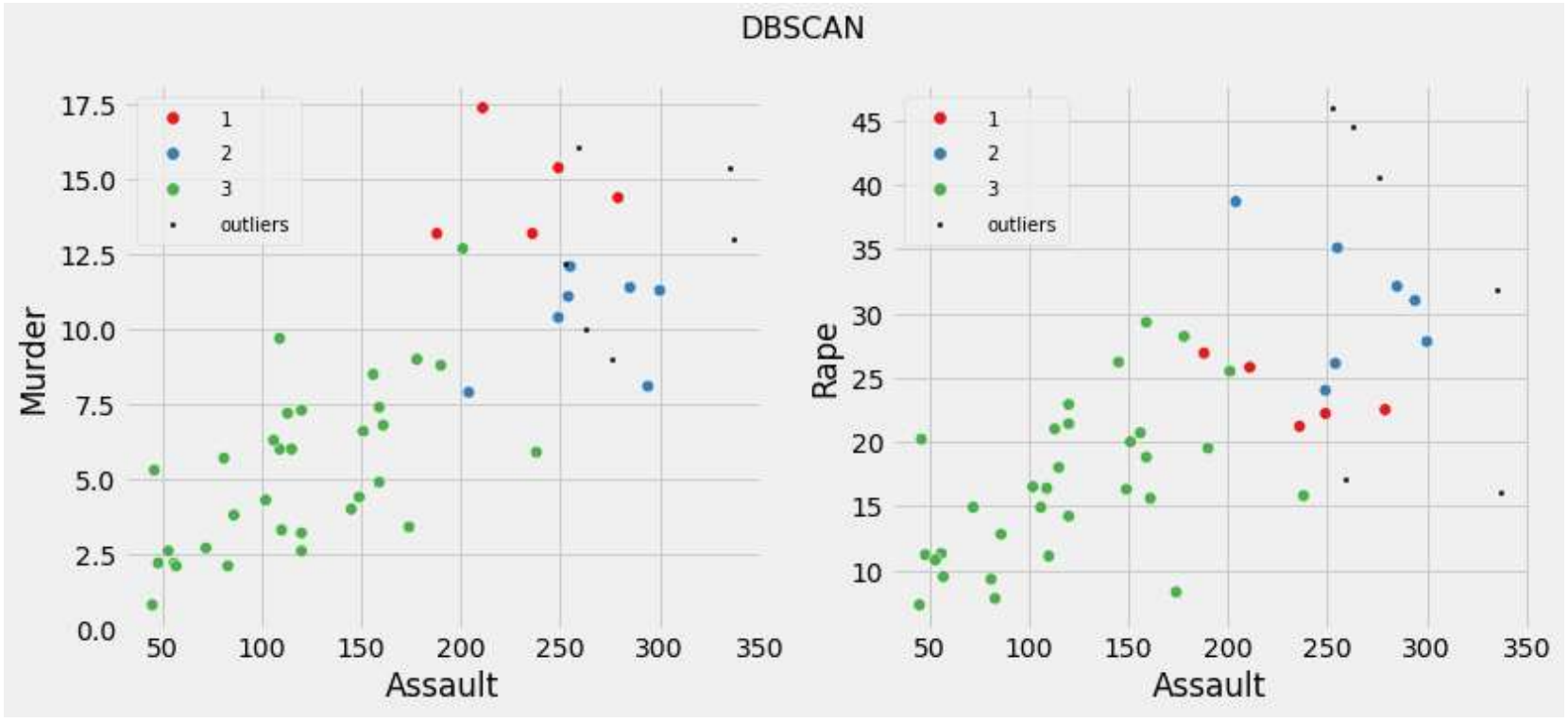


In [41]:
```
#HierarchicalCluster
fig22, (axes) = plt.subplots(1,2,figsize=(12,5))

sns.scatterplot('Assault', 'Murder', data=data_,
                hue='HierarchicalCluster', ax=axes[0], palette='Set1', legend='full')

sns.scatterplot('Assault', 'Rape', data=data_,
                hue='HierarchicalCluster', palette='Set1', ax=axes[1], legend='full')

# plotting centroids
#axes[0].scatter(kmeans.cluster_centers_[:,1], kmeans.cluster_centers_[:,0], marker='s', s=40, c="blue")
#axes[1].scatter(kmeans.cluster_centers_[:,1], kmeans.cluster_centers_[:,3], marker='s', s=40, c="blue")
fig22.suptitle('Hierarchical Clustering', fontsize=16)
plt.show()
```



In [42]:
```
fig2.suptitle('DBSCAN', fontsize=16)
fig2
```

Out[42]:



In [43]:
```
kmeans_clust_sizes = data_.groupby('Cluster').size().to_frame()
kmeans_clust_sizes.columns = ["kmeans_size"]
kmeans_clust_sizes
```

Out[43]:

Cluster	kmeans_size
1	30
2	20

In [44]:
```
HC_clust_sizes = data_.groupby('HierarchicalCluster').size().to_frame()
HC_clust_sizes.columns = ["HC_size"]
HC_clust_sizes
```

Out[44]:

HierarchicalCluster	HC_size
1	31
2	19

In [45]:
```
clusters = pd.concat([kmeans_clust_sizes, HC_clust_sizes, DBSCAN_clust_sizes],axis=1, sort=False)
clusters
```

Out[45]:

	kmeans_size	HC_size	DBSCAN_size
0	NaN	NaN	6
1	30.0	31.0	5
2	20.0	19.0	7
3	NaN	NaN	32