Import neccessery libraries

In [52]:
```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn import preprocessing
import seaborn as sns

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

Problem

Perform Clustering for the crime data and identify the number of clusters formed and draw inferences

Import data

In [2]:
```
crime_data = pd.read_csv('crime_data.csv')
crime_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

	Unnamed: 0	Murder	Assault	UrbanPop	Rape
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

Data understanding

In [3]:

```
crime_data.shape
```

Out[3]:

```
(50, 5)
```

In [4]:
```
crime_data.isna().sum()
```
Out[4]:
```
Unnamed: 0     0
Murder         0
Assault        0
UrbanPop       0
Rape           0
dtype: int64
```
In [5]:
```
crime_data.dtypes
```
Out[5]:
```
Unnamed: 0     object
Murder        float64
Assault         int64
UrbanPop        int64
Rape          float64
dtype: object
```
In [6]:
```
crime_data_1 = crime_data.copy()
```
In [7]:
```
crime_data_1.columns=['City','Murder' , 'Assault', 'Urbanpop','Rape']
```
In [10]:
```
crime_data_1.loc[:,'Total'] = crime_data_1.sum(numeric_only=True, axis=1)
```
In [12]:
```
crime_data_1.head()
```
Out[12]:

	City	Murder	Assault	Urbanpop	Rape	Total
0	Alabama	13.2	236	58	21.2	328.4
1	Alaska	10.0	263	48	44.5	365.5
2	Arizona	8.1	294	80	31.0	413.1
3	Arkansas	8.8	190	50	19.5	268.3
4	California	9.0	276	91	40.6	416.6

In [14]:
```
crime_data_1.describe()
```
Out[14]:

	Murder	Assault	Urbanpop	Rape	Total
count	50.00000	50.000000	50.000000	50.000000	50.000000
mean	7.78800	170.760000	65.540000	21.232000	265.320000
std	4.35551	83.337661	14.474763	9.366385	98.350844
min	0.80000	45.000000	32.000000	7.300000	93.400000
25%	4.07500	109.000000	54.500000	15.075000	187.950000
50%	7.25000	159.000000	66.000000	20.100000	257.450000
75%	11.25000	249.000000	77.750000	26.175000	348.500000
max	17.40000	337.000000	91.000000	46.000000	462.300000

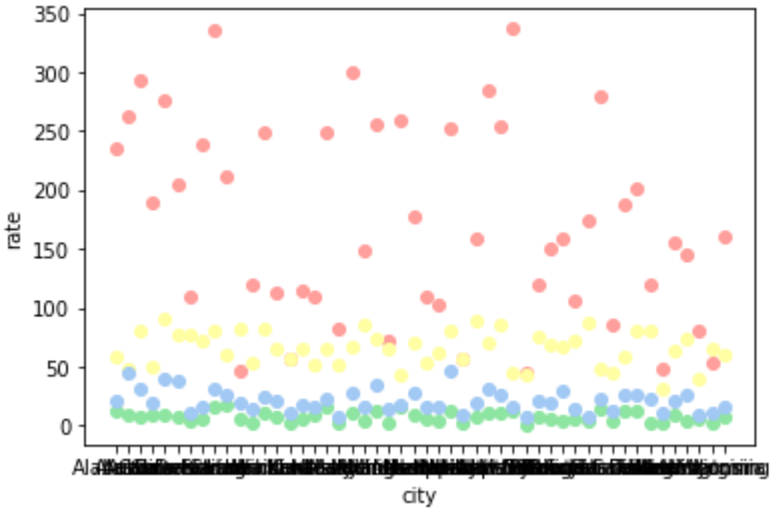
In [16]:

```
crime_data_1.shape
```

Out[16]: (50, 6)

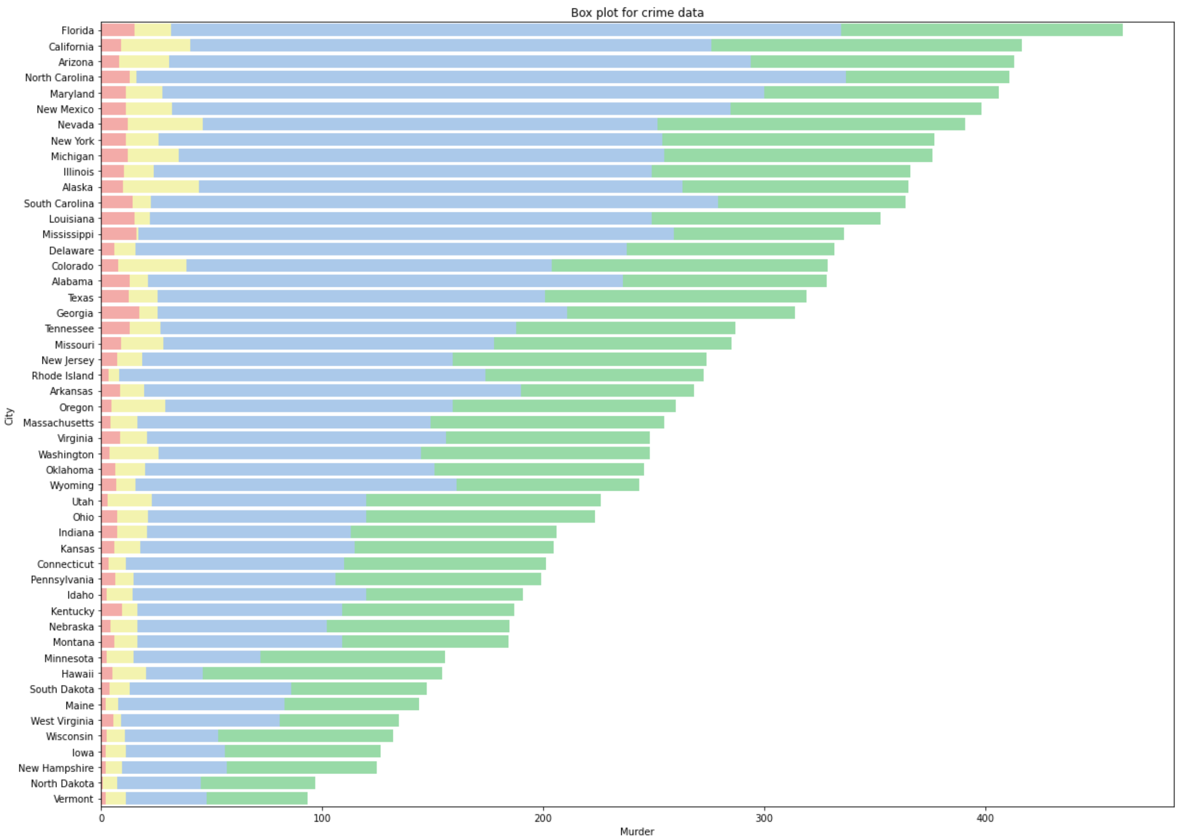
In [87]:

```
plt.scatter(crime_data_1.City, crime_data_1.Murder, color='g')
plt.scatter(crime_data_1.City, crime_data_1.Assault, color='r')
plt.scatter(crime_data_1.City, crime_data_1.Urbanpop, color='y')
plt.scatter(crime_data_1.City, crime_data_1.Rape, color='b')
plt.xlabel('city')
plt.ylabel('rate')
plt.show()
```



In [35]:

```
f, ax = plt.subplots(figsize=(20, 15))
plt.title('Box plot for crime data')


stats = crime_data_1.sort_values("Total", ascending=False)

sns.set_color_codes("pastel")
sns.barplot(x="Total", y="City", data=stats,
            label="Total", color="g")

sns.barplot(x="Assault", y="City", data=stats,
            label="Assault", color="b")

sns.barplot(x="Rape", y="City", data=stats,
            label="Rape", color="y")

sns.barplot(x="Murder", y="City", data=stats,
            label="Murder", color="r")
plt.show()
```



Data preprocessing

In [44]:
```
X = crime_data_1[['Murder', 'Assault', 'Rape', 'Urbanpop']]
```

In [45]:
```
#standardize the data to normal distribution
crime_norm = preprocessing.scale(X)
```

In [46]:
```
crime_norm = pd.DataFrame(crime_norm)
```

In [47]:
```
crime_norm.head()
```

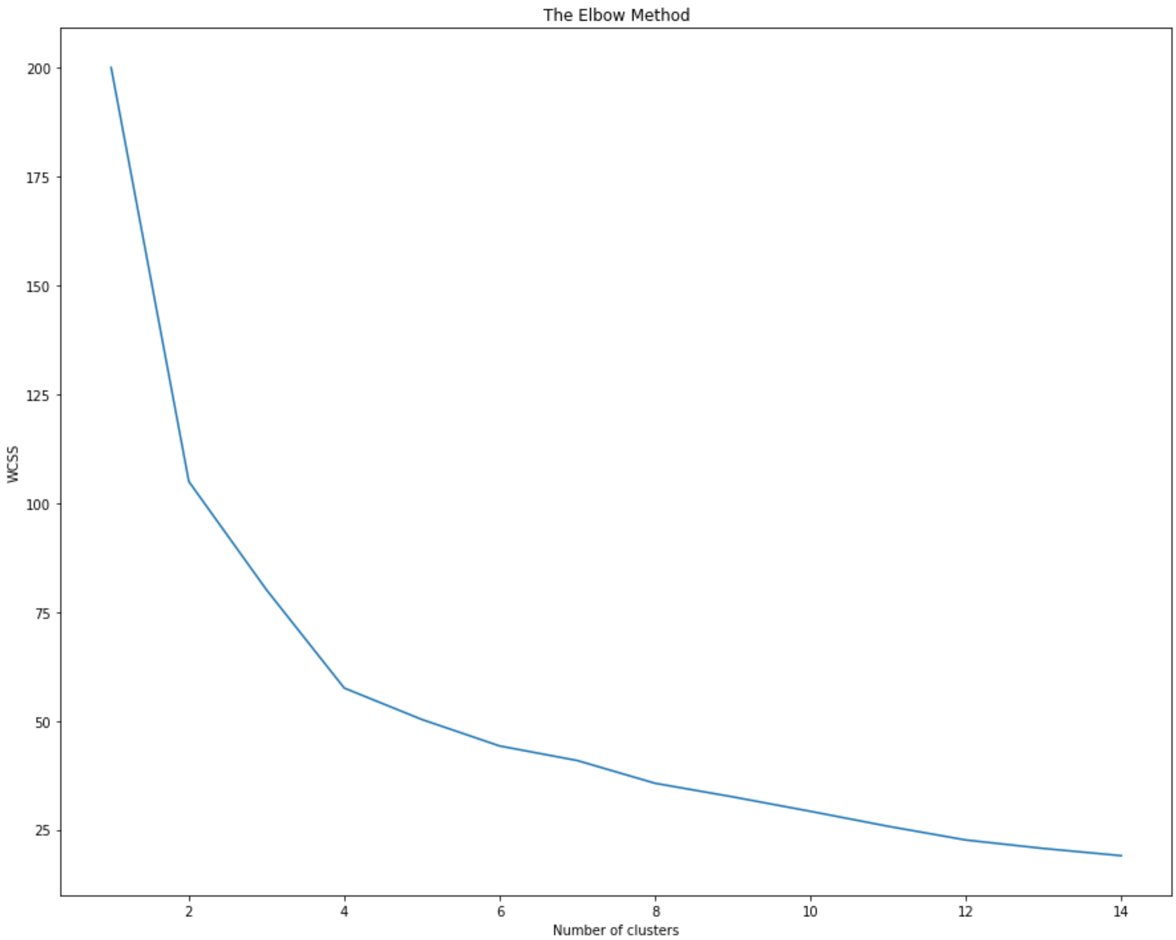
Out[47]:

	0	1	2	3
0	1.255179	0.790787	-0.003451	-0.526195
1	0.513019	1.118060	2.509424	-1.224067
2	0.072361	1.493817	1.053466	1.009122
3	0.234708	0.233212	-0.186794	-1.084492
4	0.281093	1.275635	2.088814	1.776781

Finding out the optimal number of clusters

In [56]:

```
plt.figure(figsize=(15, 12))
wcss = []
for i in range(1, 15):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(crime_norm)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 15), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



plot levels off at k=4 and let's use it to determine the clusters

Analysing the data using kmean

In [57]:

```
k_mean = KMeans(n_clusters = 4, init = 'k-means++', random_state = 42)
y_kmean = kmeans.fit_predict(crime_norm)
```

In [58]:

```
y_kmean
```

Out[58]:

```
array([ 0, 10,  4, 12,  2,  2,  1,  3,  4,  0,  9,  5,  8,  3,  5,  3, 12,
        0,  5,  4, 13,  4,  5,  6, 11, 12,  5,  2,  5, 13,  4,  8,  6,  7,
        3,  3, 11,  3,  1,  6,  7,  0,  8,  9,  7,  3, 11,  7,  5,  3])
```

In [64]:

```
y_kmean_1=y_kmean+1
cluster = list(y_kmean_1)
```

In [65]:

```
crime_data_1['cluster'] = cluster
```

In [67]:

```
kmean_cluster = pd.DataFrame(round(crime_data_1.groupby('cluster').mean(),1
kmean_cluster
```

Out[67]:

cluster	Murder	Assault	Urbanpop	Rape	Total
1	14.8	221.0	60.8	24.0	320.6
2	3.4	142.0	82.0	9.7	237.0
3	9.7	244.0	83.3	41.8	378.8
4	6.8	145.0	67.6	18.4	237.9
5	11.7	293.8	74.2	31.6	411.2
6	2.7	77.6	58.9	12.1	151.2
7	14.5	291.7	45.7	18.6	370.4
8	3.1	65.0	40.0	10.2	118.3
9	11.4	234.7	83.0	25.2	354.3
10	4.2	83.0	81.5	21.5	190.3
11	10.0	263.0	48.0	44.5	365.5
12	6.0	160.7	70.0	27.9	264.5
13	8.2	136.0	51.7	17.4	213.2
14	5.9	154.0	87.0	17.6	264.4

In [68]:

```
kmean_cluster = pd.DataFrame(round(crime_data_1.groupby('cluster').count(),
kmean_cluster
```

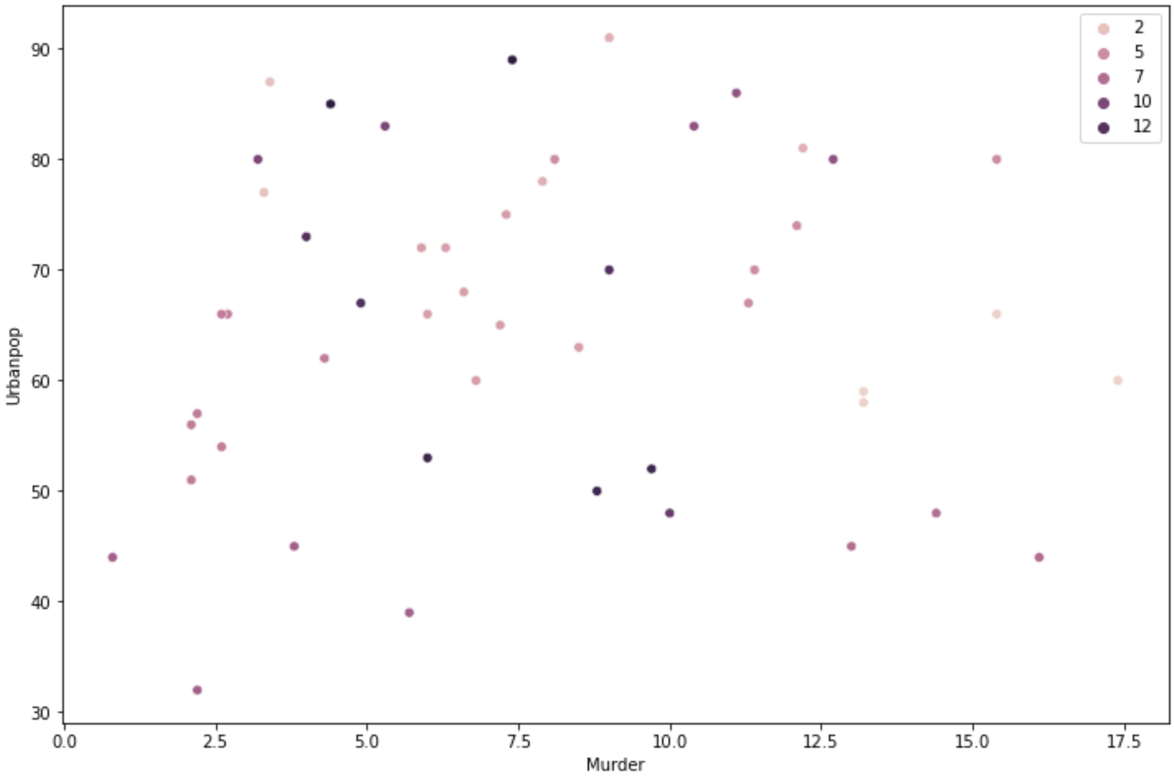
Out[68]:

cluster	City	Murder	Assault	Urbanpop	Rape	Total
1	4	4	4	4	4	4
2	2	2	2	2	2	2
3	3	3	3	3	3	3
4	8	8	8	8	8	8
5	5	5	5	5	5	5
6	7	7	7	7	7	7
7	3	3	3	3	3	3
8	4	4	4	4	4	4

	City	Murder	Assault	Urbanpop	Rape	Total
cluster						
9	3	3	3	3	3	3
10	2	2	2	2	2	2
11	1	1	1	1	1	1
12	3	3	3	3	3	3

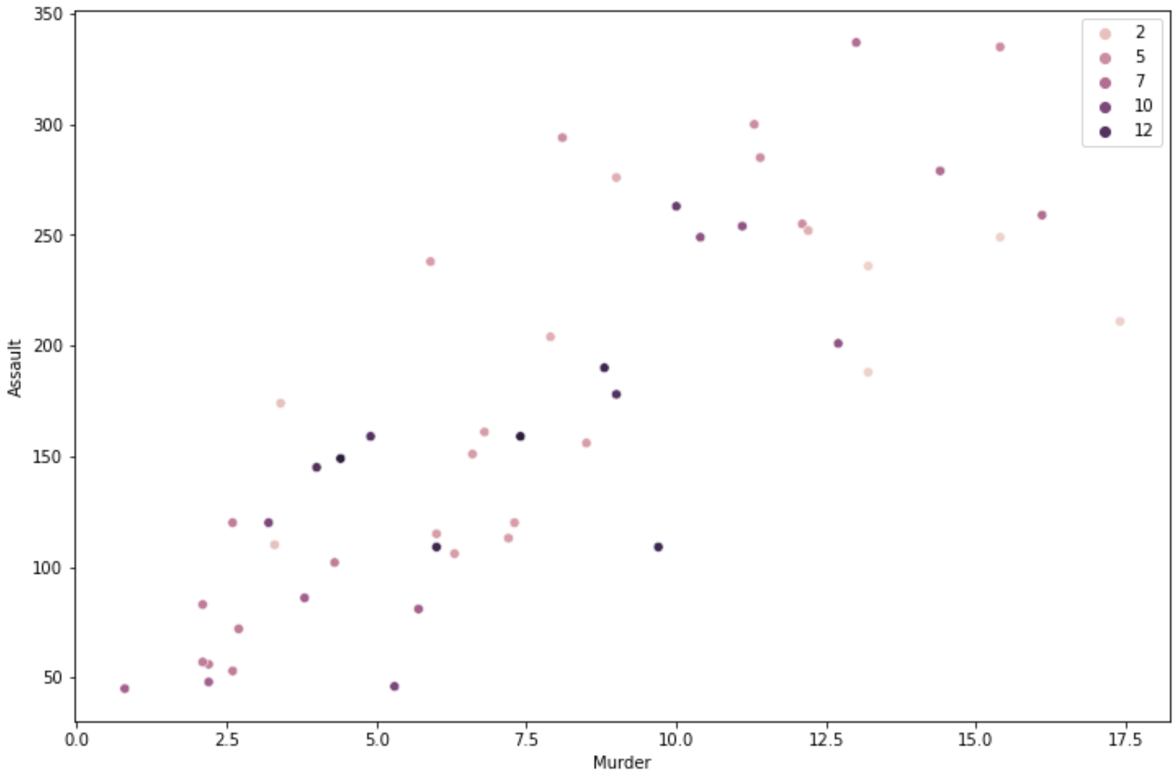
In [77]:

```
plt.figure(figsize=(12,8))
sns.scatterplot(x=crime_data_1['Murder'], y = crime_data_1['Urbanpop'],hue=
plt.show()
```



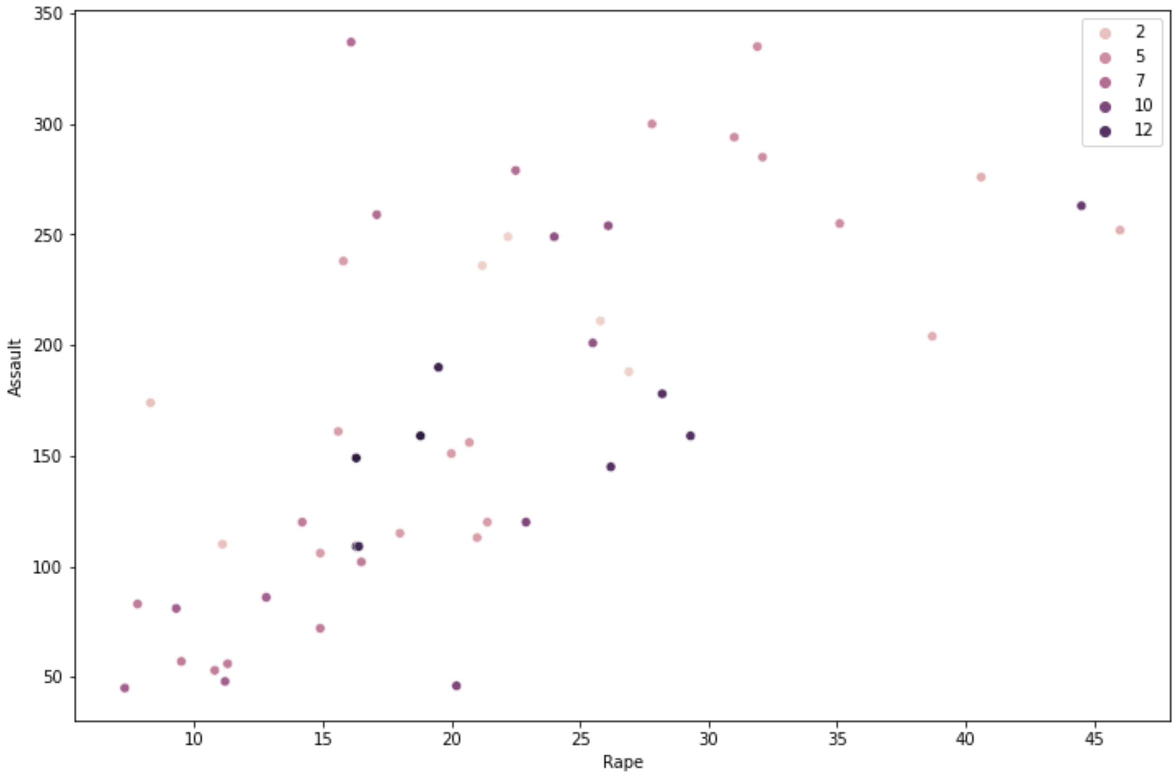
In [78]:

```
plt.figure(figsize=(12,8))
sns.scatterplot(x=crime_data_1['Murder'], y = crime_data_1['Assault'],hue=y
plt.show()
```



In [80]:

```
plt.figure(figsize=(12,8))
sns.scatterplot(x=crime_data_1['Rape'], y = crime_data_1['Assault'],hue=y_l
plt.show()
```



In [83]:

```
stats = crime_data_1.sort_values("Total", ascending=True)
crime_total= pd.DataFrame(stats)
```

In [84]:

```
crime_total.head()
```

Out[84]:

	City	Murder	Assault	Urbanpop	Rape	Total	cluster
44	Vermont	2.2	48	32	11.2	93.4	8
33	North Dakota	0.8	45	44	7.3	97.1	8
28	New Hampshire	2.1	57	56	9.5	124.6	6
14	Iowa	2.2	56	57	11.3	126.5	6
48	Wisconsin	2.6	53	66	10.8	132.4	6

1 - Analysing Murder and Assault variables shows a clearer connection between them. Higher the murder rates in a city higer the assaults and vice versa

2 = Contrary to murders and assaults, there is much more spread among the clusters when comparing murders and rapes. Some correlation is visible, but low murder rates in a city seem to indicate lower number of rapes and vice versa

3 - As with murder and assault, also rates of rape and assault show clearer correlations

In [ ]: