In [2]:
```python
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
```

In [3]:
```python
df = pd.read_csv('crime_data.csv')
```

In [4]:
```python
df1 = df.copy()
```

In [5]:
```python
df1.columns = ['City','Murder' , 'Assault', 'Urbanpop','Rape']
```

In [6]:
```python
df1.loc[:,'Total'] = df1.sum(numeric_only=True, axis=1)
```

In [7]:
```python
df1.head()
```

Out[7]:

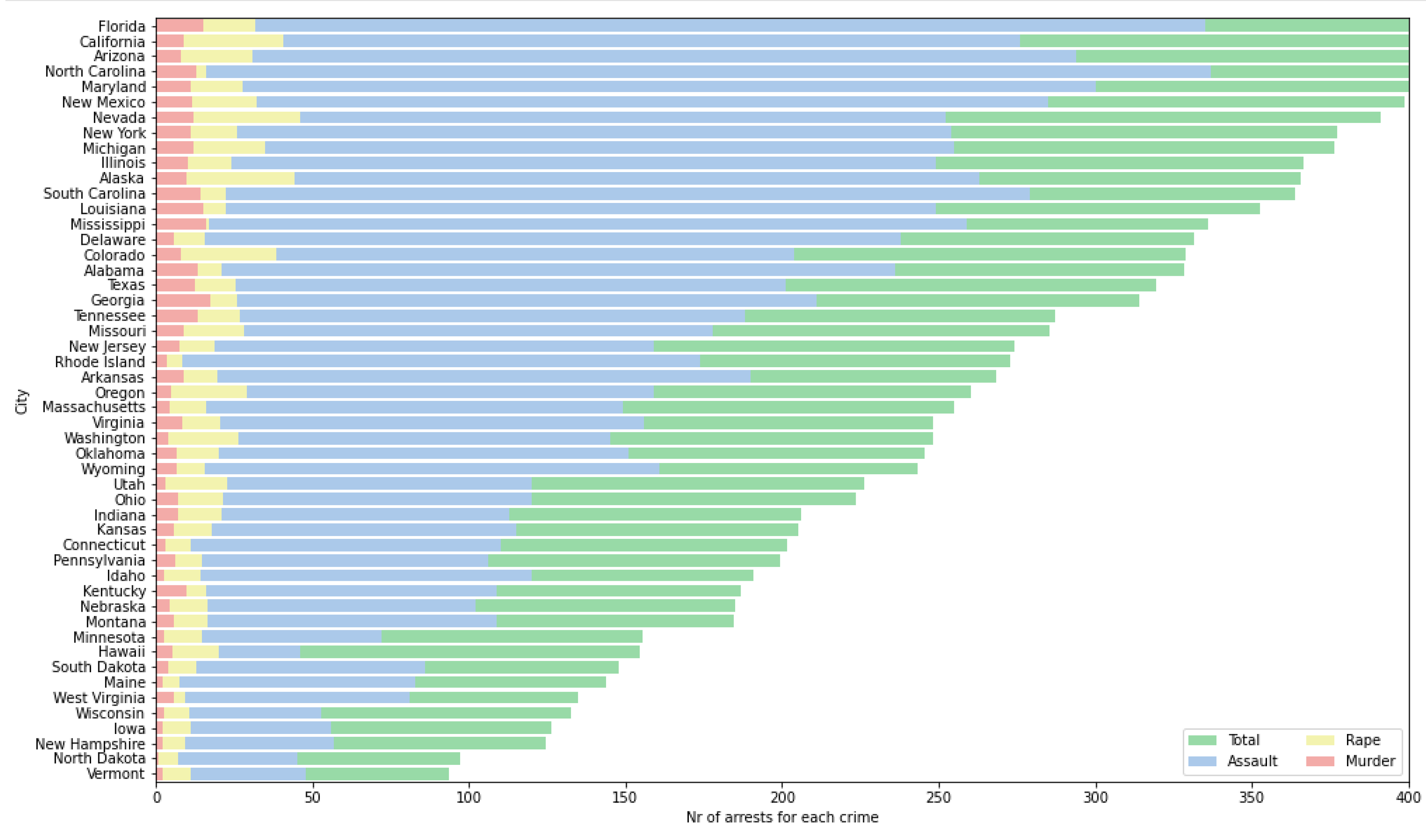
	City	Murder	Assault	Urbanpop	Rape	Total
0	Alabama	13.2	236	58	21.2	328.4
1	Alaska	10.0	263	48	44.5	365.5
2	Arizona	8.1	294	80	31.0	413.1
3	Arkansas	8.8	190	50	19.5	268.3
4	California	9.0	276	91	40.6	416.6

In [8]:
```python
df1.describe()
```

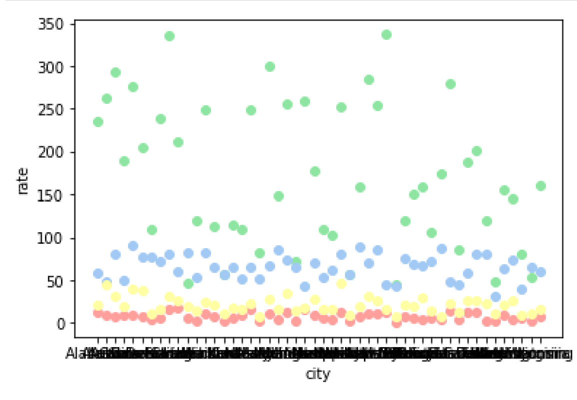
Out[8]:

	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 [9]:
```python
f, ax = plt.subplots(figsize=(16, 10))

stats = df1.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")

ax.legend(ncol=2, loc="lower right", frameon=True)
ax.set(xlim=(0, 400), ylabel="City",
       xlabel="Nr of arrests for each crime");
```



In [10]:
```python
plt.scatter(df1.City, df1.Murder, color='r')
plt.scatter(df1.City, df1.Assault, color='g')
plt.scatter(df1.City, df1.Urbanpop, color='b')
plt.scatter(df1.City, df1.Rape, color='y')
plt.xlabel('city')
plt.ylabel('rate')
plt.show()
```



Finding out the optimal numbers of clusters

In [11]:
```python
X = df1[['Murder', 'Assault', 'Rape', 'Urbanpop']]
```

In [12]:
```python
df1_norm = preprocessing.scale(X)
```

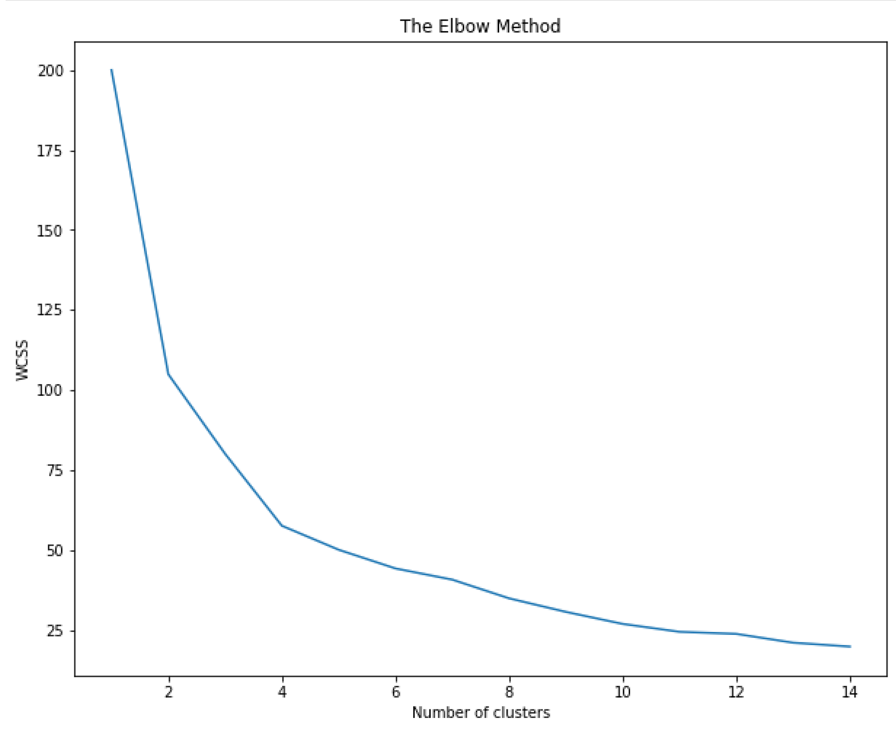
In [15]:
```python
# Standardize the data to normal distribution
df1_norm = pd.DataFrame(df1_norm)
```

In [16]:
```python
df1_norm.head()
```

Out[16]:

	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

In [17]:
```python
plt.figure(figsize=(10, 8))
wcss = []
for i in range(1, 15):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(df1_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()
```



Analysing the data

In [18]:
```python
kmeans = KMeans(n_clusters = 4, init = 'k-means++', random_state = 42)
y_kmeans = kmeans.fit_predict(df1_norm)
```

In [19]:
```python
y_kmeans
```

Out[19]:
```
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 [20]:
```python
y_kmeans1=y_kmeans+1
cluster = list(y_kmeans1)
```

In [21]:
```python
df1['cluster'] = cluster
```

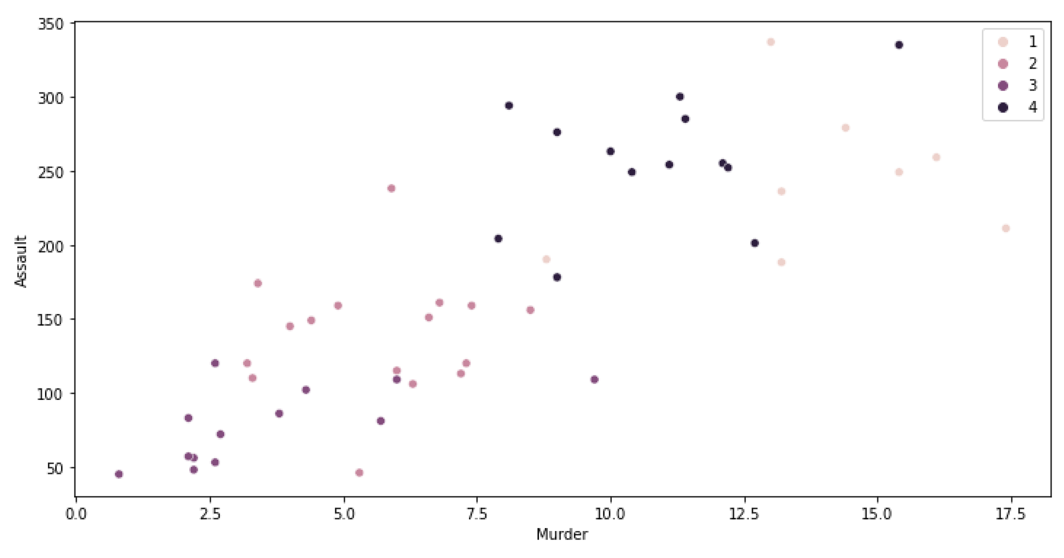
In [22]:
```python
kmeans_mean_cluster = pd.DataFrame(round(df1.groupby('cluster').mean(),1))
kmeans_mean_cluster
```

Out[22]:

	Murder	Assault	Urbanpop	Rape	Total
cluster					
1	13.9	243.6	53.8	21.4	332.7
2	5.7	138.9	73.9	18.8	237.2
3	3.6	78.5	52.1	12.2	146.4
4	10.8	257.4	76.0	33.2	377.4

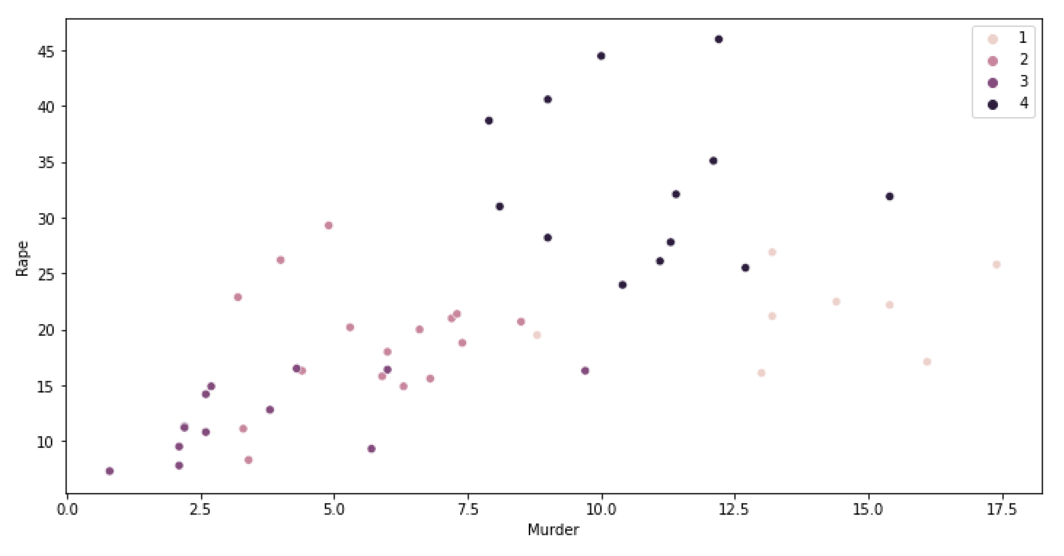
In [23]:
```python
plt.figure(figsize=(12,6))
sns.scatterplot(x=df1['Murder'], y = df1['Assault'],hue=y_kmeans1)
```

Out[23]:
```
<AxesSubplot:xlabel='Murder', ylabel='Assault'>
```



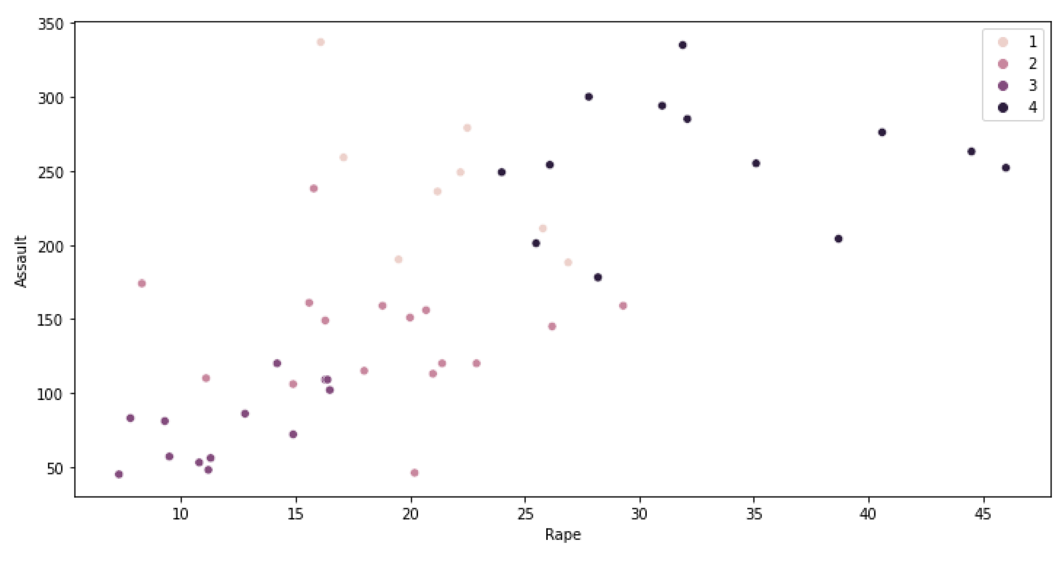
In [24]:
```python
plt.figure(figsize=(12,6))
sns.scatterplot(x=df1['Murder'], y = df1['Rape'],hue=y_kmeans1)
```

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



In [25]:
```python
plt.figure(figsize=(12,6))
sns.scatterplot(x=df1['Rape'], y = df1['Assault'],hue=y_kmeans1)
```

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



In [26]:
```python
stats = df1.sort_values("Total", ascending=True)
df1_total= pd.DataFrame(stats)
```

In [27]:
```python
df1_total.head()
```

Out[27]:

	City	Murder	Assault	Urbanpop	Rape	Total	cluster
44	Vermont	2.2	48	32	11.2	93.4	3
33	North Dakota	0.8	45	44	7.3	97.1	3
28	New Hampshire	2.1	57	56	9.5	124.6	3
14	Iowa	2.2	56	57	11.3	126.5	3
48	Wisconsin	2.6	53	66	10.8	132.4	3

Conclusion

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

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

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