In [53]:

```python
from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
from matplotlib import pyplot as plt
%matplotlib inline
```

In [80]:

```python
df=pd.read_excel(r'DataFinal17-20.xlsx',sheet_name='2018')
df1=df.drop(['Total'], axis=1)
df1=df1.iloc[:,2:15].values
#scaler = MinMaxScaler()
scaler= StandardScaler()
# transform data
df1 = scaler.fit_transform(df1)
```

In [81]:

```python
from sklearn.cluster import KMeans

#create a list for the wcss parameter
wcss = []
#test with 14 clusters
for i in range(1, 15):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state =0)
    kmeans.fit(df1)
    wcss.append(kmeans.inertia_)
```

In [82]:

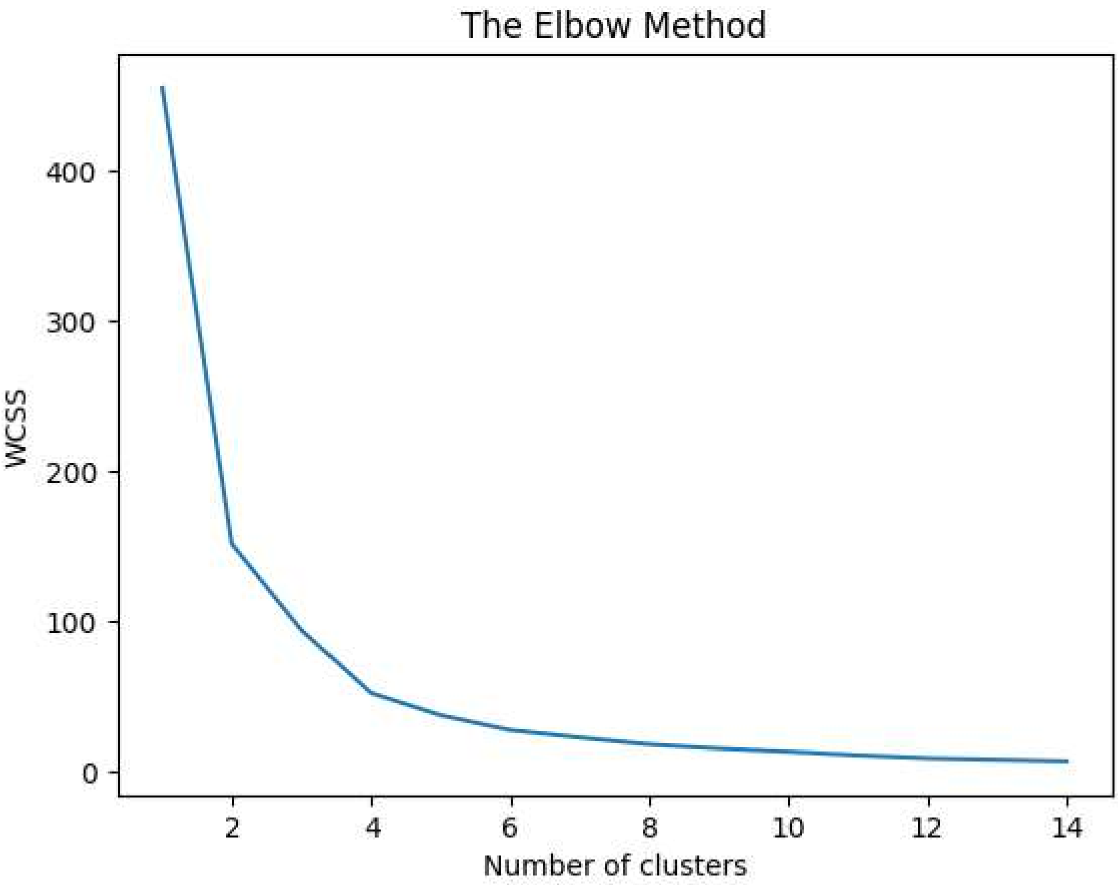
```python
wcss
```

Out[82]:

```
[455.00000000000006,
 151.88198373485318,
 94.4962974367791,
 52.36778125280657,
 37.66311578686052,
 27.85608724129876,
 22.956380978779134,
 18.411323303516994,
 15.654980301497902,
 13.505706826278605,
 11.131631283310846,
 9.189929945241067,
 8.259155435971405,
 7.249011347468659]
```

In [83]:

```python
plt.plot(range(1, 15), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



```
In [84]: km=KMeans(n_clusters = 4, init = 'k-means++', random_state = 0)
         y_kmeans=km.fit_predict(df1)
```

```
In [85]: y_kmeans
```

```
Out[85]: array([0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 0, 1, 1, 0, 1,
                1, 1, 2, 1, 1, 1, 1, 1, 0, 2, 1, 1, 1])
```

```
In [86]: df['cluster']=y_kmeans
         df.head()
```

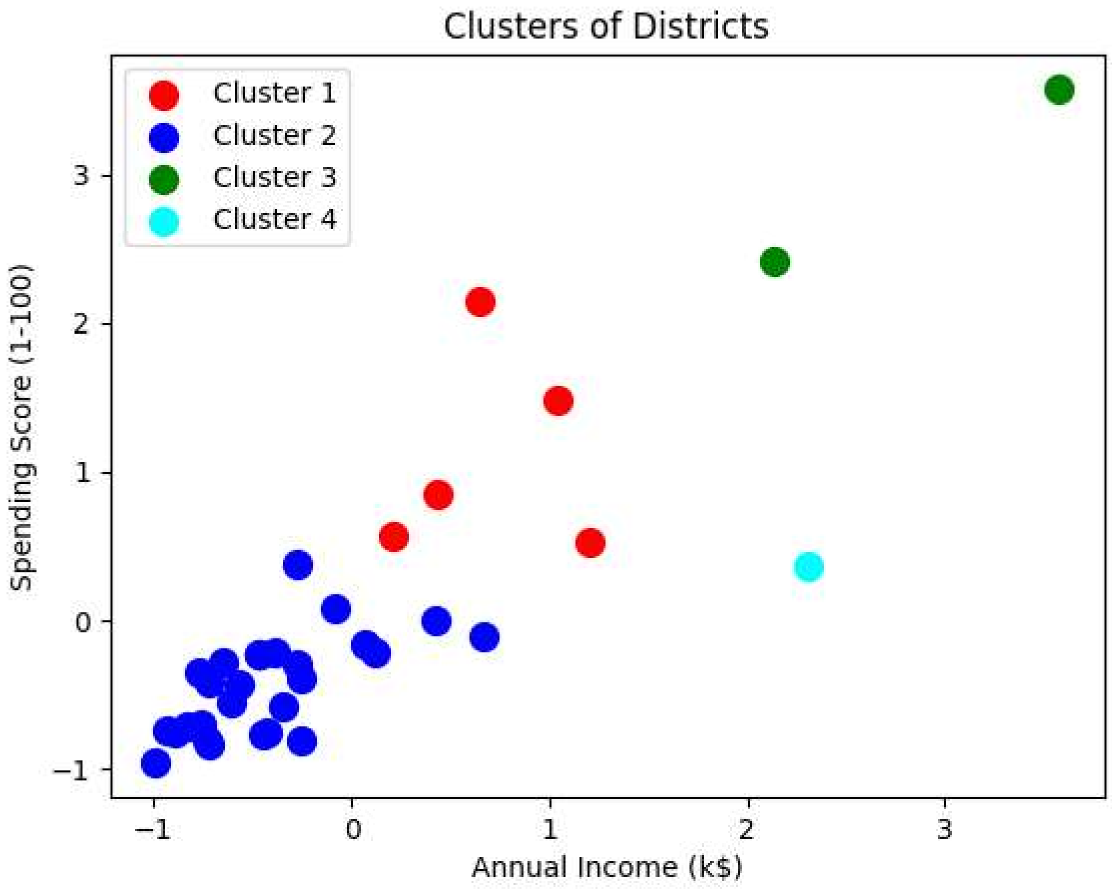
Out[86]:

	SrNo	State/UT/District	Homicide/Murder	Causing death by negligence	Hurt	Assault on woman	Kidnapping and abduction	Human trafficking	Ra
0	1	Ahmednagar	308	799	1458	476	431	0	
1	2	Akola	104	146	1203	202	133	0	
2	3	Amravati	181	302	1325	419	236	1	
3	4	Aurangbad	193	532	1852	419	281	0	
4	5	Beed	173	318	944	249	145	7	

5 rows × 23 columns

```
In [87]: #plt.scatter(df['SrNo'],df['cluster'])
```

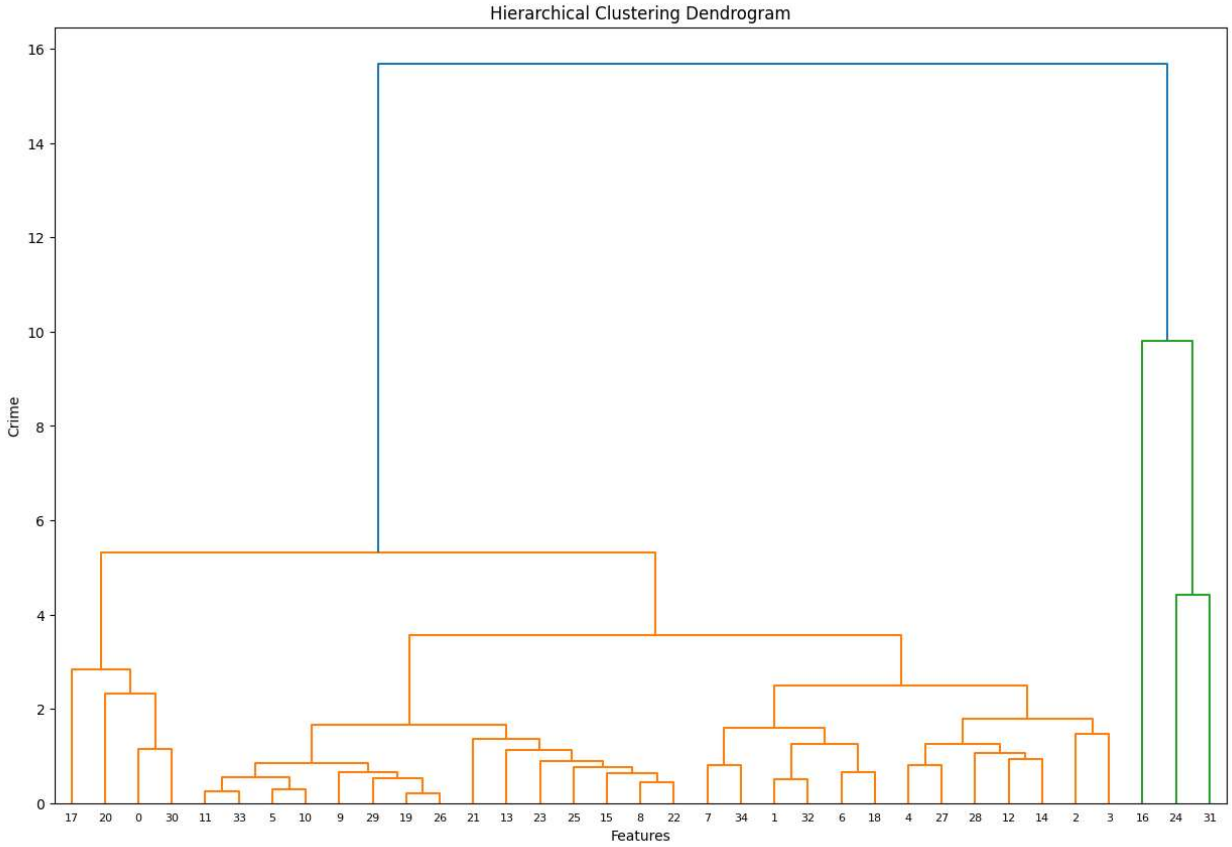
```
#for col in df.columns:
 #   print(col)
plt.scatter(df1[y_kmeans == 0, 0], df1[y_kmeans == 0, 1], s = 100, c = 'red', label =
plt.scatter(df1[y_kmeans == 1, 0], df1[y_kmeans == 1, 1], s = 100, c = 'blue', label =
plt.scatter(df1[y_kmeans == 2, 0], df1[y_kmeans == 2, 1], s = 100, c = 'green', label
plt.scatter(df1[y_kmeans == 3, 0], df1[y_kmeans == 3, 1], s = 100, c = 'cyan', label =
#plt.scatter(df1[y_kmeans == 4, 0], df1[y_kmeans == 4, 1], s = 100, c = 'magenta', lab
#plt.scatter(df1[y_kmeans == 5, 0], df1[y_kmeans == 5, 1], s = 100, c = 'black', label
#plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c
plt.title('Clusters of Districts')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```



In [88]:
```
from scipy.cluster.hierarchy import linkage
import scipy.cluster.hierarchy as sch # for creating dendrogram
```

In [89]:
```
z = linkage(df1, method="complete",metric="euclidean")
```

In [90]:
```
plt.figure(figsize=(15, 10))
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Features')
plt.ylabel('Crime')
sch.dendrogram(z,
    leaf_rotation=0.,  # rotates the x axis labels
    leaf_font_size=8.,  # font size for the x axis labels
)
plt.show()
```


Hierarchical Clustering Dendrogram
Crime
0
2
4
6
8
10
12
14
16
Features
17 20 0 30 11 33 5 10 9 29 19 26 21 13 23 25 15 8 22 7 34 1 32 6 18 4 27 28 12 14 2 3 16 24 31