In [53]:
```
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 [69]:
```
df=pd.read_excel(r'DataFinal17-20.xlsx',sheet_name='2017')
df1=df.drop(['Total'], axis=1)
df1=df1.iloc[:,2:15].values
#scaler = MinMaxScaler()
scaler= StandardScaler()
# transform data
df1 = scaler.fit_transform(df1)
```

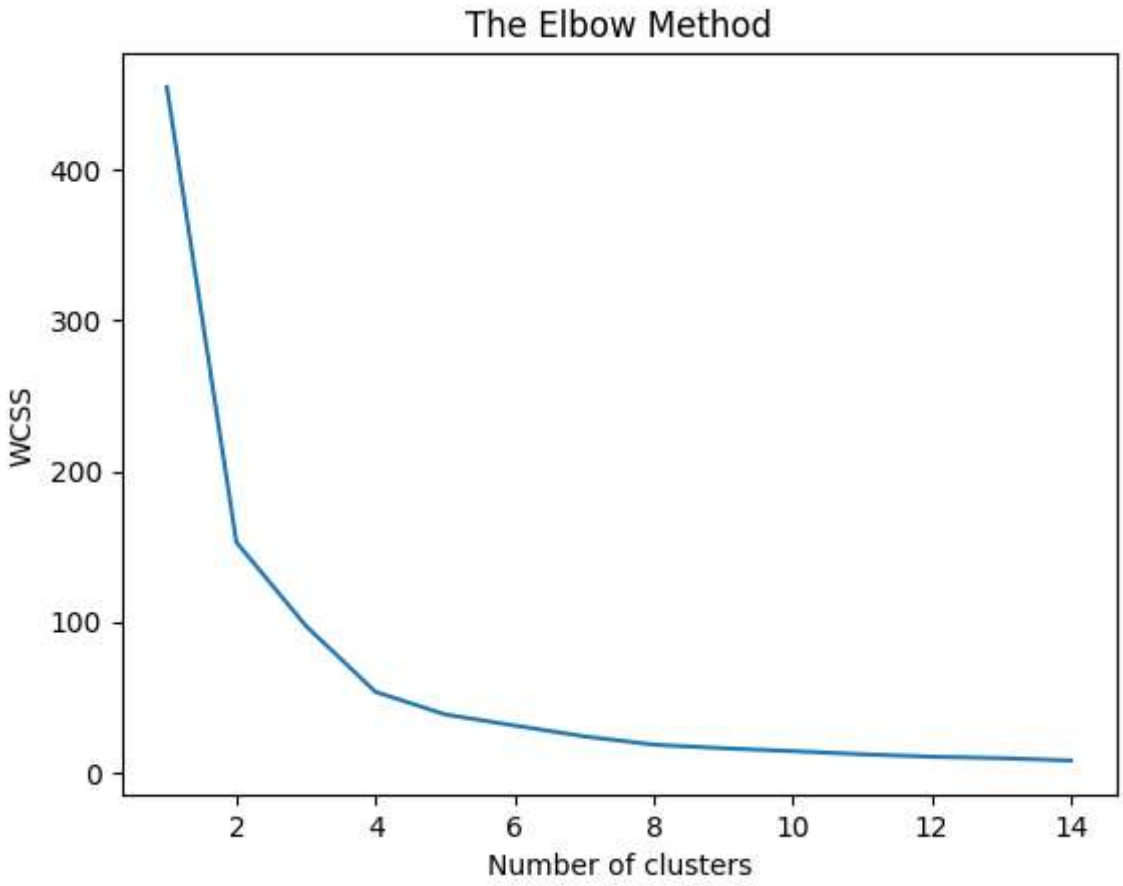
In [70]:
```
from sklearn.cluster import KMeans

#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 [71]:
```
wcss
```

Out[71]:
```
[455.0,
 152.7664789538139,
 97.21044553141512,
 53.40858546286942,
 38.40727304203008,
 31.056381108989736,
 23.84985694654172,
 18.34802812829626,
 15.95205386197719,
 14.043031124213531,
 11.969671693923948,
 10.33141102924851,
 9.348423492063928,
 7.8131790896540645]
```

In [72]:
```
plt.plot(range(1, 15), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



In [73]:
```
km=KMeans(n_clusters = 4, init = 'k-means++', random_state = 0)
y_kmeans=km.fit_predict(df1)
```

In [74]:
```
y_kmeans
```

Out[74]:
```
array([0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 0, 1, 1, 0, 1,
       1, 1, 2, 1, 1, 1, 0, 1, 0, 2, 1, 1, 1])
```

In [75]:
```
df['cluster']=y_kmeans
df.head()
```

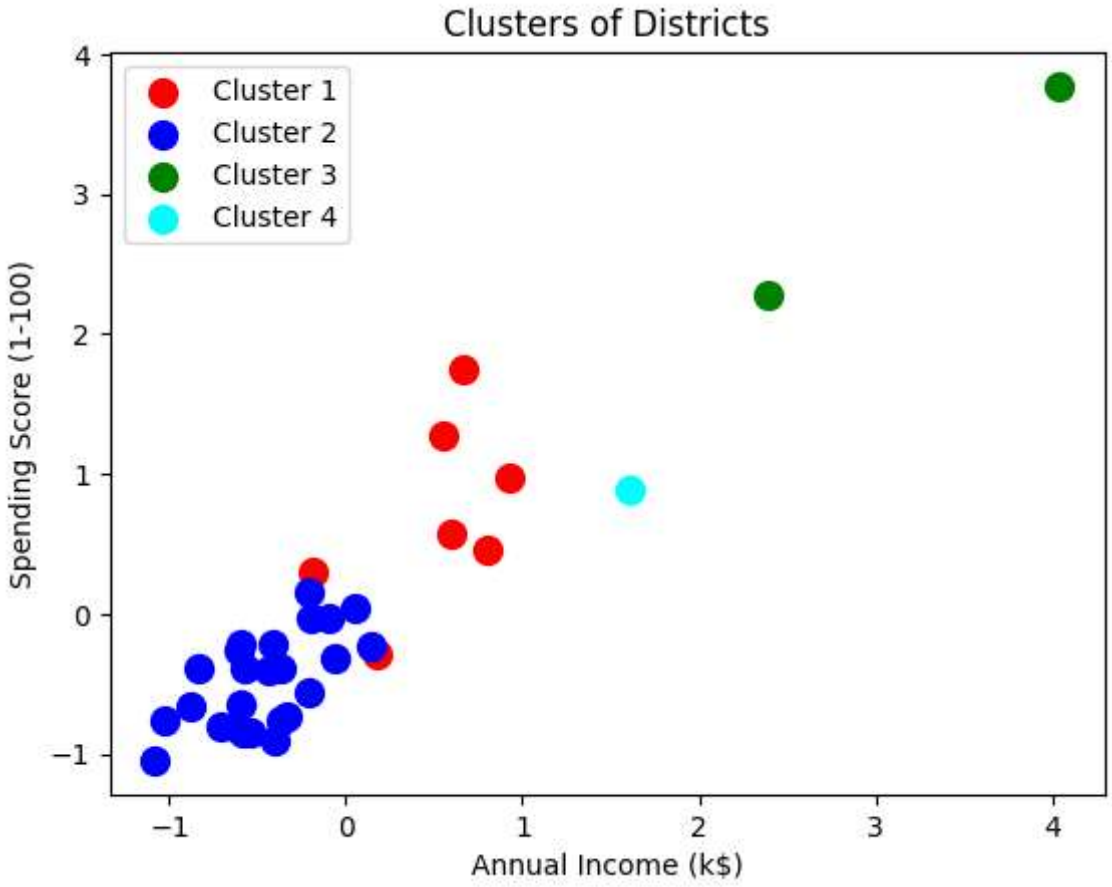
Out[75]:

	SrNo	State/UT/District	Homicide/Murder	Causing death by negligence	Hurt	Assault on woman	Kidnapping and abduction	Human trafficking	Ra
0	1	Ahmednagar	215	715	886	410	303	0	
1	2	Akola	101	143	1161	144	96	0	
2	3	Amravati	168	273	1553	345	215	2	
3	4	Aurangbad	220	515	1613	474	264	0	
4	5	Beed	164	292	925	196	134	1	

5 rows × 23 columns

In [76]:
```
#plt.scatter(df['SrNo'],df['cluster'])
```

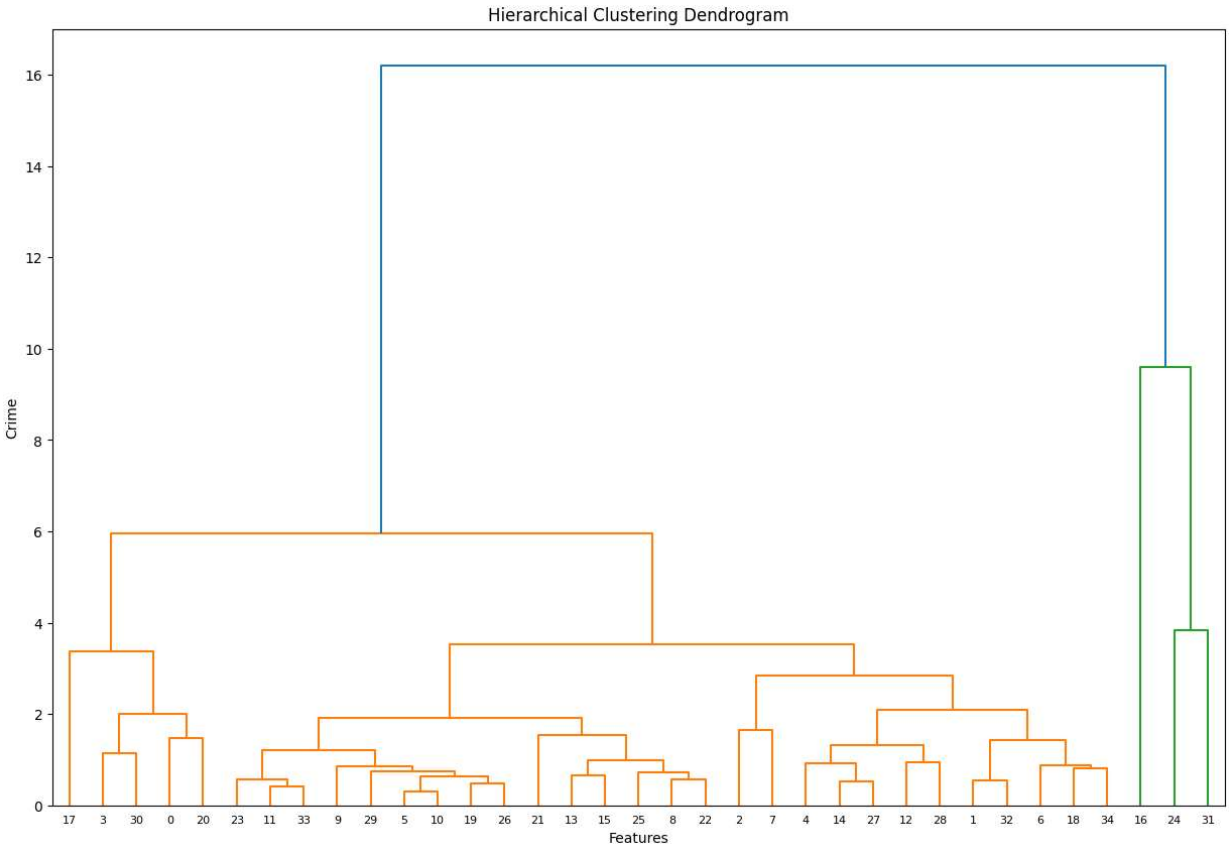
```
#for col in df.columns:
 #   print(col)
plt.scatter(df1[y_kmeans == 0, 0], df1[y_kmeans == 0, 1], s = 100, c = 'red', label =
plt.scatter(df1[y_kmeans == 1, 0], df1[y_kmeans == 1, 1], s = 100, c = 'blue', label =
plt.scatter(df1[y_kmeans == 2, 0], df1[y_kmeans == 2, 1], s = 100, c = 'green', label
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 [77]:
```
from scipy.cluster.hierarchy import linkage
import scipy.cluster.hierarchy as sch # for creating dendrogram
```

In [78]:
```
z = linkage(df1, method="complete",metric="euclidean")
```

In [79]:
```
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
Features
0
2
4
6
8
10
12
14
16
17 3 30 0 20 23 11 33 9 29 5 10 19 26 21 13 15 25 8 22 2 7 4 14 27 12 28 1 32 6 18 34 16 24 31