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 [91]:
```
df=pd.read_excel(r'DataFinal17-20.xlsx',sheet_name='2019')
df1=df.drop(['Total'], axis=1)
df1=df1.iloc[:,2:15].values
#scaler = MinMaxScaler()
scaler= StandardScaler()
# transform data
df1 = scaler.fit_transform(df1)
```

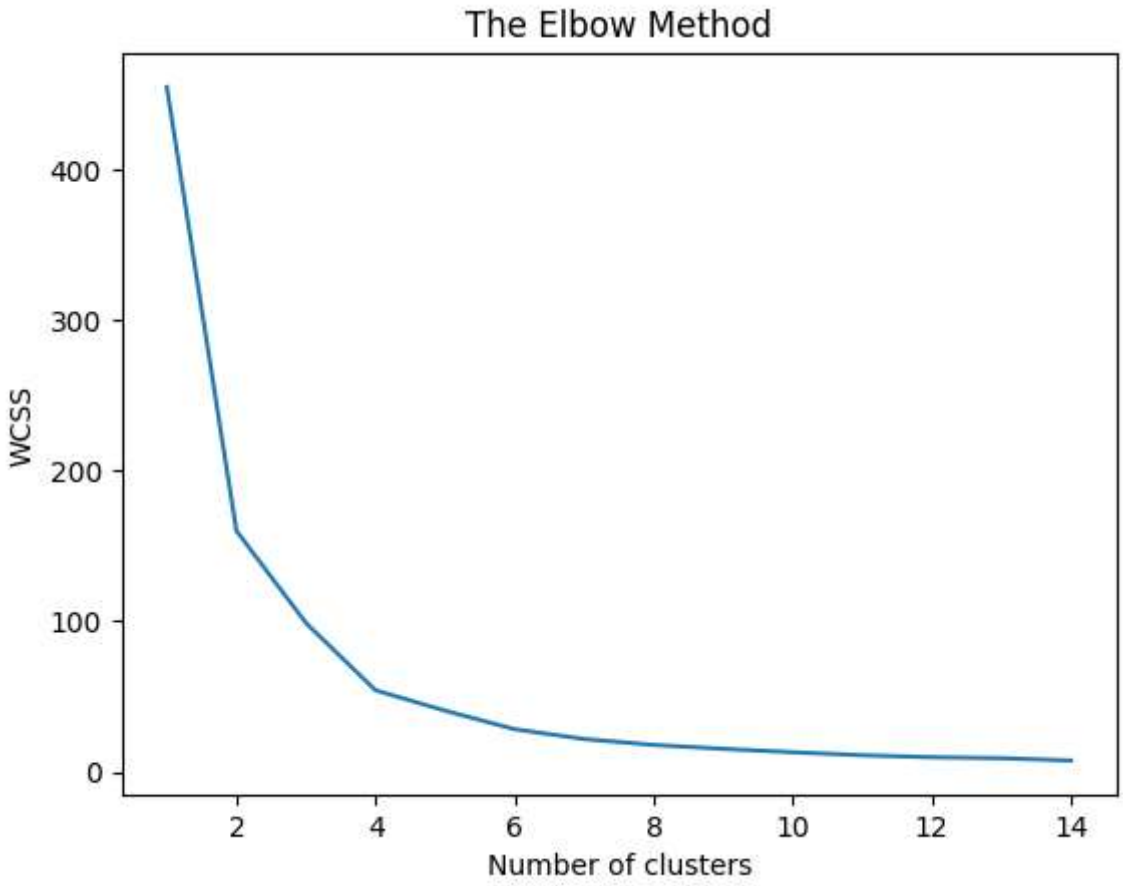
In [92]:
```
from sklearn.cluster import KMeans

#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 [93]:
```
wcss
```

Out[93]:
```
[455.0,
 160.16871868554736,
 98.85558888393712,
 54.114590464152684,
 40.579311227790214,
 28.250766109101185,
 21.7378878789146,
 17.81191386844559,
 15.197416213239427,
 12.971175003258992,
 10.963050359064402,
 9.647866972458212,
 8.99326653752742,
 7.4540376244609865]
```

In [94]:
```
plt.plot(range(1, 15), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



In [95]:
```
km=KMeans(n_clusters = 4, init = 'k-means++', random_state = 0)
y_kmeans=km.fit_predict(df1)
```

In [96]:
```
y_kmeans
```

Out[96]:
```
array([0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 0, 1, 1, 0, 1,
       1, 1, 3, 1, 1, 1, 1, 1, 0, 3, 1, 1, 1])
```

In [97]:
```
df['cluster']=y_kmeans
df.head()
```

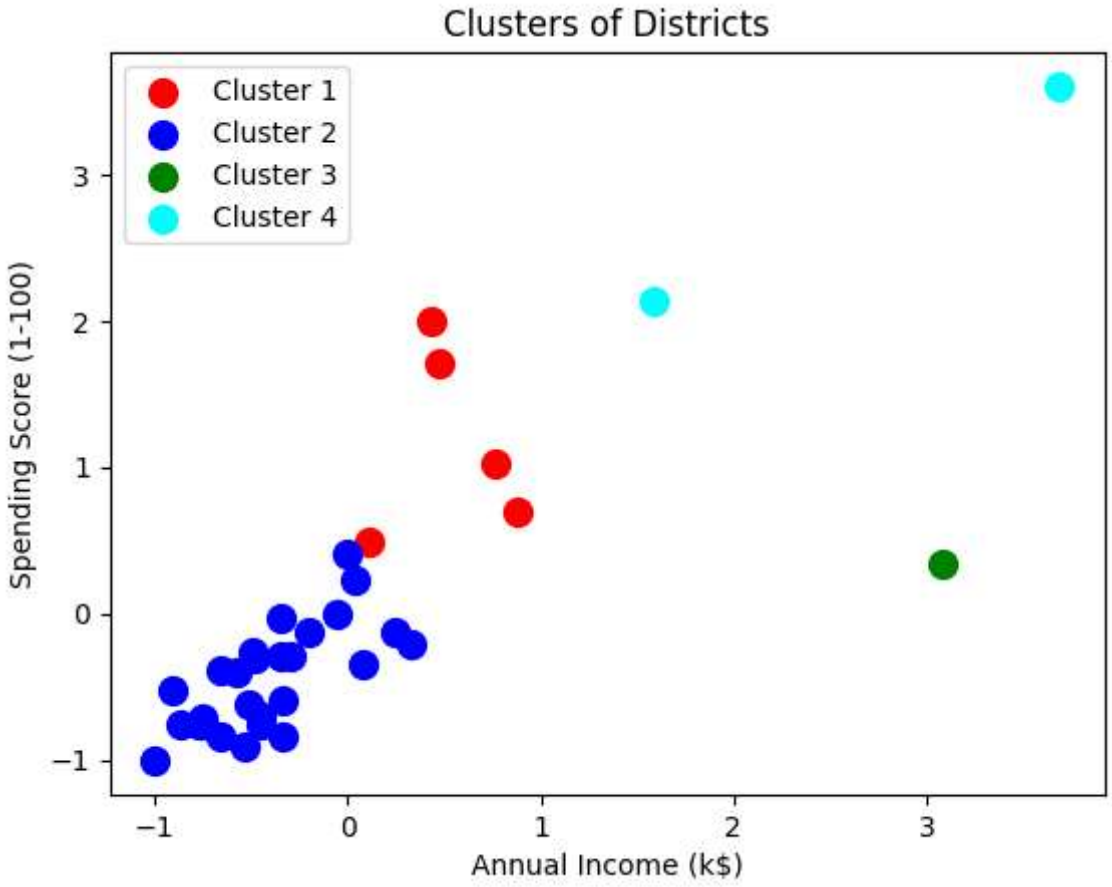
Out[97]:

	S. No	State/UT/District	Homicide/Murder(3,4,15,16)	Causing death by negligence(5 to 12)	Hurt(20,26)	Assault on woman(35)	Kidn abducti
0	1	Ahmednagar	217	832	1362	486	
1	2	Akola	96	156	1167	197	
2	3	Amravati	187	316	1681	400	
3	4	Aurangbad	170	489	1718	344	
4	5	Beed	129	316	955	204	

5 rows × 23 columns

In [98]:
```
#plt.scatter(df['SrNo'],df['cluster'])
```

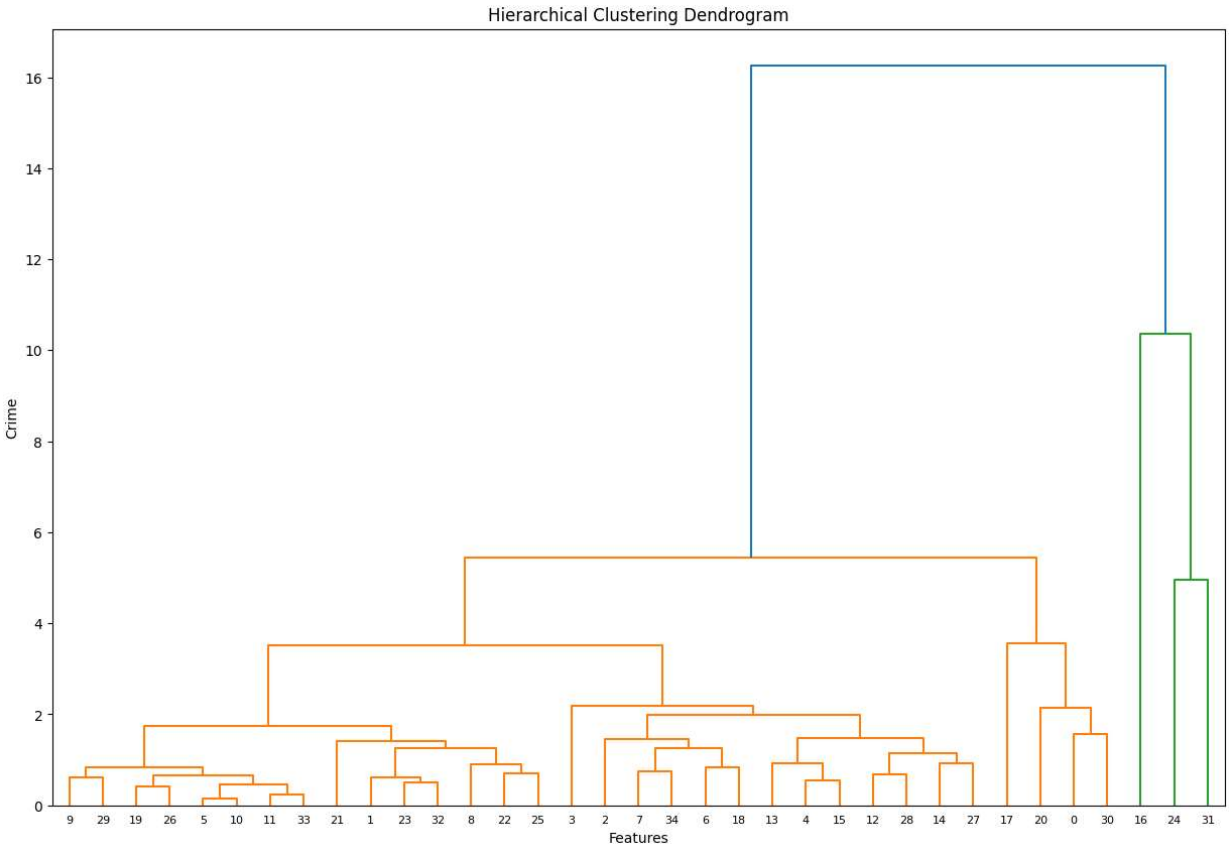
```
#for col in df.columns:
 #   print(col)
plt.scatter(df1[y_kmeans == 0, 0], df1[y_kmeans == 0, 1], s = 100, c = 'red', label =
plt.scatter(df1[y_kmeans == 1, 0], df1[y_kmeans == 1, 1], s = 100, c = 'blue', label =
plt.scatter(df1[y_kmeans == 2, 0], df1[y_kmeans == 2, 1], s = 100, c = 'green', label
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 [99]:
```
from scipy.cluster.hierarchy import linkage
import scipy.cluster.hierarchy as sch # for creating dendrogram
```

In [100…
```
z = linkage(df1, method="complete",metric="euclidean")
```

In [101…
```
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
9 29 19 26 5 10 11 33 21 1 23 32 8 22 25 3 2 7 34 6 18 13 4 15 12 28 14 27 17 20 0 30 16 24 31