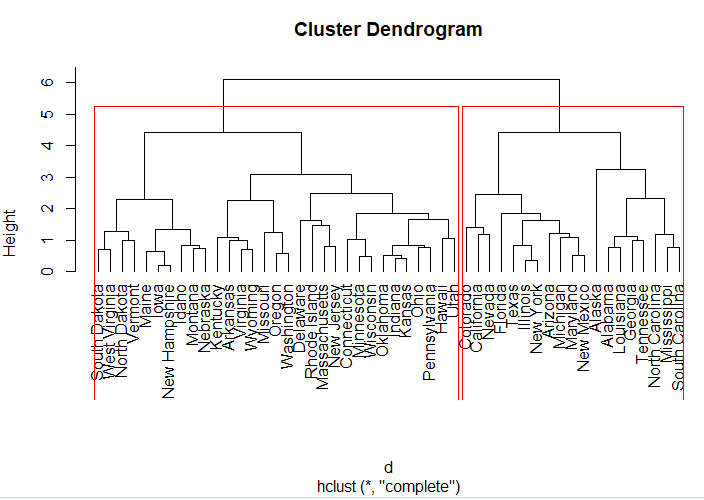
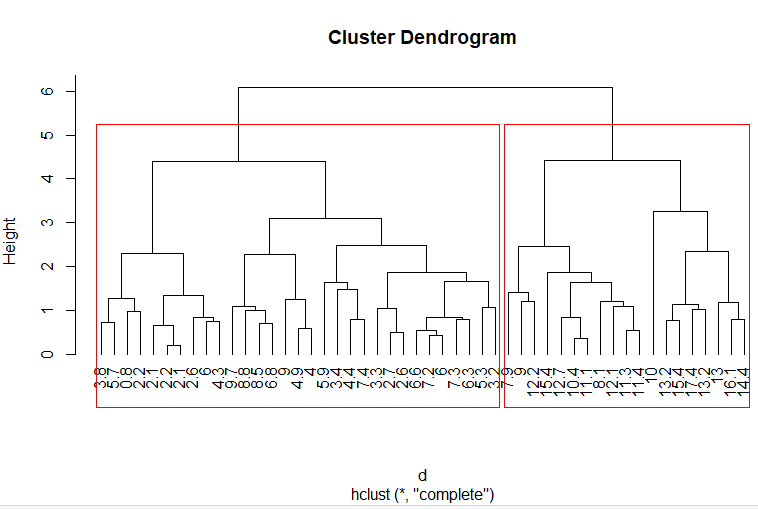
Perform Clustering for the crime data and identify the number of clusters formed and draw inferences.

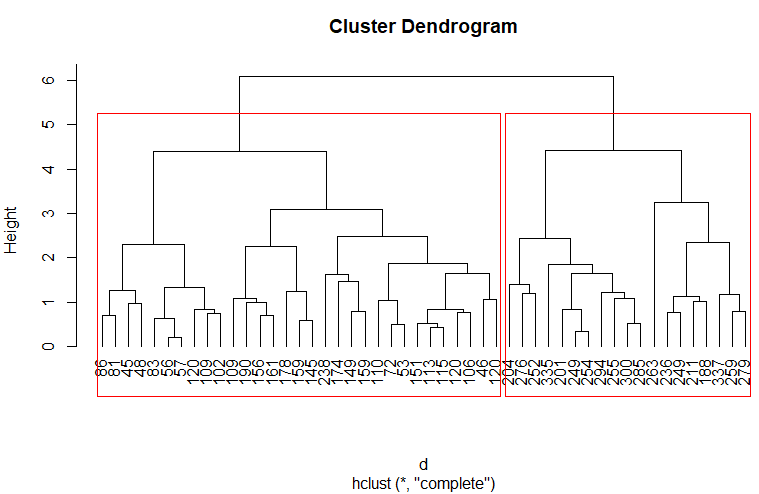
PROBLEM: To infer from crime data.



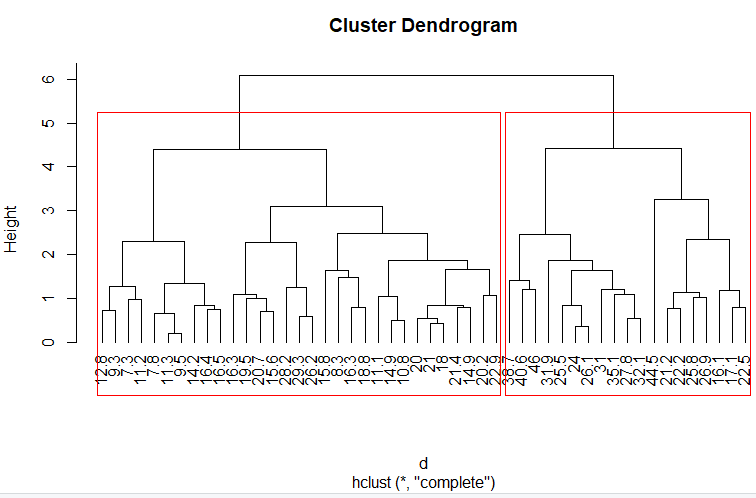
MURDER-RATE



ASSAULT-RATE



RAPE-RATE



|  |  |  |
| --- | --- | --- |
| CASES | GROUP1-1 RATE | GROUP-2 RATE |
| MURDER | 2.1-8.8 | 9.0-17.4 |
| ASSUALT | 86-178 | 188-355 |
| RAPE | 7.3-29.3 | 26.5-46 |

|  |
| --- |
| INFERENCE:  group-2 crime rate is greater than group-1 |

PYTHON CODE

import numpy as np

crime\_data=pd.read\_csv('C:/Users/USER/Desktop/hierical\_clustering/crime\_data.csv')

###for normalizing the data

def norm\_fun(i):

x=(i-i.min())/(i.max()-i.min())

return(x)

df\_norm=norm\_fun(crime\_data.iloc[:,1:])

#####for linkage function and dist

from scipy.cluster.hierarchy import linkage

import scipy.cluster.hierarchy as sch

dist=linkage(df\_norm, method="complete",metric="euclidean")

plt.figure(figsize=(15, 5));plt.title('Hierarchical Clustering Dendrogram');plt.xlabel('Index');plt.ylabel('Distance')

sch.dendrogram(

dist,

leaf\_rotation=0., # rotates the x axis labels

leaf\_font\_size=8., # font size for the x axis labels

)

plt.show()

from sklearn.cluster import AgglomerativeClustering

h\_complete = AgglomerativeClustering(n\_clusters=3, linkage='complete',affinity = "euclidean").fit(df\_norm)

cluster\_labels=pd.Series(h\_complete.labels\_)

crime\_data['clust']=cluster\_labels # creating a new column and assigning it to new column

crime\_data = crime\_data.iloc[:,[5,0,1,2,3,4]]

crime\_data.head()