Perform Clustering(Hierarchical, Kmeans & DBSCAN) for the crime data and identify the number of clusters formed and draw inferences.

Data Description:

Murder -- Muder rates in different places of United States

Assualt- Assualt rate in different places of United States

UrbanPop - urban population in different places of United States

Rape - Rape rate in different places of United States

ANS: import pandas as pd

import numpy as np

from matplotlib import pyplot as plt

import seaborn as sns

import scipy.cluster.hierarchy as sch

from sklearn.cluster import AgglomerativeClustering

crime=pd.read\_csv('E:\\crime\_data.csv')

crime.head()

def norm\_func(i):

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

return(x)

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

df\_norm.head()

dendrogram=sch.dendrogram(sch.linkage(df\_norm,method='average'))

hc = AgglomerativeClustering(n\_clusters=3, affinity = 'euclidean', linkage = 'complete')

hc

y\_hc=hc.fit\_predict(df\_norm)

y\_hc

crime['h\_clusterid']=hc.labels\_

crime.head()

# Kmeans

from sklearn.cluster import KMeans

crime1=pd.read\_csv('E:\\crime\_data.csv')

crime1.head()

def norm\_func(i):

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

return(x)

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

wcss=[]

for i in range(1,11):

kmeans=KMeans(n\_clusters=i)

kmeans.fit(df\_norm)

wcss.append(kmeans.inertia\_)

plt.plot(range(1,11),wcss)

plt.title('Elbow curv')

plt.xlabel('number of clusters')

plt.ylabel('WCSS')

plt.show()

model=KMeans(n\_clusters=4)

model.fit(df\_norm)

model.labels\_

x=pd.Series(model.labels\_)

crime1['Clust']=x

crime1.head()

crime1.iloc[:,1:5].groupby(crime1.Clust).mean()

# DBSCAN

from sklearn.cluster import DBSCAN

from sklearn.preprocessing import StandardScaler

crime2=pd.read\_csv('E:\\crime\_data.csv')

crime2.info()

df=crime2.iloc[:,1:5]

df.values

stscaler=StandardScaler().fit(df.values)

x=stscaler.transform(df.values)

x

dbscan=DBSCAN(eps=2,min\_samples=5)

dbscan.fit(x)

dbscan.labels\_

cl=pd.DataFrame(dbscan.labels\_,columns=['cluster'])

cl

pd.concat([crime2,cl],axis=1)