Experiment 3.1

Student Name: Yash Gupta **UID: 20BCS5009**
Branch: BE-CSE **Section/Group: 20BCS_DM-716 B**
Semester: 6 **Date of Performance: 26/04/23**
Subject Name: DM LAB **Subject Code: 20CSP_376**

AIM :-

To perform the hierarchical clustering using R programming.

Theory :-

Cluster analysis or clustering is a technique to find subgroups of data points within a data set. The data points belonging to the same subgroup have similar features or properties. Clustering is an unsupervised machine learning approach and has a wide variety of applications such as market research, pattern recognition, recommendation systems, and so on. The most common algorithms used for clustering are Kmeans clustering and Hierarchical cluster analysis. In this article, we will learn about hierarchical cluster analysis and its implementation in R programming.

Hierarchical cluster analysis (also known as hierarchical clustering) is a clustering technique where clusters have a hierarchy or a predetermined order. Hierarchical clustering can be represented by a treelike structure called a Dendrogram. There are two types of hierarchical clustering:

- Agglomerative hierarchical clustering: This is a bottom-up approach where each data point starts in its own cluster and as one moves up the hierarchy, similar pairs of clusters are merged.

- Divisive hierarchical clustering: This is a top-down approach where all data points start in one cluster and as one moves down the hierarchy, clusters are split recursively.

Output :-

Performing Hierarchical Cluster Analysis using R

For computing hierarchical clustering in R, the commonly used functions are as follows:

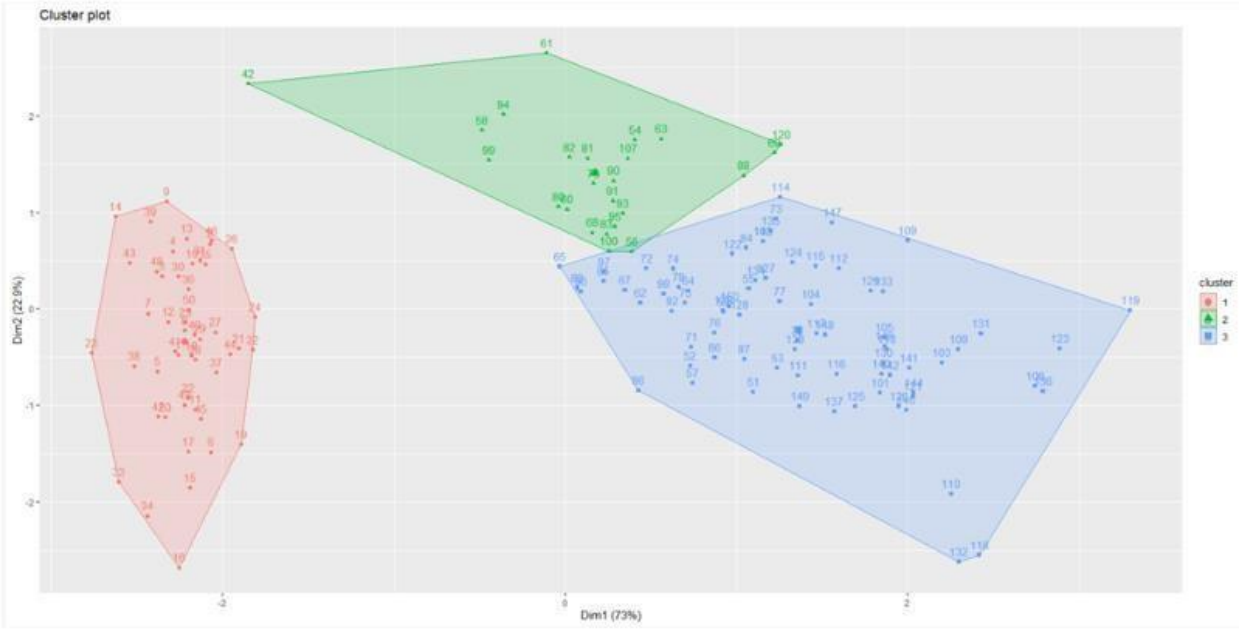
· **hclust** in the stats package and **agnes** in the cluster package for agglomerative hierarchical clustering.

· **diana** in the cluster package for divisive hierarchical clustering.

We will use the Iris flower data set from the datasets package in our implementation. We will use sepal width, sepal length, petal width, and petal length column as our data points. First, we load and normalize the data. Then the dissimilarity values are computed with *dist* function and these values are fed to clustering functions for performing hierarchical clustering.

```
# Load required packages
library(datasets) # contains iris dataset
library(cluster)  # clustering algorithms
library(factoextra) # visualization
library(purrr) # to use map_dbl() function
# Load and preprocess the dataset
df <- iris[, 1:4]
df <- na.omit(df)
df <- scale(df)
# Dissimilarity matrix
d <- dist(df, method = "euclidean")
# Hierarchical clustering using Complete Linkage
hc1 <- hclust(d, method = "complete" )
# Plot the obtained dendrogram
plot(hc1, cex = 0.6, hang = -1)
```

```
# Cut tree into 3 groups
sub_grps <- cutree(hc1, k = 3)  # Visualize the result in a scatter plot
fviz_cluster(list(data = df, cluster = sub_grps))
```



```
> #applying single link clustering algorithm to the model
>
> h1 = hclust(dist,method='single')
>
> h1

Call:
hclust(d = dist, method = "single")

Cluster method   : single
Distance         : euclidean
Number of objects: 150
```

CODE:-

```
a=studentData

a=a[,-5]

dist <- dist(a,method='euclidean')

#applying single link clustering algorithm to the model

h1 = hclust(dist,method='single') h1

#plotting the dendogram

plot(h1,hang=-1, main='single link')

#Cutting tree by height c=cutree(h,3)

ColorDendrogram(h1,y=c, main='Single Link')

#applying average link clustering algorithm to the model

h2=hclust(dist,method='average') h2

#plotting the dendogram plot(

h2, hang=1, main='Average Link') #Cutting tree by height c=cutree(h,3)

ColorDendrogram(h2,y=c, main='Average Link')

#applying complete link clustering algorithm to the model

h3=hclust(dist,method='complete') h3

#plotting the dendogram plot(h3,hang=-1,main='complete link')

#Cutting tree by height
c=cutree(h,3)
```

ColorDendrogram(h3,y=c, main='Complete Link')