Cluster Analysis with R

**1. Objective**

In our previous [**R tutorial**](http://data-flair.training/blogs/r-programming-tutorial/) we have discussed [**Graphical data analysis with R**](http://data-flair.training/blogs/graphical-data-analysis-with-r/), now we will take you through the cluster analysis in R. First of all we will see what is R Clustering, then we will see the Applications of Clustering, Clustering by Similarity Aggregation, use of R amap Package, Implementation of Hierarchical Clustering in R and examples of R clustering in various fields.

[Get the best R books to become a master in R Programming Language.](http://data-flair.training/blogs/r-best-books-to-learn-r/)

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/clusterning-in-r.jpg)

**2. Introduction to Clustering in R**

**Clustering** is a data segmentation technique that divides huge datasets into different groups on the basis of similarity in the data. It is a statistical operation of grouping objects. The resulting groups are clusters. Clusters have the following properties:

* We find them during the operation and their number is also not always fixed in advance.
* They are the combination of objects having similar characteristics.

Clustering is one of the most widespread descriptive methods of [**data analysis**](http://data-flair.training/blogs/data-analytics-comprehensive-guide/) and **data mining**. We use it when data volume is large to find homogeneous subsets that we can process and analyze in different ways.

For example, a food product manufacturing company can categorize its customers on the basis of purchased items and cost of those items.

**3. Applications of Clustering**

Following are the main Clustering applications:

* **Marketing –** In this field, clustering is useful in finding customer profiles that make customer base. After detecting clusters, a business can develop a specific strategy for each cluster base. We can use clusters to keep track of customers over months and detect a number of customers who moved from 1 cluster to other.
* **Retail –** In the retail industry, we use clustering to divide all stores of a particular company into groups of establishments on basis of type of customer, turnover etc.
* **Medical Science –** In medical, we use clustering discover a group of patients suitable for particular treatment protocols. Each group comprises all patients who react in the same way. Formation of these groups is on basis of age, type of disease etc. We can also us clustering in the classification of the *protein sequence, ct-scans* etc.
* **Sociology –** We use Clustering in performing data mining operations here. We divide the population into groups of individuals who are homogeneous in terms of social demographics, lifestyle, expectations etc. We can then use the categorization for purposes like polls, identifying criminals etc.

In different fields, clustering has different names, such as:

* **Marketing –** In marketing, ‘segmentation’ or ‘typological analyses’ term is available for clustering.
* **Medicine –** In the field of medicine, the term, nosology, for clustering.
* **Biology –** In the field of biology, numerical taxonomy is the term for clustering.

To define correct criteria for clustering and making use of efficient algorithms, the general formula is as follows:

***Bn(number of partitions for n objects)>exp(n)***

You can determine the complexity of clustering by the number of possible combinations of objects. The complexity of the cluster depends on the number of possible combinations of objects.

The basis for joining or separating objects is the distance between them. These distances are dissimilarity (when objects are far from each other) or similarity (when objects are close by).

**4. Methods for Measuring Distance between Objects**

To calculate the distance between the objects, we apply certain types of methods. Some of them are:

* **Euclidean Distance –** It is the most common method used. It is the geometric measure of distance between objects in a multidimensional space.

In general, for an n-dimensional space, the distance is

* **Squared Euclidean Distance –** Here we get the distance by *squaring Euclidean distance*. We can use it to put more weight on the objects that are at greater distances.
* **City-Block (Manhattan) Distance –** We get it by calculating average difference between 2 points in all dimensions. In most cases, it is same as Euclidean distance. It also finds use in reducing the effect of extreme individuals whose coordinates are not squared.

The total sum of squares or inertia is the weighted mean of squares of the distances of each point from the center of gravity of that cluster. Thus we calculate it by adding the within-cluster sum of squares with the between-cluster sum of squares.

e calculate it by adding the within-cluster sum of squares with the between-cluster sum of squares.

We calculate Sum of Squares of Clusters on its center of gravity as below:

***Total Sum of Squares (I) = Between-Cluster Sum of Squares (IR) + Within-Cluster Sum of Squares (IA)***

This formula is **Huygen’s Formula** and Dutch mathematician and a philosopher **Christiaan Huygens** gave it.

We calculate Between-Cluster Sum of Squares by finding the square of difference from the center of gravity for each cluster and then adding them. As it increases, the separation between clusters also increases indicating satisfactory clustering.

We calculate Within-Cluster Sum of Squares by finding the square of difference from the center of gravity for each cluster and then adding them within in a single cluster. As it diminishes, clustering of the population becomes better.

**R2 (RSQ)** is the proportion of the sum of squares explained by the clusters (between-cluster sum of squares/total sum of squares). The nearer it is to 1, the better the clustering will be, but we should not aim to maximize it at all costs because this would result in the largest number of clusters: there would be one cluster per individual. So we need an R2 that is close to 1 but without too many clusters. A good rule is that, if the last significant rise in R2 occurs when we move from k to k + 1 clusters, the partition into k+1 clusters is correct.

Below are few of the properties of Efficient Clustering:

* Detection of the structures present in the data
* Easy determination of optimal number of clusters
* Yielding of clearly differentiated clusters
* Yielding of clusters that remain stable with minor changes in data
* Processing of large data volumes efficiently
* Handling of all types of variables if required

**Note:**In the calculation of the sum of squares, either IR is large or IA is small for correct clustering.

To distinguish true clusters in data, we often have to interpret the data before transforming, adding or excluding variables, and then restart the clustering. Excluding a variable does not necessarily mean deleting it from the analysis base. Instead, we stop to take it into account in the clustering operation, while retaining it as an inactive variable to observe the distribution of its categories in the various clusters. It is no longer an ‘**active**’ variable but becomes an ‘**illustrative**’ variable (also called ‘**supplementary**’ variable).

**5. Agglomerative Hierarchical Clustering**

The**Agglomerative hierarchical clustering (AHC)** is a form of clustering that produces a sequence of nested partitions between partitions into n clusters. These nested partitions are of increasing heterogeneity. In this technique, we isolate each object and partition into 1 cluster which includes all the objects. We can use AHC if there is a distance between partitions which can be in either an individual space or a variable space. To categorize data, you must define the distance of 2 objects or clusters.

For this, the general form of the algorithm is as follows:

* Observations are the initial clusters.
* Calculate the distance between the clusters.
* Merge the two closest clusters together and replace with a single cluster.
* Repeat Step 2 and the complete process until a single cluster containing all the observations remains.

The tree generated by AHC is also known as dendogram. We can cut it to get clusters.

Usage of **Hierarchical clustering** is for identification of patterns in digital images, predicting stock exchange and in text mining and ontology. It also finds the use for research on protein sequence classification.

The working of AHC involves searching for the closest points at each step and merging them. We can define the distance in many ways but most usual definitions are as below:

**i) Main Distances –**

* **Maximum distance –** the Largest distance between 2 observations tends to generate clusters of equal diameter.
* **Minimum distance –** Least distance between 2 observations defines the nearest neighbor technique or single linkage AHC method.

The **minimum distance** between the points of different clusters should be greater than the **maximum distance** between the points of the same cluster. These distances are the main distances.

**ii) Density Estimation –**

These are among the most suitable for detecting structure of complex clusters. Below are three typical methods for density estimation in clustering:

* **The k-nearest-neighbors method –**According to this method, the density at a point x is the number k of observations in a sphere centered on x, divided by the volume of the sphere.
* **The uniform kernel method –**Here we fix the **r**adius of the sphere and not the number of neighbors.
* **The Wong hybrid method –**It finds use in a preliminary analysis.

The 3 methods are effective for detecting all types of clusters irregularly shaped ones, which are of unequal sizes and have variances. The distance between 2 clusters is inversely proportional to the density in the middle of these 2 clusters.

**Note:** The density estimation methods specify a smoothing parameter which can be a number of clusters of **k-means**, number k of neighbors of each point x or radius r of a sphere surrounding x.

**6. Clustering by Similarity Aggregation**

This is also known as **relational clustering** or voting method or the **Condorcet method**.

Clustering by Similarity Aggregation finds use to compare all the individuals in pairs at each step to build a global clustering. It follows the principle of an **equivalence relation**, where iRj, if i and j are in the same cluster.

An equivalence relation has three properties, namely **reflexivity**, **symmetry**, and **transitivity**.

Reflexivity => Mii= 1

Symmetry => Mij= Mij

Transitivity => Mij +Mjk –Mik<=1

The clustering by similarity aggregation follows an intuitive approach. Here we assign a pair of individual values (A, B) to two vectors m(A, B) and d(A, B). m(A, B)contains the same values for both A  and B, while d(A,B) contains different values for A and B.

The **Condorcet criterion**for two individuals A and B is as follows:

c(A, B) = m(A, B)-d(A, B)

The **Condorcet criterion** of an individual A and a cluster S is as follows:

c(A,S) = Σic(A,Bi)

The summation is being over all the Bi ∈ S.

Given the preceding conditions, you start constructing clusters by placing each individual A in the cluster S for which c(A, S) is largest and at least 0.

Calculate Global Condorcet criterion by summing all individuals in A and clusters SAto which they have been assigned. In practice, 2 iterations will be enough to provide good results.

**Note**: Several iterations follow until we reach the specified largest number of iterations or the global Condorcet criterion no more improves.

**7. Use of the R amap Package**

For clustering by similarity aggregation, R provides the amap package. First, we load the amap package from the R library, after that, we use it for clustering.

**Loading the amap Package**

>library(amap)

[](https://data-flair.training/blogs/r-interview-questions-and-answers/)

**Performing Similarity Aggregation**

> pop(matrix)

**Note**: Only after transforming the data into factors and converting the values into whole numbers, we can apply similarity aggregation.

**8. K-Means Clustering**

The**k-means**is the most widely used method for customer segmentation of numerical data. This technique partitions n units into k ≤ n distinct clusters, S = {S1, S2, . . . , Sk }, to reduce the within-cluster sum of squares. You can use kmeans function in R package stats. This algorithm is fast and reliable. But there is no guarantee that it converges to the global optimum. Its final result may depend on how is the starting of the algorithm.

For k-means clustering, two common initialization approaches are as follows:

* You can randomly choose k units from the dataset and use these as the initial cluster means.
* You can randomly assign one of the k clusters to each unit and then proceed to the update step, and compute initial means as the centroids of the clusters’ randomly assigned units. This random partition method is preferable generally.

The **k-means clustering** is the most common R clustering technique. Some of the applications of this technique are as follows:

* Predicting the price of products for a specific period or for specific seasons or occasions such as summers, New Year or any particular festival.
* Extracting information from electric price by time series models.

**9. Example of R Clustering**

**9.1. European Protein Consumption**

Consider 25 European countries (n=25 units) and their protein intakes (in percent) from nine major food sources (p= 9).

Let us start with clustering on the first 2 features, protein intake from red and white meat, to cluster the 25 countries into 3 groups.

The following code applies the R clustering technique on the dataset:

**a) Application of Clustering on European Protein Consumption Data**

1. food <- read.csv("C:/DataMining/Data/protein.csv") //Reads data available in protein.csv file
2. set.seed(1) // Fixes random starting for clusters from 1.
3. grpMeat <- kmeans(food[,c("WhiteMeat","RedMeat")], centers=3, + nstart=10) //Specifies clustering of data of red and white meat with 3 clusters.

**b) Application of Clustering on European Protein Consumption Data**

1. o=order(grpMeat$cluster) // Sorts the cluster on basis of number clustering
2. data.frame(food$Country[o],grpMeat$cluster[o]) // Lists the clusters along with country name.

Now you can use the following command to plot the scatter plot:

**c) Plotting the Scatterplot for the Clusters**

plot(food$Red, food$White, type="n", xlim=c(3,19), xlab ="Red Meat",+ ylab="White Meat")

In R, you can easily generate different numbers of clusters for the same data by assigning different values to the centers attribute in the **kmeans()** command.

**[Test Your R Knowledge](https://data-flair.training/blogs/category/quiz/r-quiz/" \t "_blank)**

**9.2. Monthly US Unemployment Rates**

This example analyzes monthly seasonally adjusted unemployment rates covering the period **January 1976 through August 2010**for the **50 US states**(n = 50).

Here you will use summary statistics for each state, such as average and standard deviation of employment rates and then use these 2 calculated features of monthly unemployment rates as attributes of clustering.

You can read unemp.csv and set seed for the cluster as done earlier.

Apply k-means Clustering in R by below command:

grpunemp <- kmeans(unemp[,c("mean","stddev")], centers=3, + nstart=10)

Sort the clusters by below way:

o=order(grpunemp$cluster)

You can now plot the clusters on the graph by using below command:

1. plot(unemp$mean,unemp$stddev,type="n",xlab="mean", + ylab="stddev")
2. text(x=unemp$mean,y=unemp$stddev,labels=unemp$state,+
3. col=grpunemp$cluster+1)

**11. Implementing Hierarchical Clustering in R**

Hierarchical clustering is an approach of clustering n units wherefore each described by p features into a smaller number of groups. Furthermore, It creates a hierarchy of clusters that we can represent in a tree-like diagram, called a dendrogram.

AHC uses a bottom-up approach where each unit starts in its own cluster and merge pairs of clusters as you move up in the hierarchy. However, For conglomerative clustering, you can use **agnes()** [**function in R**](http://data-flair.training/blogs/r-programming-functions/) package or **hclust()** function in stats package.

Let us consider the example of European protein consumption in the context of hierarchical clustering.

**Applying Hierarchical Clustering**

foodagg=agnes(food,diss=FALSE,metric="euclidian")

Here argument dist=FALSE indicates that you have used dissimilarity matrix that is being calculated from raw data.

Argument metric=”euclidian” indicates that Euclidian distance is being used.

You can generate dendrogram as below:

plot(foodagg)

A distance dP between two clusters is inversely proportional to the density in the middle of these two clusters. Here the assumption is that dP = ∞ if the two clusters are not adjacent.

If you have any query related to R clustering, so feel free to share with us. We will be happy to solve your queries.