# Agglomerative Hierarchical Clustering

**Filename:** AHC.R

## Functions used:

* singleLinkage–
  + take data matrix
  + compute the data matrix based on minimum distance between clusters
  + output -> final dendrogram/ tree
* completeLinkage–
  + take data matrix
  + compute the data matrix based on maximum distance between clusters
  + output -> final dendrogram/ tree
* meanLinkage–
  + take data matrix
  + compute the data matrix based on average distance between clusters
  + output -> final dendrogram/ tree
* centroidLinkage–
  + take data matrix
  + compute the data matrix based on centroid of clusters
  + output -> final dendrogram/ tree

## Algorithm:

Load the ncidata set

Compute distance matrix which will be a 64 \* 64 matrix of distances Set diagonal + lower triangle values to ‘NA’

Initialize leaf nodes of the dendrogram by setting initial obs as index values and creating leaf list of 64 initial leaf nodes

Compute the dendrogram for each of the following types of linkages:

* singleLinkage
* completeLinkage
* meanLinkage
* centroidLinkage



Set current matrix as main ncidata distance matrix

While the computed node doesn’t have nrow(ncidata) number of values (just one cluster remaining): Calculate min element

Get location of element ( n \* m )

Replace n row and m column with ‘NA’ Generate parent node

Add observations of parent node as concatenated obs of left and right nodes Add min dist element as the height

Append to leaf list

Generate an empty compare matrix

For each observation in parent node:

Get distance row from main ncidata matrix Initialize distance vactor

For each data element in nrow(ncidata) (first 64 leaves):

If obs != data element and data is not in the obs list of parent node: Compute distance of observation and data element

Append to distance array Else append ‘NA’

For each data element from nrow(ncidata)+1:nrow(current matrix) Create data cluster array and observation array

If observation list is not equal to sample list: For each cluster:

Compute distance to nodes of other cluster Append to cluster array

Get min element from cluster Append to distance array

Else append ‘NA’

Add all distance arrays to compare matrix

Compute minimum distances for each column that form the final min distance list Append min distance list to current matrix as a row and column

Set parent node values to ‘NA’ for new bound vectors

If length of parent node’s observations == nrow(ncidata): (only one cluster remaining) Break

## Observations:

1. Final observations

The final cluster has 64 elements forming the root of the dendrogram. The clusters are formed from the bottom up by adding one observation at a time and creating parent nodes one by one from left to right.

Graphical user interface, text, application

Description automatically generated

1. Dendrogram/ tree:

As shown below the generated tree has 64 total observations and the leaves are broken down into clusters. The height is calculated using the minimum distance (single linkage) distance. The tree splits into 1:63, 1:62 going downwards. We can observe that the dendrogram is a bottom up tree built using the setClass in R. It begins at the leaves and groups observations into clusters as the level of the tree increases. Each child is stored at the left/ right node whereas their observations (indices) are stored by grouping them in the parent’s observations.

Table

Description automatically generated