Hierarchical Clustering - Dendrogram

Drew Sandberg | [drew.sandberg@gmail.com](mailto:drew.sandberg@gmail.com)

4/25/2020

Dataset column names and sampling probabilities derived from copyrighted (@2017)data by DecisionPro, Inc. used in MIS 722 - Marketing Engineering and Analytics lesson plan Dr. LinLin Chai, NDSU. Data set values are generated from sampling 0 - 10 using probabilities

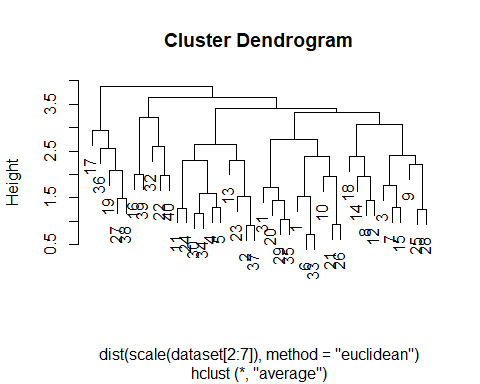
#set seed so dataset values, as calcuated from sampling probabilities, are generated with the same values each time.  
set.seed(141)  
  
#construct the dataset; i.e the "segmentation data"  
#observations = customer/respondent feedback  
#sample function says to generate 40 numbers between 0 and 10 where each of the numbers (i.e. 0, 1,2,3 etc.)   
#has the following probability of occuring in the sample.  
dataset = data.frame(  
 observation = seq(1,40,1), #generate customers 1 through 40  
 variety = sample(0:10,40,replace = TRUE, prob = c(0, 0, 0, 0, 0.1, 0.075, 0.1, 0.225, 0.175, 0.075, 0.25)),  
 electronics = sample(0:10,40,replace = TRUE, prob = c(0, 0.075, 0.05, 0.15, 0.175, 0.225, 0.225, 0.025, 0.075, 0, 0)),  
 furniture = sample(0:10,40,replace = TRUE, prob = c(0.1, 0.125, 0.25, 0.075, 0.075, 0.1, 0.125, 0.15, 0, 0, 0)),  
 QoS = sample(0:10,40,replace = TRUE, prob = c(0, 0.15, 0.2, 0.125, 0.25, 0.075, 0, 0.025, 0.05, 0.125, 0)),  
 price = sample(0:10,40,replace = TRUE, prob = c(0, 0.025, 0.225, 0.1, 0.125, 0.175, 0, 0.125, 0.075, 0.075, 0.075)),  
 returns = sample(0:10,40,replace = TRUE, prob = c(0, 0, 0.15, 0.225, 0.1, 0.25, 0.125, 0.075, 0.075, 0, 0))  
 )

Next, normalize the values in the dataset, but omit the observation column. The dendrogram is created with the hclust function

#normalizing the data using scale function  
#omitting the first column; normalizing only the variety, electornics... returns values  
scaled\_df = scale(dataset[2:7])  
  
#hc = hierarchical clustering  
#'dist' argument of hclust function computes a distance matrix between all values  
hc = hclust(dist(scale(dataset[2:7]), method = "euclidean"), method = "average")

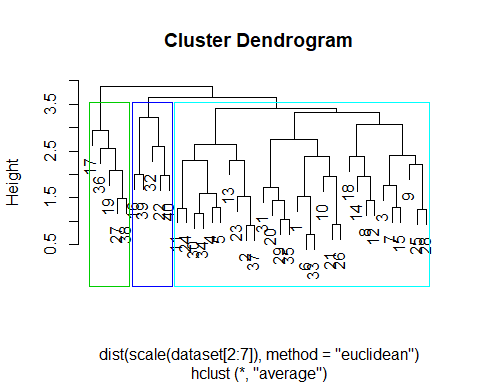
Next, print/plot the dendrogram

plot(hc)



To better visualize the clusters, let R draw colored rectangles around the clusters – based upon your desired number of clusters.

#draw pretty colors around clusters; you pick the number of clusters using the k argument (e.g. k = 3)  
#border defines the colors you want to use, or you can used   
plot(hc)  
rect.hclust(hc, k = 3, border = 3:9)



Alternatively, you can define border color using an index of colors

plot(hc)  
rect.hclust(hc, k = 4, border = 2:6)

