Assignment 3: Big Data Analytics
Hierarchical Clustering on Cereals
By: Valay Shah

Purpose

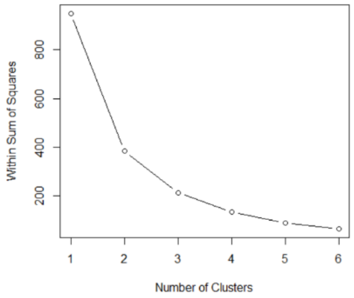
The purpose of this report is to analyze the data file Cereals.csv which contains information on 77 different cereals and their characteristics. For the analysis, all cereals with missing values were removed, hierarchical clustering was applied to the data using Euclidean distance, and normalised measurements. Finally, the dendograms for single and complete linkages were compared and a recommendation for the number of clusters and cluster membership was determined.

Removing all cereals with missing values

There are three rows which will be removed because of carbo, sugar, or potass columns containing blank cells. This will lead to the database going from 76 observations to 74 observations.

Hierarchical clustering

Before we perform hierarchical clustering, non-numeric columns must be removed which means mfr and type columns will be removed from the database. These two variables are not entirely relevant to the database as the manufacturer of the cereal does not matter and almost all the cereal are the same type (type C). Using sapply will change all the numbers to standardized numbers and the distance between the variables will be normalized. Within sum of squares (wss) plot was used to help determine the number of clusters which is ideal for a lower variance between the points within the clusters.

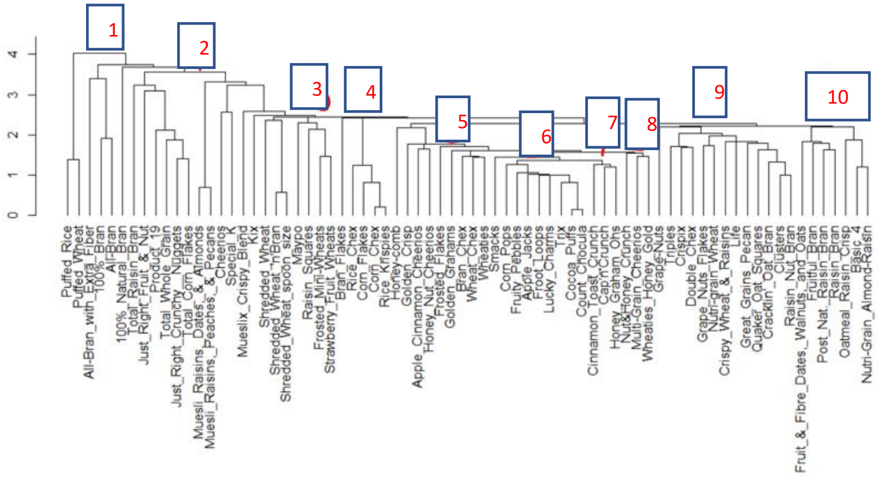


From the plot we can observe that 5 or 6 clusters are ideal for having a low sum of squares which is ideal due to low variance between the points within a cluster.

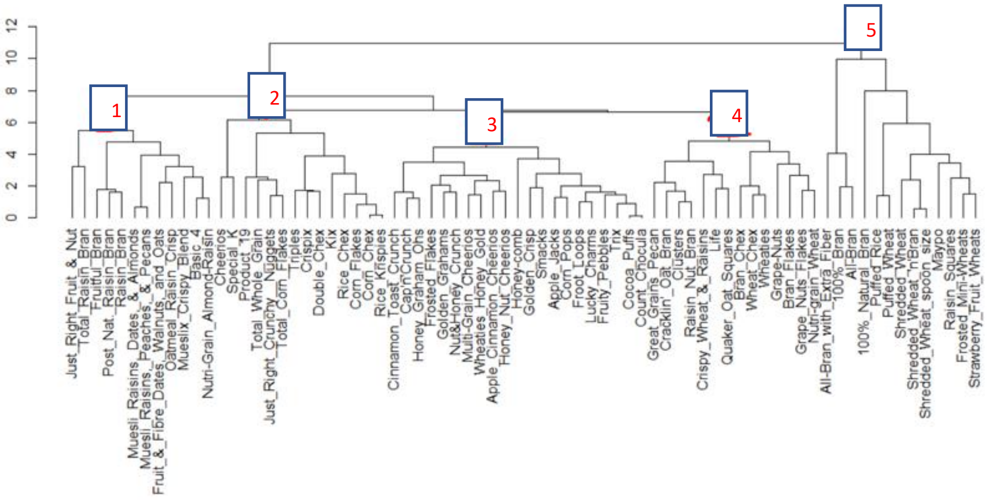
From this, we can move forward with using the single and complete clustering methods to analyze the dendograms.

Single Clustering vs Complete Clustering

The single clustering graph with a hang of -6 is displayed below:



From this plot we can see that there are 10 distinct groups. Group 1 is comprised of nut bran type of cereals, group 2 contains crispy, wheat and bran type of cereals, group 3 has a healthier version of sugary/honey cereal, group 4 has unhealthy and sugary crunchy cereals, group 5 contains sugary cereals, group 6 has wheat, bran, and honey healthy cereals, group 7 has plain cereals, group 8 has frosty and fruity cereals, group 9 possesses cereals with nuts and fruits, and lastly group 10 is 100% natural and bran which are the healthiest type of cereals. From this dendogram, we can cut the tree into 10 clusters. The results of that show that most cereals fall in cluster 4 and there are not a lot of cereals in clusters 2,3 7, 8, and 10. For this reason, we can stick to a cutree of 6 clusters which once again shows that most cereals fall into cluster 4 but there are quite a few data points in clusters 5 and 6 as well. If we were to divide the above dendogram into 6 clusters, the clusters which are labeled 3,4,5,6 would likely all fall into cluster 4.

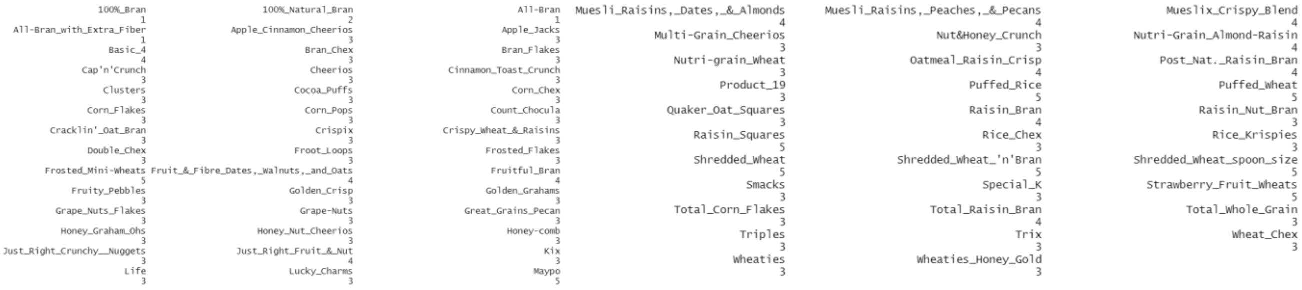


Complete Cluster Plot

From the complete cluster plot above, it paints a much clearer picture for which group go together than single cluster plot. In this plot, we can clearly see 5 overarching distinct groups: Group 1 – Brans and wheats, Group 2: Flakes, Chex and Raisins, Group 3 – Sugary cereals, Group 4 – Plain cereals, Group 5 – Fruits, nuts, and dry fruit cereals. This complete cluster graph is much easier to read than the single cluster graph as it groups cereals with maximum distance from one another apart from each other as opposed to the two closest pairings we find in the single cluster graph. The complete linkage dendogram tells a much clearer story by breaking the clusters in 5 distinct groups and one can clearly determine healthy cereal groups and unhealthy groups.

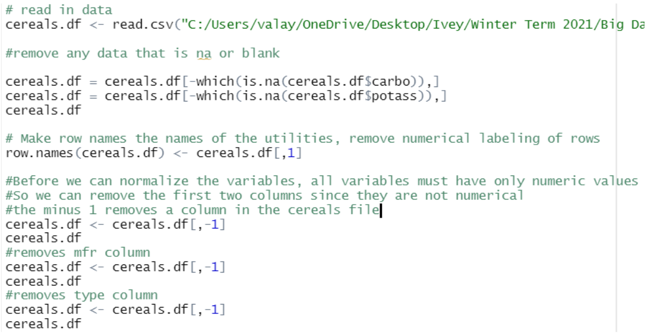
Recommendation

As noted above, the complete linkage dendogram shows a much clearer picture than the single linkage dendogram and the groups can be easily identified and sorted into 5 distinct cereal groups. As a result, the complete linkage dendogram's results are what I would recommend. We should cut the data into 5 clusters using cutree which gives the cluster membership as follows:

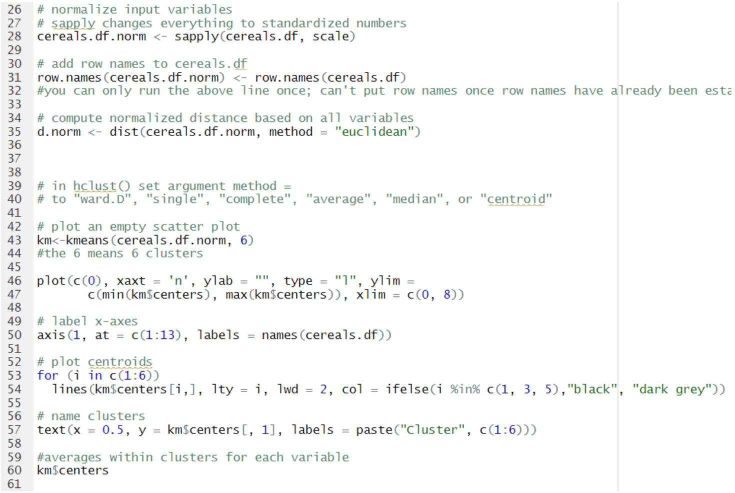


```
                  100%_Bran                        100%_Natural_Bran                       All-Bran Muesli_Raisins,_Dates,_&_Almonds Muesli_Raisins,_Peaches,_&_Pecans        Mueslix_Crispy_Blend
                          1                                        2                              1                                4                                 4                           4
   All-Bran_with_Extra_Fiber                  Apple_Cinnamon_Cheerios                    Apple_Jacks             Multi-Grain_Cheerios                  Nut&Honey_Crunch   Nutri-Grain_Almond-Raisin
                          1                                        3                              3                                3                                 3                           4
                    Basic_4                                Bran_Chex                    Bran_Flakes               Nutri-grain_Wheat              Oatmeal_Raisin_Crisp      Post_Nat._Raisin_Bran
                          4                                        3                              3                                3                                 4                           4
               Cap'n'Crunch                                 Cheerios          Cinnamon_Toast_Crunch                       Product_19                       Puffed_Rice                Puffed_Wheat
                          3                                        3                              3                                3                                 5                           5
                   Clusters                              Cocoa_Puffs                      Corn_Chex               Quaker_Oat_Squares                       Raisin_Bran             Raisin_Nut_Bran
                          3                                        3                              3                                3                                 4                           3
                Corn_Flakes                                Corn_Pops                  Count_Chocula                   Raisin_Squares                         Rice_Chex               Rice_Krispies
                          3                                        3                              3                                5                                 3                           3
         Cracklin'_Oat_Bran                                  Crispix         Crispy_Wheat_&_Raisins                   Shredded_Wheat            Shredded_Wheat_'n'Bran   Shredded_Wheat_spoon_size
                          3                                        3                              3                                5                                 5                           5
                Double_Chex                              Froot_Loops                 Frosted_Flakes                           Smacks                         Special_K     Strawberry_Fruit_Wheats
                          3                                        3                              3                                3                                 3                           5
       Frosted_Mini-Wheats Fruit_&_Fibre_Dates,_Walnuts,_and_Oats                  Fruitful_Bran                Total_Corn_Flakes                 Total_Raisin_Bran           Total_Whole_Grain
                          5                                        4                              4                                3                                 4                           3
             Fruity_Pebbles                             Golden_Crisp                 Golden_Grahams                          Triples                              Trix                 Wheat_Chex
                          3                                        3                              3                                3                                 3                           3
          Grape_Nuts_Flakes                               Grape-Nuts             Great_Grains_Pecan                         Wheaties               Wheaties_Honey_Gold
                          3                                        3                              3                                3                                 3
          Honey_Graham_Ohs                       Honey_Nut_Cheerios                     Honey-comb
                          3                                        3                              3
Just_Right_Crunchy__Nuggets                   Just_Right_Fruit_&_Nut                            Kix
                          3                                        4                              3
                       Life                             Lucky_Charms                          Maypo
                          3                                        3                              5
```

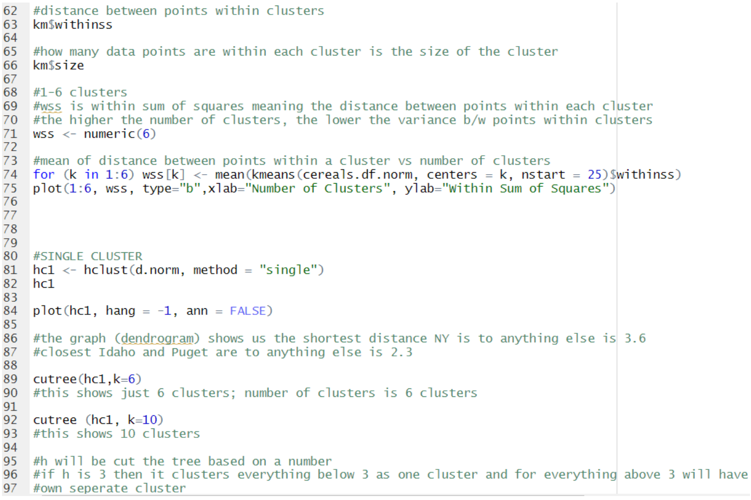
Code:



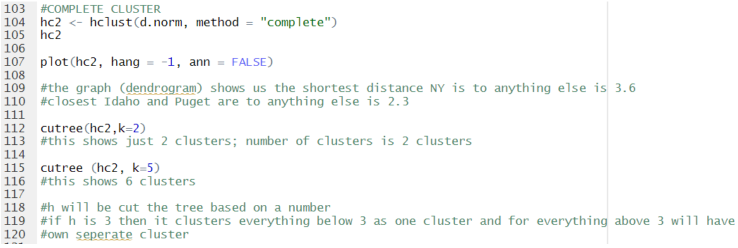
```
# read in data
cereals.df <- read.csv("C:/Users/valay/OneDrive/Desktop/Ivey/Winter Term 2021/Big Da

#remove any data that is na or blank

cereals.df = cereals.df[-which(is.na(cereals.df$carbo)),]
cereals.df = cereals.df[-which(is.na(cereals.df$potass)),]
cereals.df

# Make row names the names of the utilities, remove numerical labeling of rows
row.names(cereals.df) <- cereals.df[,1]

#Before we can normalize the variables, all variables must have only numeric values
#So we can remove the first two columns since they are not numerical
#the minus 1 removes a column in the cereals file
cereals.df <- cereals.df[,-1]
cereals.df
#removes mfr column
cereals.df <- cereals.df[,-1]
cereals.df
#removes type column
cereals.df <- cereals.df[,-1]
cereals.df
```



```
# normalize input variables
# sapply changes everything to standardized numbers
cereals.df.norm <- sapply(cereals.df, scale)

# add row names to cereals.df
row.names(cereals.df.norm) <- row.names(cereals.df)
#you can only run the above line once; can't put row names once row names have already been esta

# compute normalized distance based on all variables
d.norm <- dist(cereals.df.norm, method = "euclidean")



# in hclust() set argument method =
# to "ward.D", "single", "complete", "average", "median", or "centroid"

# plot an empty scatter plot
km<-kmeans(cereals.df.norm, 6)
#the 6 means 6 clusters

plot(c(0), xaxt = 'n', ylab = "", type = "l", ylim =
       c(min(km$centers), max(km$centers)), xlim = c(0, 8))

# label x-axes
axis(1, at = c(1:13), labels = names(cereals.df))

# plot centroids
for (i in c(1:6))
  lines(km$centers[i,], lty = i, lwd = 2, col = ifelse(i %in% c(1, 3, 5),"black", "dark grey"))

# name clusters
text(x = 0.5, y = km$centers[, 1], labels = paste("Cluster", c(1:6)))

#averages within clusters for each variable
km$centers

```



```
#distance between points within clusters
km$withinss

#how many data points are within each cluster is the size of the cluster
km$size

#1-6 clusters
#wss is within sum of squares meaning the distance between points within each cluster
#the higher the number of clusters, the lower the variance b/w points within clusters
wss <- numeric(6)

#mean of distance between points within a cluster vs number of clusters
for (k in 1:6) wss[k] <- mean(kmeans(cereals.df.norm, centers = k, nstart = 25)$withinss)
plot(1:6, wss, type="b",xlab="Number of Clusters", ylab="Within Sum of Squares")




#SINGLE CLUSTER
hc1 <- hclust(d.norm, method = "single")
hc1

plot(hc1, hang = -1, ann = FALSE)

#the graph (dendrogram) shows us the shortest distance NY is to anything else is 3.6
#closest Idaho and Puget are to anything else is 2.3

cutree(hc1,k=6)
#this shows just 6 clusters; number of clusters is 6 clusters

cutree (hc1, k=10)
#this shows 10 clusters

#h will be cut the tree based on a number
#if h is 3 then it clusters everything below 3 as one cluster and for everything above 3 will have
#own seperate cluster
```



```
#COMPLETE CLUSTER
hc2 <- hclust(d.norm, method = "complete")
hc2

plot(hc2, hang = -1, ann = FALSE)

#the graph (dendrogram) shows us the shortest distance NY is to anything else is 3.6
#closest Idaho and Puget are to anything else is 2.3

cutree(hc2,k=2)
#this shows just 2 clusters; number of clusters is 2 clusters

cutree (hc2, k=5)
#this shows 6 clusters

#h will be cut the tree based on a number
#if h is 3 then it clusters everything below 3 as one cluster and for everything above 3 will have
#own seperate cluster
```