Assignment 5

Yousef

2023-04-12

#Load the data set and convert it into a data frame  
Cereals <- read\_csv("/Users/binsalim/Downloads/Cereals.csv")

## Rows: 77 Columns: 16  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (3): name, mfr, type  
## dbl (13): calories, protein, fat, sodium, fiber, carbo, sugars, potass, vita...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

df <- Cereals  
df <- as.data.frame(df)  
df <- na.omit(df) # Remove NA (missing) values  
Cereals\_clean <- na.omit(Cereals)  
head(df) # Examine the dataset

## name mfr type calories protein fat sodium fiber carbo  
## 1 100%\_Bran N C 70 4 1 130 10.0 5.0  
## 2 100%\_Natural\_Bran Q C 120 3 5 15 2.0 8.0  
## 3 All-Bran K C 70 4 1 260 9.0 7.0  
## 4 All-Bran\_with\_Extra\_Fiber K C 50 4 0 140 14.0 8.0  
## 6 Apple\_Cinnamon\_Cheerios G C 110 2 2 180 1.5 10.5  
## 7 Apple\_Jacks K C 110 2 0 125 1.0 11.0  
## sugars potass vitamins shelf weight cups rating  
## 1 6 280 25 3 1 0.33 68.40297  
## 2 8 135 0 3 1 1.00 33.98368  
## 3 5 320 25 3 1 0.33 59.42551  
## 4 0 330 25 3 1 0.50 93.70491  
## 6 10 70 25 1 1 0.75 29.50954  
## 7 14 30 25 2 1 1.00 33.17409

#Clean the dataframe and examine it  
df <- na.omit(df) # Remove NA (missing) values  
Cereals\_clean <- na.omit(Cereals)  
head(df) # Examine the dataset

## name mfr type calories protein fat sodium fiber carbo  
## 1 100%\_Bran N C 70 4 1 130 10.0 5.0  
## 2 100%\_Natural\_Bran Q C 120 3 5 15 2.0 8.0  
## 3 All-Bran K C 70 4 1 260 9.0 7.0  
## 4 All-Bran\_with\_Extra\_Fiber K C 50 4 0 140 14.0 8.0  
## 6 Apple\_Cinnamon\_Cheerios G C 110 2 2 180 1.5 10.5  
## 7 Apple\_Jacks K C 110 2 0 125 1.0 11.0  
## sugars potass vitamins shelf weight cups rating  
## 1 6 280 25 3 1 0.33 68.40297  
## 2 8 135 0 3 1 1.00 33.98368  
## 3 5 320 25 3 1 0.33 59.42551  
## 4 0 330 25 3 1 0.50 93.70491  
## 6 10 70 25 1 1 0.75 29.50954  
## 7 14 30 25 2 1 1.00 33.17409

#Normalize the numrical columns  
df <-- df[,4:16]  
df <- scale(df)

#Reassign the nonnumrical column to the dataframe after normalization  
Normalized\_df\_Data <- cbind(df, name = Cereals\_clean$name)  
Normalized\_df\_Data <- cbind(df, mfr = Cereals\_clean$mfr)  
Normalized\_df\_Data <- cbind(df, type = Cereals\_clean$type)  
head(df) #re-examine the scaled data

## calories protein fat sodium fiber carbo sugars  
## 1 1.8659155 -1.3817478 0.0000000 0.3910227 -3.22866747 2.5001396 0.2542051  
## 2 -0.6537514 -0.4522084 -3.9728810 1.7804186 0.07249167 1.7292632 -0.2046041  
## 3 1.8659155 -1.3817478 0.0000000 -1.1795987 -2.81602258 1.9862220 0.4836096  
## 4 2.8737823 -1.3817478 0.9932203 0.2702057 -4.87924705 1.7292632 1.6306324  
## 6 -0.1498180 0.4773310 -0.9932203 -0.2130625 0.27881412 1.0868662 -0.6634132  
## 7 -0.1498180 0.4773310 0.9932203 0.4514312 0.48513656 0.9583868 -1.5810314  
## potass vitamins shelf weight cups rating  
## 1 -2.5605229 0.1818422 -0.9419715 0.2008324 2.0856582 -1.8549038  
## 2 -0.5147738 1.3032024 -0.9419715 0.2008324 -0.7567534 0.5977113  
## 3 -3.1248675 0.1818422 -0.9419715 0.2008324 2.0856582 -1.2151965  
## 4 -3.2659536 0.1818422 -0.9419715 0.2008324 1.3644493 -3.6578436  
## 6 0.4022862 0.1818422 1.4616799 0.2008324 0.3038480 0.9165248  
## 7 0.9666308 0.1818422 0.2598542 0.2008324 -0.7567534 0.6553998

#Compute with agnes and with different linkage methods  
hc\_single <- agnes(df, method = "single")  
hc\_complete <- agnes(df, method = "complete")  
hc\_average <- agnes(df, method = "average")  
hc\_ward.D <- agnes(df, method = "ward")  
#Compare Agglomerative coefficients  
print(hc\_single$ac)

## [1] 0.6067859

print(hc\_complete$ac)

## [1] 0.8353712

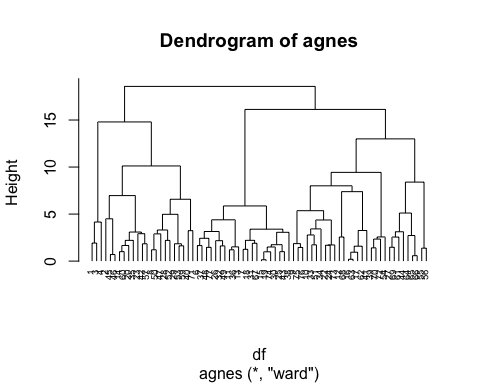
print(hc\_average$ac)

## [1] 0.7766075

print(hc\_ward.D$ac) # is the best method as it classify 0.9046042 into their actual cluster and the closer to 1 is best.

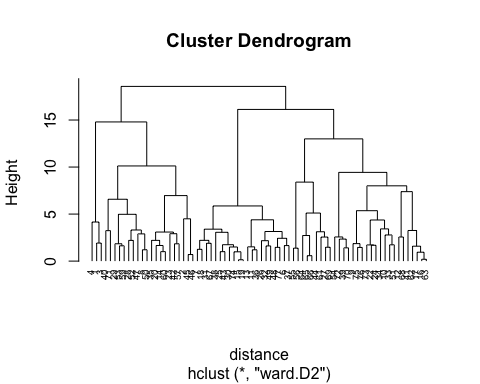
## [1] 0.9046042

#Plot with the best method in this case ward is the best  
pltree(hc\_ward.D, cex = 0.6, hang = -1, main = "Dendrogram of agnes")



#Calculate the euclidean to use in the clustering using ward since it is the best method  
distance <- dist(df, method = "euclidean")  
# Hierarchical clustering using ward method  
hc1 <- hclust(distance, method = "ward.D2" )

#Now plot using the euclidean distance and ward method  
# Plot the obtained dendrogram  
plot(hc1, cex = 0.6, hang = -1)



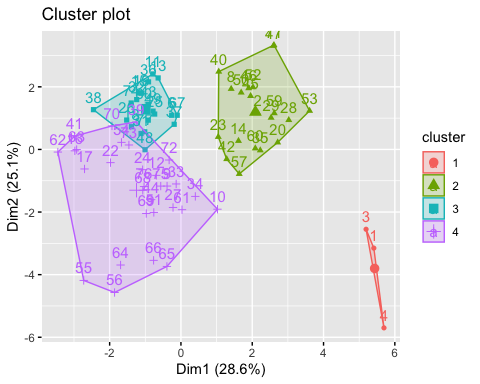
How many clusters would you choose? 4 clusters

#Cut the tree into four group and show how many member is in each group   
grp <- cutree(hc1, k = 4)  
# Number of members in each cluster  
table(grp)

## grp  
## 1 2 3 4   
## 3 20 21 30

#now bind the group membership to each record   
df <- as.data.frame(cbind(df,grp))

#visulaize the cereals and their cluster membership  
fviz\_cluster(list(data = df, cluster = grp))



#Now using the numrical and the group membership show each clusters members  
Newdf = Cereals\_clean[,4:16]  
clust <- cbind(Newdf, grp)  
clust[clust$grp==1,]

## calories protein fat sodium fiber carbo sugars potass vitamins shelf weight  
## 1 70 4 1 130 10 5 6 280 25 3 1  
## 3 70 4 1 260 9 7 5 320 25 3 1  
## 4 50 4 0 140 14 8 0 330 25 3 1  
## cups rating grp  
## 1 0.33 68.40297 1  
## 3 0.33 59.42551 1  
## 4 0.50 93.70491 1

clust[clust$grp==2,]

## calories protein fat sodium fiber carbo sugars potass vitamins shelf weight  
## 2 120 3 5 15 2.0 8.0 8 135 0 3 1.00  
## 8 130 3 2 210 2.0 18.0 8 100 25 3 1.33  
## 14 110 3 2 140 2.0 13.0 7 105 25 3 1.00  
## 20 110 3 3 140 4.0 10.0 7 160 25 3 1.00  
## 23 100 2 1 140 2.0 11.0 10 120 25 3 1.00  
## 28 120 3 2 160 5.0 12.0 10 200 25 3 1.25  
## 29 120 3 0 240 5.0 14.0 12 190 25 3 1.33  
## 35 120 3 3 75 3.0 13.0 4 100 25 3 1.00  
## 40 140 3 1 170 2.0 20.0 9 95 100 3 1.30  
## 42 100 4 2 150 2.0 12.0 6 95 25 2 1.00  
## 45 150 4 3 95 3.0 16.0 11 170 25 3 1.00  
## 46 150 4 3 150 3.0 16.0 11 170 25 3 1.00  
## 47 160 3 2 150 3.0 17.0 13 160 25 3 1.50  
## 50 140 3 2 220 3.0 21.0 7 130 25 3 1.33  
## 52 130 3 2 170 1.5 13.5 10 120 25 3 1.25  
## 53 120 3 1 200 6.0 11.0 14 260 25 3 1.33  
## 57 100 4 1 135 2.0 14.0 6 110 25 3 1.00  
## 59 120 3 1 210 5.0 14.0 12 240 25 2 1.33  
## 60 100 3 2 140 2.5 10.5 8 140 25 3 1.00  
## 71 140 3 1 190 4.0 15.0 14 230 100 3 1.50  
## cups rating grp  
## 2 1.00 33.98368 2  
## 8 0.75 37.03856 2  
## 14 0.50 40.40021 2  
## 20 0.50 40.44877 2  
## 23 0.75 36.17620 2  
## 28 0.67 40.91705 2  
## 29 0.67 41.01549 2  
## 35 0.33 45.81172 2  
## 40 0.75 36.47151 2  
## 42 0.67 45.32807 2  
## 45 1.00 37.13686 2  
## 46 1.00 34.13976 2  
## 47 0.67 30.31335 2  
## 50 0.67 40.69232 2  
## 52 0.50 30.45084 2  
## 53 0.67 37.84059 2  
## 57 0.50 49.51187 2  
## 59 0.75 39.25920 2  
## 60 0.50 39.70340 2  
## 71 1.00 28.59278 2

clust[clust$grp==3,]

## calories protein fat sodium fiber carbo sugars potass vitamins shelf weight  
## 6 110 2 2 180 1.5 10.5 10 70 25 1 1  
## 7 110 2 0 125 1.0 11.0 14 30 25 2 1  
## 11 120 1 2 220 0.0 12.0 12 35 25 2 1  
## 13 120 1 3 210 0.0 13.0 9 45 25 2 1  
## 15 110 1 1 180 0.0 12.0 13 55 25 2 1  
## 18 110 1 0 90 1.0 13.0 12 20 25 2 1  
## 19 110 1 1 180 0.0 12.0 13 65 25 2 1  
## 25 110 2 1 125 1.0 11.0 13 30 25 2 1  
## 26 110 1 0 200 1.0 14.0 11 25 25 1 1  
## 30 110 1 1 135 0.0 13.0 12 25 25 2 1  
## 31 100 2 0 45 0.0 11.0 15 40 25 1 1  
## 32 110 1 1 280 0.0 15.0 9 45 25 2 1  
## 36 120 1 2 220 1.0 12.0 11 45 25 2 1  
## 37 110 3 1 250 1.5 11.5 10 90 25 1 1  
## 38 110 1 0 180 0.0 14.0 11 35 25 1 1  
## 43 110 2 1 180 0.0 12.0 12 55 25 2 1  
## 48 100 2 1 220 2.0 15.0 6 90 25 1 1  
## 49 120 2 1 190 0.0 15.0 9 40 25 2 1  
## 67 110 2 1 70 1.0 9.0 15 40 25 2 1  
## 74 110 1 1 140 0.0 13.0 12 25 25 2 1  
## 77 110 2 1 200 1.0 16.0 8 60 25 1 1  
## cups rating grp  
## 6 0.75 29.50954 3  
## 7 1.00 33.17409 3  
## 11 0.75 18.04285 3  
## 13 0.75 19.82357 3  
## 15 1.00 22.73645 3  
## 18 1.00 35.78279 3  
## 19 1.00 22.39651 3  
## 25 1.00 32.20758 3  
## 26 0.75 31.43597 3  
## 30 0.75 28.02576 3  
## 31 0.88 35.25244 3  
## 32 0.75 23.80404 3  
## 36 1.00 21.87129 3  
## 37 0.75 31.07222 3  
## 38 1.33 28.74241 3  
## 43 1.00 26.73451 3  
## 48 1.00 40.10596 3  
## 49 0.67 29.92429 3  
## 67 0.75 31.23005 3  
## 74 1.00 27.75330 3  
## 77 0.75 36.18756 3

clust[clust$grp==4,]

## calories protein fat sodium fiber carbo sugars potass vitamins shelf weight  
## 9 90 2 1 200 4 15 6 125 25 1 1.00  
## 10 90 3 0 210 5 13 5 190 25 3 1.00  
## 12 110 6 2 290 2 17 1 105 25 1 1.00  
## 16 110 2 0 280 0 22 3 25 25 1 1.00  
## 17 100 2 0 290 1 21 2 35 25 1 1.00  
## 22 110 2 0 220 1 21 3 30 25 3 1.00  
## 24 100 2 0 190 1 18 5 80 25 3 1.00  
## 27 100 3 0 0 3 14 7 100 25 2 1.00  
## 33 100 3 1 140 3 15 5 85 25 3 1.00  
## 34 110 3 0 170 3 17 3 90 25 3 1.00  
## 39 110 2 1 170 1 17 6 60 100 3 1.00  
## 41 110 2 1 260 0 21 3 40 25 2 1.00  
## 44 100 4 1 0 0 16 3 95 25 2 1.00  
## 51 90 3 0 170 3 18 2 90 25 3 1.00  
## 54 100 3 0 320 1 20 3 45 100 3 1.00  
## 55 50 1 0 0 0 13 0 15 0 3 0.50  
## 56 50 2 0 0 1 10 0 50 0 3 0.50  
## 61 90 2 0 0 2 15 6 110 25 3 1.00  
## 62 110 1 0 240 0 23 2 30 25 1 1.00  
## 63 110 2 0 290 0 22 3 35 25 1 1.00  
## 64 80 2 0 0 3 16 0 95 0 1 0.83  
## 65 90 3 0 0 4 19 0 140 0 1 1.00  
## 66 90 3 0 0 3 20 0 120 0 1 1.00  
## 68 110 6 0 230 1 16 3 55 25 1 1.00  
## 69 90 2 0 15 3 15 5 90 25 2 1.00  
## 70 110 2 1 200 0 21 3 35 100 3 1.00  
## 72 100 3 1 200 3 16 3 110 100 3 1.00  
## 73 110 2 1 250 0 21 3 60 25 3 1.00  
## 75 100 3 1 230 3 17 3 115 25 1 1.00  
## 76 100 3 1 200 3 17 3 110 25 1 1.00  
## cups rating grp  
## 9 0.67 49.12025 4  
## 10 0.67 53.31381 4  
## 12 1.25 50.76500 4  
## 16 1.00 41.44502 4  
## 17 1.00 45.86332 4  
## 22 1.00 46.89564 4  
## 24 0.75 44.33086 4  
## 27 0.80 58.34514 4  
## 33 0.88 52.07690 4  
## 34 0.25 53.37101 4  
## 39 1.00 36.52368 4  
## 41 1.50 39.24111 4  
## 44 1.00 54.85092 4  
## 51 1.00 59.64284 4  
## 54 1.00 41.50354 4  
## 55 1.00 60.75611 4  
## 56 1.00 63.00565 4  
## 61 0.50 55.33314 4  
## 62 1.13 41.99893 4  
## 63 1.00 40.56016 4  
## 64 1.00 68.23588 4  
## 65 0.67 74.47295 4  
## 66 0.67 72.80179 4  
## 68 1.00 53.13132 4  
## 69 1.00 59.36399 4  
## 70 1.00 38.83975 4  
## 72 1.00 46.65884 4  
## 73 0.75 39.10617 4  
## 75 0.67 49.78744 4  
## 76 1.00 51.59219 4

#now based on the rating columns show the mean rating of each cluster to determine which cluster have the highes rating  
mean(clust[clust$grp==1,"rating"])

## [1] 73.84446

mean(clust[clust$grp==2,"rating"])

## [1] 38.26161

mean(clust[clust$grp==3,"rating"])

## [1] 28.84825

mean(clust[clust$grp==4,"rating"])

## [1] 51.43111

from the rating we could tell that cluster one has the highest rating therefore it is the cluster with the best breakfast cereals.

Should the data be normalized? you need to first define what “healthy” means. Depending on the criteria, you may or may not need to normalize the data. For example, if you define “healthy” as cereals that have high fiber content and low sugar content, then you should normalize the data