Assignment\_5

Kruthi Tatavarthy

4/17/2022

#loading all the required libraries  
library(factoextra)

## Warning: package 'factoextra' was built under R version 4.1.3

## Loading required package: ggplot2

## Warning in register(): Can't find generic `scale\_type` in package ggplot2 to  
## register S3 method.

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(dendextend)

## Warning: package 'dendextend' was built under R version 4.1.3

##   
## ---------------------  
## Welcome to dendextend version 1.15.2  
## Type citation('dendextend') for how to cite the package.  
##   
## Type browseVignettes(package = 'dendextend') for the package vignette.  
## The github page is: https://github.com/talgalili/dendextend/  
##   
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues  
## You may ask questions at stackoverflow, use the r and dendextend tags:   
## https://stackoverflow.com/questions/tagged/dendextend  
##   
## To suppress this message use: suppressPackageStartupMessages(library(dendextend))  
## ---------------------

##   
## Attaching package: 'dendextend'

## The following object is masked from 'package:stats':  
##   
## cutree

library(cluster)  
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.1.3

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble 3.1.6 v dplyr 1.0.8  
## v tidyr 1.1.4 v stringr 1.4.0  
## v readr 2.1.2 v forcats 0.5.1  
## v purrr 0.3.4

## Warning: package 'forcats' was built under R version 4.1.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(knitr)

#Import Data

cereals<- read.csv('C:/Users/Kruthi Tatavarthy/Desktop/Machine-Learning/Assignment 5/Cereals.csv')  
numericaldata = data.frame(cereals[,4:16])

#1. Apply hierarchical clustering to the data using Euclidean distance to the normalized measurements. Use Agnes to compare the clustering from single linkage, complete linkage, average linkage, and Ward. Choose the best method.

#omitting all the missing values present in the data

OmitMissing = na.omit(numericaldata)

#normalizing and scaling the data

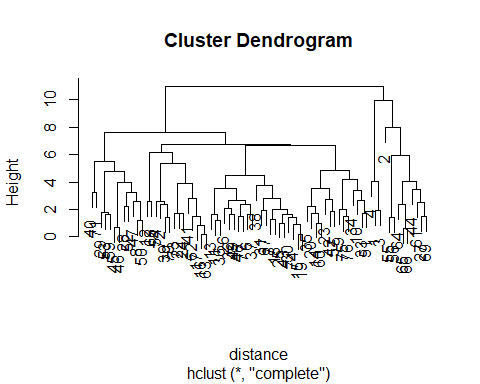
Normalise = scale(OmitMissing)

#measuring the distance using the euclidian distance and computing the dissimilarity matrix

distance = dist(Normalise, method = "euclidian")

#performing hierarchial clustering using complete linkage and representing in plot

clustering\_heirarchial = hclust(distance,method = "complete")  
plot(clustering\_heirarchial)



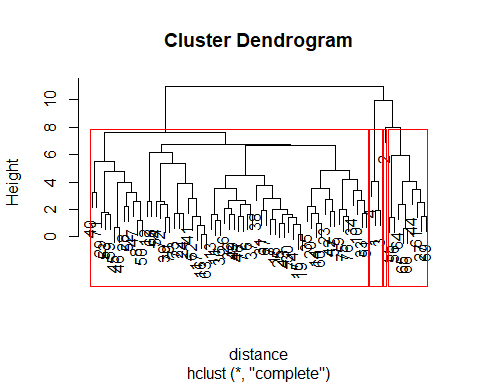
#rounding off the decimals

round(clustering\_heirarchial$height, 3)

## [1] 0.143 0.196 0.575 0.698 0.828 0.904 1.003 1.004 1.201 1.203  
## [11] 1.254 1.378 1.408 1.421 1.454 1.463 1.474 1.517 1.608 1.611  
## [21] 1.616 1.625 1.650 1.687 1.692 1.720 1.730 1.795 1.839 1.897  
## [31] 1.919 1.982 2.015 2.046 2.203 2.224 2.339 2.381 2.394 2.522  
## [41] 2.563 2.574 2.579 2.668 2.682 2.734 2.776 2.787 3.229 3.236  
## [51] 3.385 3.451 3.510 3.535 3.717 3.866 3.957 4.005 4.031 4.168  
## [61] 4.456 4.779 4.839 5.342 5.488 5.920 6.169 6.669 6.731 7.650  
## [71] 7.964 9.979 10.984

#determining the optimla clusters and highlighting with colours

plot(clustering\_heirarchial)  
rect.hclust(clustering\_heirarchial,k = 4, border = "red")



#performing clustering using AGNES

singleCH = agnes(Normalise, method = "single")  
completeCH = agnes(Normalise, method = "complete")  
averageCH = agnes(Normalise, method = "average")  
wardCH = agnes(Normalise, method = "ward")

#comparing the agglomerative cosfficients of single , complete, average, ward

print(singleCH$ac)

## [1] 0.6067859

print(completeCH$ac)

## [1] 0.8353712

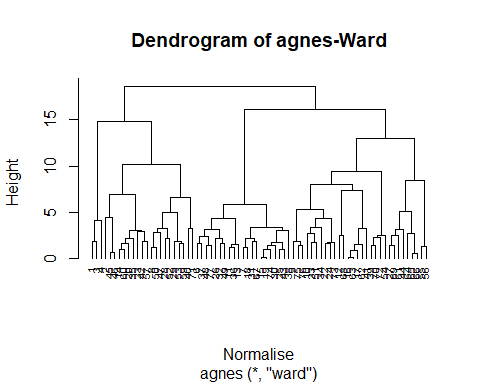
print(averageCH$ac)

## [1] 0.7766075

print(wardCH$ac)

## [1] 0.9046042

pltree(wardCH, cex = 0.6, hang = -1, main = "Dendrogram of agnes-Ward")

 #according to the above values, wards method is the best with the value of 0.904.plotting ward using agnes and the dendogram

#using the ward method for hierarchial clustering

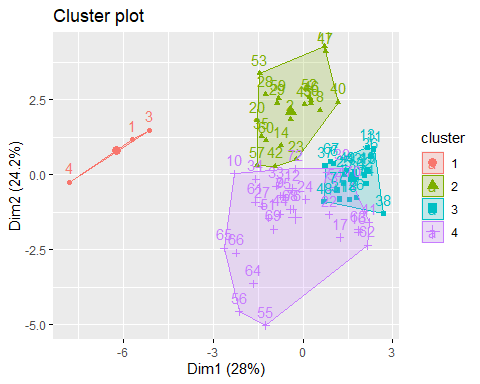
HCuster1 <- hclust(distance, method = "ward.D2" )  
subgrp <- cutree(HCuster1, k = 4)  
table(subgrp)

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

cereals <- as.data.frame(cbind(Normalise,subgrp))

#visualising the results on scatterplot

fviz\_cluster(list(data = Normalise, cluster = subgrp))



#selecting the best breakfast cereal cluster with high protein, fibre and low in sugar and sodium. #choosing the healthy cereal cluster

Newcereals = numericaldata  
Newcereals\_omit = na.omit(Newcereals)  
Clust = cbind(Newcereals\_omit, subgrp)  
Clust[Clust$subgrp==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 subgrp  
## 1 0.33 68.40297 1  
## 3 0.33 59.42551 1  
## 4 0.50 93.70491 1

Clust[Clust$subgrp==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 subgrp  
## 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$subgrp==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 subgrp  
## 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$subgrp==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 subgrp  
## 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

#here we calculate the mean rating in order determine the healthy cluster cereals

mean(Clust[Clust$subgrp==1,"rating"])

## [1] 73.84446

mean(Clust[Clust$subgrp==2,"rating"])

## [1] 38.26161

mean(Clust[Clust$subgrp==3,"rating"])

## [1] 28.84825

mean(Clust[Clust$subgrp==4,"rating"])

## [1] 51.43111

#From the above results it is clearly evident that mean rating is highest for subgroup 1. #so, it is recommended to choose subgrp 1 as the healthy diet cluster.