HC assignment 05

Sai Supriya

2022-11-28

## Hierarchial Clustering

library(readr)

## Warning: package 'readr' was built under R version 4.1.2

Cereals <- read\_csv("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.

str(Cereals)

## spec\_tbl\_df [77 × 16] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ name : chr [1:77] "100%\_Bran" "100%\_Natural\_Bran" "All-Bran" "All-Bran\_with\_Extra\_Fiber" ...  
## $ mfr : chr [1:77] "N" "Q" "K" "K" ...  
## $ type : chr [1:77] "C" "C" "C" "C" ...  
## $ calories: num [1:77] 70 120 70 50 110 110 110 130 90 90 ...  
## $ protein : num [1:77] 4 3 4 4 2 2 2 3 2 3 ...  
## $ fat : num [1:77] 1 5 1 0 2 2 0 2 1 0 ...  
## $ sodium : num [1:77] 130 15 260 140 200 180 125 210 200 210 ...  
## $ fiber : num [1:77] 10 2 9 14 1 1.5 1 2 4 5 ...  
## $ carbo : num [1:77] 5 8 7 8 14 10.5 11 18 15 13 ...  
## $ sugars : num [1:77] 6 8 5 0 8 10 14 8 6 5 ...  
## $ potass : num [1:77] 280 135 320 330 NA 70 30 100 125 190 ...  
## $ vitamins: num [1:77] 25 0 25 25 25 25 25 25 25 25 ...  
## $ shelf : num [1:77] 3 3 3 3 3 1 2 3 1 3 ...  
## $ weight : num [1:77] 1 1 1 1 1 1 1 1.33 1 1 ...  
## $ cups : num [1:77] 0.33 1 0.33 0.5 0.75 0.75 1 0.75 0.67 0.67 ...  
## $ rating : num [1:77] 68.4 34 59.4 93.7 34.4 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. name = col\_character(),  
## .. mfr = col\_character(),  
## .. type = col\_character(),  
## .. calories = col\_double(),  
## .. protein = col\_double(),  
## .. fat = col\_double(),  
## .. sodium = col\_double(),  
## .. fiber = col\_double(),  
## .. carbo = col\_double(),  
## .. sugars = col\_double(),  
## .. potass = col\_double(),  
## .. vitamins = col\_double(),  
## .. shelf = col\_double(),  
## .. weight = col\_double(),  
## .. cups = col\_double(),  
## .. rating = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

## Installing & calling required packages

library(tidyverse)## Data manipulation

## Warning: package 'tidyverse' was built under R version 4.1.2

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ dplyr 1.0.10  
## ✔ tibble 3.1.8 ✔ stringr 1.4.0   
## ✔ tidyr 1.2.1 ✔ forcats 0.5.2   
## ✔ purrr 0.3.4

## Warning: package 'ggplot2' was built under R version 4.1.2

## Warning: package 'tibble' was built under R version 4.1.2

## Warning: package 'tidyr' was built under R version 4.1.2

## Warning: package 'dplyr' was built under R version 4.1.2

## Warning: package 'forcats' was built under R version 4.1.2

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(cluster) ## Clustering Algorithms

## Warning: package 'cluster' was built under R version 4.1.2

library(factoextra) ## Clustering Visualization

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

library(dendextend) ## for comparing 2 dendograms

## Warning: package 'dendextend' was built under R version 4.1.2

##   
## ---------------------  
## Welcome to dendextend version 1.16.0  
## 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

Cereals1<- Cereals[, unlist(lapply(Cereals, is.numeric))]  
Cereals1

## # A tibble: 77 × 13  
## calories protein fat sodium fiber carbo sugars potass vitamins shelf weight  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 70 4 1 130 10 5 6 280 25 3 1   
## 2 120 3 5 15 2 8 8 135 0 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   
## 5 110 2 2 200 1 14 8 NA 25 3 1   
## 6 110 2 2 180 1.5 10.5 10 70 25 1 1   
## 7 110 2 0 125 1 11 14 30 25 2 1   
## 8 130 3 2 210 2 18 8 100 25 3 1.33  
## 9 90 2 1 200 4 15 6 125 25 1 1   
## 10 90 3 0 210 5 13 5 190 25 3 1   
## # … with 67 more rows, and 2 more variables: cups <dbl>, rating <dbl>

Removing any missing values

Cereals2 <- na.omit(Cereals1)

Scaling the data & Normalizing data

Cereals3<- Cereals2[,1:13]  
Cereals3<- scale(Cereals2, center = TRUE, scale = TRUE)  
head(Cereals3)

## calories protein fat sodium fiber carbo  
## [1,] -1.8659155 1.3817478 0.0000000 -0.3910227 3.22866747 -2.5001396  
## [2,] 0.6537514 0.4522084 3.9728810 -1.7804186 -0.07249167 -1.7292632  
## [3,] -1.8659155 1.3817478 0.0000000 1.1795987 2.81602258 -1.9862220  
## [4,] -2.8737823 1.3817478 -0.9932203 -0.2702057 4.87924705 -1.7292632  
## [5,] 0.1498180 -0.4773310 0.9932203 0.2130625 -0.27881412 -1.0868662  
## [6,] 0.1498180 -0.4773310 -0.9932203 -0.4514312 -0.48513656 -0.9583868  
## sugars potass vitamins shelf weight cups  
## [1,] -0.2542051 2.5605229 -0.1818422 0.9419715 -0.2008324 -2.0856582  
## [2,] 0.2046041 0.5147738 -1.3032024 0.9419715 -0.2008324 0.7567534  
## [3,] -0.4836096 3.1248675 -0.1818422 0.9419715 -0.2008324 -2.0856582  
## [4,] -1.6306324 3.2659536 -0.1818422 0.9419715 -0.2008324 -1.3644493  
## [5,] 0.6634132 -0.4022862 -0.1818422 -1.4616799 -0.2008324 -0.3038480  
## [6,] 1.5810314 -0.9666308 -0.1818422 -0.2598542 -0.2008324 0.7567534  
## rating  
## [1,] 1.8549038  
## [2,] -0.5977113  
## [3,] 1.2151965  
## [4,] 3.6578436  
## [5,] -0.9165248  
## [6,] -0.6553998

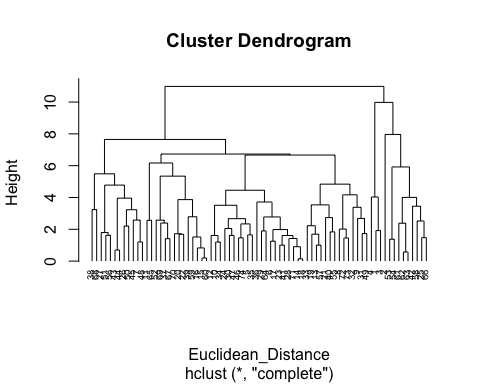
##Task-1.Apply hierarchical clustering to the data using Euclidean distance to the normalized measurements. #Use Agnes to compare the clustering from single linkage, complete #linkage, average linkage, and Ward. Choose the best method

##Computing the distance matrix Dissimilarity matrix

## Euclidean distance to the normalized measurements.   
  
Euclidean\_Distance<- dist(Cereals3, method = "euclidean")

###Hierarchial clustering using complete linkage & plotting the obtained dendogram

HC1<- hclust(Euclidean\_Distance, method = "complete")  
  
plot(HC1, cex=0.6, hang= -1)



round(HC1$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

## Using agnes and other to find best option

library(cluster)  
## Use Agnes to compare the clustering from single linkage, complete linkage, average linkage, and Ward.  
  
  
HC\_single <- agnes(Cereals3, method="single")  
HC\_complete <- agnes(Cereals3, method = "complete")  
HC\_average <- agnes(Cereals3, method = "average")  
HC\_ward<- agnes(Cereals3, method = "ward")  
  
print(HC\_single$ac)

## [1] 0.6067859

print(HC\_complete$ac)

## [1] 0.8353712

print(HC\_average$ac)

## [1] 0.7766075

print(HC\_ward$ac)

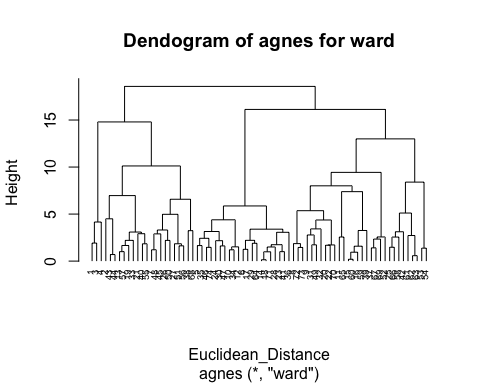
## [1] 0.9046042

It can be infered that the ac ( agglomerate coefficient for ward method is greater than the other two methods.)

## Using the dendrogram to find optimal no. of clusters

##Cluster partition A ● Use the cluster centroids from A to assign each record in partition B (each record is assigned to the cluster with the closest centroid). ● Assess how consistent the cluster assignments are compared to the assignments based on all the data

HC\_ward<- agnes(Euclidean\_Distance, method = "ward")  
pltree(HC\_ward, cex= 0.6, hang= -1 , main= "Dendogram of agnes for ward")

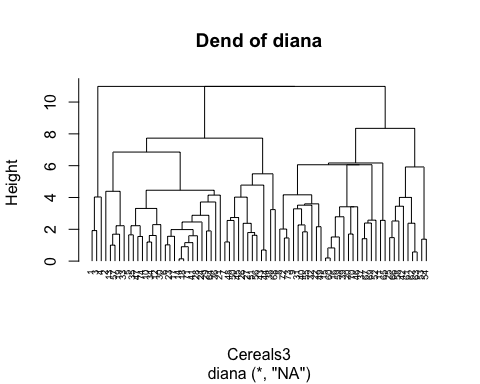


#### Divisive clustering

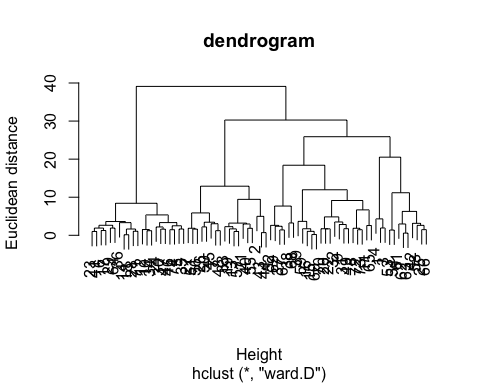
hc\_4<- diana(Cereals3)  
hc\_4$dc

## [1] 0.8302148

pltree(hc\_4, cex=0.6, hang=-1, main="Dend of diana")

 ##“Srength of the endogram is observed to be 83%”

dendrogram= hclust(dist(Cereals3, method = 'euclidean'), method = 'ward.D')  
plot(dendrogram, main= paste('dendrogram'),  
xlab= 'Height',  
ylab= 'Euclidean distance')

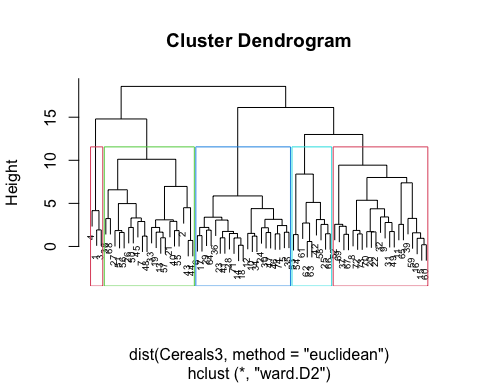
 The largest vertical distance is observed to have 5 clusters at the same level.

##Fitting hierarchial clustering model

hc5= hclust(dist(Cereals3, method = 'euclidean'), method = 'ward.D2')  
  
  
## Cut tree into 5 groups  
  
sub\_grp = cutree(hc5, k=5)  
  
##Number of in each clusters  
table(sub\_grp)

## sub\_grp  
## 1 2 3 4 5   
## 3 20 21 21 9

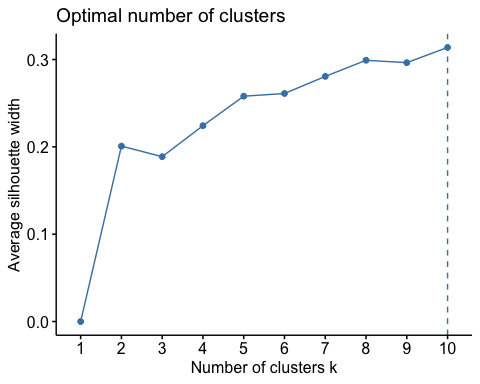
##  
plot(hc5, cex=0.6)  
rect.hclust(hc5, k=5, border = 2:5,)



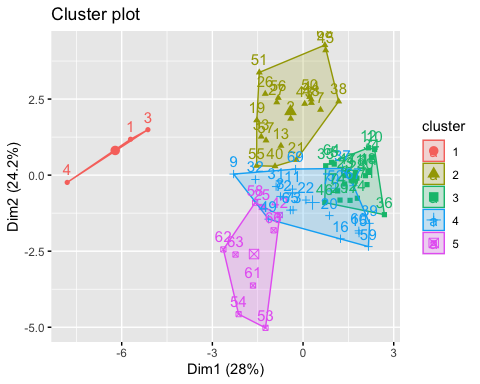
Bind\_clus<- cbind.data.frame(cbind(Cereals3, sub\_grp))

##Determing optimal clusters

library(cluster)  
library(factoextra)  
fviz\_nbclust(Cereals3, FUN = hcut, method = "silhouette")



fviz\_cluster(list(data= Cereals3, cluster= sub\_grp))

 ###3)The elementary public schools would like to choose a set of cereals to include in their #daily cafeterias. Every day a different cereal is offered, but all cereals should support a #healthy diet. For this goal, you are requested to find a cluster of “healthy cereals.”

Healthy\_cereals<- cbind(Cereals2,sub\_grp)  
mean(Healthy\_cereals[Healthy\_cereals$sub\_grp==1, "rating"])

## [1] 73.84446

mean(Healthy\_cereals[Healthy\_cereals$sub\_grp==2,"rating"])

## [1] 38.26161

mean(Healthy\_cereals[Healthy\_cereals$sub\_grp==3,"rating"])

## [1] 28.84825

mean(Healthy\_cereals[Healthy\_cereals$sub\_grp==4,"rating"])

## [1] 46.46513

mean(Healthy\_cereals[Healthy\_cereals$sub\_grp==5,"rating"])

## [1] 63.0184

##Therefore from the above observations it can be infered that the first cluster is given the high rating as the healthiest cereals of all the other 4.