assignment-5

pavan

29/11/2021

```
library(cluster)
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(dendextend)

##
## ---------------------
## 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(factoextra)

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

library(purrr)

##
## Attaching package: 'purrr'
```

```
## The following object is masked from 'package:caret':
##
##     lift
```

Q1.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.

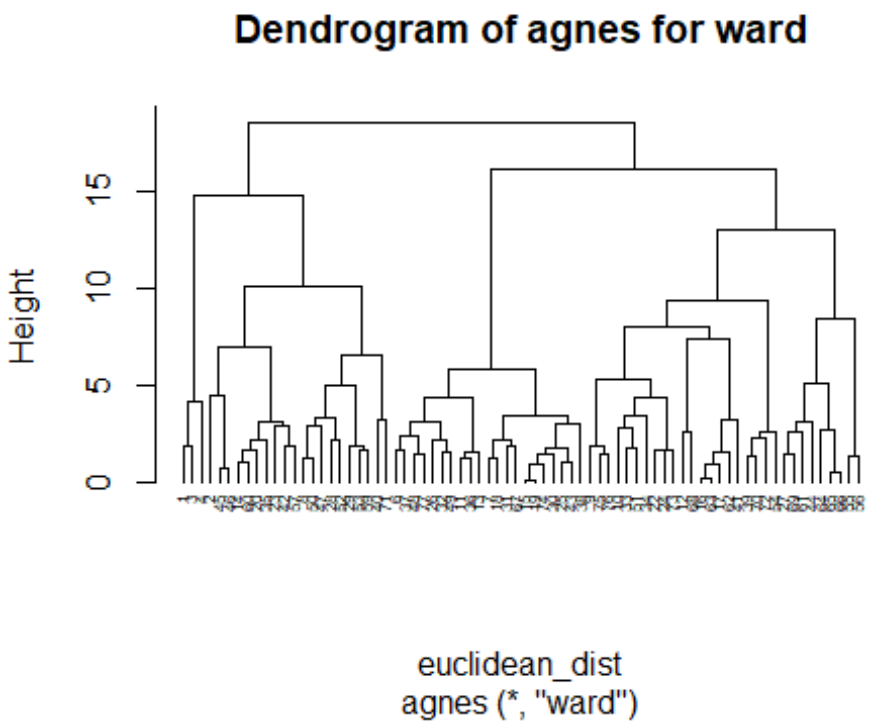
```
setwd("C:/Users/pavankumar pendela/Desktop/R/ppendela-74790/ppendela-74790/assignment 5")
cereals_hc <- read.csv("Cereals.csv")
sum(is.na(cereals_hc))
```

```
## [1] 4
```

```
cereals_hc <- na.omit(cereals_hc)##dataset with omitted rows with missing values
cereals_hc <- cereals_hc[,4:16]
cereals_hc <- scale(cereals_hc, center = T, scale = T)
set.seed(123)
# Dissimilarity matrix
euclidean_dist <- dist(cereals_hc, method = "euclidean")
method <- c( "average", "single", "complete", "ward")
names(method) <- c( "average", "single", "complete", "ward")
ac_values <- function(x) {

  agnes(euclidean_dist, method = x)$ac
}
map_dbl(method, ac_values)
```

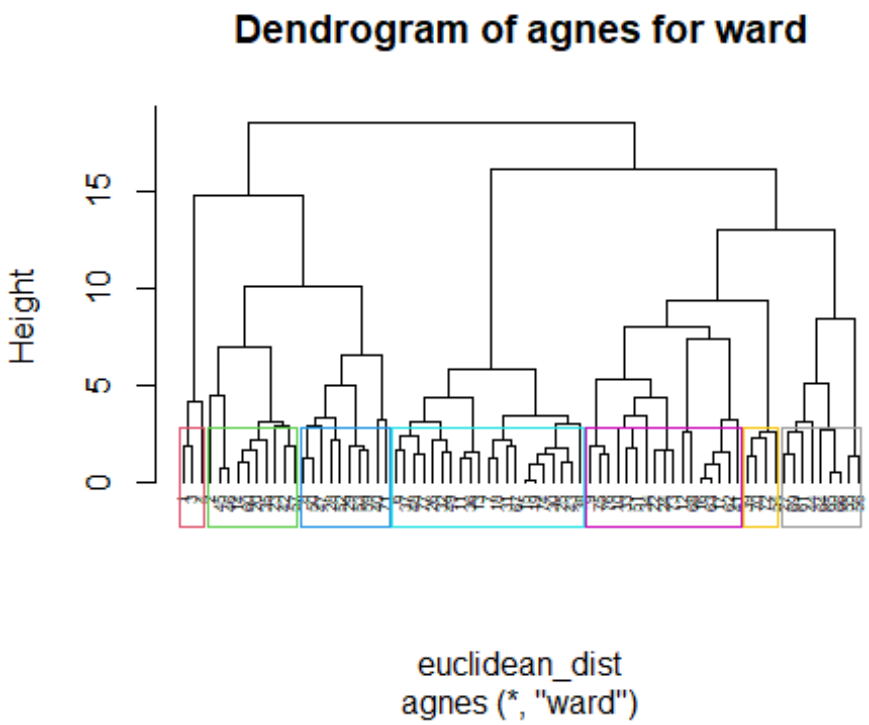
```
##   average    single  complete      ward
## 0.7766075 0.6067859 0.8353712 0.9046042
```

```
#The agglomerative coefficient obtained by Ward's method is the largest.
#Let's take a peek at the dendogram.
hc_ward <- agnes(euclidean_dist, method = "ward")
pltree(hc_ward, cex = 0.5, hang = -1, main = "Dendrogram of agnes for ward")
```



#Q2.How many clusters would you choose?

```
#install.packages("NbClust")
hc_ward <- agnes(euclidean_dist, method = "ward")
pltree(hc_ward, cex = 0.5, hang = -1, main = "Dendrogram of agnes for ward")
#install.packages("NbClust")
library(NbClust)
num_of_clust = NbClust(cereals_hc, distance = "euclidean", min.nc = 5, max.nc
= 10, method = "ward.D",index = 'dunn')
num_of_clust$Best.nc

## Number_clusters     Value_Index
##          7.0000          0.2604

#After checking NbClust value for best number of clusters, the best fits is
with K=7
rect.hclust(hc_ward, k = 7, border = 2:10)
```

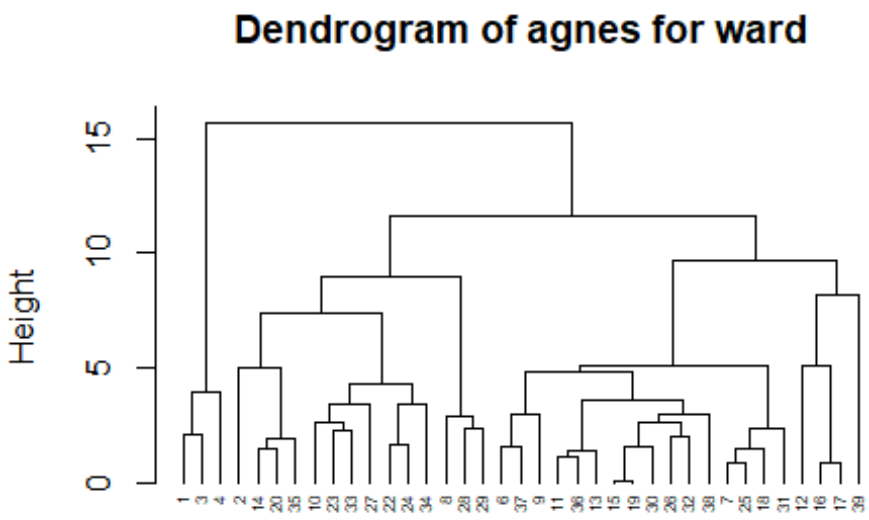



```
clust_comp <- cutree(hc_ward, k = 7)
temp3 <- cbind(as.data.frame(cbind(cereals_hc,clust_comp)))
```

Q3.Comment on the structure of the clusters and on their stability. Hint: To check stability, partition the data and see how well clusters formed based on one part apply to the other part. To do this:● 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

```
cereals_hc <- read.csv("Cereals.csv")
sum(is.na(cereals_hc))

## [1] 4

cereals_hc <- na.omit(cereals_hc)
cereals_hc <- cereals_hc[,4:16]
# Creating Partitions for into two data
c_partition_A <- cereals_hc[1:37,]
c_partition_B <- cereals_hc[38:74,]
c_partition_A <- scale(c_partition_A, center = T, scale = T)
c_partition_B <- scale(c_partition_B, center = T, scale = T)
euclidean_dist_partition_A <- dist(c_partition_A, method = "euclidean")
names(method) <- c( "average", "single", "complete", "ward")
ac_values1 <- function(x) {
  agnes(euclidean_dist_partition_A, method = x)$ac
```

```
}
map_dbl(method, ac_values1)

##   average    single  complete      ward
## 0.7091020 0.6724675 0.7706708 0.8570846

#The agglomerative coefficient obtained by Ward's method is the largest.
#Let's take a peek at the dendogram.
set.seed(123)
hc_ward_partition_A <- agnes(euclidean_dist_partition_A, method = "ward")
pltree(hc_ward_partition_A, cex = 0.5, hang = -1, main = "Dendrogram of agnes
for ward")
```




```
clust_comp_partition_A <- cutree(hc_ward_partition_A, k = 7)
result<-as.data.frame(cbind(c_partition_A,clust_comp_partition_A))
#result[result$clust_comp_partition_A==1,]
#center1<-colMeans(result[result$clust_comp_partition_A==1,])
klust <- 1:7
for (i in klust) {
  assign(paste0("center_",i),
colMeans(result[result$clust_comp_partition_A==i,]))
}
centroids <-
rbind(center_1,center_2,center_3,center_4,center_5,center_6,center_7
                    )
combined <- as.data.frame(rbind(centroids[,-14], c_partition_B))
temp1<-get_dist(combined)
```

```r
temp2<-as.matrix(temp1)
data1<-
data.frame(data=seq(1,nrow(c_partition_B),1),clusters=rep(0,nrow(c_partition_
B)))
for(i in 1:nrow(c_partition_B))
{
  data1[i,2]<-which.min(temp2[i+7,1:7])
}
cbind(temp3$clust_comp[38:74],data1$clusters)
```

```
##       [,1] [,2]
##  [1,]    4    4
##  [2,]    5    6
##  [3,]    2    2
##  [4,]    3    3
##  [5,]    6    5
##  [6,]    2    2
##  [7,]    2    2
##  [8,]    4    4
##  [9,]    3    3
## [10,]    3    3
## [11,]    4    4
## [12,]    5    5
## [13,]    4    2
## [14,]    4    4
## [15,]    7    5
## [16,]    6    5
## [17,]    6    5
## [18,]    2    5
## [19,]    4    4
## [20,]    2    2
## [21,]    6    5
## [22,]    5    6
## [23,]    5    6
## [24,]    6    5
## [25,]    6    5
## [26,]    6    5
## [27,]    3    3
## [28,]    5    6
## [29,]    6    5
## [30,]    7    7
## [31,]    4    4
## [32,]    7    5
## [33,]    5    5
## [34,]    3    3
## [35,]    5    5
## [36,]    5    6
## [37,]    3    3
```

```r
table(temp3$clust_comp[38:74]==data1$clusters)
```

```
##
## FALSE  TRUE
##    17    20

#We get 17 FALSE and 20 TRUE, indicating that the model is only partly
stable.
```

Q4.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." Should the data be normalized? If not, how should they be used in the cluster analysis?

```
Nutri_cereal <- na.omit(read.csv("Cereals.csv"))
Nutri_cereal<- cbind(Nutri_cereal,clust_comp)
for (i in 1:7){
  assign(paste0("Cluster",i),
mean(Nutri_cereal[Nutri_cereal$clust_comp==i,"rating"]))
}
a<-cbind(Cluster1,Cluster2,Cluster3,Cluster4,Cluster5,Cluster6,Cluster7)
paste("clearly cluster 1 has maximum rating", max(a)," hence we'll choose
it")

## [1] "clearly cluster 1 has maximum rating 73.8444633333333  hence we'll
choose it"
```

We must normalize data since we are using the distance metic algorithm. Since the features of data vary, we must scale it to similar features. And Yes, data should be normalized. Having non-normalized Data would simply disregard the attribute with the smaller range.