Assignment 5

Lukas van der Watt

11/26/2021

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v stringr 1.4.0
## v tidyr   1.1.3     v forcats 0.5.1
## v readr   2.0.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(cluster)
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

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

```
library(factoextra)
```

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

```
library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.15.1
## 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
## Or contact: <tal.galili@gmail.com>
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attaching package: 'dendextend'
```

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

PART 1

```
data <- read.csv("Cereals.csv")
data <- na.omit(data) #This omits the rows with NA values in some of the column
newdata <- data[,4:16]
str(newdata)
```

```
## 'data.frame':    74 obs. of  13 variables:
##  $ calories: int  70 120 70 50 110 110 130 90 90 120 ...
##  $ protein : int  4 3 4 4 2 2 3 2 3 1 ...
##  $ fat     : int  1 5 1 0 2 0 2 1 0 2 ...
##  $ sodium  : int  130 15 260 140 180 125 210 200 210 220 ...
##  $ fiber   : num  10 2 9 14 1.5 1 2 4 5 0 ...
##  $ carbo   : num  5 8 7 8 10.5 11 18 15 13 12 ...
##  $ sugars  : int  6 8 5 0 10 14 8 6 5 12 ...
##  $ potass  : int  280 135 320 330 70 30 100 125 190 35 ...
##  $ vitamins: int  25 0 25 25 25 25 25 25 25 25 ...
##  $ shelf   : int  3 3 3 3 1 2 3 1 3 2 ...
##  $ weight  : num  1 1 1 1 1 1 1.33 1 1 1 ...
##  $ cups    : num  0.33 1 0.33 0.5 0.75 1 0.75 0.67 0.67 0.75 ...
##  $ rating  : num  68.4 34 59.4 93.7 29.5 ...
```

```r
#Standardizing/Scaling the data.Distance measures used in clustering are highly influenced by the scale
dataCereals <- as.data.frame(scale(newdata))
str(dataCereals)
```

```
## 'data.frame':    74 obs. of  13 variables:
##  $ calories: num  -1.866 0.654 -1.866 -2.874 0.15 ...
##  $ protein : num  1.382 0.452 1.382 1.382 -0.477 ...
##  $ fat     : num  0 3.973 0 -0.993 0.993 ...
##  $ sodium  : num  -0.391 -1.78 1.18 -0.27 0.213 ...
##  $ fiber   : num  3.2287 -0.0725 2.816 4.8792 -0.2788 ...
##  $ carbo   : num  -2.5 -1.73 -1.99 -1.73 -1.09 ...
##  $ sugars  : num  -0.254 0.205 -0.484 -1.631 0.663 ...
##  $ potass  : num  2.561 0.515 3.125 3.266 -0.402 ...
##  $ vitamins: num  -0.182 -1.303 -0.182 -0.182 -0.182 ...
##  $ shelf   : num  0.942 0.942 0.942 0.942 -1.462 ...
##  $ weight  : num  -0.201 -0.201 -0.201 -0.201 -0.201 ...
##  $ cups    : num  -2.086 0.757 -2.086 -1.364 -0.304 ...
##  $ rating  : num  1.855 -0.598 1.215 3.658 -0.917 ...
```

```r
# Dissimilarity matrix
d <- dist(dataCereals, method = "euclidean")

# Hierarchical clustering using different types of Linkage
hc1 <- agnes(d, method = "complete" )
hc2 <- agnes(d, method = "single" )
hc3 <- agnes(d, method = "average" )
hc4 <- agnes(d, method = "ward" )
# Looking at the Aglomerative Coefficients of the different linkage methods
print(hc1$ac)
```

```
## [1] 0.8353712
```

```r
print(hc2$ac)
```

```
## [1] 0.6067859
```

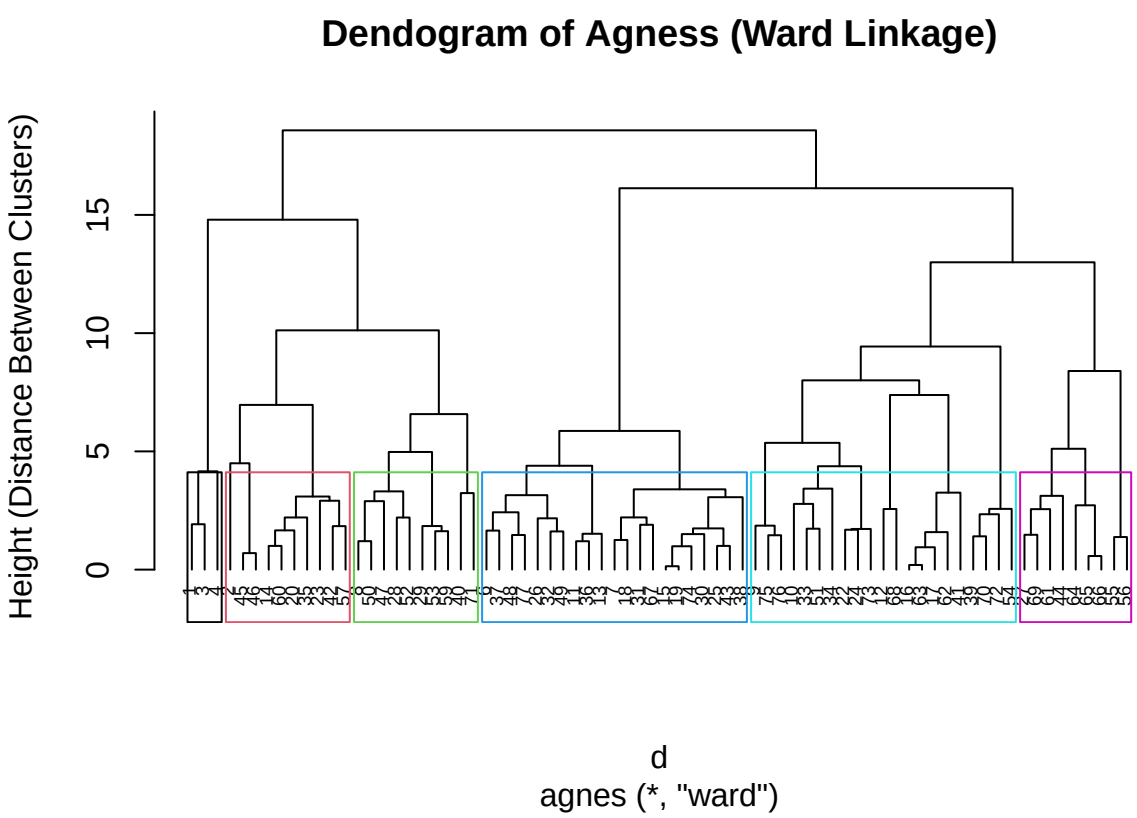
```r
print(hc3$ac)
```

```
## [1] 0.7766075
```

```r
print(hc4$ac) #Since this is the highest aglomerative coeffiecient (value:0.9046042), it has the strong
```

```
## [1] 0.9046042
```

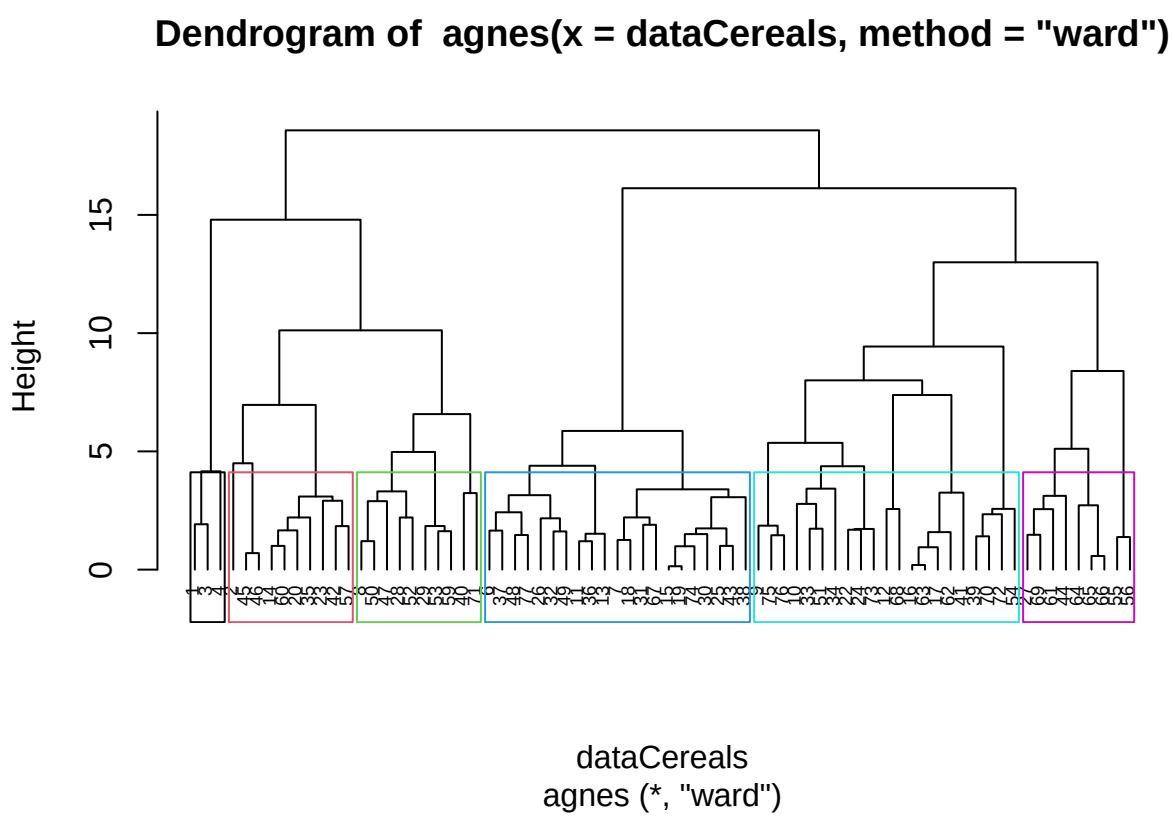
```r
# Plot the obtained dendogram
pltree(hc4, cex = 0.6, hang = -1, main = "Dendogram of Agness (Ward Linkage)",ylab = "Height (Distance
rect.hclust(hc4,k=6,border = 1:6) # k=6 is chosen
```



We can see that 6 clusters are formed at the height of approx 4.5

HOW MANY CLUSTERS SHOULD BE CHOSEN? PART 2

```
clustercount <- agnes(dataCereals, method="ward")
#Plotting the obtained dendogram with the agnes linkage method using the dataCereals dataset.
pltree(clustercount,cex = 0.6,hang = -1)
rect.hclust(clustercount,k=6,border = 1:6)
```




```
#Creating a cutoff on the dendogram at the height (Distance Between Clusters) of 10
Cutoff = cutree(clustercount,h=10)
Cereal_Clust <- mutate(dataCereals,cluster=Cutoff)
max(Cereal_Clust$cluster) #Showing the max number of the cluster column which will be the number of clu
```

```
## [1] 6
```

In the above output we see that 6 clusters should be chosen at the the Height of 10.

COMMENT ON THE STRUCTURE & STABILITY OF THE CLUSTER FORMED PART 3 Firstly the structure of the dataset is determined to be the ward linkage method as that proved to have the highest algomertive coefficient.

```
set.seed(123) #randomize
train.ind <- createDataPartition(Cereal_Clust$cluster,p = 0.65, list = FALSE)
training_set <- Cereal_Clust[train.ind,] #Training set making up 65% of the data
valid_set <- Cereal_Clust[-train.ind,] #Validation set making up 25% of the data

#Determining Centroid of A
Clust_A <- training_set %>% gather("Atribute","value",-cluster) %>% group_by(cluster,Atribute) %>% summa
```

```
## `summarise()` has grouped output by 'cluster'. You can override using the `.groups` argument.
```

```r
head (Clust_A)
```

```
## # A tibble: 6 x 3
## # Groups:   cluster [1]
##   cluster Atribute mean_values
##     <int> <chr>          <dbl>
## 1       1 calories       -1.87
## 2       1 carbo          -1.99
## 3       1 cups           -2.09
## 4       1 fat             0
## 5       1 fiber           2.82
## 6       1 potass          3.12
```

```r
#I know that I have to calcuate the distance between the centroids of A to Each observation of B for th
```

PART 4

```r
Healthy_data <- dataCereals[,c(1,5,9,13)]
str(Healthy_data)
```

```
## 'data.frame':    74 obs. of  4 variables:
##  $ calories: num  -1.866 0.654 -1.866 -2.874 0.15 ...
##  $ fiber   : num  3.2287 -0.0725 2.816 4.8792 -0.2788 ...
##  $ vitamins: num  -0.182 -1.303 -0.182 -0.182 -0.182 ...
##  $ rating  : num  1.855 -0.598 1.215 3.658 -0.917 ...
```

```r
Healthy_data <- mutate(Healthy_data,data$name,)
d_health <- dist(Healthy_data, method = "euclidean")
```

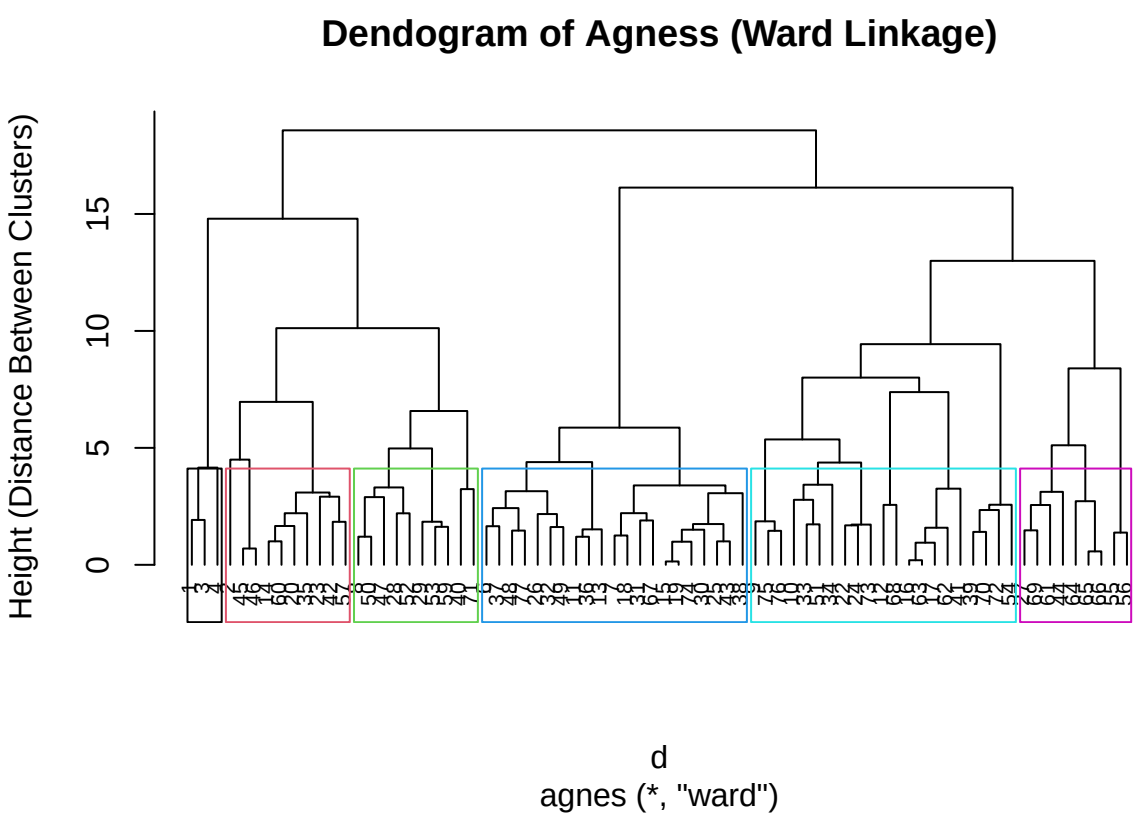
```
## Warning in dist(Healthy_data, method = "euclidean"): NAs introduced by coercion
```

```r
hc_health <- agnes(d_health, method = "ward" )

print(hc_health$ac) #Since this is the highest aglomerative coeffiecient, it has the strongest clusteri
```

```
## [1] 0.9698245
```

```r
# Plot the obtained dendogram
pltree(hc4, cex = 0.7, hang = -1, main = "Dendogram of Agness (Ward Linkage)",ylab = "Height (Distance
rect.hclust(hc4,k=6,border = 1:6) # k=6 is chosen
```



The data should be normalized as the different “healthy”(Column such as Protein, Vitamins and rating) columns should carry the same weight in the data whilst being on different measurements. If however there is a case that we do not have to normalize it would be because we want certain aspect/columns to cary more weight in the analysis. We would pair the sronger/desired variables to that of weaker variables without scaling and then run the hierarical clustering algorythms.