

Assignment5

```
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 ggplot2 3.3.5    v purrr   0.3.4  
## v tibble  3.1.6    v dplyr  1.0.8  
## v tidyr   1.2.0    v stringr 1.4.0  
## v readr   2.1.2    v forcats 0.5.1
```

```
## Warning: package 'tidyr' was built under R version 4.1.3
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
## 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(factoextra)
```

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

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

```
library(ggplot2)  
library(NbClust)
```

```
cereal_ex = read_csv("C:/Users/amuna/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.
```

```
#omitting all the missing values present in the data  
cereal_ex = na.omit(cereal_ex)
```

```
cereal_ex <- cereal_ex[,4:16]  
cereal_ex <- scale(cereal_ex, center = T, scale = T)  
set.seed(64060)  
euc_distance <- dist(cereal_ex, method = "euclidean")  
methods <- c("average", "single", "complete", "ward")  
names(methods) <- c("average", "single", "complete", "ward")  
values <- function(x) {  
  agnes(euc_distance, method = x)$ac  
}  
map_dbl(methods, values)
```

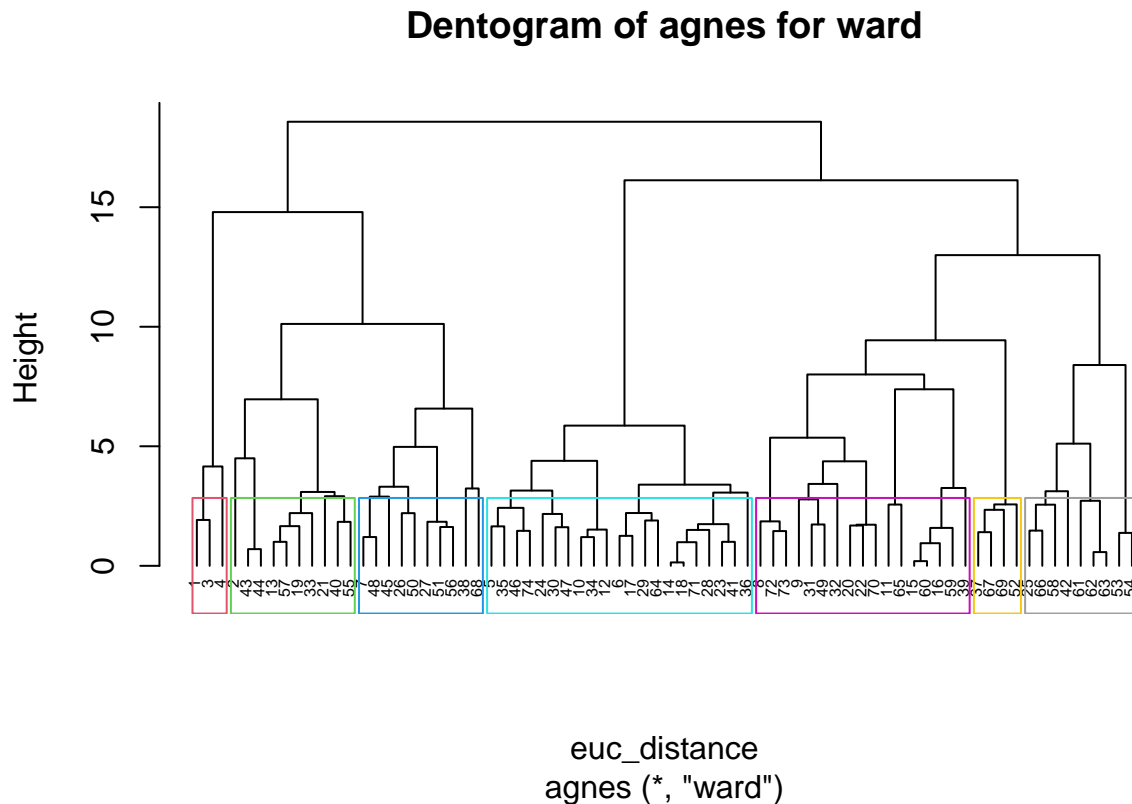
```
## average single complete ward  
## 0.7766075 0.6067859 0.8353712 0.9046042
```

```
ward_fn <- agnes(euc_distance, method = "ward")  
pltree(ward_fn, cex = .5, hang = -1, main = "Dentogram of agnes for ward")
```

```
ward_fn <- agnes(euc_distance, method = "ward")  
pltree(ward_fn, cex = .5, hang = -1, main = "Dentogram of agnes for ward")  
clusters_1 <- NbClust(cereal_ex, distance = "euclidean", min.nc = 5, max.nc = 10, method = "ward.D", in  
clusters_1$Best.nc
```

```
## Number_clusters      Value_Index
##           9.0000      0.2982
```

```
# Number of clusters with K=7
rect.hclust(ward_fn, k = 7, border = 2:10)
```



```
cluster_compt <- cutree(ward_fn, k = 7)
temp_variables <- cbind(as.data.frame(cbind(cereal_ex, cluster_compt)))
```

```
cereal_ex = read_csv("C:/Users/amuna/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.
```

```
sum(is.na(cereal_ex))
```

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

```

cereal_ex <- na.omit(cereal_ex)
cereal_ex <- cereal_ex[,4:16]
# Creating Partitions for into two data A, B
cluster_A <- cereal_ex[1:37,]
cluster_B <- cereal_ex[38:74,]
cluster_A <- scale(cluster_A, center = T, scale = T)
cluster_B <- scale(cluster_B, center = T, scale = T)
eucl_distance_partition_A <- dist(cluster_A, method = "euclidean")
names(methods) <- c("average", "single", "complete", "ward")
values1 <- function(x) {
  agnes(eucl_distance_partition_A, method = x)$ac
}
map_dbl(methods, values1)

```

```

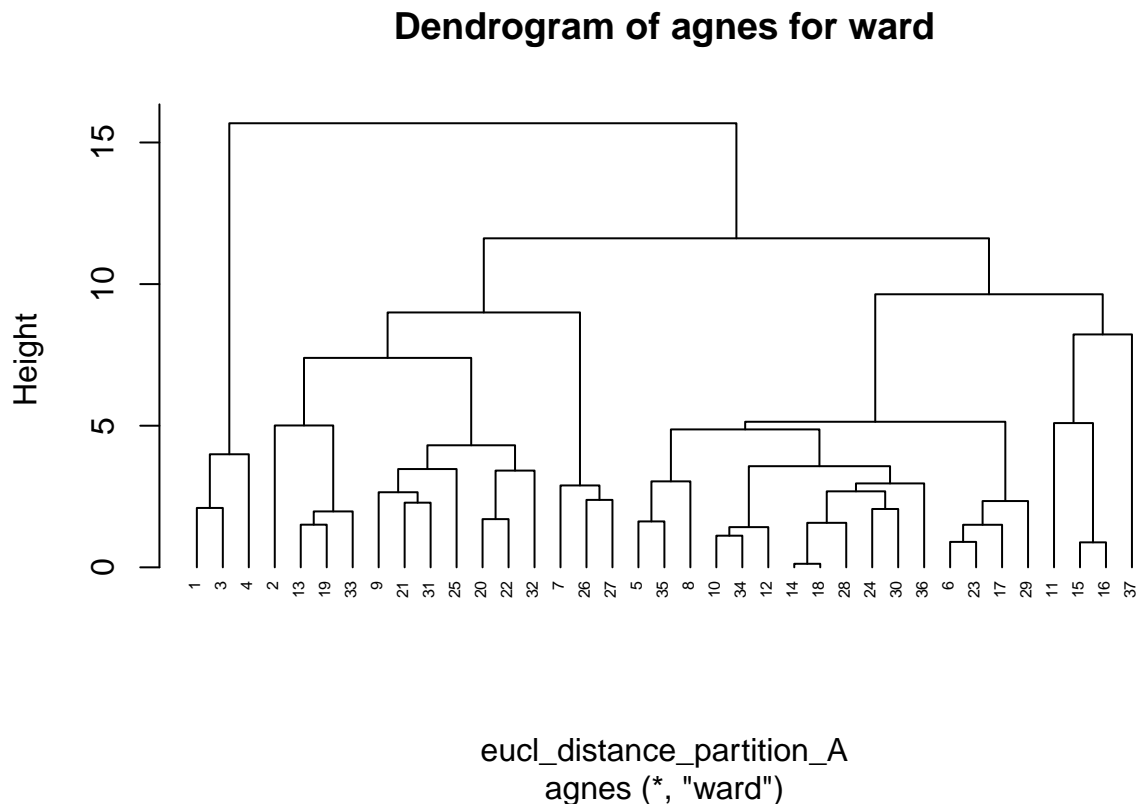
## average single complete ward
## 0.7091020 0.6724675 0.7706708 0.8570846

```

```

#Let's take a peek at the dendrogram.
set.seed(64060)
ward_partition_A <- agnes(eucl_distance_partition_A, method = "ward")
pltree(ward_partition_A, cex = 0.5, hang = -1, main = "Dendrogram of agnes for ward")

```



```

cluster_comp_part_A <- cutree(ward_partition_A, k = 7)
results<-as.data.frame(cbind(cluster_A,cluster_comp_part_A))

```

```

kluster <- 1:7
for (i in kluster) {
assign(paste0("center_",i), colMeans(results[results$cluster_comp_part_A==i,]))
}
centroid <- rbind(center_1,center_2,center_3,center_4,center_5,center_6,center_7)
combined_data <- as.data.frame(rbind(centroid[, -14], cluster_B))
temp_var1<-get_dist(combined_data)
temp_var2<-as.matrix(temp_var1)
data_df<-data.frame(data=seq(1,nrow(cluster_B),1),clusters=rep(0,nrow(cluster_A)))
for(i in 1:nrow(cluster_B))
{
data_df[i,2]<-which.min(temp_var2[i+7,1:7])
}
cbind(temp_variables$cluster_comp[38:74],data_df$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

```

```
table(temp_variables$cluster_comp[38:74]==data_df$clusters)
```

```
##
```

```
## FALSE  TRUE
```

```
##    17    20
```