Assignment 4

ManojKumar Chalamala

11/13/2019

```
library(readr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v ggplot2 3.2.1     v forcats 0.4.0
```

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

```
library(factoextra)
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```
library(cluster)
```

Read the data

```
Cereals <- read_csv("Cereals.csv")
```

```
## Parsed with column specification:
## 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()
## )
```

```
head(Cereals)
```

```
## # A tibble: 6 x 16
##   name  mfr   type  calories protein   fat sodium fiber carbo sugars potass
##   <chr> <chr> <chr>    <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>  <dbl>  <dbl>
## 1 100%~ N     C           70       4     1    130  10     5        6    280
## 2 100%~ Q     C          120       3     5     15   2     8        8    135
## 3 All-~ K     C           70       4     1    260   9     7        5    320
## 4 All-~ K     C           50       4     0    140  14     8        0    330
## 5 Almo~ R     C          110       2     2    200   1    14        8     NA
## 6 Appl~ G     C          110       2     2    180   1.5  10.5     10     70
## # ... with 5 more variables: vitamins <dbl>, shelf <dbl>, weight <dbl>,
## #   cups <dbl>, rating <dbl>
```

Data Preprocessing. Remove all cereals with missing values

```
# Number of missing values
sum(is.na(Cereals))
```

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

```
# Remove all cereals with missing values
MyData <- na.omit(Cereals)
#str(MyData)
```

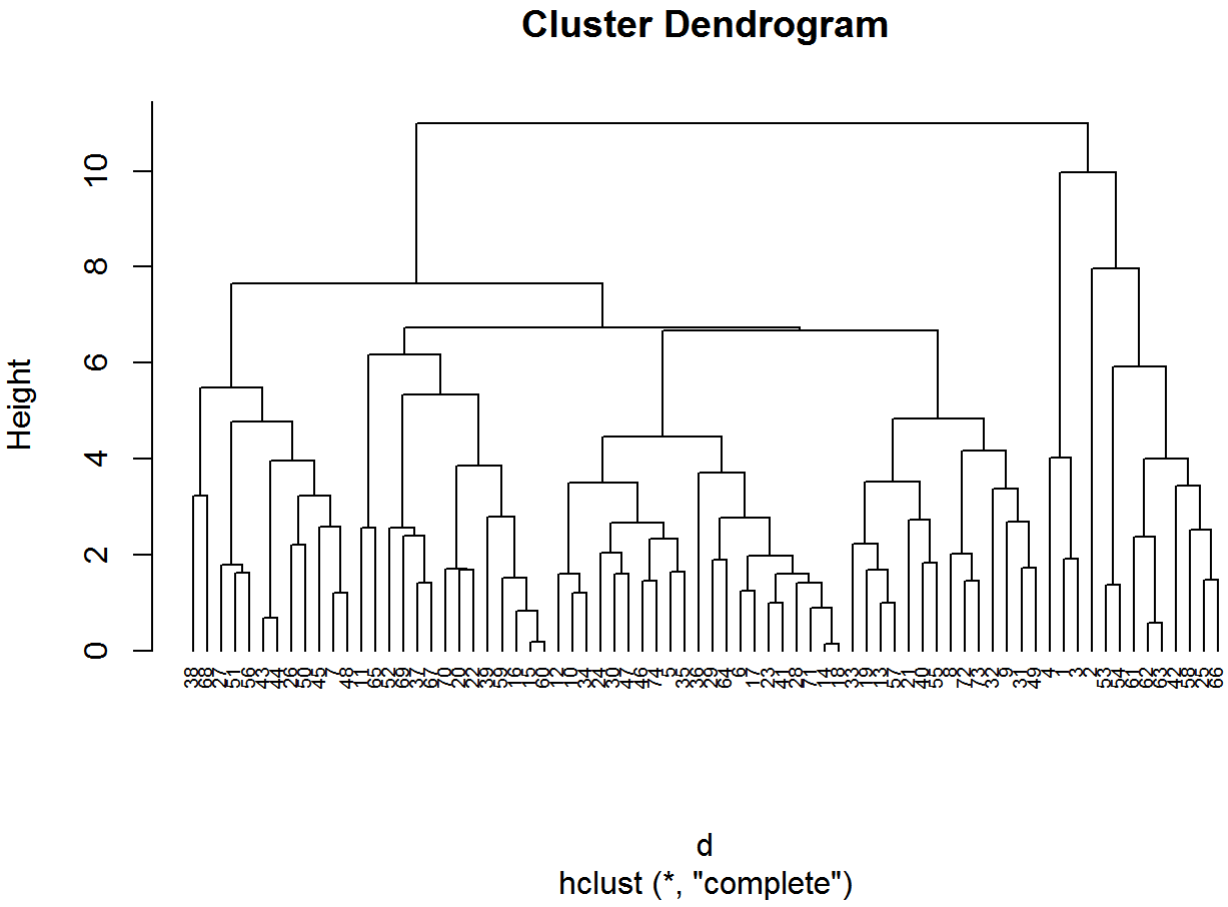
Normalization and Scale the Data

```
Cerealnames <- MyData$name
# Drop the Categorical Columns
MyData <- MyData[, c(-1, -2, -3)]
MyData <- scale(MyData, center = T, scale = T)
head(MyData)
```

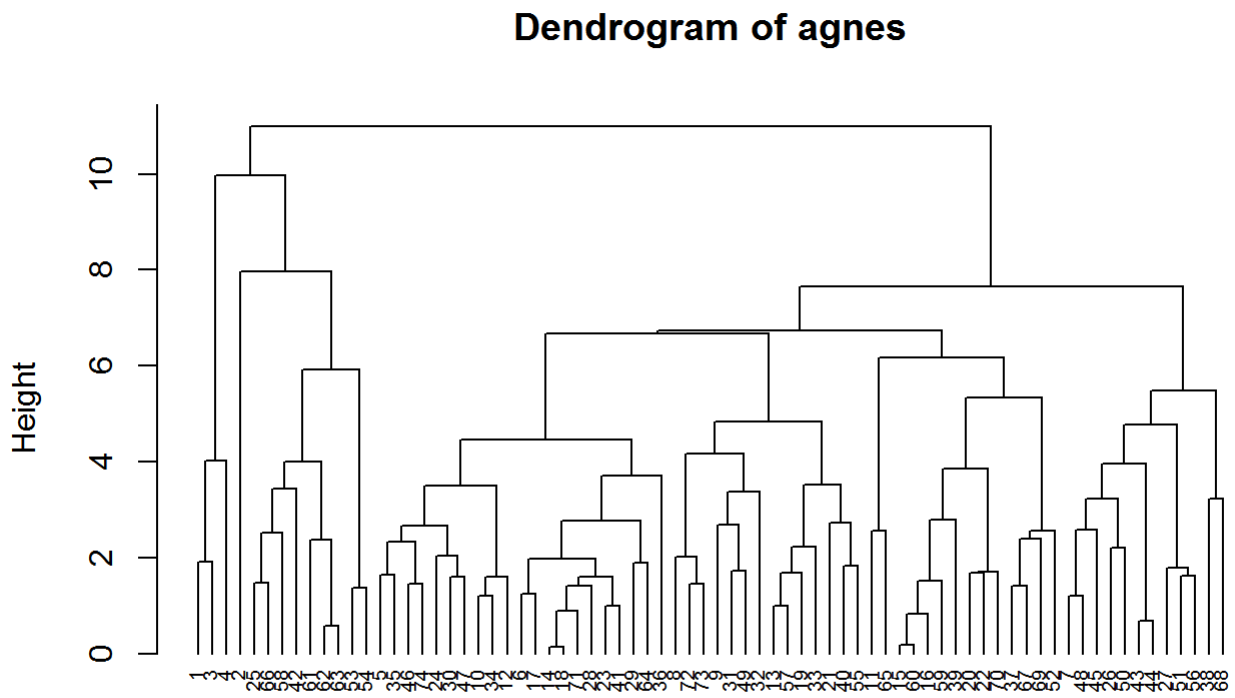
```
##        calories    protein        fat     sodium       fiber      carbo
## [1,] -1.8659155  1.3817478  0.0000000 -0.3910227  3.22866747 -2.5001396
## [2,]  0.6537514  0.4522084  3.9728810 -1.7804186 -0.07249167 -1.7292632
## [3,] -1.8659155  1.3817478  0.0000000  1.1795987  2.81602258 -1.9862220
## [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
```

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

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

# Hierarchical clustering using Complete Linkage
hc1 <- hclust(d, method = "complete" )

# Plot the obtained dendrogram
plot(hc1, cex = 0.6, hang = -1)
```




```
# Compute with agnes and with different linkage methods
hc_single <- agnes(MyData, method = "single")
hc_complete <- agnes(MyData, method = "complete")
hc_average <- agnes(MyData, method = "average")
hc_ward <- agnes(MyData, method = 'ward')

pltree(hc_complete, cex = 0.6, hang = -1, main = "Dendrogram of agnes")
```

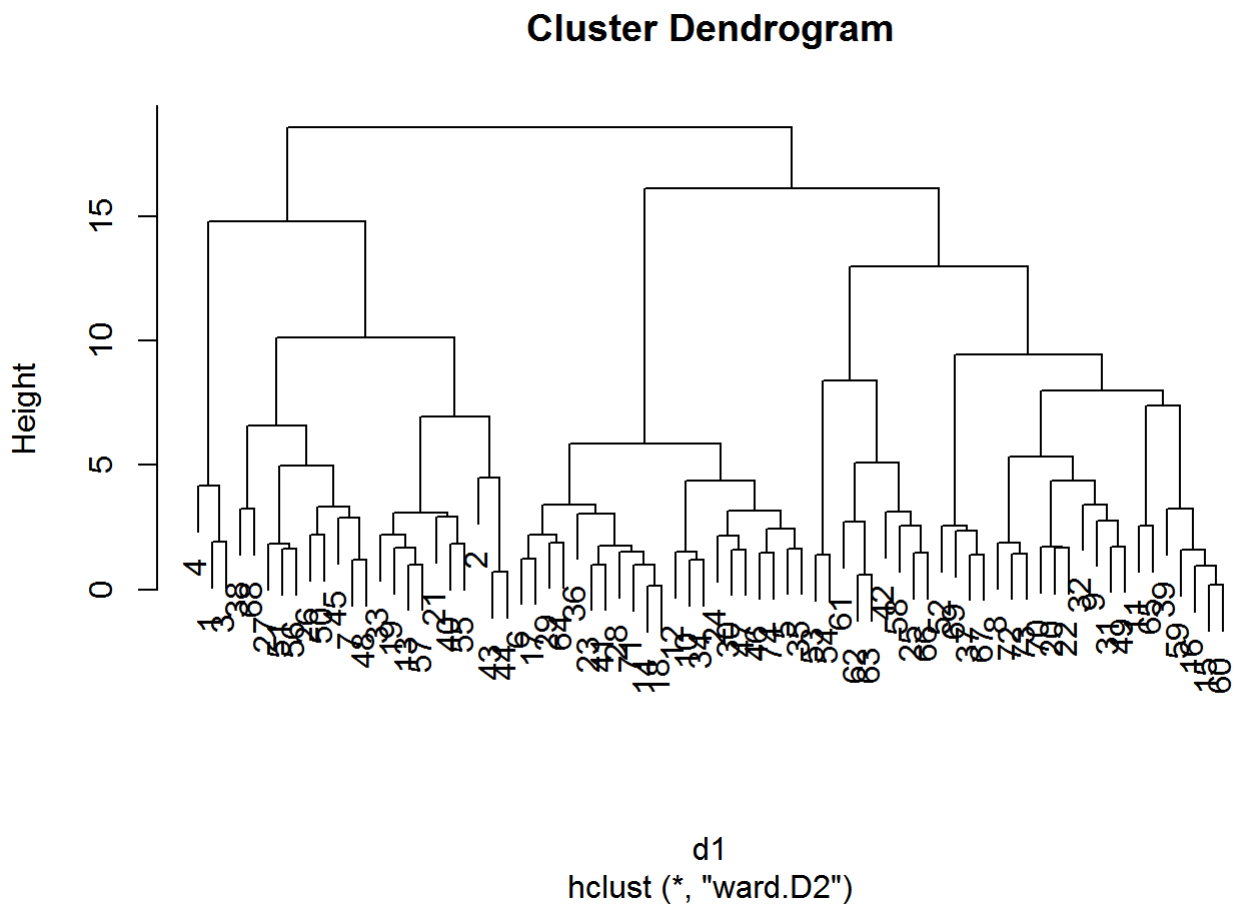



```
# Compare Agglomerative Coefficients

m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")

# function to compute coefficient

ac <- function(x) {
  agnes(MyData, method = x)$ac
}

map_dbl(m, ac)
```

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

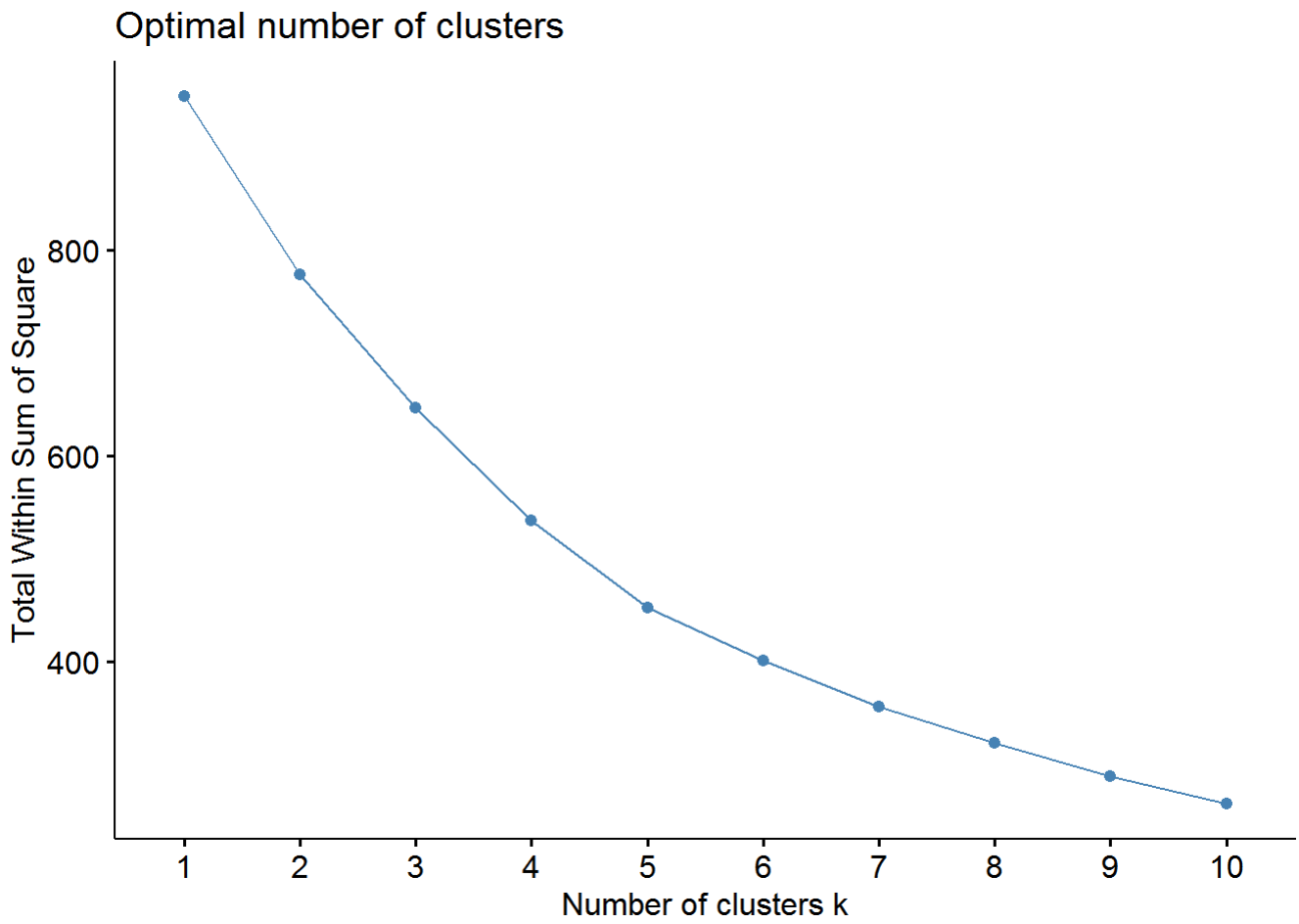
On comparing the best method is ‘Ward’

```
# Dissimilarity matrix
d1 <- dist(MyData, method = "euclidean")

# Hierarchical clustering using Ward Linkage
hc2 <- hclust(d1, method = "ward.D2" )

# Plot the obtained dendrogram
plot(hc2)
```



2. How many clusters would you choose?

```
# Elbow method to chosse K
fviz_nbclust(MyData, FUN = hcut, method = "wss")
```



Let's choose 6 clusters and cluster the data

```
# Cut the tree to 6 clusters, using the cutree() function
hc3 <- cutree(hc2, k = 6)

# Number of Cereals in each cluster
table(hc3)
```

```
## hc3
##  1  2  3  4  5  6
##  3 10 21 10 21  9
```

```
# Store the clusters in a data frame along with the cereals data

cereals_hc <- cbind(hc3, MyData)

# We can also use the cutree output to add the the cluster each observation belongs to to our or
iginal data.

colnames(cereals_hc)[1] <- "cluster"

head(cereals_hc)
```

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

```
plot(hc2)

rect.hclust(hc2, k = 6, border = 2:6)
```