## Clustering

### Anthony Sumter

**Libraries**

#install.packages("tidyverse","cluster", "factoextra", "dendextend")  
options(tidyverse.quiet = TRUE)  
library(tidyverse)  
library(cluster)  
library(factoextra)

## Warning: package 'factoextra' was built under R version 3.6.2

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

library(dendextend)

##   
## ---------------------  
## Welcome to dendextend version 1.13.4  
## 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

**Dataset**

trucks = read.csv("trucks.csv")  
str(trucks)

## 'data.frame': 4000 obs. of 3 variables:  
## $ Driver\_ID: num 3.42e+09 3.42e+09 3.42e+09 3.42e+09 3.42e+09 ...  
## $ Distance : num 71.2 52.5 64.5 55.7 54.6 ...  
## $ Speeding : int 28 25 27 22 25 10 20 8 34 19 ...

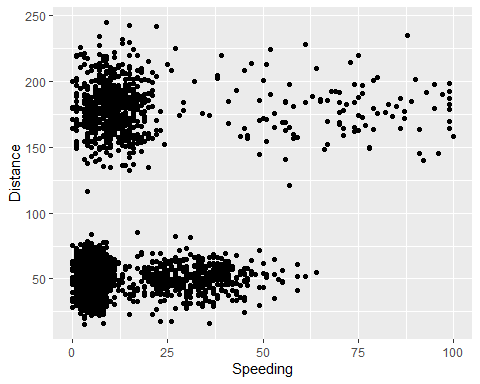
summary(trucks)

## Driver\_ID Distance Speeding   
## Min. :3.423e+09 Min. : 15.52 Min. : 0.00   
## 1st Qu.:3.423e+09 1st Qu.: 45.25 1st Qu.: 4.00   
## Median :3.423e+09 Median : 53.33 Median : 6.00   
## Mean :3.423e+09 Mean : 76.04 Mean : 10.72   
## 3rd Qu.:3.423e+09 3rd Qu.: 65.63 3rd Qu.: 9.00   
## Max. :3.423e+09 Max. :244.79 Max. :100.00

#### Task 1: Plot the relationship between Distance and Speeding. Describe this relationship. Does there appear to be any natural clustering of drivers?

It does appear that there is natural clustering of drivers as the first cluster closer to the bottom of the X axis shows that speeding is occuring in a range of 0-100 miles driven. Compared to the second cluster which shows that speeding is occuring during approximately 125-225 miles driven.

ggplot(trucks,aes(x=Speeding,y=Distance)) + geom\_point()



#### Task 2: Create a new data frame (called trucks2) that excludes the Driver\_ID variable and includes scaled versions of the Distance and Speeding variables. NOTE: Wrap the scale(trucks2) command in an as.data.frame command to ensure that the resulting object is a data frame. By default, scale converts data frames to lists

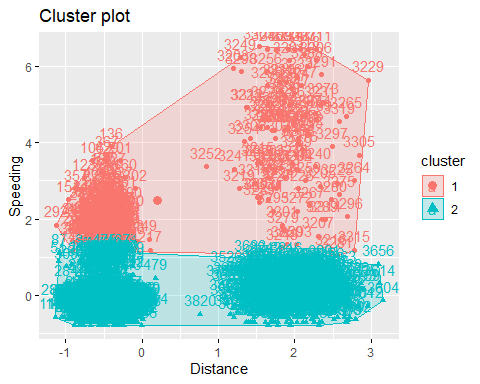
trucks2 = trucks %>% select("Distance","Speeding")  
trucks2\_scaled = scale(as.data.frame(trucks2))  
summary(trucks2\_scaled)

## Distance Speeding   
## Min. :-1.1319 Min. :-0.7821   
## 1st Qu.:-0.5759 1st Qu.:-0.4903   
## Median :-0.4248 Median :-0.3444   
## Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.1947 3rd Qu.:-0.1255   
## Max. : 3.1560 Max. : 6.5127

#### Task 3 Use k-Means clustering with two clusters (k=2) to cluster the trucks2 data frame. Use a random number seed of 1234. Visualize the clusters using the fviz\_cluster function. Comment on the clusters.

The larger cluster out the two vizualized is cluster 1. Even with that being the case neither of the clusters are well defined or good as they are overlapping each other.

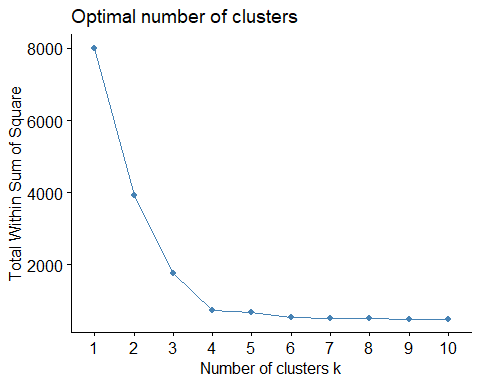
set.seed(1234)  
clusters1 <- kmeans(trucks2\_scaled, 2)  
fviz\_cluster(clusters1, trucks2\_scaled)



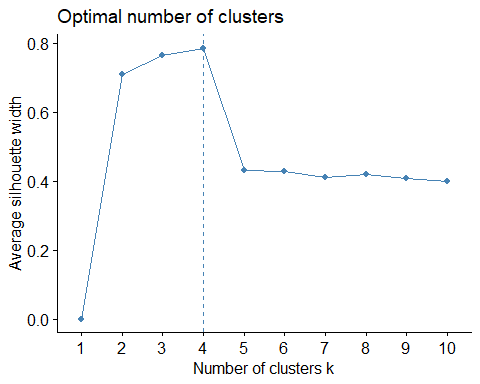
#### Task 4: Use the two methods from the k-Means lecture to identify the optimal number of clusters. Use a random number seed of 123 for these methods. Is there consensus between these two methods as the optimal number of clusters?

Based upon the two methods that are used to find the optimal number of clusters there is a consensus that the optimal number of clusters is 4.

set.seed(123)  
fviz\_nbclust(trucks2\_scaled, kmeans, method = "wss")

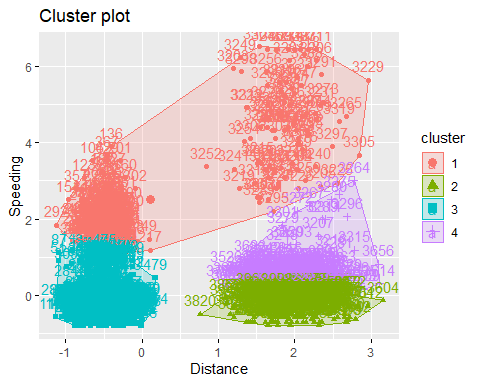


set.seed(123)  
fviz\_nbclust(trucks2\_scaled, kmeans, method = "silhouette")



#### Task 5: Use the optimal number of clusters that you identified in Task 4 to create k-Means clusters. Use a random number seed of 1234. Use the fviz\_cluster function to visualize the clusters.

set.seed(1234)  
clusters2 <- kmeans(trucks2\_scaled, 4)  
fviz\_cluster(clusters2, trucks2\_scaled)



#### Task 6: In words, how would you characterize the clusters you created in Task 5?

It appears the clusters 1 and 3 as well as clusters 2 and 4 are overlapping each other. This is an indication that they share the same characteristics but are not well defined or good.

**Dataset 2**

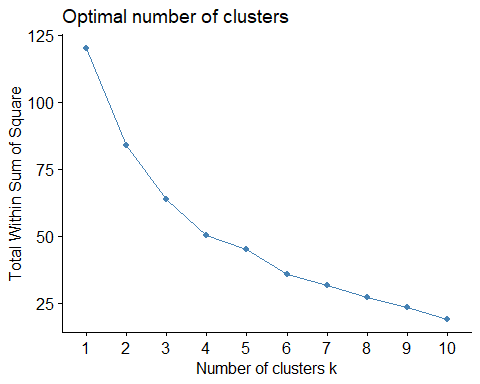
wine = read.csv("wineprice.csv")  
wine2 = wine %>% select("Price","WinterRain","AGST","HarvestRain", "Age")  
wine2\_scaled = scale(as.data.frame(wine2))  
summary(wine2\_scaled)

## Price WinterRain AGST   
## Min. :-1.32596 Min. :-1.73332 Min. :-2.25947   
## 1st Qu.:-0.84329 1st Qu.:-0.52375 1st Qu.:-0.45801   
## Median : 0.08284 Median :-0.03992 Median : 0.03548   
## Mean : 0.00000 Mean : 0.00000 Mean : 0.00000   
## 3rd Qu.: 0.65777 3rd Qu.: 0.69339 3rd Qu.: 0.82524   
## Max. : 2.19343 Max. : 1.69885 Max. : 1.68888   
## HarvestRain Age   
## Min. :-1.4856 Min. :-1.586   
## 1st Qu.:-0.8003 1st Qu.:-0.806   
## Median :-0.2494 Median :-0.026   
## Mean : 0.0000 Mean : 0.000   
## 3rd Qu.: 0.5165 3rd Qu.: 0.754   
## Max. : 1.9275 Max. : 1.794

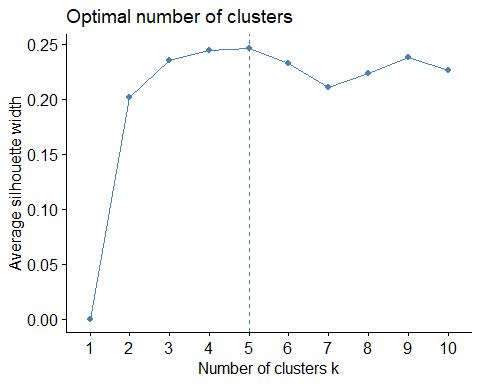
#### Task 7: Use the two methods from Task 4 to determine the optimal number of k-Means clusters for this data. Use a random number seed of 123. Is there consensus between these two methods as the optimal number of clusters?

Based upon the two methods that are used to find the optimal number of clusters there is a consensus that the optimal number of clusters is 5.

set.seed(123)  
fviz\_nbclust(wine2\_scaled, kmeans, method = "wss")

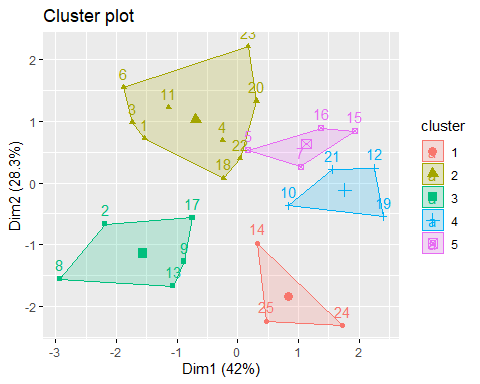


set.seed(123)  
fviz\_nbclust(wine2\_scaled, kmeans, method = "silhouette")



#### Task 8: Use the optimal number of clusters that you identified in Task 4 to create k-Means clusters. Use a random number seed of 1234. Use the fviz\_cluster function to visualize the clusters.

set.seed(1234)  
clusters3 <- kmeans(wine2\_scaled, 5)  
fviz\_cluster(clusters3, wine2\_scaled)

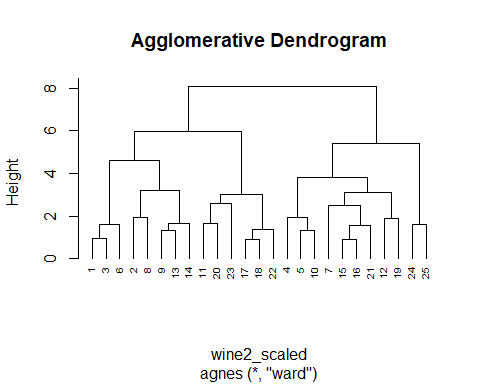


#### Task 9: Use agglomerative clustering to develop a dendogram for the scaled wine data. Follow the same process from the lecture where we used a custom function to identify the distance metric that maximizes the “agglomerative coefficient”. Plot the dendogram.

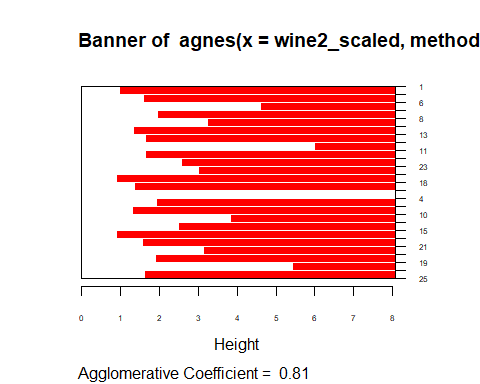
m = c( "average", "single", "complete", "ward")  
names(m) = c( "average", "single", "complete", "ward")  
  
ac = function(x) {  
 agnes(wine2\_scaled, method = x)$ac  
}  
map\_dbl(m, ac)

## average single complete ward   
## 0.5666719 0.2920143 0.7196616 0.8112139

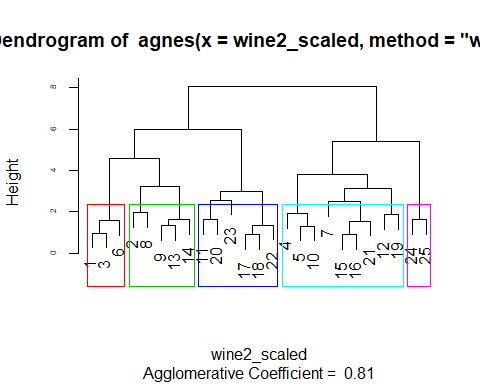
hc = agnes(wine2\_scaled, method = "ward")  
pltree(hc, cex = 0.6, hang = -1, main = "Agglomerative Dendrogram")



plot(hc, cex.axis= 0.5)

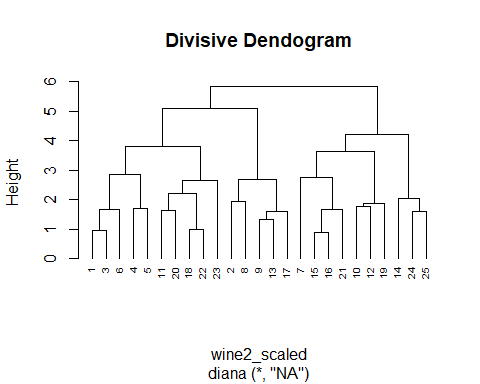


rect.hclust(hc, k = 5, border = 2:6)

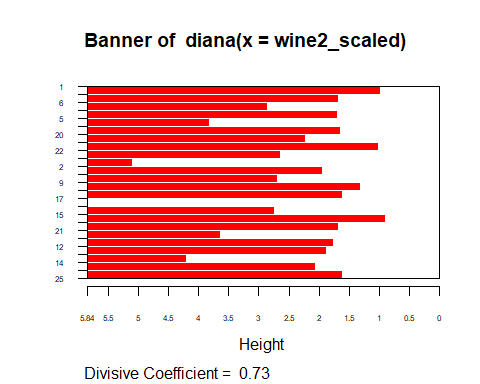


#### Task 10: Repeat Task 9, but with divisive clustering.

hc2 = diana(wine2\_scaled)  
pltree(hc2, cex = 0.6, hang = -1, main = "Divisive Dendogram")



plot(hc2, cex.axis= 0.5)



rect.hclust(hc2, k = 5, border = 2:6)

