Kraft\_MOD6\_BAN502

Madison

3/1/2020

library(tidyverse)

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

## -- Attaching packages ---------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(cluster)

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

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

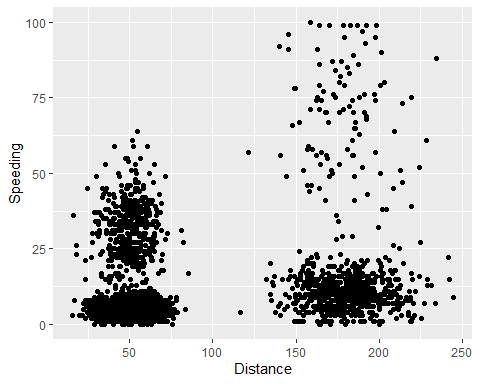
trucks <- read\_csv("trucks.csv")

## Parsed with column specification:  
## cols(  
## Driver\_ID = col\_double(),  
## Distance = col\_double(),  
## Speeding = col\_double()  
## )

# TASK ONE

In the plot below, we can see that there appears to be a significantly less amount of speeding drivers in-between the distance of about 75 to about 150. This could be explained that the drivers who drove a distance between 75-150miles did not speed hardly as much as drivers who drove less than 75 miles or more than 150 miles.

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



# TASK TWO

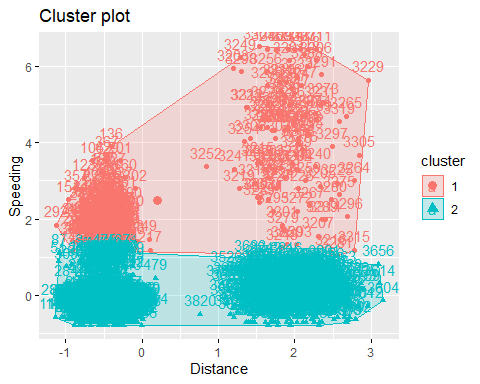
trucks2 <- trucks %>% select(Distance, Speeding)  
trucks2 <- as.data.frame(scale(trucks2))

# TASK THREE

I was expecting the data to be clustered by the distance, rather than by speeding, but I definitely think that there is a better clustering approach.

set.seed(1234)  
clusters <- kmeans(trucks2, 2)

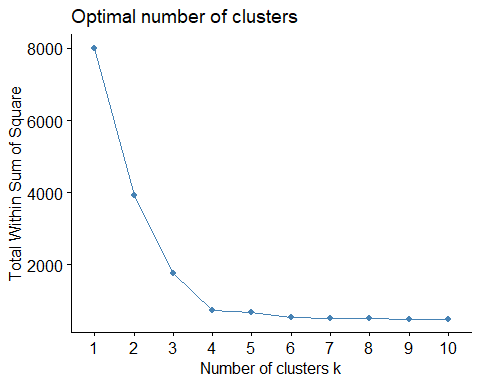
fviz\_cluster(clusters, trucks2)



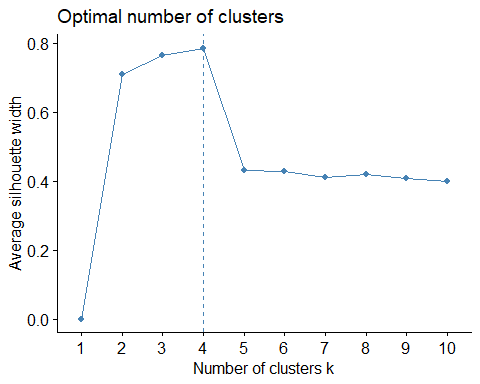
# TASK FOUR

After running the WSS and Silhouette methods, there does seem to be a consensus that 4 is the optimal number of clusters.

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



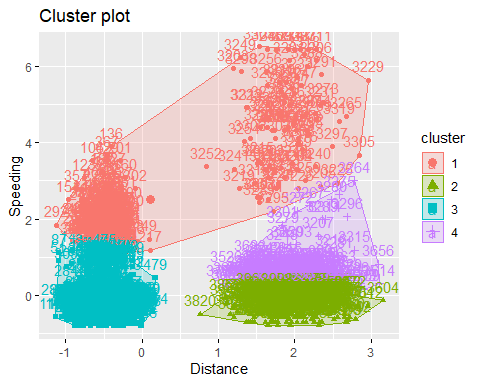
fviz\_nbclust(trucks2, kmeans, method = "silhouette")



# TASK FIVE

set.seed(1234)  
clusters4 <- kmeans(trucks2, 4)

fviz\_cluster(clusters4, trucks2)



# TASK SIX

The clusters made in task five appear to be more accurate and condensed. Although for cluster 1, it looks like a lot of the data is far away from the center.

wine <- read\_csv("wine.csv")

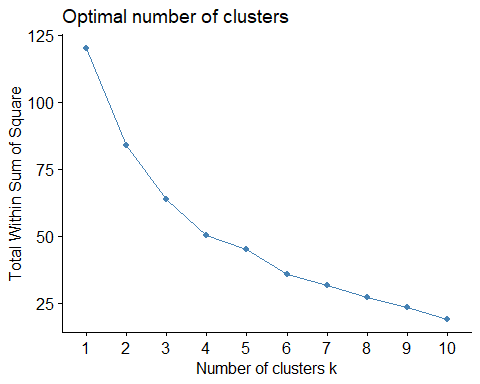
## Parsed with column specification:  
## cols(  
## Year = col\_double(),  
## Price = col\_double(),  
## WinterRain = col\_double(),  
## AGST = col\_double(),  
## HarvestRain = col\_double(),  
## Age = col\_double(),  
## FrancePop = col\_double()  
## )

wine2<- wine %>% select(-c("Year","FrancePop"))  
wine2<- as.data.frame(scale(wine2))

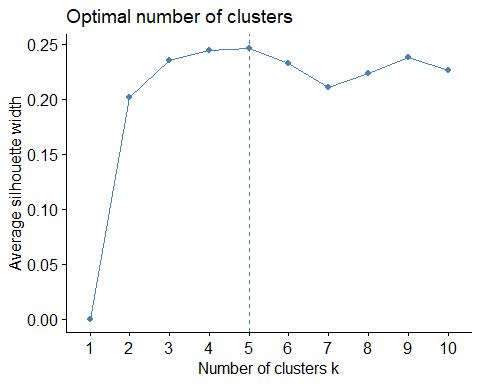
# TASK SEVEN

Based on the WSS method, I would assume that the optimal number of clusters would be 4, but it could be 5 as well. Based on the Silhouette approach, 5 would be the optimal number of clusters. So there is a little bit of inconsistency here.

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

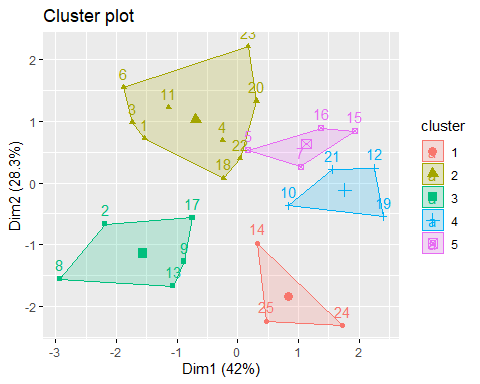


fviz\_nbclust(wine2, kmeans, method = "silhouette")



# TASK EIGHT

set.seed(1234)  
wine\_clusters <- kmeans(wine2, 5)  
fviz\_cluster(wine\_clusters, wine2)

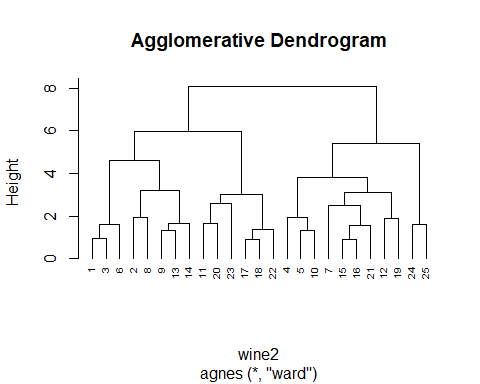


# TASK NINE HIERARCHICAL CLUSTERS

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

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

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



# TASK TEN DIVISIVE CLUSTERING

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

