## Stuart Broach

## Assignment 6

#install.packages("cluster")  
#install.packages("factoextra")  
#install.packages("dendextend")  
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.5.2

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

## v ggplot2 3.1.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.7  
## v tidyr 0.8.2 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.3.0

## -- 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 3.5.2

library(factoextra)

## Warning: package 'factoextra' was built under R version 3.5.2

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

library(dendextend)

## Warning: package 'dendextend' was built under R version 3.5.2

##   
## ---------------------  
## Welcome to dendextend version 1.9.0  
## 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))  
## ---------------------

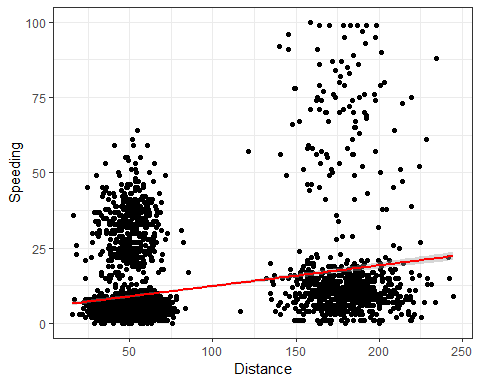
##   
## Attaching package: 'dendextend'

## 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\_integer()  
## )

ggplot(trucks,aes(x=Distance,y=Speeding)) + geom\_point() + geom\_smooth(method = "lm", color = "red") + theme\_bw()



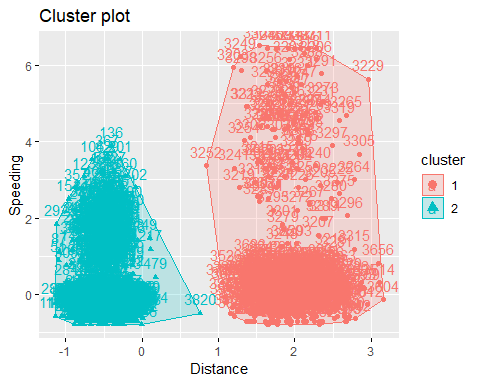
Yes there seems to be 4 groups of clusters. The driver seems to be that the further a driver drives in a day, the more likely they’re to speed to get to where they need to go.

trucks2 = trucks %>% select(-Driver\_ID)

trucks2= as.data.frame(scale(trucks2))  
summary(trucks2)

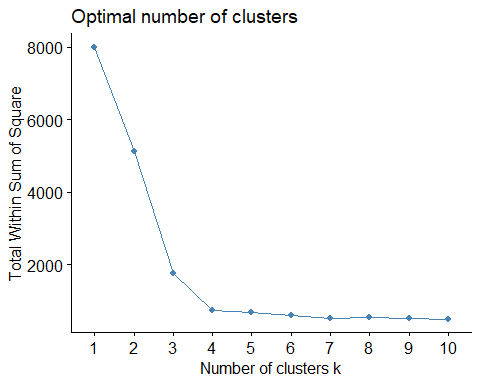
## Distance Speeding   
## Min. :-1.1319 Min. :-0.7821   
## 1st Qu.:-0.5759 1st Qu.:-0.4903   
## Median :-0.4248 Median :-0.3444   
## Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.1947 3rd Qu.:-0.1255   
## Max. : 3.1560 Max. : 6.5127

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

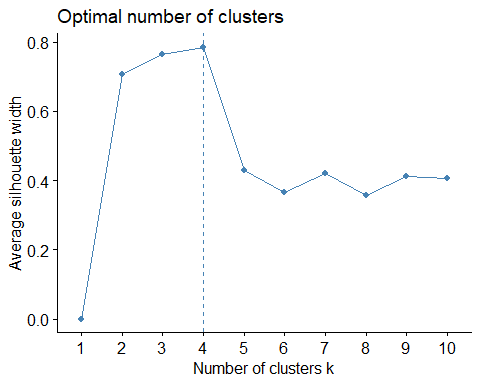


It defiantly split the clusters are the 125 mile mark that was so obvious in the intial chart, but the clusters are not tight at all, especially cluster 1. It would do better if there was 4 clusters.

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

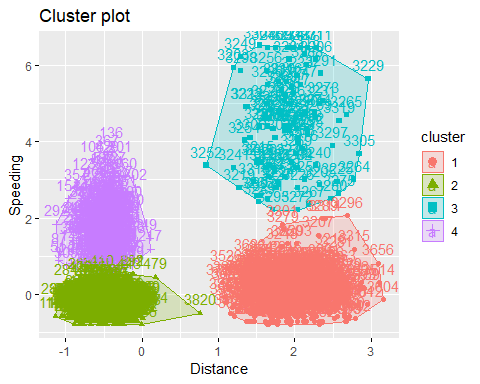


set.seed(123)  
fviz\_nbclust(trucks2, kmeans, method = "silhouette" )



Yes, the conclusion of both methods is that 4 clusters is the best option.

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



I would classify these clusters as those who are risk takers and those who are risk averse. Green and red are both risk averse while purple and blue are both risk takers. But there comes a point in distance where (about 125 mile mark) where drivers feel more pressure to complete a task in a given day so they speed more. This is why there are 4 groups instead of just 2.

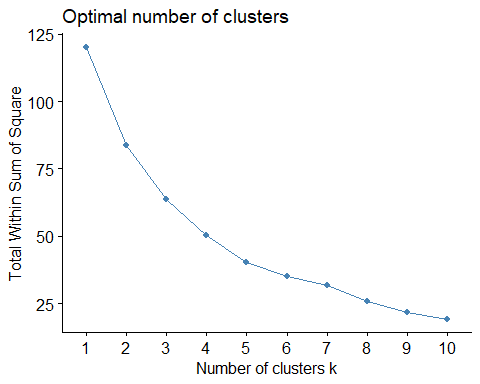
wine = read\_csv("wineprice.csv")

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

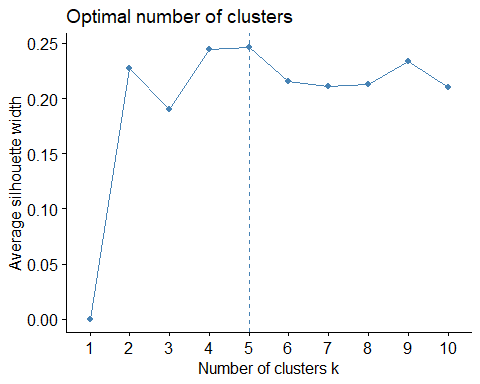
wine2 = wine %>% select(-Year, -FrancePop)  
wine2= as.data.frame(scale(wine2))  
summary(wine2)

## 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

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

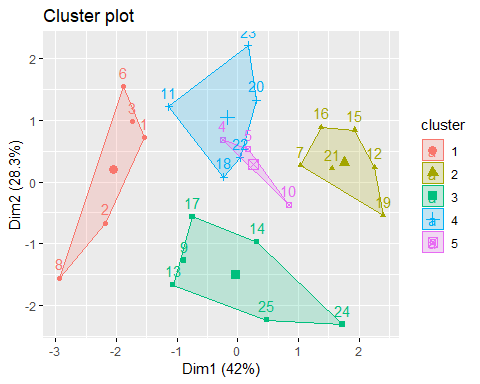


set.seed(123)  
fviz\_nbclust(wine2, kmeans, method = "silhouette" )



This time, there is no consensus as to the optimal number of clusters betweent he two types. One is clearly 5 while the other never comes back up so 10 could be assumed.

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

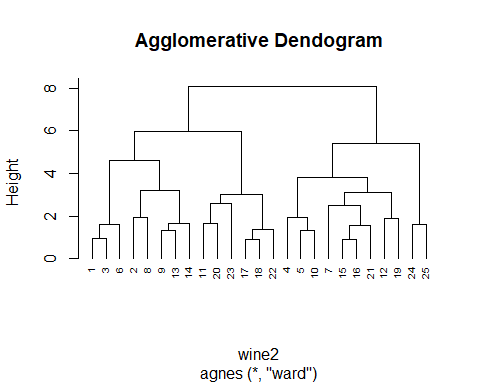


m = c("average", "single", "complete", "ward")  
names(m) = c("average", "single", "complete", "ward")  
  
ac = function(x) {  
 agnes(wine2, method = x)$ac  
}  
map\_dbl(m, ac)

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

Ward method gives the highest coeficient.

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



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

