# Alex Lugo

## Clustering

library(tidyverse)

## -- 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 1.0.2 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)  
library(factoextra)

## 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()  
## )

summary(trucks)

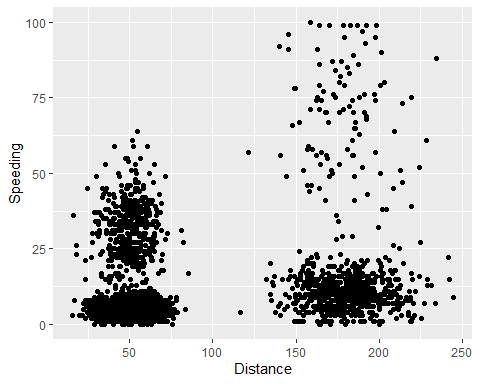
## Driver\_ID Distance Speeding   
## Min. :3.423e+09 Min. : 15.52 Min. : 0.00   
## 1st Qu.:3.423e+09 1st Qu.: 45.25 1st Qu.: 4.00   
## Median :3.423e+09 Median : 53.33 Median : 6.00   
## Mean :3.423e+09 Mean : 76.04 Mean : 10.72   
## 3rd Qu.:3.423e+09 3rd Qu.: 65.63 3rd Qu.: 9.00   
## Max. :3.423e+09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

str(trucks)

## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and '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 : num 28 25 27 22 25 10 20 8 34 19 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. Driver\_ID = col\_double(),  
## .. Distance = col\_double(),  
## .. Speeding = col\_double()  
## .. )

**TASK 1** The below plot between the Distance and Speeding variables allows us to compare the relationship between them. We can see that shorter distances typically have drivers who don’t speed 5 miles per hour over the speed limit during the trip. Drivers going longer distances will speed more of the time. We can also see that there appears to be about 4 natural clusters of drivers (short distance speeders and short distance “safe” drivers as well as long distance speeders and long distance “safe” drivers).

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

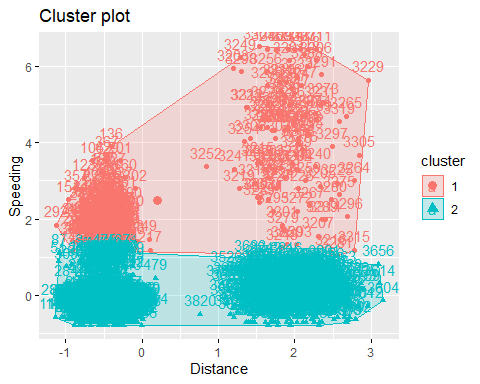


**TASK 2**

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

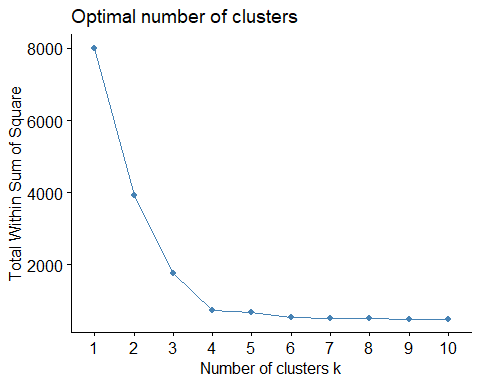
**TASK 3** The two clusters created below are separated by the percentage of drivers who speed over the speed limit vs the percentage of drivers who don’t speed over the limit. Distance does not factor into the two clusters.

set.seed(1234)  
cluster = kmeans(trucks2, 2)  
fviz\_cluster(cluster, trucks2)

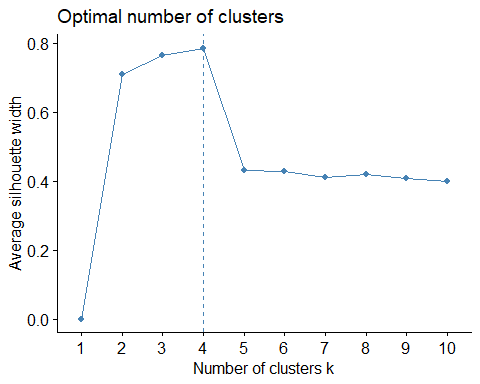


**TASK 4** The two methods below provide similar results in that 4 is the optimal number of clusters to use.

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

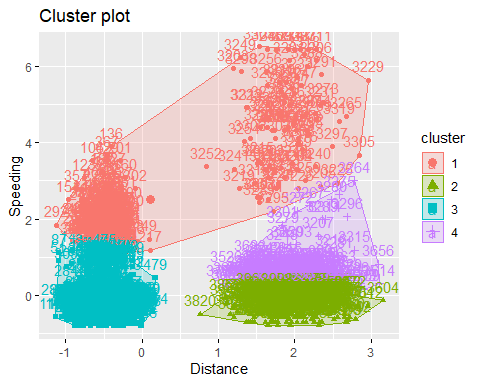


set.seed(123)  
fviz\_nbclust(trucks2, kmeans, method = "silhouette")



**TASK 5**

set.seed(1234)  
cluster2 = kmeans(trucks2, 4)  
fviz\_cluster(cluster2, trucks2)



**TASK 6** The clusters created in the previous task were unexpected by myself. It seems that there are high and low percentage of drivers who speed when the distance is short but there are 3 types of drivers when the distance is far, low percentage, medium percentage, and high percentage of speeders.

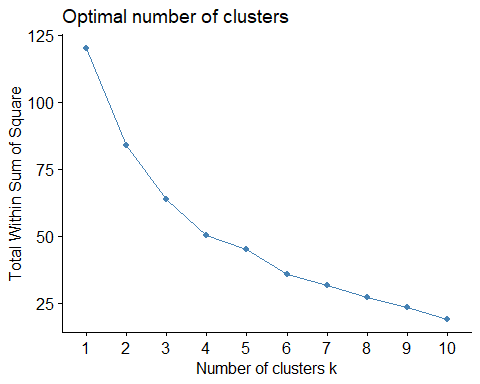
wine = read\_csv("wineprice.csv")

## 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()  
## )

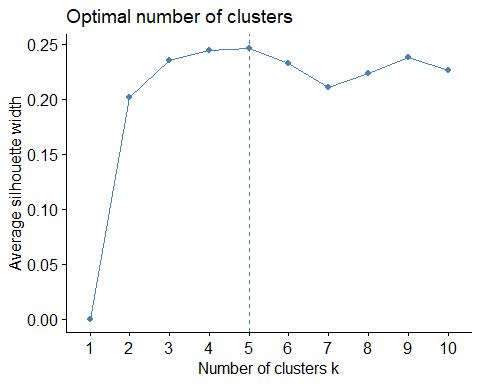
wine2 = wine %>% select("Price", "WinterRain", "AGST", "HarvestRain", "Age")  
wine2 = as.data.frame(scale(wine2))

**TASK 7** Both of the below methods are similar in that 5 clusters is the optimal amount.

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

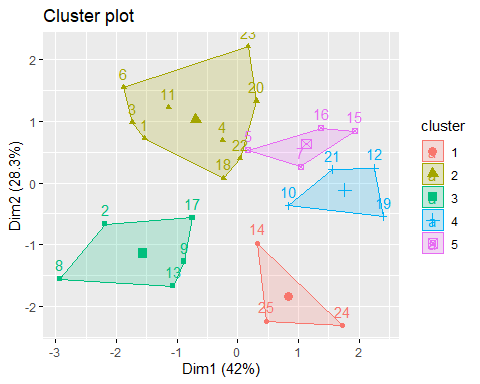


set.seed(123)  
fviz\_nbclust(wine2, kmeans, method = "silhouette")



**TASK 8**

set.seed(1234)  
cluster3 = kmeans(wine2, 5)  
fviz\_cluster(cluster3, wine2)

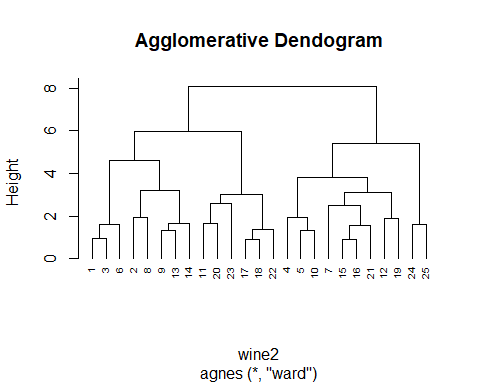


**TASK 9**

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 Dendogram")



**TASK 10**

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

