options(tidyverse.quiet=TRUE)  
library(tidyverse)  
library(cluster)  
library(dendextend)  
library(factoextra)

trucks=read\_csv("trucks.csv")

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

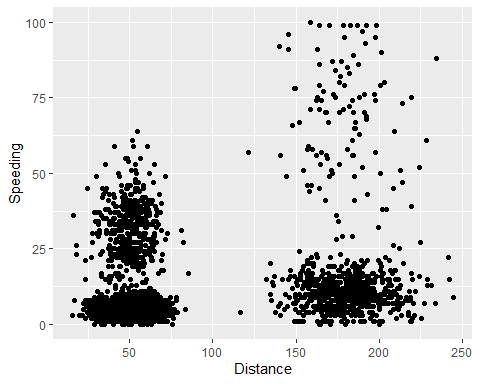
str(trucks)

## 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()  
## .. )

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

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



There does appear to be some natural clustering based on the ggplot visualization.

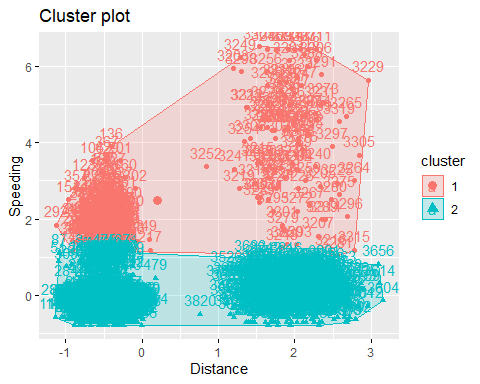
trucks2=as.data.frame(trucks%>%select(-Driver\_ID))

trucks2=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

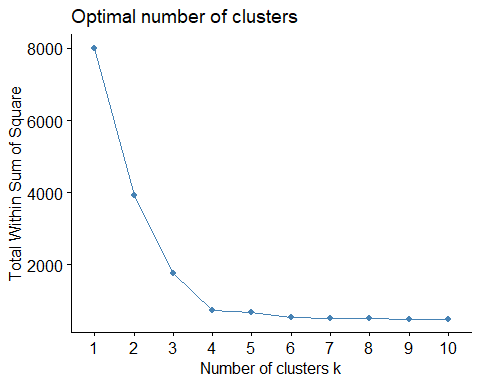
set.seed(1234)  
cluster1<-kmeans(trucks2,2)

fviz\_cluster(cluster1,trucks2)



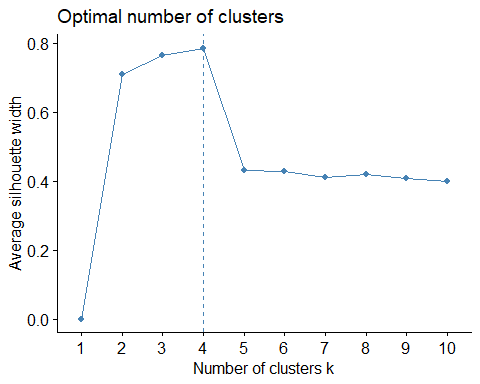
Two clusters looks a little forced. It looks more like it should be four.

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



Look like possibly four is the best per this method.

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

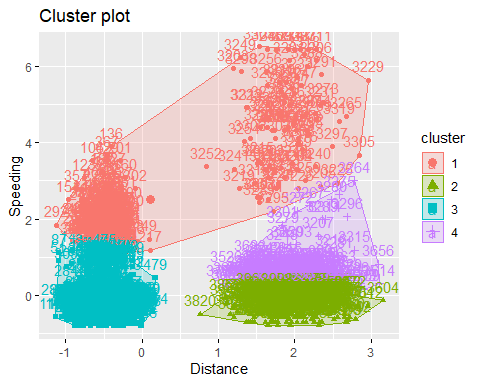


This one ALSO looks like four is optimal! There is a consensus between the two methods.

set.seed(1234)  
cluster2<-kmeans(trucks2,4)  
cluster2

## K-means clustering with 4 clusters of sizes 399, 463, 2900, 238  
##   
## Cluster means:  
## Distance Speeding  
## 1 0.1099584 2.5111614  
## 2 1.9532586 -0.2399259  
## 3 -0.4874490 -0.3463710  
## 4 1.9553359 0.4773451  
##   
## Clustering vector:  
## [1] 1 3 1 3 3 3 3 3 1 3 1 1 3 3 1 1 1 1 3 1 3 1 3 3 1 1 1 3 3 1 1 1 1 1 1 1 3  
## [38] 1 1 1 1 3 3 1 3 3 3 1 1 3 3 3 3 1 1 1 1 1 3 1 1 3 3 1 3 1 1 1 1 1 3 3 1 1  
## [75] 3 1 1 3 1 1 3 3 1 3 3 3 3 1 1 1 3 3 1 3 1 3 1 1 1 1 3 3 3 1 1 1 1 3 3 3 1  
## [112] 3 3 3 1 3 1 3 1 1 1 1 1 3 1 3 1 3 1 1 1 1 1 3 1 1 1 3 3 3 3 1 1 1 3 1 3 1  
## [149] 1 3 1 1 3 1 3 1 3 3 3 1 1 3 3 1 3 3 1 1 3 1 1 3 3 3 1 1 1 3 1 1 3 1 1 3 1  
## [186] 1 1 3 3 1 3 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 3 3 3 1 1 1 3 1 1 3 1 1 1 1  
## [223] 1 3 1 1 1 3 1 1 1 1 1 1 3 1 1 1 3 3 1 1 1 1 1 1 3 1 1 3 1 1 1 3 1 1 1 1 1  
## [260] 1 1 1 3 3 1 3 1 1 3 3 1 1 1 1 1 3 3 1 1 3 1 1 1 3 3 1 3 3 3 3 1 1 1 3 1 3  
## [297] 1 1 1 1 3 1 1 3 3 1 3 3 3 1 3 1 1 1 1 1 1 3 1 1 1 1 1 1 3 3 1 1 3 1 1 1 1  
## [334] 1 1 3 1 3 1 1 3 1 1 1 1 3 3 3 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 3 1 1 1 1  
## [371] 1 1 3 3 1 3 1 1 1 3 1 1 1 1 1 3 1 3 1 1 3 3 3 3 1 1 1 1 1 3 3 1 1 3 1 1 1  
## [408] 1 3 3 1 3 3 1 3 3 3 3 1 3 1 3 1 3 1 1 3 3 1 1 3 3 3 1 3 3 1 3 3 3 3 3 3 1  
## [445] 1 3 1 1 1 3 1 1 3 1 3 1 1 1 3 1 1 1 1 1 1 1 1 1 3 1 3 3 1 1 3 1 1 1 3 1 3  
## [482] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [519] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [556] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [593] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [630] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [667] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [704] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [741] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [778] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [815] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [852] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [889] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [926] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [963] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1000] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1037] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1074] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1111] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1148] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1185] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1222] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1259] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1296] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1333] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1370] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1407] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1444] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1481] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1518] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1555] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1592] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1629] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1666] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1703] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1740] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1777] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1814] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1851] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1888] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1925] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1962] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1999] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2036] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2073] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2110] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2147] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2184] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2221] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2258] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2295] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2332] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2369] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2406] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2443] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2480] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2517] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2554] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2591] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2628] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2665] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2702] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2739] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2776] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2813] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2850] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2887] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2924] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2961] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [2998] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [3035] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [3072] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [3109] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [3146] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [3183] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 4 1 1 1 1 1 1 2 1 4 1 1 1  
## [3220] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 1 1 1 1 1 4 1 4 1 1 1 1 1 1  
## [3257] 1 1 1 1 1 4 1 4 1 1 4 1 1 1 1 4 1 1 4 1 1 1 4 4 4 1 1 1 1 1 1 1 4 1 1 1 4  
## [3294] 1 1 4 1 1 1 1 4 1 1 1 1 1 1 1 1 1 1 1 1 4 4 1 1 1 1 1 2 2 4 4 2 2 2 4 4 4  
## [3331] 2 4 2 2 4 4 2 2 2 4 2 4 2 2 2 2 2 2 2 2 2 4 2 2 2 4 4 4 4 2 2 2 2 2 2 4 4  
## [3368] 4 2 2 4 4 4 2 2 2 2 4 2 2 2 2 4 2 2 4 4 2 2 2 2 2 4 2 4 2 4 2 2 2 2 2 2 2  
## [3405] 2 4 4 4 2 2 2 2 2 4 2 2 2 2 4 2 2 4 2 2 4 2 4 2 2 2 2 4 4 4 2 4 2 2 2 2 4  
## [3442] 2 4 4 2 4 2 2 2 4 2 2 4 2 2 2 2 2 2 2 4 2 4 2 2 4 2 2 2 2 4 2 4 2 4 2 4 2  
## [3479] 2 2 2 2 4 2 4 2 4 2 2 2 4 2 4 2 2 2 4 2 2 2 4 4 2 4 2 2 4 2 2 2 2 2 4 2 2  
## [3516] 4 4 2 2 2 2 2 4 2 4 4 2 4 4 2 2 4 2 2 2 2 2 4 4 4 4 2 2 4 2 2 2 2 2 4 2 2  
## [3553] 2 2 2 2 4 2 2 2 2 4 4 2 2 2 2 4 2 2 2 2 2 4 2 4 2 2 4 4 4 2 2 4 4 4 2 2 2  
## [3590] 2 2 2 4 2 2 4 2 2 2 2 2 4 4 2 2 2 2 2 2 2 2 2 4 4 2 2 4 2 4 2 4 4 2 2 4 4  
## [3627] 4 2 4 4 2 2 2 2 4 2 4 2 2 4 2 4 2 4 2 4 2 2 2 2 2 4 2 4 2 4 2 4 2 2 2 2 2  
## [3664] 2 2 2 2 4 4 4 2 2 2 2 2 2 2 4 2 2 4 2 4 4 4 2 2 4 2 4 2 4 4 2 2 4 2 4 4 2  
## [3701] 2 2 2 2 2 4 2 4 4 2 2 2 4 2 2 2 2 2 4 4 4 2 2 2 2 2 2 4 2 4 2 2 2 4 2 4 2  
## [3738] 4 2 2 2 4 2 2 2 2 2 2 2 2 2 2 4 2 2 4 2 2 4 2 2 2 4 2 2 2 4 2 2 4 2 4 2 2  
## [3775] 4 2 2 2 4 2 4 2 4 2 2 2 4 2 2 2 2 2 2 2 2 4 2 2 4 4 4 2 2 4 4 2 4 2 2 4 2  
## [3812] 2 2 2 4 4 2 4 2 2 2 2 2 4 2 4 2 2 2 4 2 2 4 4 4 4 2 2 2 2 2 4 2 2 2 2 2 2  
## [3849] 4 2 2 2 4 2 2 2 2 2 4 2 4 2 2 2 2 2 4 4 2 2 4 2 2 2 2 2 2 2 4 4 2 2 2 4 2  
## [3886] 2 2 2 2 2 2 2 2 2 4 2 2 4 4 2 2 2 2 2 2 2 2 4 2 4 4 2 4 2 4 2 2 4 2 4 2 2  
## [3923] 2 2 2 2 2 2 4 2 4 2 4 2 4 4 2 2 4 2 2 2 2 4 4 2 2 2 2 2 2 2 2 2 4 2 2 2 2  
## [3960] 2 2 4 4 4 4 2 4 2 4 2 4 2 4 2 4 4 2 2 2 2 2 2 4 2 2 2 2 2 2 4 4 2 4 4 4 2  
## [3997] 2 2 2 2  
##   
## Within cluster sum of squares by cluster:  
## [1] 1135.73431 79.87293 385.95989 81.27803  
## (between\_SS / total\_SS = 79.0 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

fviz\_cluster(cluster2,trucks2)



That was NOT how I expected this to look. Now it looks almost like maybe 5 would have been better? Cluster 1 (the salmon colored one) seems a little forced to me.

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()  
## )

str(wine)

## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and 'data.frame': 25 obs. of 7 variables:  
## $ Year : num 1952 1953 1955 1957 1958 ...  
## $ Price : num 7.5 8.04 7.69 6.98 6.78 ...  
## $ WinterRain : num 600 690 502 420 582 485 763 830 697 608 ...  
## $ AGST : num 17.1 16.7 17.1 16.1 16.4 ...  
## $ HarvestRain: num 160 80 130 110 187 187 290 38 52 155 ...  
## $ Age : num 31 30 28 26 25 24 23 22 21 20 ...  
## $ FrancePop : num 43184 43495 44218 45152 45654 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. Year = col\_double(),  
## .. Price = col\_double(),  
## .. WinterRain = col\_double(),  
## .. AGST = col\_double(),  
## .. HarvestRain = col\_double(),  
## .. Age = col\_double(),  
## .. FrancePop = col\_double()  
## .. )

summary(wine)

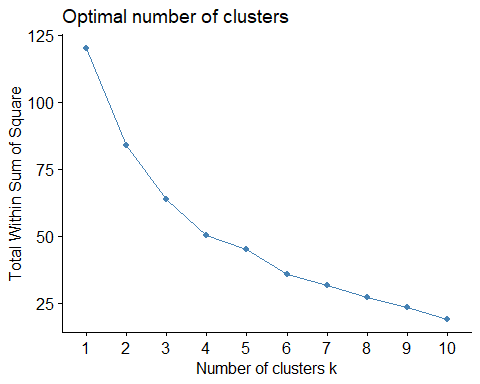
## Year Price WinterRain AGST HarvestRain   
## Min. :1952 Min. :6.205 Min. :376.0 Min. :14.98 Min. : 38.0   
## 1st Qu.:1960 1st Qu.:6.519 1st Qu.:536.0 1st Qu.:16.20 1st Qu.: 89.0   
## Median :1966 Median :7.121 Median :600.0 Median :16.53 Median :130.0   
## Mean :1966 Mean :7.067 Mean :605.3 Mean :16.51 Mean :148.6   
## 3rd Qu.:1972 3rd Qu.:7.495 3rd Qu.:697.0 3rd Qu.:17.07 3rd Qu.:187.0   
## Max. :1978 Max. :8.494 Max. :830.0 Max. :17.65 Max. :292.0   
## Age FrancePop   
## Min. : 5.0 Min. :43184   
## 1st Qu.:11.0 1st Qu.:46584   
## Median :17.0 Median :50255   
## Mean :17.2 Mean :49694   
## 3rd Qu.:23.0 3rd Qu.:52894   
## Max. :31.0 Max. :54602

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

wine2=scale(wine2)  
summary(wine2)

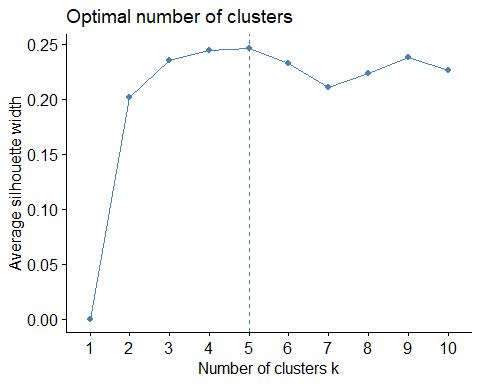
## Price WinterRain AGST HarvestRain   
## Min. :-1.32596 Min. :-1.73332 Min. :-2.25947 Min. :-1.4856   
## 1st Qu.:-0.84329 1st Qu.:-0.52375 1st Qu.:-0.45801 1st Qu.:-0.8003   
## Median : 0.08284 Median :-0.03992 Median : 0.03548 Median :-0.2494   
## Mean : 0.00000 Mean : 0.00000 Mean : 0.00000 Mean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

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



Somewhere around five or six is what it looks like here, I think.

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

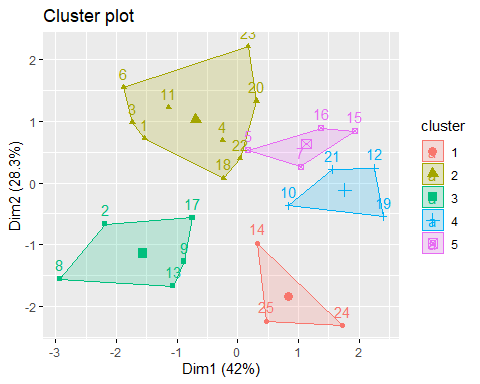


This method identifies five, so there is consensus.

set.seed(1234)  
cluster2<-kmeans(wine2,5)  
cluster2

## K-means clustering with 5 clusters of sizes 3, 9, 5, 4, 4  
##   
## Cluster means:  
## Price WinterRain AGST HarvestRain Age  
## 1 -0.4700981 1.2150172 -0.9351085 -0.84959852 -1.0660444  
## 2 0.3739208 -0.9420566 0.8224590 -0.00155276 0.1473395  
## 3 1.2047467 0.9549587 0.3020210 -1.06907515 0.4420184  
## 4 -1.0784329 -0.1911127 -1.3586990 0.57028091 -0.3510146  
## 5 -0.9162487 0.2057788 -0.1680286 1.40675562 0.2665111  
##   
## Clustering vector:  
## [1] 2 3 2 2 5 2 5 3 3 4 2 4 3 1 5 5 3 2 4 2 4 2 2 1 1  
##   
## Within cluster sum of squares by cluster:  
## [1] 3.498321 24.609221 7.991457 5.549316 4.752018  
## (between\_SS / total\_SS = 61.3 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

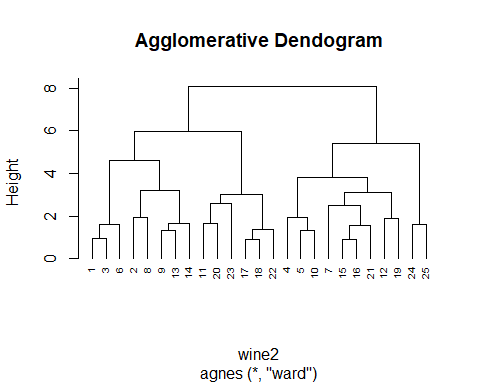
fviz\_cluster(cluster2,wine2)



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

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="Diversive Dendogram")

