## Module 6 Assignment 1 Clustering

libraries

options(tidyverse.quiet=TRUE)  
library(tidyverse)

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

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

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

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

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

library(cluster) #algorithms for clustering

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

library(factoextra) #visualization

## 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

library(ggplot2)

Read in Data

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

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

str(trucks)

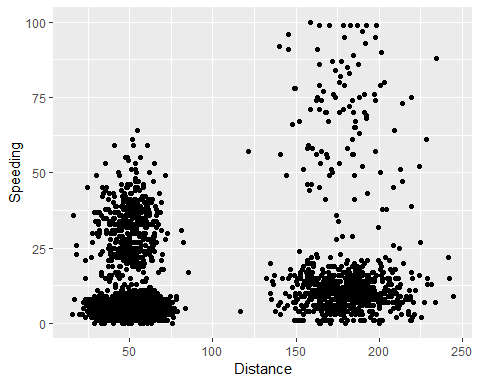
## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and 'data.frame': 4000 obs. of 3 variables:  
## $ Driver\_ID: num 3.42e+09 3.42e+09 3.42e+09 3.42e+09 3.42e+09 ...  
## $ Distance : num 71.2 52.5 64.5 55.7 54.6 ...  
## $ Speeding : num 28 25 27 22 25 10 20 8 34 19 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. 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

Task 1: Plot relationship between Distance and Speeding

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



In analyzing the scatterplot, there does appear to be a natural clustering of drivers. This can be seen be two areas on the graph that are heavily populated as compared to the rest of the graph.

Task 2: Create a new data frame (trucks2)

trucks2 = trucks %>% select("Distance","Speeding")  
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

str(trucks2)

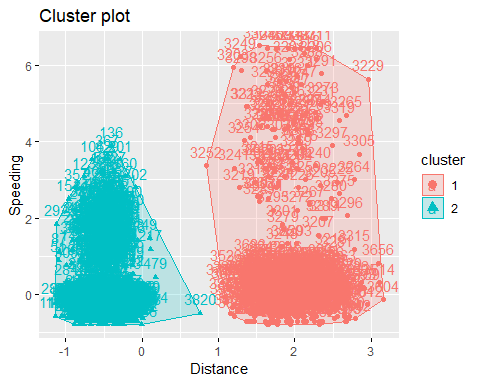
## 'data.frame': 4000 obs. of 2 variables:  
## $ Distance: num -0.0898 -0.4397 -0.2151 -0.3806 -0.4014 ...  
## $ Speeding: num 1.26 1.042 1.188 0.823 1.042 ...

Task 3: Use k-Means clustering with two clusters (k=2)

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

visualize the clusters

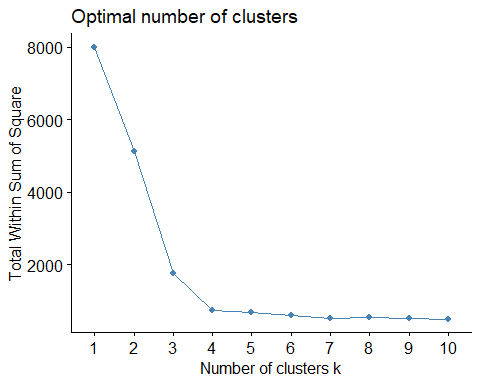
fviz\_cluster(clusters1, trucks2)



The clusters in the k-Means clustering look very similar to those found in the scatterplot done earlier. There appear to be two areas on the graph that are very heavily populated.

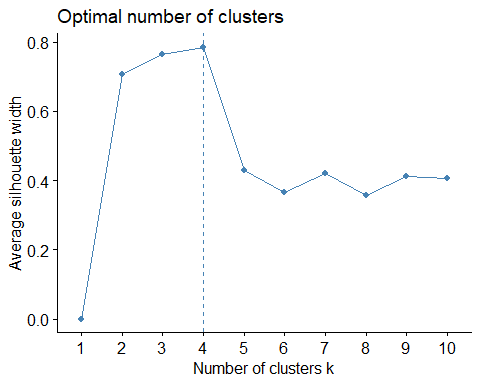
Task 4: Identify the optimal number of clusters.

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



Method 2

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



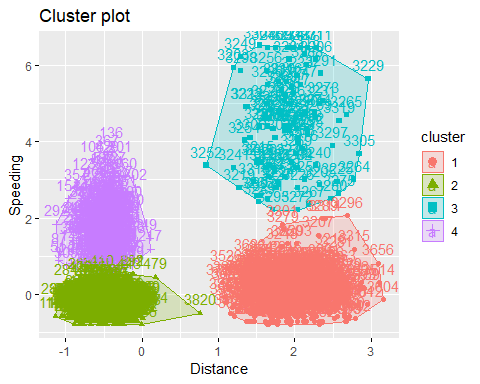
There is a consesus between the two methods for the optimal number of clusters to be used, both showing 4 clusters.

Task 5: Use optimal number of clusters to create k-Means clusters.

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

Visualize the clusters.

fviz\_cluster(clusters2, trucks2)



Task 6: desribe the clusters from task 5 In analyzing the clusters created, it appears that there are 4 distinct clusters of similar drivers. There are two clusters of people who traveled short distances, one group that didn’t speed found in the yellow group and one that did speed in the purple group. Likewise, there are two clusters for people who traveled longer distances. Again one group that didn’t speed (pink) and a second cluster of people who did speed (blue).

Read in data

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

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

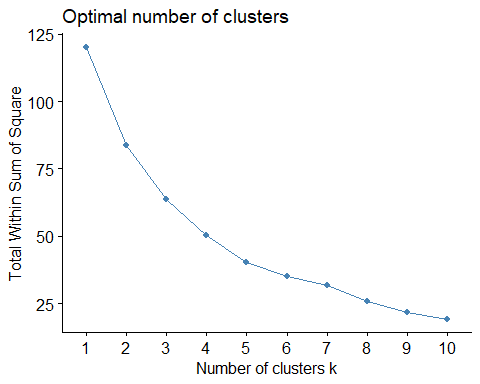
## WinterRain AGST HarvestRain Age   
## Min. :-1.73332 Min. :-2.25947 Min. :-1.4856 Min. :-1.586   
## 1st Qu.:-0.52375 1st Qu.:-0.45801 1st Qu.:-0.8003 1st Qu.:-0.806   
## Median :-0.03992 Median : 0.03548 Median :-0.2494 Median :-0.026   
## Mean : 0.00000 Mean : 0.00000 Mean : 0.0000 Mean : 0.000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

str(wine2)

## 'data.frame': 25 obs. of 5 variables:  
## $ WinterRain : num -0.0399 0.6405 -0.7808 -1.4007 -0.176 ...  
## $ AGST : num 0.899 0.332 0.949 -0.557 -0.137 ...  
## $ HarvestRain: num 0.154 -0.921 -0.249 -0.518 0.517 ...  
## $ Age : num 1.79 1.66 1.4 1.14 1.01 ...  
## $ Price : num 0.658 1.495 0.951 -0.127 -0.446 ...

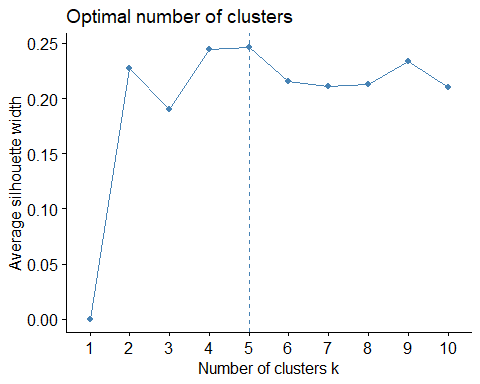
Task 7: determine optimal number of k-Means clusters

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



method 2

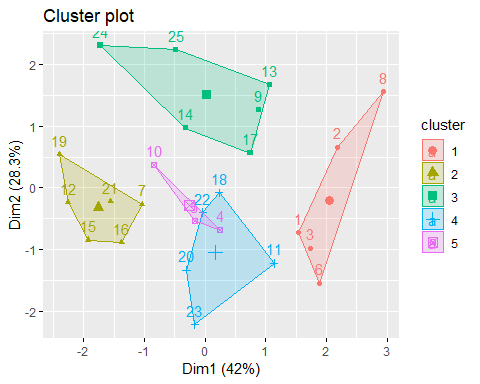
set.seed(123)  
fviz\_nbclust(wine2, kmeans, method = "silhouette")



There is not a perfect consensus on the optimal number of clusters between the two methods. It seems as if 5 clusters is a good number to go with.

Task 8: Use optimal number of clusters to create k-Means clusters

set.seed(1234)  
clusters2 <- kmeans(wine2, 5)  
fviz\_cluster(clusters2, wine2)



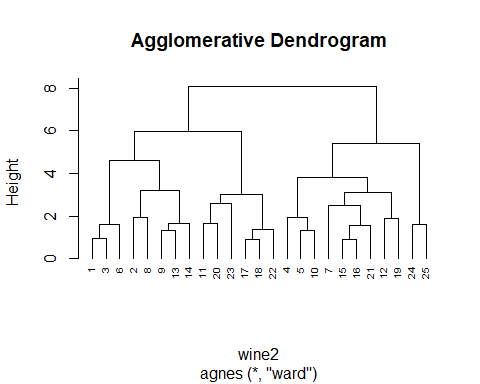
Task 9: Agglomerative clustering

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 has the highest agglomerative coefficiant. This will be used to develop the clusters.

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



Task 10: divisive clustering

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

