# Tatyana Zhabina

# Clustering

### Libraries: For this assignment you will need the following libraries: tidyverse, cluster, factoextend, and dendextend.

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

### Before beginning the assignment tasks, you should read-in the data for the assignment into a data frame called “trucks”. In this dataset, Driver\_ID is a unique identifier for each delivery driver, Distance is the average mileage driven by each driver in a day, and Speeding is the percentage of the driver’s time in which he is driving at least 5 miles per hour over the speed limit.

### Task 1: Plot the relationship between Distance and Speeding.

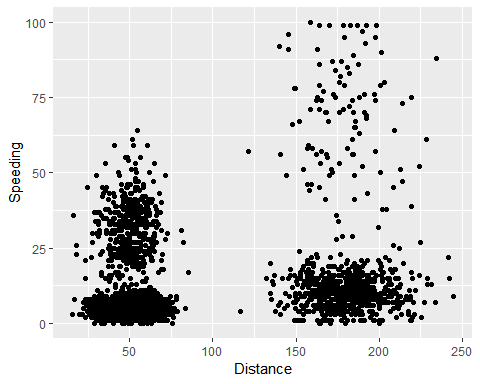
trucks <- read\_csv("C:/Users/Tanushka/Desktop/MSBA/BAN - 502/Week 6/Assignment 6/trucks.csv")

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

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()



### Describe this relationship. Does there appear to be any natural clustering of drivers?

### It looks like there are four natural clusters.

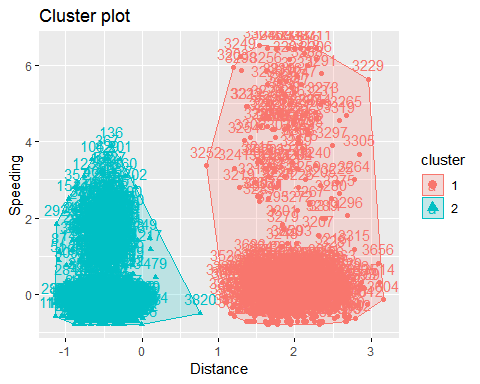
### Task 2: Create a new data frame (called trucks2) that excludes the Driver\_ID variable and includes scaled versions of the Distance and Speeding variables. NOTE: Wrap the scale(trucks2) command in an as.data.frame command to ensure that the resulting object is a data frame. By default, scale converts data frames to lists.

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

## 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

### Task 3 Use k-Means clustering with two clusters (k=2) to cluster the trucks2 data frame. Use a random number seed of 1234. Visualize the clusters using the fviz\_cluster function. Comment on the clusters.

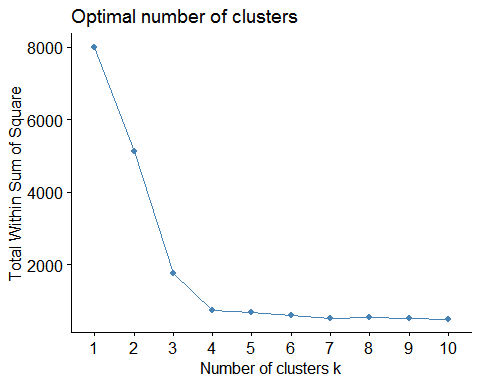
set.seed(1234)  
clusters1 <- kmeans(trucks2\_scaled, 2)  
  
fviz\_cluster(clusters1, trucks2\_scaled)



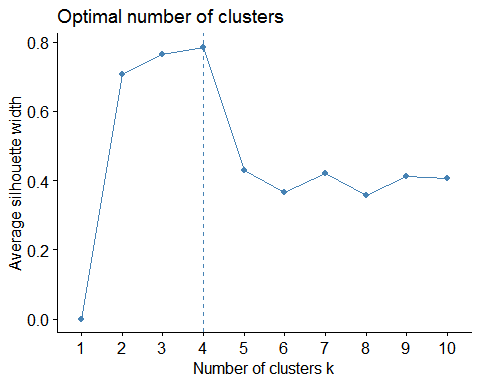
### On the graph it is noticeble that there are more than two clusters.

### Task 4: Use the two methods from the k-Means lecture to identify the optimal number of clusters. Use a random number seed of 123 for these methods. Is there consensus between these two methods as the optimal number of clusters?

set.seed(123)  
fviz\_nbclust(trucks2\_scaled, kmeans, method = "wss") #minimize within-cluster variation



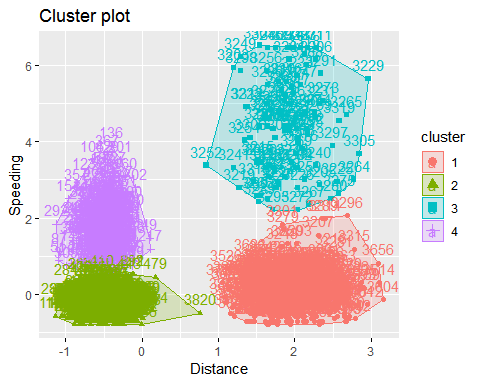
set.seed(123)  
fviz\_nbclust(trucks2\_scaled, kmeans, method = "silhouette") #maximize how well points sit in their clusters



### Yes, there is a consensus between these two methods that the optimal number of clusters is 4.

### Task 5: Use the optimal number of clusters that you identified in Task 4 to create k-Means clusters. Use a random number seed of 1234. Use the fviz\_cluster function to visualize the clusters.

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



### I think this graph looks very good. It looks like this dataset has a perfect cluster breakdown.

### Task 6: In words, how would you characterize the clusters you created in Task 5? Before starting Task 7, read in the “wineprice.csv” file into a data frame called wine. This is a small dataset containing wine characteristics and the price of wine at auction. WinterRain refers to the amount of rain received in winter, AGST refers to the average growing season temperature, HarvestRain refers to the amount of rain received in the harvest season, Age refers to the age of the wine when sold at auction, and FrancePop refers to the population of France at the time the wine was sold at auction.

wine <- read\_csv("C:/Users/Tanushka/Desktop/MSBA/BAN - 502/Week 6/Assignment 6/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()  
## )

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

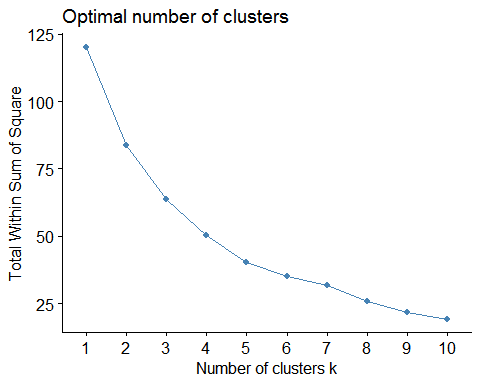
### Create a new data frame called wine2 that removes the Year and FrancePop variables and scales the other variables.

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

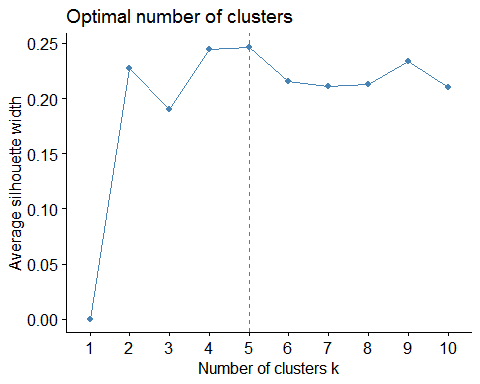
## Price WinterRain AGST   
## Min. :-1.32596 Min. :-1.73332 Min. :-2.25947   
## 1st Qu.:-0.84329 1st Qu.:-0.52375 1st Qu.:-0.45801   
## Median : 0.08284 Median :-0.03992 Median : 0.03548   
## Mean : 0.00000 Mean : 0.00000 Mean : 0.00000   
## 3rd Qu.: 0.65777 3rd Qu.: 0.69339 3rd Qu.: 0.82524   
## 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

### Task 7: Use the two methods from Task 4 to determine the optimal number of k-Means clusters for this data.

set.seed(123)  
fviz\_nbclust(wine2\_scaled, kmeans, method = "wss") #minimize within-cluster variation



set.seed(123)  
fviz\_nbclust(wine2\_scaled, kmeans, method = "silhouette") #maximize how well points sit in their clusters

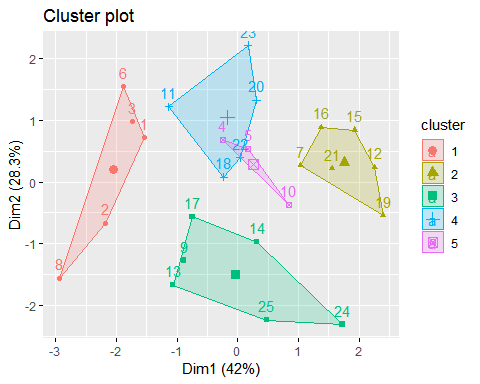


### Use a random number seed of 123. Is there a consensus between these two methods as the optimal number of clusters?

### In this dataset, the consensus is not clear as it was in the trucks dataset. That is hard to see, in the first method it looks like 5 or 9 can be an optimal number. However, in the second method it is clearly 5.

### Task 8: Use the optimal number of clusters that you identified in Task 4 to create k-Means clusters. Use a random number seed of 1234. Use the fviz\_cluster function to visualize the clusters.

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



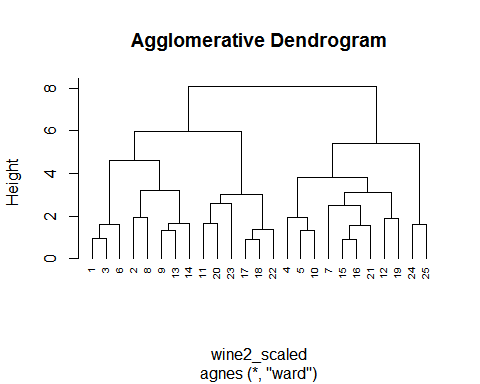
### Task 9: Use agglomerative clustering to develop a dendogram for the scaled wine data. Follow the same process from the lecture where we used a custom function to identify the distance metric that maximizes the “agglomerative coefficient”. Plot the dendogram.

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

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

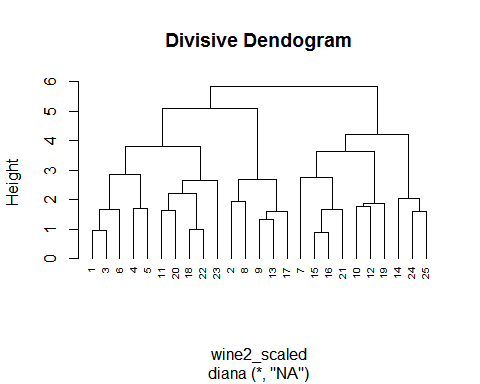
### Ward’s is the highest. Use this to develop clusters.

hc = agnes(wine2\_scaled, method = "ward") #change ward to other method if desired  
pltree(hc, cex = 0.6, hang = -1, main = "Agglomerative Dendrogram")

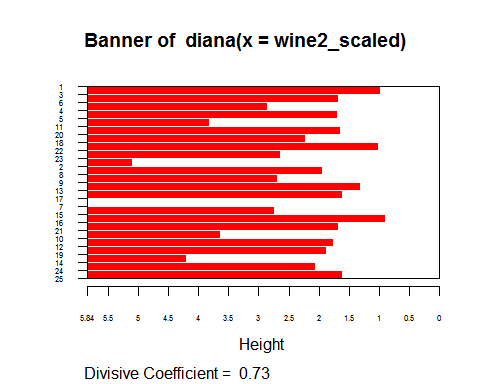


#### Task 10: Repeat Task 9, but with divisive clustering.

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



plot(hc2, cex.axis= 0.5)



rect.hclust(hc2, k = 5, border = 2:6) #border selects colors for the boxes

