# Clustering Assignment

## BAN502 Predictive Analytics

### Susan Wiggins

The libraries needed for this assignment:

library(tidyverse)

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

## v ggplot2 3.1.1 v purrr 0.3.2  
## v tibble 2.1.1 v dplyr 0.8.1  
## v tidyr 0.8.3 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(dendextend)

##   
## ---------------------  
## Welcome to dendextend version 1.12.0  
## 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(factoextra) #visualization

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

library(readr)  
trucks <- read\_csv("trucks.csv")

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

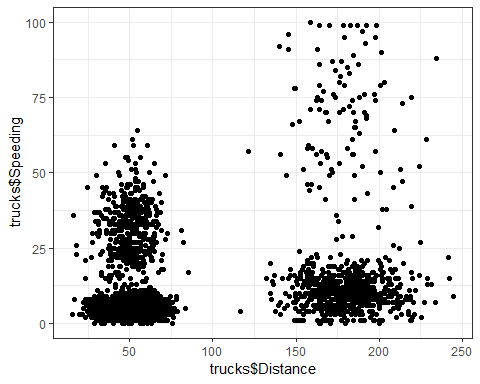
View(trucks)

### Task 1

Plot the relationship between Distance and Speeding. Describe this relationship. Does there appear to be any natural clustering of drivers?

Yes, there seems to be a natural clustering around the distance of 150 to 250 and at 25 to 75 as well as Speeds 25 to 50.

ggplot(trucks,aes(trucks$Distance, trucks$Speeding)) + geom\_point()+ theme\_bw()



### 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(scale(trucks2))

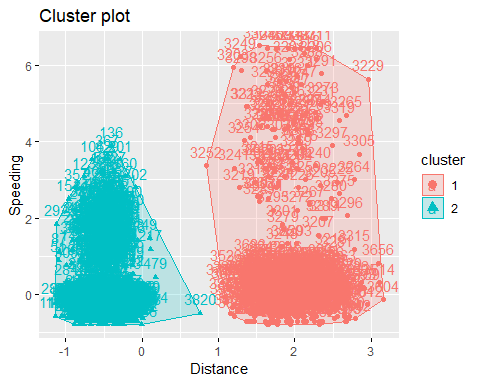
### Task 3

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)  
clustersT1 <- kmeans(trucks2, 2)

Visualization of Clustering:

fviz\_cluster(clustersT1, trucks2)



Task 3 Continued

Comments on Clustering:

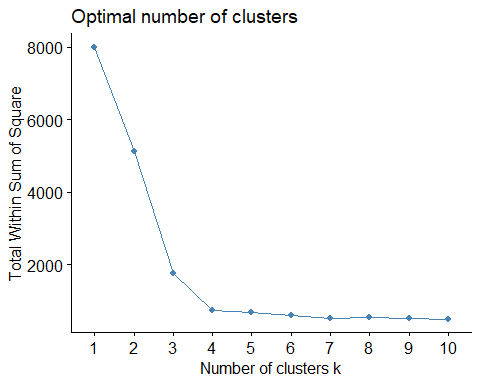
Cluster one tends to be a large group of data between 1 to 3 on the y-axis on the Distance scale, cluster two looks as though there are two clusters of large grouping of data ranging from below -1 and 0 on the Distance scale.Based on how the graphical analysis looks k might need to be increased in order to address the presence of possibly 3 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?

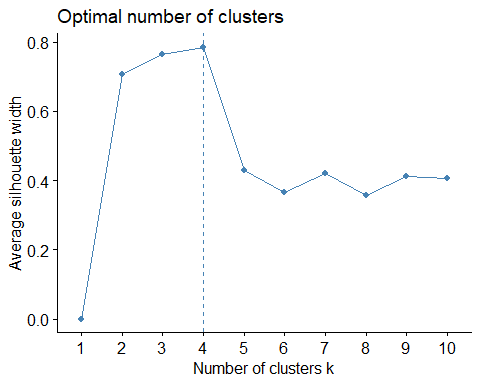
Visually identify optimal number of clusters

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



Another method

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



Task 4 Continued:

Is there consensus between these two methods as the optimal number of clusters?

Yes, there is consensus between these two methods a bend occurs at around three and then straightens at the number of cluster of 4 which indicates the optimal cluster for k. In addition, the maximum silhouette of clusters is indicated in the second method as k = 4.

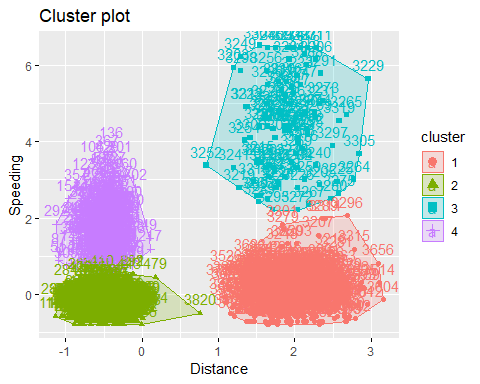
### Task 5

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)  
clustersT2 <- kmeans(trucks2, 4)

Visualization of Clustering:

fviz\_cluster(clustersT2, trucks2)



### Task 6

In words, how would you characterize the clusters you created in Task 5?

There seems to be a little overlapping in clusters for 2 and 4, whereas no overlapping occurs within the other clusters. In addition, two clusters are formulated at -1 to .75 and the other two cluster appear on the same axis at 1 to 3 for Distance. By looking at these clusters you can go back and look into the data and see why the clusters are occurring at these different positions in the graph.

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

Reading in the data wineprice:

library(readr)  
wineprice <- 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()  
## )

View(wineprice)

Creation of new data frame:

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

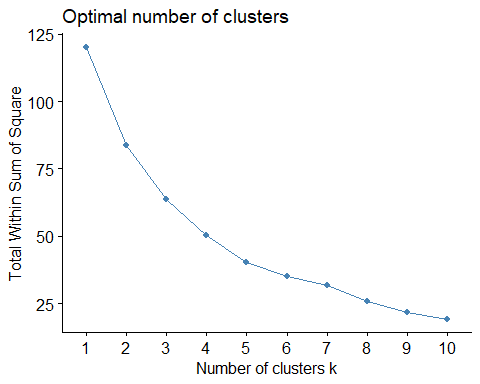
### Task 7

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

set.seed(123)  
clustersW2 <-kmeans(wine2\_scaled, 2)

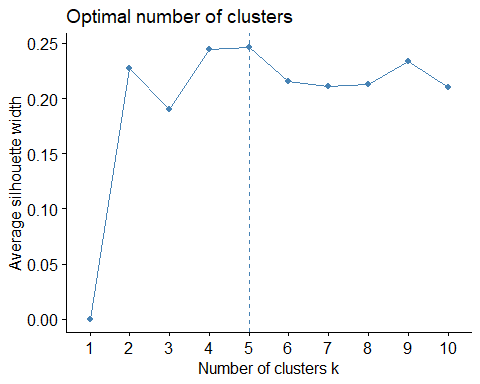
Visually identify optimal number of clusters

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



Another Method:

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



Task 7 Continued:

Comments:

Yes, there is consensus between these two methods a bend occurs at around five and then straightens at the number of cluster of 5 which indicates the optimal cluster for k. In addition, the maximum silhouette of clusters is indicated in the second method as k = 5.

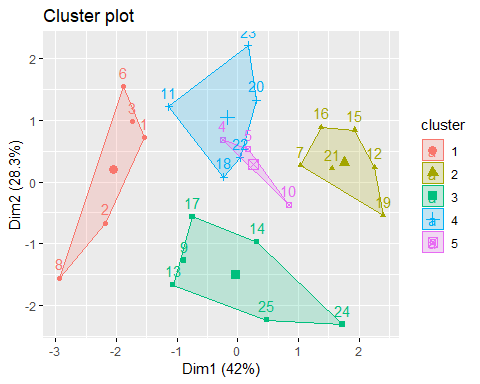
### Task 8

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)  
clustersW3 <- kmeans(wine2\_scaled, 5)

Visualize the Cluster Analysis:

fviz\_cluster(clustersW3, 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.

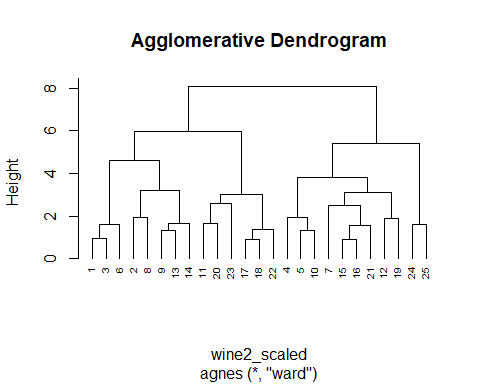
Agglomerative clustering

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 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")

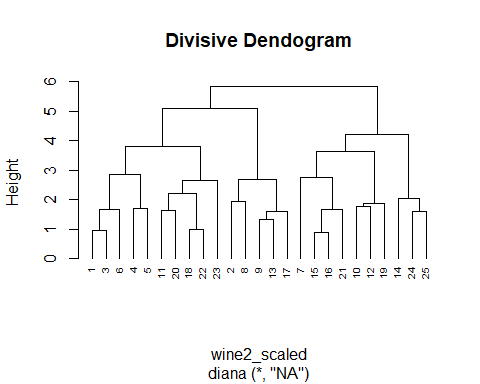


### Task 10

Repeat Task 9, but with divisive clustering.

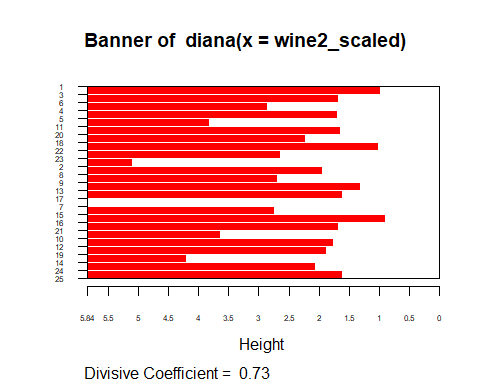
Divisive clustering

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



How do we actually use dendograms?

plot(hc2, cex.axis= 0.5)



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

