## Daniel Sanchez

## BAN 502

## Dr. Stephen Hill

## Module 6 Assignment

## June 22nd, 2020

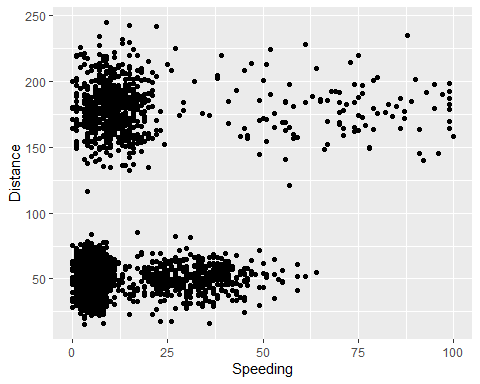
library(tidyverse)  
library(cluster)  
library(factoextra)  
library(dendextend)

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

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

**Task 1: Plot the relationship between Distance and Speeding. Describe this relationship. Does there appear to be any natural clustering of drivers?**

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



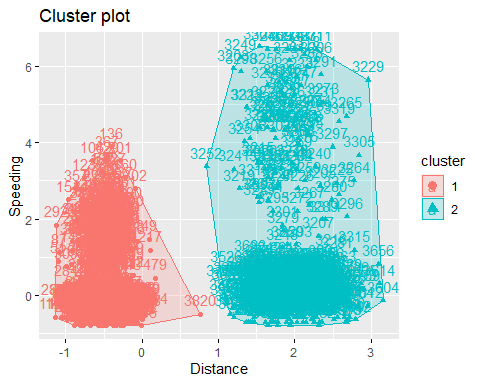
In terms of distance alone, there is a cluster of drivers traveling between 25 and 75 miles. There is another cluster of drivers traveling between approximately 135 and 225 miles. Combining distance and speed, there is a cluster of 0-12% time speeding with less than 75 miles driven. Another cluster exists between 135 and 225 miles driven, with 0-20% time speeding. The longer distances driven have a higher occurrence of speeding.

**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(c(-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 64. Visualize the clusters using the fviz\_cluster function. Comment on the clusters.**

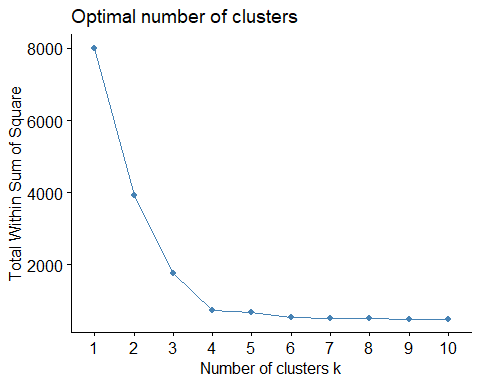
set.seed(64)  
clusters1 <- kmeans(trucks2, 2)  
fviz\_cluster(clusters1, trucks2)



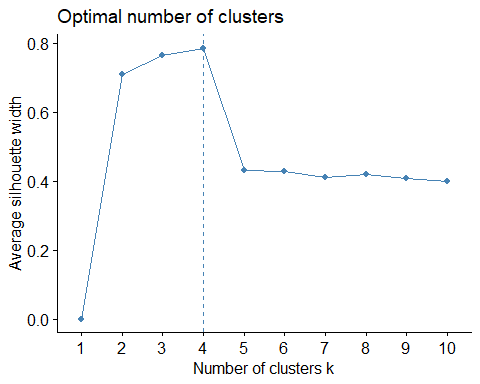
The two clusters are similar to the simple analysis from Task 1. The clusters are separated primarily by distance.

**Task 4: Use the two methods from the k-Means lecture to identify the optimal number of clusters. Use a random number seed of 64 for these methods. Is there consensus between these two methods as the optimal number of clusters?**

set.seed(64)  
fviz\_nbclust(trucks2, kmeans, method = "wss") #minimize within-cluster variation



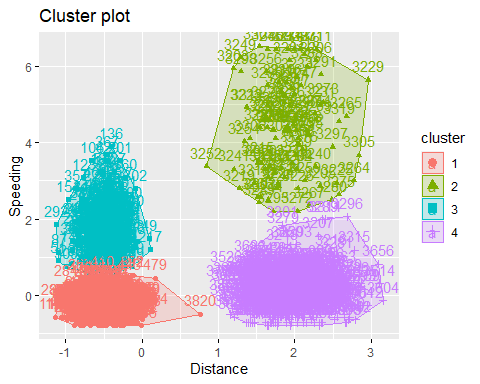
set.seed(64)  
fviz\_nbclust(trucks2, kmeans, method = "silhouette") #maximize how well points sit in their clusters



Both of these methods indicate that four clusters would provide a more optimal representation of the data.

**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 64. Use the fviz\_cluster function to visualize the clusters.**

set.seed(64)  
clusters2 <- kmeans(trucks2, 4)  
fviz\_cluster(clusters2, trucks2)



**Task 6: In words, how would you characterize the clusters you created in Task 5?**

The four clusters are represented by two clusters for range of distance. There is one cluster for short distance/low speeding, short distance/high speeding, long distance/low speeding, and long distance/high speeding.

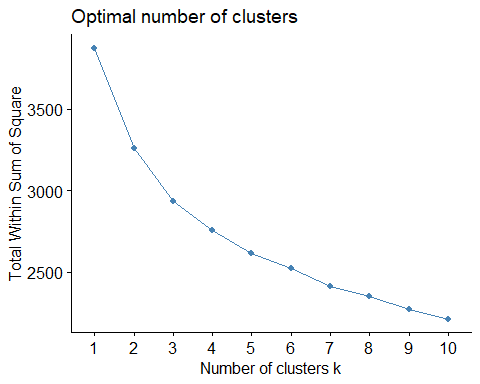
**Task 7: Create a new data frame called “bball2” that excludes team name and scales the variables. Then use the two methods from Task 4 to determine the optimal number of k-Means clusters for this data. Use a random number seed of 123. Is there consensus between these two methods as the optimal number of clusters?**

bball=read\_csv("kenpom20.csv")

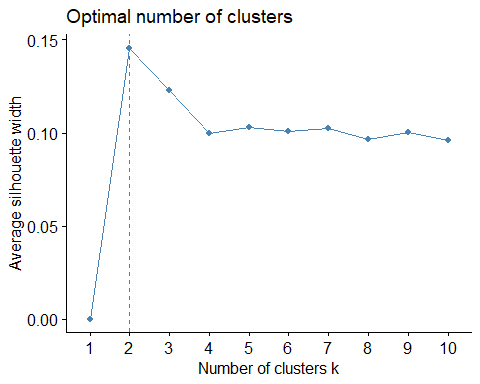
## Parsed with column specification:  
## cols(  
## TeamName = col\_character(),  
## AdjTempo = col\_double(),  
## AdjOE = col\_double(),  
## AdjDE = col\_double(),  
## eFGPct = col\_double(),  
## TOPct = col\_double(),  
## ORPct = col\_double(),  
## FTRate = col\_double(),  
## eFGPctD = col\_double(),  
## TOPctD = col\_double(),  
## ORPctD = col\_double(),  
## FTRateD = col\_double()  
## )

bball2 = bball %>% select(c(-TeamName))  
bball2 = as.data.frame(scale(bball2))

set.seed(123)  
fviz\_nbclust(bball2, kmeans, method = "wss") #minimize within-cluster variation



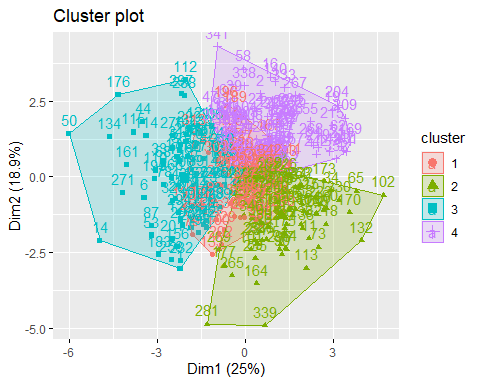
set.seed(123)  
fviz\_nbclust(bball2, kmeans, method = "silhouette") #maximize how well points sit in their clusters



The Silhouette method makes it easier to see a more clear recommendation for the optimal number of two clusters. This is also reflected in the WSS method, in which there is a steep curve from 1 to 2 cluster. However, the gradual downward trend using WSS does not provide a clear distinction between 2 to 3 or beyond.

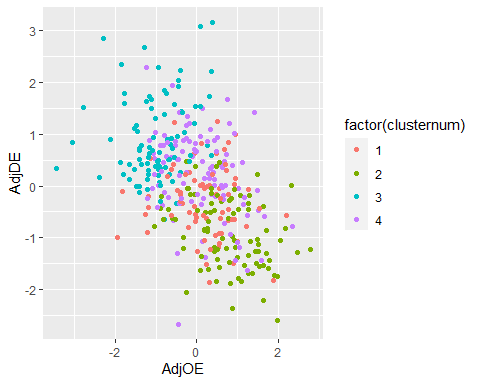
**Task 8: Create k-Means clusters with a k of 4. Use a random number seed of 1234. Use the fviz\_cluster function to visualize the clusters.**

set.seed(1234)  
clusters3 <- kmeans(bball2, 4)  
fviz\_cluster(clusters3, bball2)



**Task 9: Extract the cluster number from the k-means algorithm and attach as a new column to your “bball” data frame. Use the code as shown below, but replace XXX with the name of your k-means object. Plot “AdjOE” vs. “AdjDE” (use a scatterplot) and assign point color based on “clusternum”. What patterns do you see?**

bball2 = bball2 %>% mutate(clusternum = clusters3$cluster)  
ggplot(bball2,aes(x=AdjOE,y=AdjDE,color=factor(clusternum)))+  
 geom\_point()



The clusters here seem to represent outcomes of games. Cluster 1 in red shows close games, with points scored and allowed being close to each other. Cluster 2 in green represents likely wins, with more points scored than allowed. Cluster 3 in blue is likely to represent losses, with more points allowed than scored. Meanwhile, Cluster 4 in purple reflect high-scoring games on both sides.