# Module 6 Assignment 1

## Towery, Chip

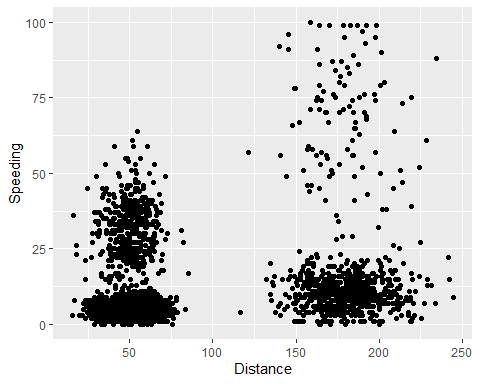
Libraries and Data

library(tidyverse)  
library(cluster)  
library(factoextra)  
library(dendextend)  
  
trucks <- read\_csv("trucks.csv")

### 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(Distance, Speeding)) +  
 geom\_point()

 There appears to be some natural clustering. Intuitively, it seems those who travel less distance each day speed less on average while longer distances per day may speed more on average. There also appears to be two groups for each category (totaling 4 groups): Short Distance per Day, Long Distance Per Day, Less/No Speeding, More/Always Speeding.

### Task 2:

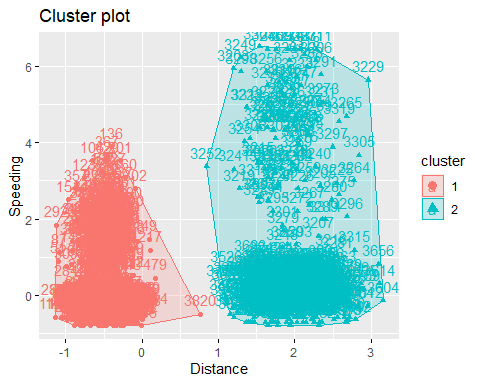
Create a new data frame (called trucks2) that excludes the Driver\_ID variable and includes scaled versions of the Distance and Speeding variables.

trucks2 <- trucks %>%  
 select(-Driver\_ID)  
trucks2 <- as.data.frame(scale(trucks2))  
#summary(trucks2) # check for 0 mean and 1 sd

### 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.

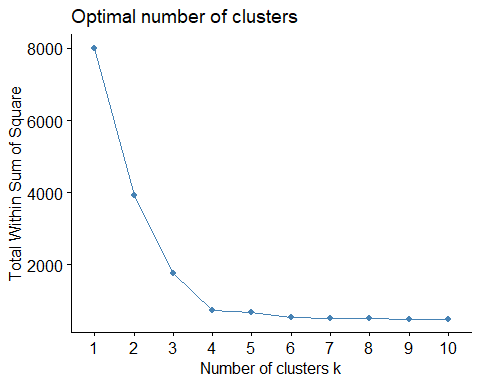
set.seed(64)  
  
cluster1 <- kmeans(trucks2, 2)  
fviz\_cluster(cluster1, trucks2)

 Logically, this clustering by 2 groups could have gone one of two ways. Either by distance or by speeding average. In this case, the clusters show a distinguished separation betwene shorter and longer distances traveled per day.

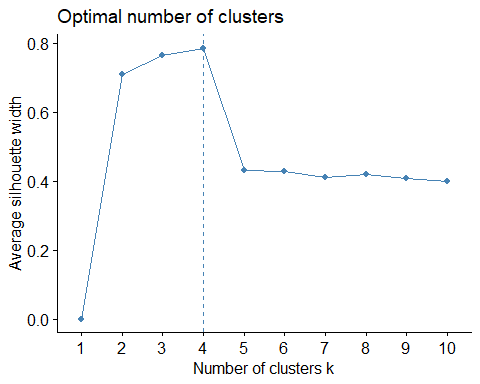
### Task 4:

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



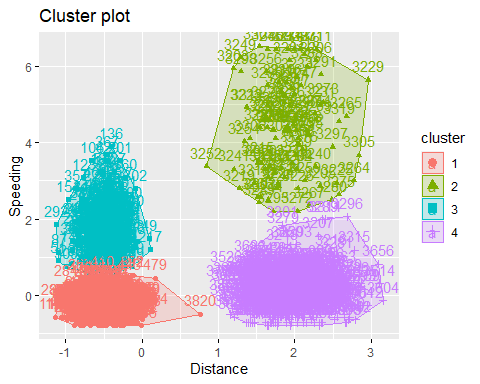
fviz\_nbclust(trucks2, kmeans, method = "silhouette") # maximize how well a point sits in it's cluster

 Both methods point to 4 clusters as the optimal number of clusters.

### 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)  
  
cluster2 <- kmeans(trucks2, 4)  
fviz\_cluster(cluster2, trucks2)



### Task 6:

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

The clusters from Task 5 match our intuition from the initial plot. There are 4 distinct groups. Less distance with minimal speeding, less distance with more speeding, more distance wirh less speeding, and more distance with more speeding.

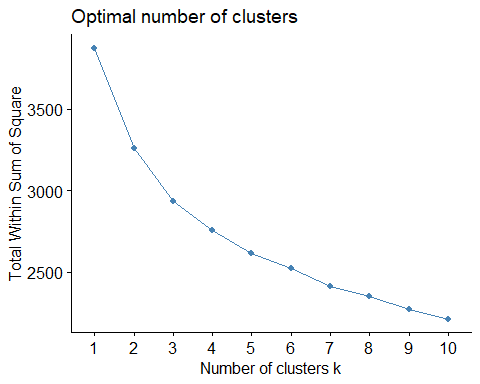
Prep for remaining Tasks:

bball <- read\_csv("kenpom20.csv")

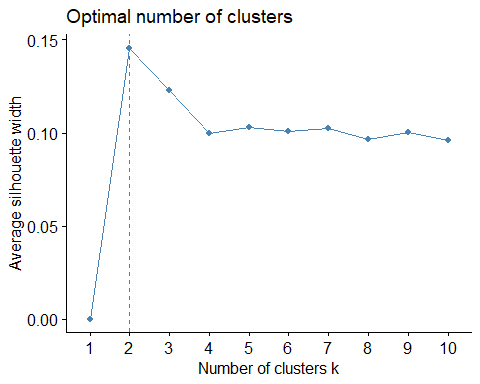
### Task 7:

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?

bball2 <- bball %>%  
 select(-TeamName)  
bball2 <- as.data.frame(scale(bball2))  
set.seed(123)  
fviz\_nbclust(bball2, kmeans, method = "wss") # again, minimize within cluster variation



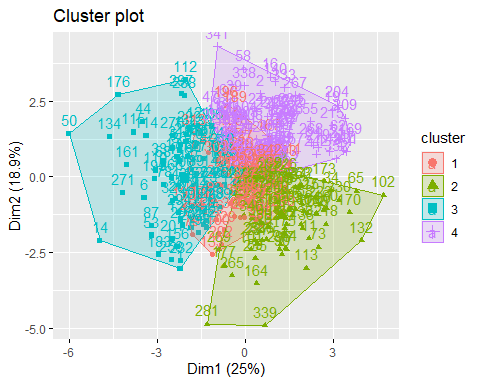
fviz\_nbclust(bball2, kmeans, method = "silhouette") # maximize goodness of point in a cluster

 Within cluster sums of squares is difficult to interperet. However, 2 clusters seems to indicate a sudden change in “within cluster variation”. This choice is strengthened when viewing the silhouette chart. Maximum wellness of fit for silhouette shows 2 clusters as the optimal choice. In summary, 2 clusters appears to be the optimal selection for clusters.

### 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)  
cluster3 <- kmeans(bball2, 4)  
fviz\_cluster(cluster3, bball2)

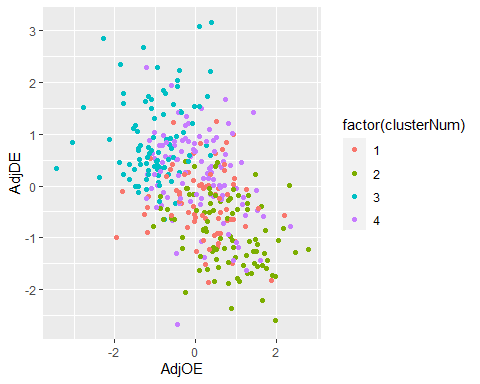


### Task 9:

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”. Hint: You should use color = factor(clusternum) to assign colors to the points. What patterns do you see?

clusterNum <- data.frame(cluster3$cluster)  
#bball2 <- bind\_cols(bball2, clusterNum)  
bball2 <- bball2 %>%  
 mutate(clusterNum = cluster3$cluster)  
#str(bball2)

ggplot(bball2, aes(AdjOE, AdjDE, color = factor(clusterNum))) +  
 geom\_point()

 As points scored goes up points allowed goes down and vice-versa (as a whole). By clustering with 4 clusters we see two obvious groups: high point scored and low points allowed, and low points scored high points allowed. This separates the good form the bad teams. The middle two clusters are separated as follows: average points scored and slightly higher points allowed, and average points scored and slighter few points allowed.