# Clustering

### Miguel Fernandez

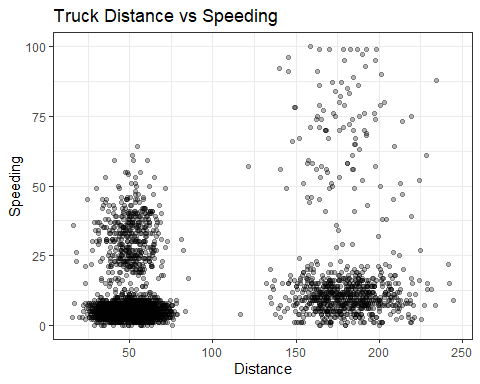
### BAN-502

# Import libraries  
library(tidyverse)  
library(cluster)  
library(factoextra)  
library(dendextend)  
  
# Set global theme  
theme\_set(theme\_bw())

# Read in data  
trucks = read\_csv("trucks.csv")

#### Task 1

# Create scatterplot  
ggplot(trucks, aes(x = Distance, y = Speeding)) +  
 geom\_point(alpha = 0.3) +  
 ggtitle("Truck Distance vs Speeding")



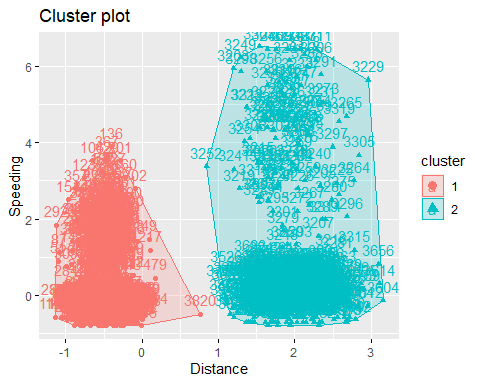
Looking at the relationship between distance and speeding, there are obvious clusters in the data set. There is a dense cluster in the lower left of the plot of drivers who tend not to speed. There is a cluster of drivers who drive longer routes and speed slightly more. Using the alpha parameter, we can see that the first cluster represents many more drivers. There appears to be a loose cluster of drivers who like to speed but do not travel long distances. There is also a very unorganized cluster of points in the upper right of the plot where drivers spend most of their time speeding. In this final group there is very little overlap with the data points and no discernible pattern.

#### Task 2

# Remove driver ID  
trucks2 = trucks %>%  
 select(-Driver\_ID)  
  
# Scale variables  
trucks2 = as.data.frame(scale(trucks2))

#### Task 3

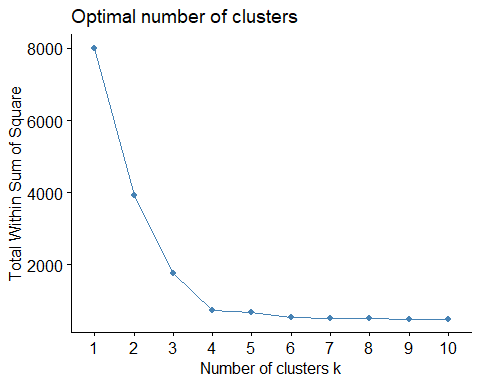
set.seed(64)  
# Create clusters  
clstr = kmeans(trucks2, 2)  
# Plot the clusters  
fviz\_cluster(clstr, trucks2)



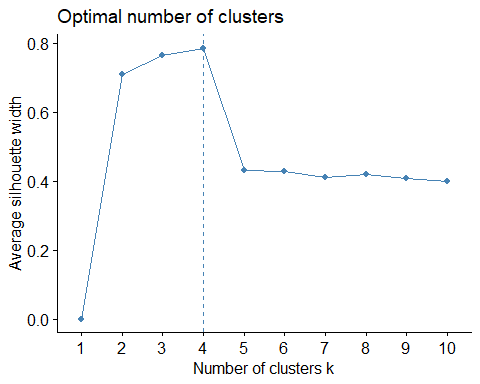
The plot above reveals the 2 clusters identified by the k-means algorithm. The cluster 1 on the left appears more defined than cluster 2 on the right. Interestingly, the point at (0.8, 0) was assigned to cluster 1. That assignment speaks to the weight of cluster 1.

#### Task 4

set.seed(64)  
# WSS method  
fviz\_nbclust(trucks2, kmeans, method = "wss")



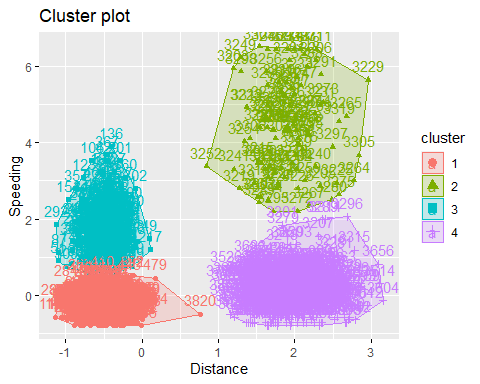
set.seed(64)  
# Silhouette method  
fviz\_nbclust(trucks2, kmeans, method = "silhouette")



Both methods identified 4 clusters as the optimal number for the data set. This is the same number identified in the scatter plot above.

#### Task 5

# Use the optimal number  
set.seed(64)  
# Create clusters  
clstr1 = kmeans(trucks2, 4)  
# Plot the clusters  
fviz\_cluster(clstr1, trucks2)



#### Task 6

Using the four clusters, the plot reveals the same four clusters identified in the scatter plot above. In cluster 1 in the bottom left of the plot, drivers average between 25 and 75 daily miles and tend to speed between 0 and 10 percent of the time. These might be drivers on local routes with little to no time on the highway. Above cluster 1 in Cluster 3, drivers average the same daily miles but spend more of their time speeding. These drivers might operate in the suburbs where the roads are more open while cluster 1 drivers are possibly confined to city streets with complex routing and many stop lights. Cluster 4 in the bottom right of the plot has drivers who average much longer distances and tend to speed up to 25 percent of the time. These drivers are probably on the highway for most, if not all of their route. Cluster 4 drivers travel between 125 miles and 225 miles a day. Above cluster 4, cluster 2 consists of drivers who have long routes and have little regard for the speed limit.

### 

### 

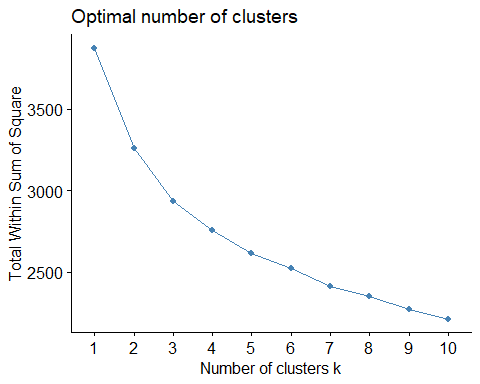
### 

# Read in kenpom data set  
bball = read\_csv("kenpom20.csv")

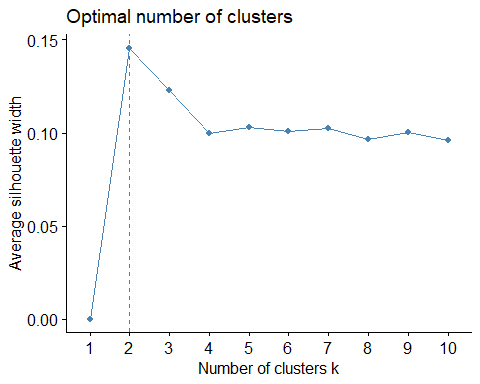
#### Task 7

# Remove Team Name  
bball2 = bball %>%  
 select(-TeamName)  
  
# Scale variables  
bball2 = as.data.frame(scale(bball2))

set.seed(123)  
# WSS method  
fviz\_nbclust(bball2, kmeans, method = "wss")



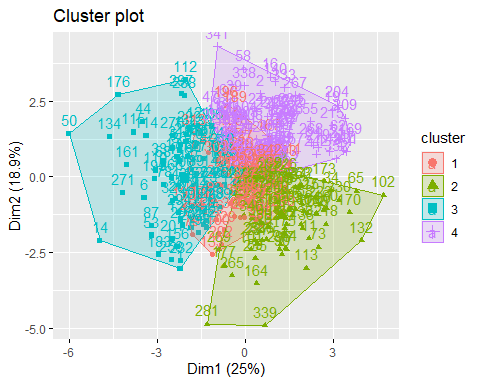
set.seed(123)  
# Silhouette method  
fviz\_nbclust(bball2, kmeans, method = "silhouette")



The elbow method identifies 7 or 8 clusters as the optimal number for the bball2 data set. Using the silhouette method, the algorithm recommends 2 clusters. The two methods show very different results.

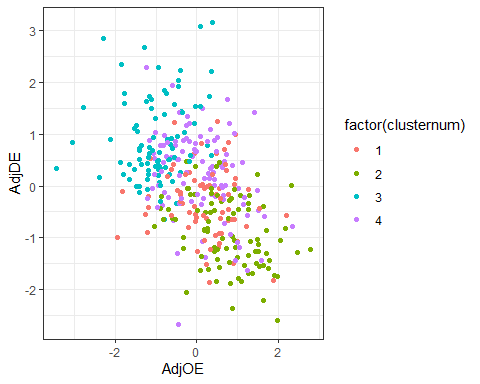
#### Task 8

set.seed(1234)  
# Create clusters  
clstr2 = kmeans(bball2, 4)  
# Plot the clusters  
fviz\_cluster(clstr2, bball2)



#### Task 9

bball2 = bball2 %>%  
 mutate(clusternum = clstr2$cluster)  
  
ggplot(bball2, aes(x = AdjOE,  
 y = AdjDE,  
 color = factor(clusternum))) +  
 geom\_point()



The plot above shows the relationship between points scored on offense per 100 possessions and points allowed per 100 possessions. While there is a negative relationship with the overall data, there is significant overlap and no distinct boundaries when visualized by cluster. Cluster 3 is easily recognized in the upper-left of the plot but also exists in the middle of the plot with all other clusters. Cluster 2 tends to gather in the bottom right of the plot but again is lost in the middle of the plot. Clusters 1 and 4 appear randomly distributed in the middle of the plot.