# Fields, Alex

## BAN 502

### Module 6 - Assignmet 1

#### Q&A - Task 1

When viewing geom\_smooth and geom\_point for the relationship for Distance and Speeding, I can see a partial relationship. I only see it with geom\_smooth, geom\_point seems to just cluster within groups. Nothing is linear.

#### Q&A - Task 4

Is there consensus between these two methods as the optimal number of clusters? *Yes, it seems that with the "descent" method, we are seeing the optimal number technically being the highest number which to me, doesn't seem correct. With the Ascent method, it draws a vertical line to give you another look into what it thinks is optimal. I agree with R when it states that 4 is the optimal number of clusters.*

#### Q&A - Task 5

In words, how would you characterize the clusters you created in Task 5? *After viewing my optimal cluster number, and applying that to the dataset, it seems to fit very well. There is no overlapping and the data seems to match, intuitively.*

#### Q&A - Task 9

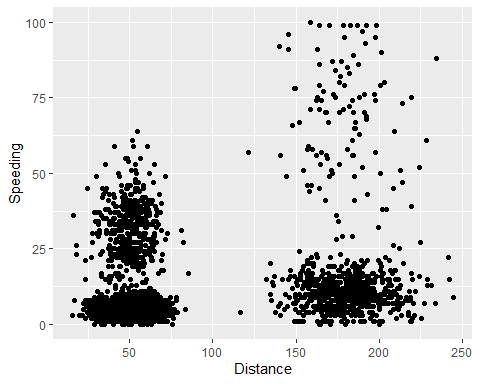
What patterns do you see? *I see the same pattern regarding the clusters that you would see with the KMeans algorithm. This shows me that we can produce the same output as the KMeans algorithm*

#### Library

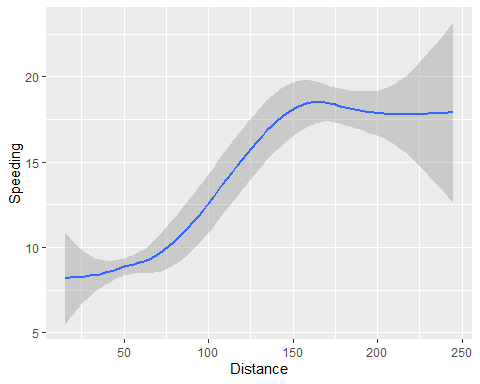
options(tidyverse.quiet=TRUE)  
library(tidyverse)  
library(cluster) #algorithms for clustering  
library(factoextra) #visualization  
library(dendextend)  
library(caret)

### Task 1

trucks <- read\_csv("trucks.csv")  
  
ggplot(trucks, aes(Distance, Speeding)) + geom\_point()



ggplot(trucks, aes(Distance, Speeding)) + geom\_smooth()



### Task 2

summary(trucks)#viewing the summary before pre-scaling

## 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

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

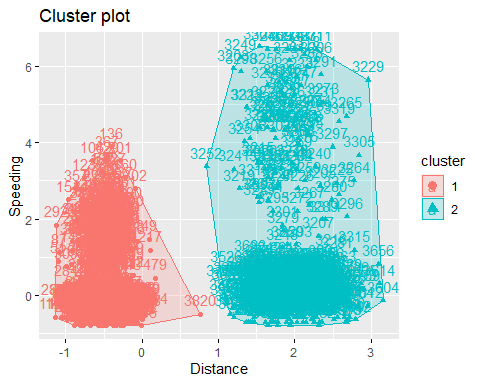
## Distance Speeding   
## Min. :-1.1319 Min. :-0.7821   
## 1st Qu.:-0.5759 1st Qu.:-0.4903   
## Median :-0.4248 Median :-0.3444   
## Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.1947 3rd Qu.:-0.1255   
## Max. : 3.1560 Max. : 6.5127

### Task 3

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

Visualize the clustering

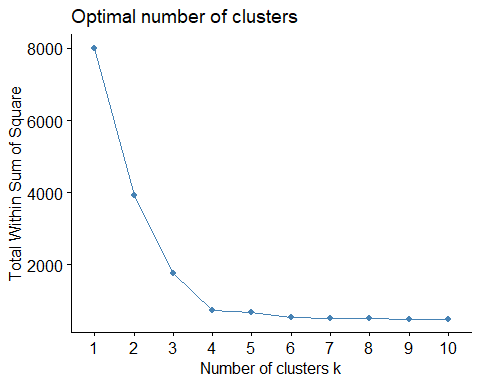
fviz\_cluster(clusters1, trucks2)



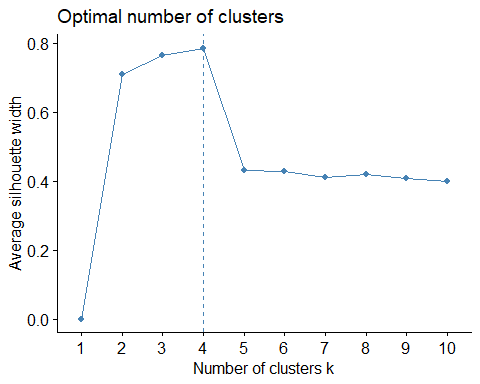
### Task 4

Visually identify optimal number of clusters

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

 Another method

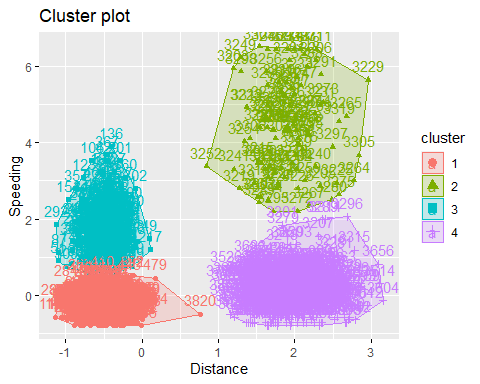
set.seed(64)  
fviz\_nbclust(trucks2, kmeans, method = "silhouette") #maximize how well points sit in their clusters



### Task 5

#### Using optimal #of clusters (4)

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



### Task 6 (Basketball)

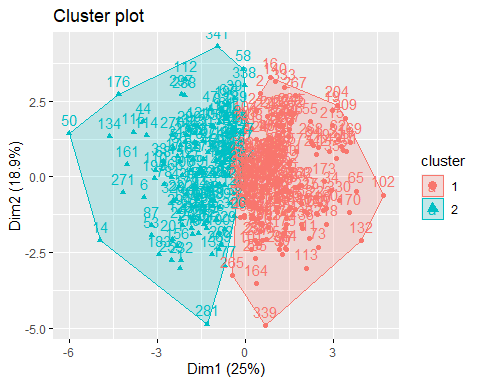
bball <- read\_csv("kenpom20.csv")  
  
  
bball2 <- bball %>% dplyr::select(-TeamName)  
  
bball2 = as.data.frame(scale(bball2))  
summary(bball2)

## AdjTempo AdjOE AdjDE eFGPct   
## Min. :-3.39100 Min. :-3.45199 Min. :-2.6678124 Min. :-3.68381   
## 1st Qu.:-0.67255 1st Qu.:-0.73679 1st Qu.:-0.6475704 1st Qu.:-0.69461   
## Median :-0.02436 Median : 0.01391 Median :-0.0002663 Median :-0.01141   
## Mean : 0.00000 Mean : 0.00000 Mean : 0.0000000 Mean : 0.00000   
## 3rd Qu.: 0.58621 3rd Qu.: 0.67648 3rd Qu.: 0.6199679 3rd Qu.: 0.68688   
## Max. : 3.31633 Max. : 2.79447 Max. : 3.1546440 Max. : 3.62001   
## TOPct ORPct FTRate eFGPctD   
## Min. :-2.61437 Min. :-3.420873 Min. :-2.25314 Min. :-3.03315   
## 1st Qu.:-0.57807 1st Qu.:-0.676410 1st Qu.:-0.72045 1st Qu.:-0.71103   
## Median :-0.05269 Median :-0.005597 Median :-0.01699 Median :-0.02033   
## Mean : 0.00000 Mean : 0.000000 Mean : 0.00000 Mean : 0.00000   
## 3rd Qu.: 0.66307 3rd Qu.: 0.690177 3rd Qu.: 0.65294 3rd Qu.: 0.68576   
## Max. : 3.78116 Max. : 3.061926 Max. : 3.17766 Max. : 3.17549   
## TOPctD ORPctD FTRateD   
## Min. :-2.1515 Min. :-3.0625871 Min. :-2.1989   
## 1st Qu.:-0.6650 1st Qu.:-0.6857607 1st Qu.:-0.6469   
## Median :-0.1072 Median :-0.0001725 Median :-0.1504   
## Mean : 0.0000 Mean : 0.0000000 Mean : 0.0000   
## 3rd Qu.: 0.6255 3rd Qu.: 0.6534689 3rd Qu.: 0.5913   
## Max. : 3.9482 Max. : 3.0506866 Max. : 3.4186

set.seed(123)  
clusters1 <- kmeans(bball2, 2)

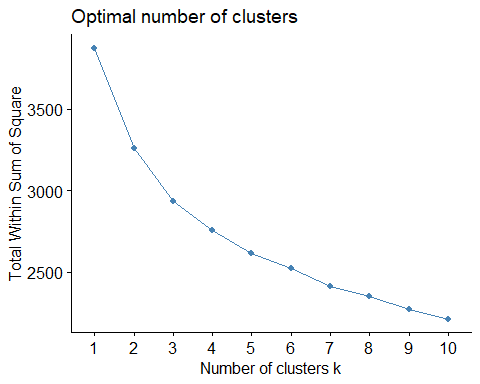
Visualize the clustering

fviz\_cluster(clusters1, bball2)

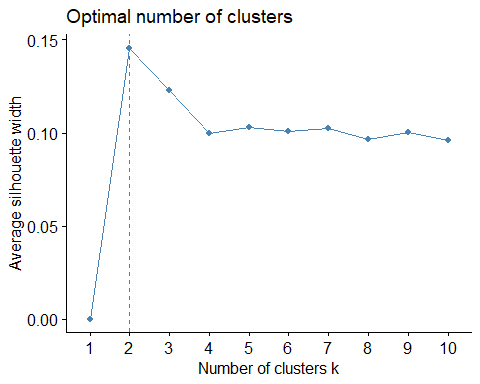


Visually identify optimal number of clusters

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

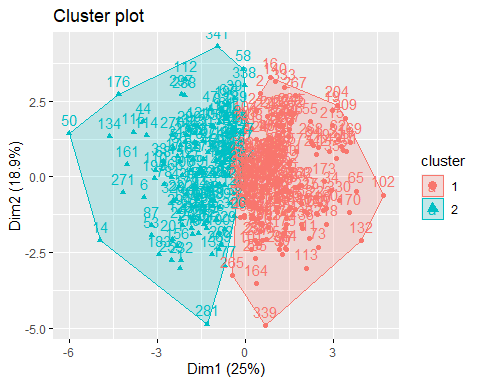
 Another method

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



Optimal Cluster

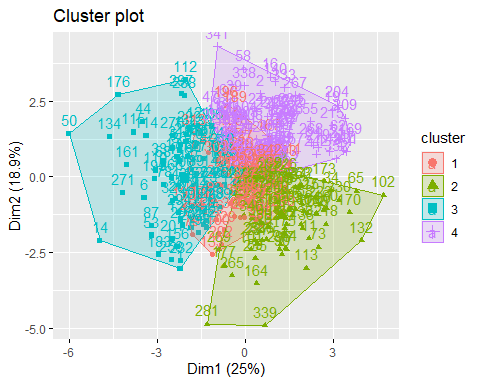
set.seed(123)  
clusters2 <- kmeans(bball2, 2)  
fviz\_cluster(clusters2, bball2)



### Task 8

Cluster of 4

set.seed(1234)  
clusters2 <- kmeans(bball2, 4)  
fviz\_cluster(clusters2, bball2)



### Task 9

Attach cluster to dataset

bball2 = bball2 %>% mutate(clusternum = clusters2$cluster)  
str(bball2)

## 'data.frame': 353 obs. of 12 variables:  
## $ AdjTempo : num 0.0183 0.1577 0.4078 2.4344 -0.235 ...  
## $ AdjOE : num -0.492 0.796 0.898 1.266 -2.132 ...  
## $ AdjDE : num -0.337 1.214 -0.531 -0.449 0.888 ...  
## $ eFGPct : num 0.0254 1.1821 0.7709 0.9326 -2.5489 ...  
## $ TOPct : num 0.8177 -0.4197 -0.48 0.1371 -0.0689 ...  
## $ ORPct : num 0.227 -1.349 -0.135 0.503 -0.746 ...  
## $ FTRate : num 1.73346 0.487 0.59672 0.86277 0.00375 ...  
## $ eFGPctD : num 0.278 2.386 -1.323 -0.619 0.61 ...  
## $ TOPctD : num 3.176 -1.094 -0.202 -0.33 -0.327 ...  
## $ ORPctD : num 0.507 -1.769 0.416 0.973 -0.342 ...  
## $ FTRateD : num 2.719 -0.501 -0.622 0.518 0.405 ...  
## $ clusternum: int 3 4 2 2 3 3 4 3 1 1 ...

ggplot(bball2, aes(x=AdjOE,y=AdjDE,color=factor(clusternum))) + geom\_point()

