library(tidyverse)

## ── Attaching packages ──────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.0 ✓ purrr 0.3.3  
## ✓ tibble 3.0.1 ✓ dplyr 0.8.5  
## ✓ tidyr 1.0.2 ✓ stringr 1.4.0  
## ✓ readr 1.3.1 ✓ forcats 0.5.0

## ── Conflicts ─────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(cluster)  
library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(dendextend)

##   
## ---------------------  
## Welcome to dendextend version 1.13.4  
## 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

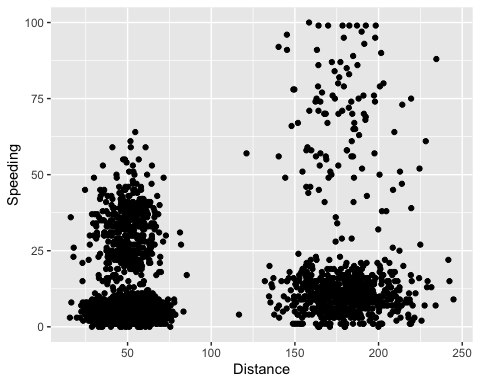
library(readr)  
trucks <- read\_csv("trucks.csv")

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

trucks = trucks %>% drop\_na()

Task 1.

library(ggplot2)  
ggplot(trucks, aes(x = Distance, y = Speeding)) +  
 geom\_point()

 Based on the scatterplot, there does seem to be some natural clustering around speeding when traveling 25-75 miles and another natural cluster when traveling between 150-200 miles. Both have large clusters that fall below 25% of time spent traveling but the shorter distance seems to have another cluster above 25% of the time as well.

Task 2.

trucks2 = trucks %>% 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

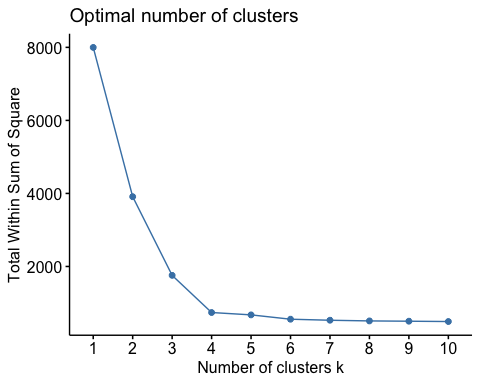
Task 3.

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

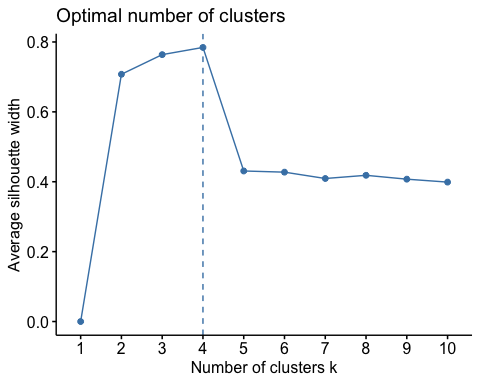
 Both clusters seem to be independent of each other and similar to the scatterplot I graphed earlier in task 2. Cluster 2 is more scattered than cluster 1.

Task 4.

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



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

 Yes, there is a consensus between these 2 methods to use k=4 clusters.

Task 5.

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



Task 6. Clusters 1 and 3 seem to have similar characteristics of speeding, distance-wise. Both groups, who seem to not travel over 100 miles, tend to speed under 50% of the time. Clusters 2 and 4 also share similar characteristics of speeding distance-wise. While the two clusters do not overlap, a larger group in each cluster shows that the longer distances they have traveled (over 100 miles), the longer time they tend to drive over the speed limit. Cluster 4 looks to be the biggest one but, similar to clusters 1 and 3, they tend to not speed more than 50% of the time. Cluster 2 on the other hand is a bit more spread out, so less drivers seem to be grouped into this cluster, but these drivers tend to speed 50-100% of the time they are driving.

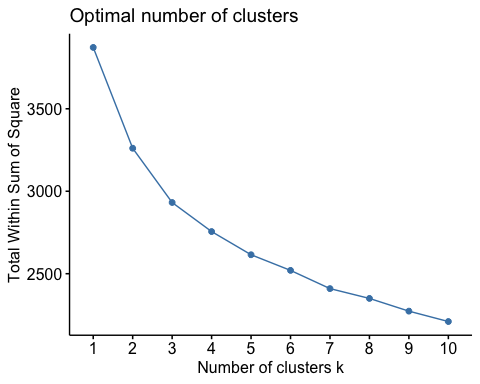
Task 7.

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

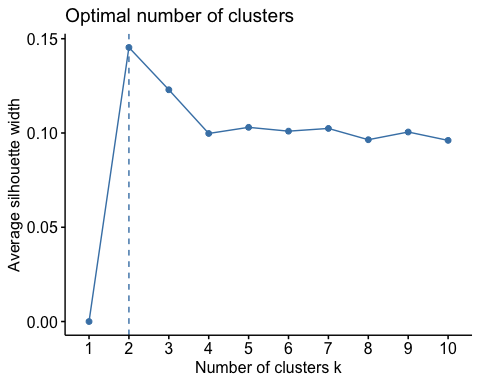
## Parsed with column specification:  
## cols(  
## TeamName = col\_character(),  
## AdjTempo = col\_double(),  
## AdjOE = col\_double(),  
## AdjDE = col\_double(),  
## eFGPct = col\_double(),  
## TOPct = col\_double(),  
## ORPct = col\_double(),  
## FTRate = col\_double(),  
## eFGPctD = col\_double(),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

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

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

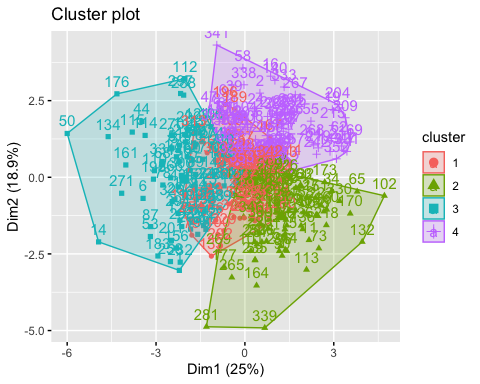


set.seed(123)  
fviz\_nbclust(bball2, kmeans, method = "silhouette")

 There’s not a clear consensus between the 2 methods to use k = 2, and the wss method doesn’t seem to depict that very clearly like the silhouette method does. I think the agreement between the two methods is that a big shift happened between k 1 and 2 based on a large sum, so k = 2 could be used. But k = 4 could be another solution based on the “elbows” of both graphs.

Task 8.

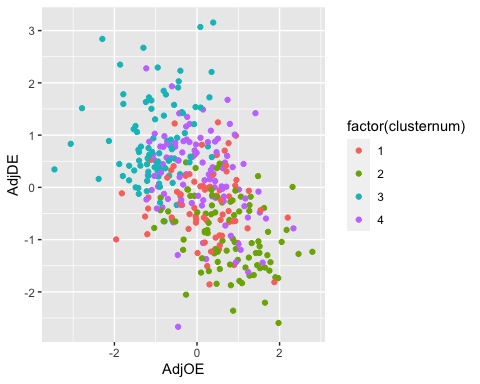
set.seed(1234)  
clusters3 <- kmeans(bball2, 4)  
fviz\_cluster(clusters3, bball2)



Task 9.

bball2 = bball2 %>% mutate(clusternum = clusters3$cluster)

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

 The majority of the clusters, especially clsuters 1 and 4, score about 0 - 1 points per 100 possessions on both the offensive and defensive ends. Cluster 2 tends to score more points when playing offense while cluster 3 is even more spread out and tends to score more points playing defense.