Module6Assign1

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.6.2

## -- Attaching packages ------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

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

library(cluster)  
library(factoextra)

## Warning: package 'factoextra' was built under R version 3.6.2

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

library(dendextend)

## Warning: package 'dendextend' was built under R version 3.6.2

##   
## ---------------------  
## Welcome to dendextend version 1.13.3  
## 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

Read in data

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

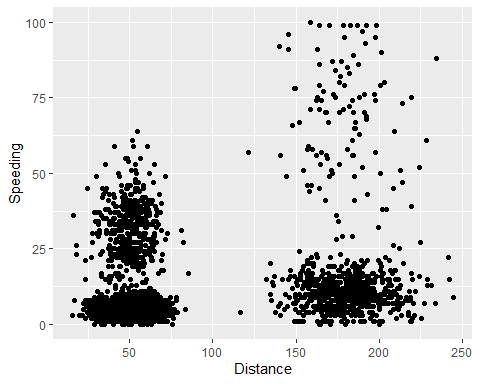
## Parsed with column specification:  
## cols(  
## Driver\_ID = col\_double(),  
## Distance = col\_double(),  
## Speeding = col\_double()  
## )

str(trucks)

## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and 'data.frame': 4000 obs. of 3 variables:  
## $ Driver\_ID: num 3.42e+09 3.42e+09 3.42e+09 3.42e+09 3.42e+09 ...  
## $ Distance : num 71.2 52.5 64.5 55.7 54.6 ...  
## $ Speeding : num 28 25 27 22 25 10 20 8 34 19 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. Driver\_ID = col\_double(),  
## .. Distance = col\_double(),  
## .. Speeding = col\_double()  
## .. )

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

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



ANSWER: There do appear to be 4 clusters of data when comparing distance and speeding.

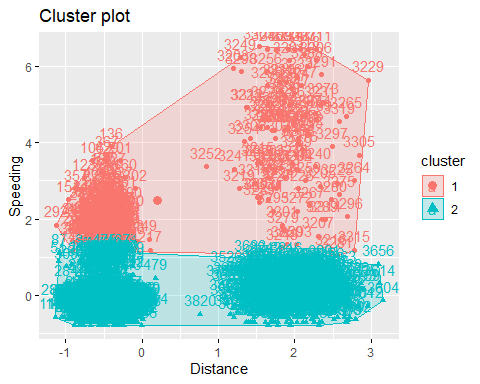
**Task 2** Create a new data frame (called trucks2) that excludes the Driver\_ID variable and includes scaled versions of the Distance and Speeding variables. NOTE: Wrap the scale(trucks2) command in an as.data.frame command to ensure that the resulting object is a data frame. By default, scale converts data frames to lists

trucks2 = trucks %>% select(-Driver\_ID)  
trucks2 = scale(as.data.frame(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** Use k-Means clustering with two clusters (k=2) to cluster the trucks2 data frame. Use a random number seed of 1234. Visualize the clusters using the fviz\_cluster function. Comment on the clusters.

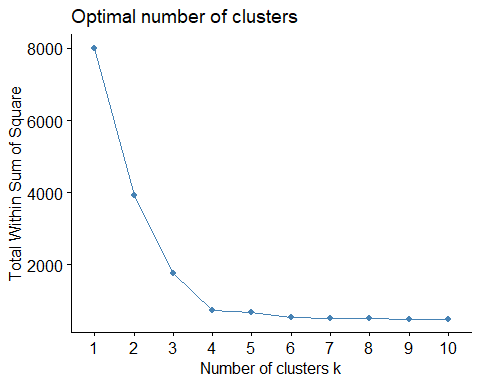
set.seed(1234)  
cluster\_truck <- kmeans(trucks2, 2)  
fviz\_cluster(cluster\_truck, trucks2)



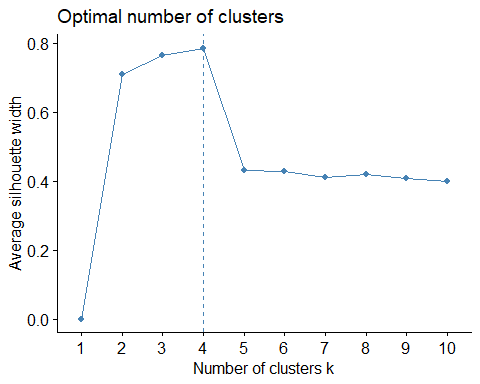
ANSWER: It’s not the clusters you’d think it would be. The clusters are divided by the normalized value of “Speeding > 1” and “Speeding < 1” (approximately)

**Task 4** Use the two methods from the k-Means lecture to identify the optimal number of clusters. Use a random number seed of 123 for these methods. Is there consensus between these two methods as the optimal number of clusters?

set.seed(123)  
fviz\_nbclust(trucks2, kmeans, method = "wss")



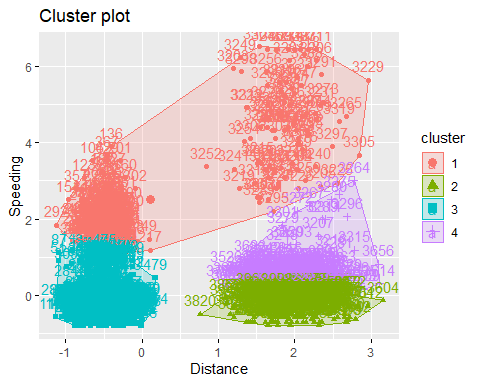
set.seed(123)  
fviz\_nbclust(trucks2, kmeans, method = "silhouette")



ANSWER: Both the silhouette and wss methods indicate that 4 clusters is optimal.

**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 1234. Use the fviz\_cluster function to visualize the clusters.

set.seed(1234)  
cluster\_truck4 <- kmeans(trucks2, 4)  
fviz\_cluster(cluster\_truck4, trucks2)



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

Cluster 1: Drivers who speed the most often. It doesn’t matter if they are on short or long trips. Cluster 2: Drivers who have long-distance routes who seldom speed. Cluster 3: Drivers who have short-distance routes who seldom speed. Cluster 4: Drivers who have long distance routes who speed sometimes.

**Time to wine** Create a new data frame called wine2 that removes the Year and FrancePop variables and scales the other variables.

wine = read\_csv("wineprice.csv")

## Parsed with column specification:  
## cols(  
## Year = col\_double(),  
## Price = col\_double(),  
## WinterRain = col\_double(),  
## AGST = col\_double(),  
## HarvestRain = col\_double(),  
## Age = col\_double(),  
## FrancePop = col\_double()  
## )

wine2 = wine %>% select(-Year, -FrancePop)  
wine2 = scale(as.data.frame(wine2))   
summary(wine2)

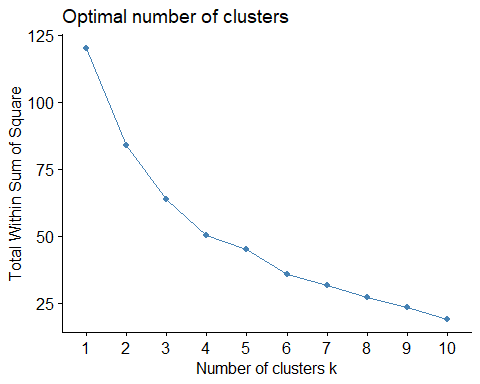
## Price WinterRain AGST   
## Min. :-1.32596 Min. :-1.73332 Min. :-2.25947   
## 1st Qu.:-0.84329 1st Qu.:-0.52375 1st Qu.:-0.45801   
## Median : 0.08284 Median :-0.03992 Median : 0.03548   
## Mean : 0.00000 Mean : 0.00000 Mean : 0.00000   
## 3rd Qu.: 0.65777 3rd Qu.: 0.69339 3rd Qu.: 0.82524   
## Max. : 2.19343 Max. : 1.69885 Max. : 1.68888   
## HarvestRain Age   
## Min. :-1.4856 Min. :-1.586   
## 1st Qu.:-0.8003 1st Qu.:-0.806   
## Median :-0.2494 Median :-0.026   
## Mean : 0.0000 Mean : 0.000   
## 3rd Qu.: 0.5165 3rd Qu.: 0.754   
## Max. : 1.9275 Max. : 1.794

str(wine2)

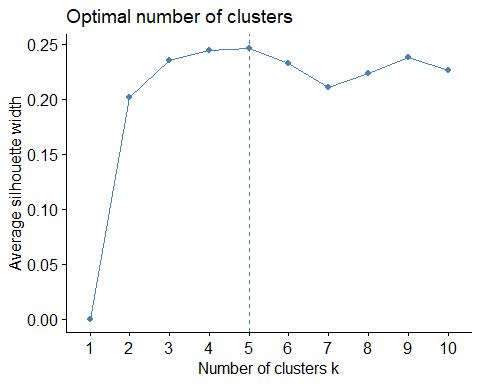
## num [1:25, 1:5] 0.658 1.495 0.951 -0.127 -0.446 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr [1:5] "Price" "WinterRain" "AGST" "HarvestRain" ...  
## - attr(\*, "scaled:center")= Named num [1:5] 7.07 605.28 16.51 148.56 17.2  
## ..- attr(\*, "names")= chr [1:5] "Price" "WinterRain" "AGST" "HarvestRain" ...  
## - attr(\*, "scaled:scale")= Named num [1:5] 0.65 132.278 0.675 74.419 7.692  
## ..- attr(\*, "names")= chr [1:5] "Price" "WinterRain" "AGST" "HarvestRain" ...

**Task 7** 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?

set.seed(123)  
fviz\_nbclust(wine2, kmeans, method = "wss")



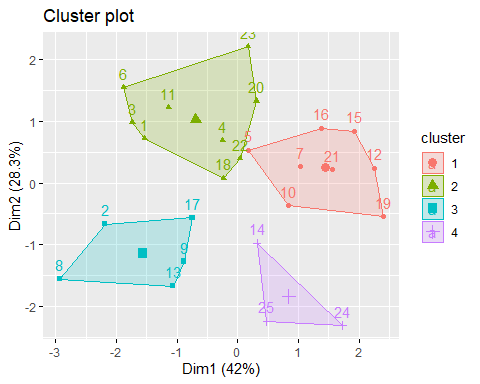
set.seed(123)  
fviz\_nbclust(wine2, kmeans, method = "silhouette")



ANSWER: Using wss there really isn’t a definitive answer. It starts to flatten out at 4. Using silhouette determines the number is 5, but 4 is pretty close too. We’ll go with 5.

**Task 8** Use the optimal number of clusters that you identiﬁed in Task 4 to create k-Means clusters. Use a random number seed of 1234. Use the fviz\_cluster function to visualize the clusters.

set.seed(1234)  
cluster\_wine4 <- kmeans(wine2, 4)  
fviz\_cluster(cluster\_wine4, wine2)



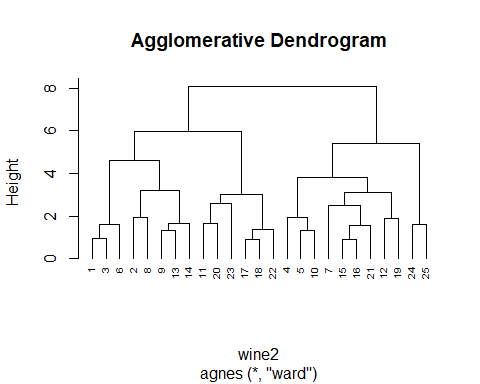
**Task 9** Use agglomerative clustering to develop a dendogram for the scaled wine data. Follow the same process from the lecture where we used a custom function to identify the distance metric that maximizes the “agglomerative coeﬃcient”. Plot the dendogram.

m = c( "average", "single", "complete", "ward")  
names(m) = c( "average", "single", "complete", "ward")  
  
ac = function(x) {  
 agnes(wine2, method = x)$ac  
}  
map\_dbl(m, ac)

## average single complete ward   
## 0.5666719 0.2920143 0.7196616 0.8112139

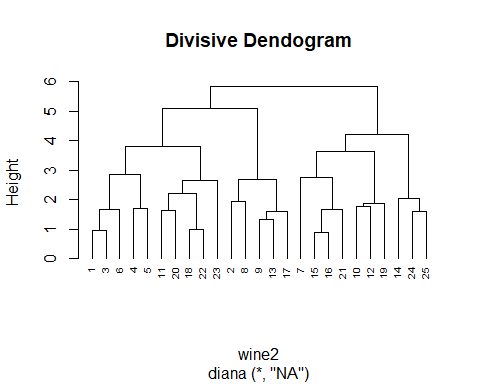
Ward is the highest with a value of 0.8112139

hc = agnes(wine2, method = "ward")   
pltree(hc, cex = 0.6, hang = -1, main = "Agglomerative Dendrogram")



Task 10: Repeat Task 9, but with divisive clustering

hc2 = diana(wine2)  
pltree(hc2, cex = 0.6, hang = -1, main = "Divisive Dendogram")



That’s it