M6 Assignment1

# Module 6: Assignment 1 – Clustering

## Kellie McLiverty

### Assignment Needs & Data Importation

Libraries & dataset needed for Assignemnt

options(tidyverse.quiet=TRUE)  
library(tidyverse)  
library(cluster) #algorithms for clustering

## Warning: package 'cluster' was built under R version 3.5.2

library(factoextra) #visualization

## Warning: package 'factoextra' was built under R version 3.5.2

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ

#first part dataset  
trucks <- read\_csv("trucks.csv")

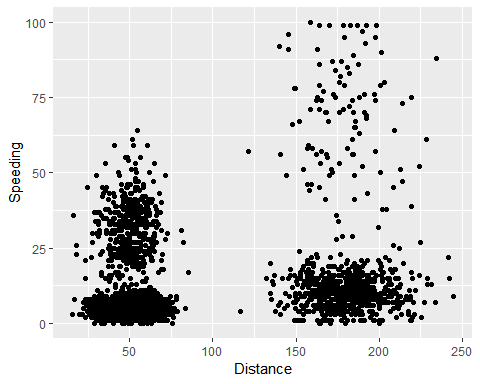
## Parsed with column specification:  
## cols(  
## Driver\_ID = col\_double(),  
## Distance = col\_double(),  
## Speeding = col\_double()  
## )

str(trucks)

## Classes '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()  
## .. )

### Examining the relationship between Distance and Speeding

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



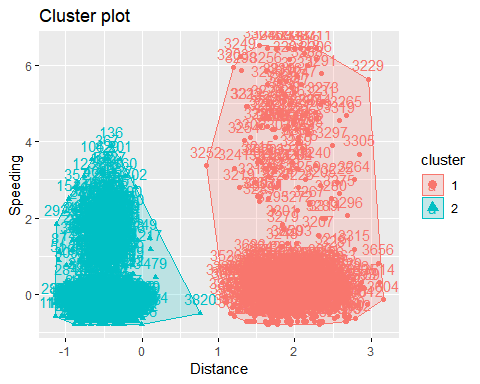
Examining the scatterplot, we can see that the data clusters around two points with very few points mingled inbetween. The frist groupd is tightly close together, while the other is more loose.

trucks2<-select(trucks, -Driver\_ID) #exclused Driver\_ID from dataset  
  
trucks2 <- as.data.frame(scale(trucks2)) #scales variables and keeps as dataframe  
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

Next, I’ll perform k-means clustering with 2 clusters and a set.seed to ensure same clusters.

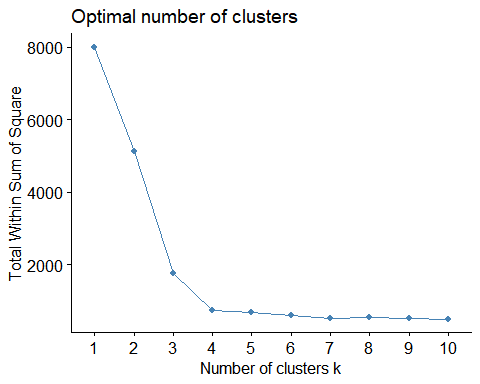
set.seed(1234)  
clusters1 <- kmeans(trucks2, 2)  
  
fviz\_cluster(clusters1, trucks2) #Visualizes cluster



After the k-mean clustering, we can see two distinct clusters. Cluster 1 is a wide, loosely fitted cluster, while cluster two is tightly clustered together and is much smaller in size than cluster 1.

Now, I’ll try two other k-means methods to view the clusters. For this section, I changed the set.seed to 123. First Method

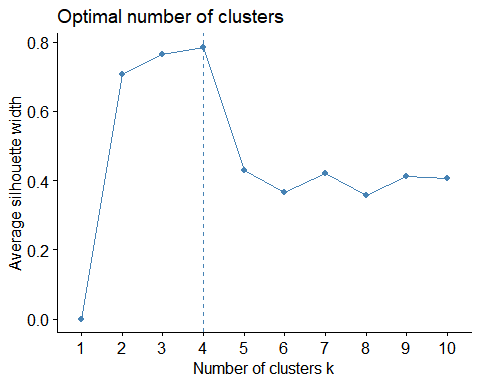
set.seed(123)  
fviz\_nbclust(trucks2, kmeans, method = "wss") #minimize within-cluster variation



Looking at the optimal cluster diagram, we can see the “eblow” in the graph between 3 and 4. This would mean to me that this data should actually have around 4 clusters instead of just 2.

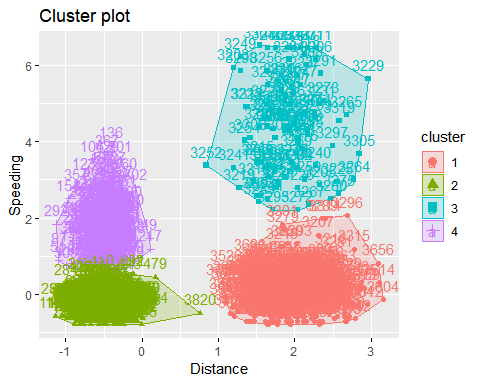
Second method

set.seed(123)  
fviz\_nbclust(trucks2, kmeans, method = "silhouette") #maximize how well points sit in their clusters



Reviewing this method, we can see that our optimal number of clusters is 4 instead of two. These two diagrams do have some consensus on the ideal optimal number of clusters, which would be 4. With this known, I will repeat the initial k-means cluster with the optimal number of clusters.

set.seed(1234)  
clusters2 <- kmeans(trucks2, 4) #uses the optimal number of clusters  
  
fviz\_cluster(clusters2, trucks2) #Visualizes cluster



As we can see from the visualization of these 4 new clusters. The first cluster, colored red, is tightly grouped together with some outlaying points. The second cluster, colored green, is so tightly packed together near its centerpoint it is basically a giant blob. The third cluster, colored teal, is loosely grouped together with no clear discernible center point. The last cluster, colored purple, is a tall, closely grouped cluster with a discernible center point [that also looks roughly shaped like the UK].

### Examining Wine Price Dataset

#second part dataset  
wineprice <- 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()  
## )

str(wineprice)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 25 obs. of 7 variables:  
## $ Year : num 1952 1953 1955 1957 1958 ...  
## $ Price : num 7.5 8.04 7.69 6.98 6.78 ...  
## $ WinterRain : num 600 690 502 420 582 485 763 830 697 608 ...  
## $ AGST : num 17.1 16.7 17.1 16.1 16.4 ...  
## $ HarvestRain: num 160 80 130 110 187 187 290 38 52 155 ...  
## $ Age : num 31 30 28 26 25 24 23 22 21 20 ...  
## $ FrancePop : num 43184 43495 44218 45152 45654 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. Year = col\_double(),  
## .. Price = col\_double(),  
## .. WinterRain = col\_double(),  
## .. AGST = col\_double(),  
## .. HarvestRain = col\_double(),  
## .. Age = col\_double(),  
## .. FrancePop = col\_double()  
## .. )

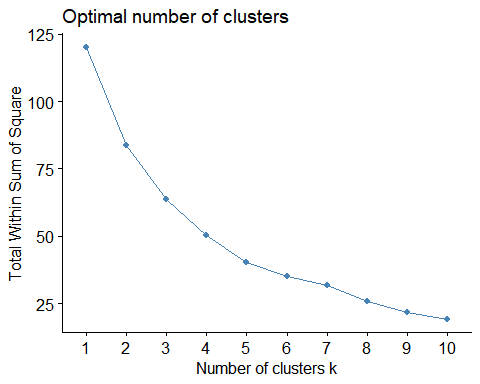
To begin, I will be creating a new dataset that removes the Year and FrancePop variables and scales the remaining variables.

wine2<-select(wineprice, -Year, -FrancePop) #exclused unwanted variables from dataset  
  
wine2 <- as.data.frame(scale(wine2)) #scales variables and keeps as dataframe  
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

Next, I will test to find my optimal number of clusters for the wine data. First Method

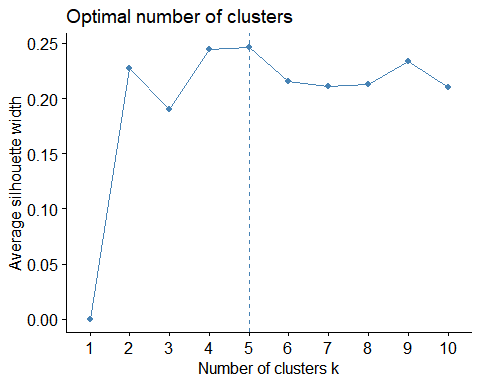
set.seed(123)  
fviz\_nbclust(wine2, kmeans, method = "wss") #minimize within-cluster variation



For this method, we have 5 to 6 optimal number of clusters. We can see this from the “elbow” the curve between the 5th and 6th datapoint.

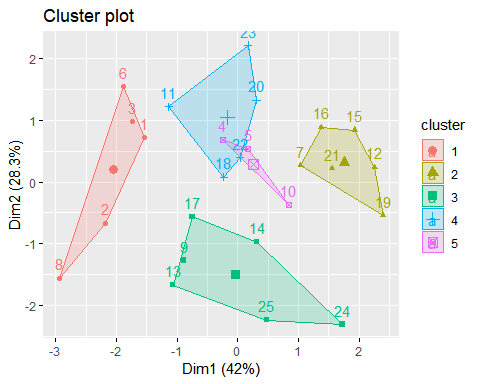
Second method

set.seed(123)  
fviz\_nbclust(wine2, kmeans, method = "silhouette") #maximize how well points sit in their clusters



This method shows that we have 5 optimal number of clusters. Both of these diagrams do have roughly similar consensus at 5 clusters, as such I will use this number to find my optimal clusters for the data. Below, you can see a cluster plot of the data. Clusters 4 and 5 overlap slighly, while the others are spreadout from each other.

set.seed(1234)  
clusters3 <- kmeans(wine2, 5) #uses the optimal number of clusters  
  
fviz\_cluster(clusters3, wine2) #Visualizes cluster



Lastly, I’ll use agglomerative & divisive clustering to develop dendograms for the scaled wine data. I’ll create the agglomerative dendogram first.

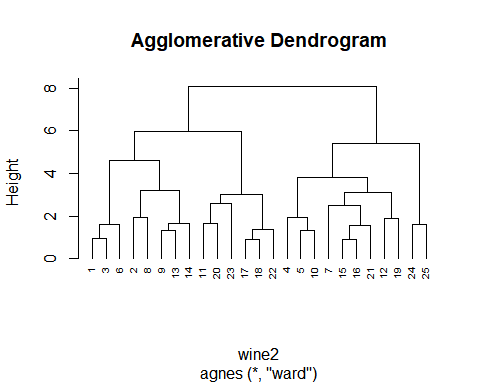
Agglomerative clustering  
Start by identifying best dissimilarity measure. This is given by highest “agglomerative coefficient”.

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

As ward’s method is the highest, I’ll use this to develop clusters for the dendogram.

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



Divisive clustering

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

