# Daniel Rauscher

## Module 6 Assignment 1

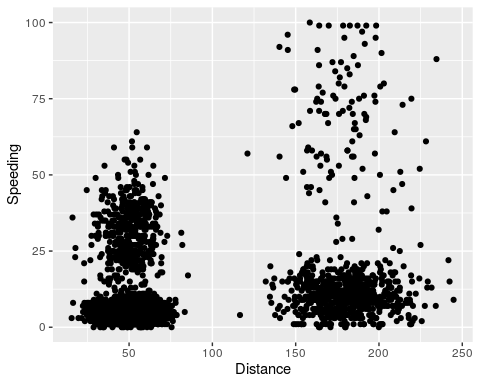
trucks <- read\_csv("trucks.csv")

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## Driver\_ID = col\_double(),  
## Distance = col\_double(),  
## Speeding = col\_double()  
## )

## Task 1

There does appear to be some natural clustering between these two variables. There seems to be a distinct clustering divided between shorter and longer distances and also higher and lower percentage of speeding over 5mph.

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



## Task 2

kmeans\_recipe = recipe(~ Distance + Speeding, trucks)  
  
trucks\_dummy = kmeans\_recipe %>%  
 step\_scale(all\_numeric()) %>%  
 step\_center(all\_numeric())  
  
trucks\_dummy = prep(trucks\_dummy, trucks)  
  
trucks\_cleaned = bake(trucks\_dummy, trucks)

## Task 3

The clusters are divided how I thought they would be when looking at the graph in task 1. The data is clustered between shorter and higher distance. It looks as though we could add a few more clusters to more clearly distinguish the data points and make sure each cluster has similar data points.

set.seed(64)  
clusts=   
 tibble(k=2) %>%  
 mutate(kclust=map(k, ~kmeans(trucks\_cleaned, .x)), tidied = map(kclust, tidy), glanced = map(kclust, glance), augmented = map(kclust, augment, trucks\_cleaned))  
  
clusts

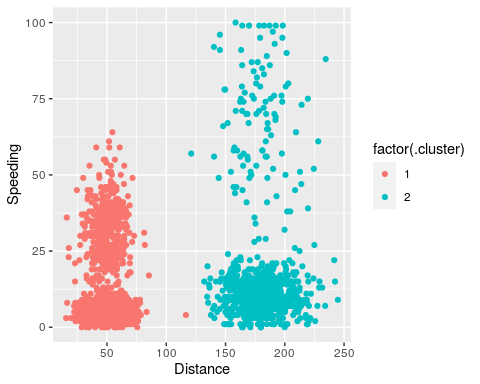
## # A tibble: 1 x 5  
## k kclust tidied glanced augmented   
## <dbl> <list> <list> <list> <list>   
## 1 2 <kmeans> <tibble [2 × 5]> <tibble [1 × 4]> <tibble [4,000 × 3]>

set.seed(64)  
clusters = kmeans(trucks\_cleaned, 2)

trucks = augment(clusters, trucks)  
str(trucks)

## tibble [4,000 × 4] (S3: tbl\_df/tbl/data.frame)  
## $ Driver\_ID: num [1:4000] 3.42e+09 3.42e+09 3.42e+09 3.42e+09 3.42e+09 ...  
## $ Distance : num [1:4000] 71.2 52.5 64.5 55.7 54.6 ...  
## $ Speeding : num [1:4000] 28 25 27 22 25 10 20 8 34 19 ...  
## $ .cluster : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. Driver\_ID = col\_double(),  
## .. Distance = col\_double(),  
## .. Speeding = col\_double()  
## .. )

p1 =   
 ggplot(trucks, aes(x=Distance, y=Speeding, color=factor(.cluster))) +   
 geom\_point()   
p1



## Task 4

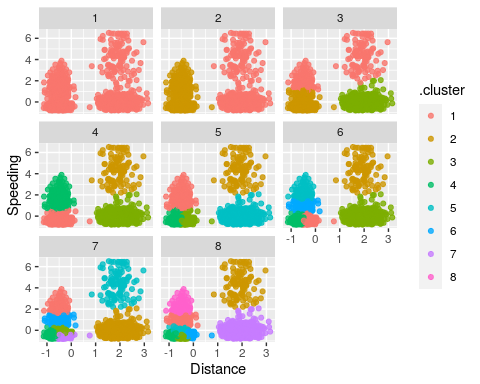
I would choose between 4-5 as the best value of k. At those values there appears to be enough clusters to better distinguish data and make each cluster contain very similar data, but not too many clusters where it becomes confusing.

set.seed(412)  
clusts=   
 tibble(k=1:8) %>%  
 mutate(kclust=map(k, ~kmeans(trucks\_cleaned, .x)), tidied = map(kclust, tidy), glanced = map(kclust, glance), augmented = map(kclust, augment, trucks\_cleaned))  
  
clusts

## # A tibble: 8 x 5  
## k kclust tidied glanced augmented   
## <int> <list> <list> <list> <list>   
## 1 1 <kmeans> <tibble [1 × 5]> <tibble [1 × 4]> <tibble [4,000 × 3]>  
## 2 2 <kmeans> <tibble [2 × 5]> <tibble [1 × 4]> <tibble [4,000 × 3]>  
## 3 3 <kmeans> <tibble [3 × 5]> <tibble [1 × 4]> <tibble [4,000 × 3]>  
## 4 4 <kmeans> <tibble [4 × 5]> <tibble [1 × 4]> <tibble [4,000 × 3]>  
## 5 5 <kmeans> <tibble [5 × 5]> <tibble [1 × 4]> <tibble [4,000 × 3]>  
## 6 6 <kmeans> <tibble [6 × 5]> <tibble [1 × 4]> <tibble [4,000 × 3]>  
## 7 7 <kmeans> <tibble [7 × 5]> <tibble [1 × 4]> <tibble [4,000 × 3]>  
## 8 8 <kmeans> <tibble [8 × 5]> <tibble [1 × 4]> <tibble [4,000 × 3]>

clusters =   
 clusts %>%  
 unnest(cols = c(tidied))  
  
assignments =   
 clusts %>%   
 unnest(cols = c(augmented))  
  
clusterings =   
 clusts %>%  
 unnest(cols = c(glanced))

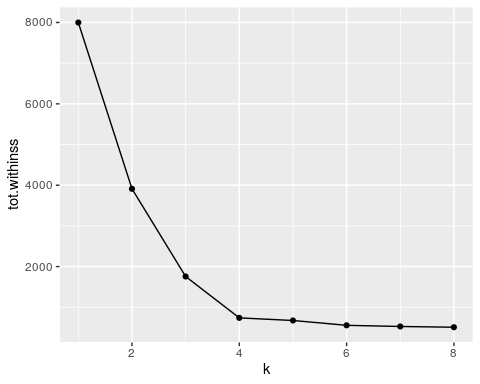
p2 =   
 ggplot(assignments, aes(x=Distance, y=Speeding)) +   
 geom\_point(aes(color = .cluster), alpha =.8) +   
 facet\_wrap(~k)  
p2



## Task 5

After looking at this graph the best k value appears to be 4, as that is the bottom of the elbow on the graph.

ggplot(clusterings, aes(k, tot.withinss)) +   
 geom\_line() +  
 geom\_point()



## Task 6

The clusters shown in the visualization below seem to make the most sense. The shorter and longer distances are broken into different clusters and then the higher % of time speeding over 5mph is broken up into different clusters as well. This creates 4 different clusters which seems to fit the data points logically. If you added anymore clusters I think clusters would start to become too similar to other clusters and thus the data points would be similar across two clusters.

set.seed(64)  
clusts=   
 tibble(k=4) %>%  
 mutate(kclust=map(k, ~kmeans(trucks\_cleaned, .x)), tidied = map(kclust, tidy), glanced = map(kclust, glance), augmented = map(kclust, augment, trucks\_cleaned))  
  
clusts

## # A tibble: 1 x 5  
## k kclust tidied glanced augmented   
## <dbl> <list> <list> <list> <list>   
## 1 4 <kmeans> <tibble [4 × 5]> <tibble [1 × 4]> <tibble [4,000 × 3]>

set.seed(64)  
clusters = kmeans(trucks\_cleaned, 4)

trucks = augment(clusters, trucks)  
str(trucks)

## tibble [4,000 × 4] (S3: tbl\_df/tbl/data.frame)  
## $ Driver\_ID: num [1:4000] 3.42e+09 3.42e+09 3.42e+09 3.42e+09 3.42e+09 ...  
## $ Distance : num [1:4000] 71.2 52.5 64.5 55.7 54.6 ...  
## $ Speeding : num [1:4000] 28 25 27 22 25 10 20 8 34 19 ...  
## $ .cluster : Factor w/ 4 levels "1","2","3","4": 3 3 3 3 3 1 3 1 3 3 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. Driver\_ID = col\_double(),  
## .. Distance = col\_double(),  
## .. Speeding = col\_double()  
## .. )

p3 =   
 ggplot(trucks, aes(x=Distance, y=Speeding, color=factor(.cluster))) +   
 geom\_point()   
p3

