# Assignment 1

## Katherine Schumann

### Clustering

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

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

str(trucks)

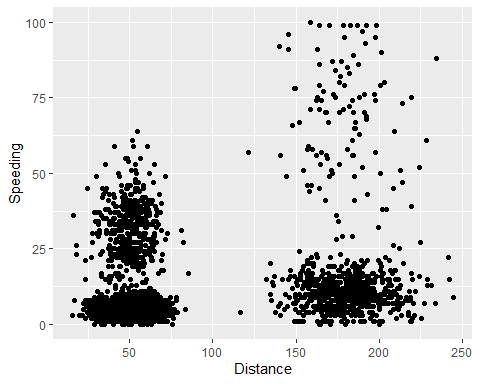
## tibble [4,000 x 3] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ Driver\_ID: num [1:4000] 3.42e+09 3.42e+09 3.42e+09 3.42e+09 3.42e+09 ...  
## $ Distance : num [1:4000] 71.2 52.5 64.5 55.7 54.6 ...  
## $ Speeding : num [1:4000] 28 25 27 22 25 10 20 8 34 19 ...  
## - attr(\*, "spec")=  
## ..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

summary(trucks)

## 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

### Task 1

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



Yes, there seems to be a natural devide between speeding and distance. There is a tighter cluster towards the bottom and a looser cluster towards the top of the speeding.

### Task 2

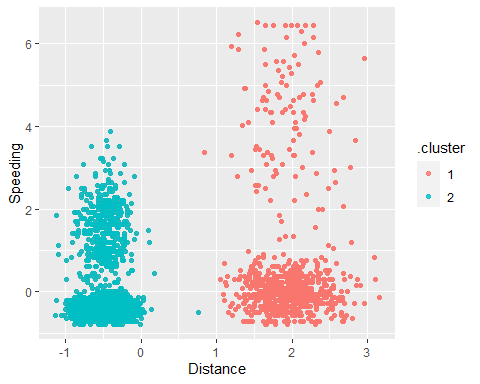
kmeans\_recipe = recipe(~ Distance + Speeding, trucks)   
  
truck\_step1 = kmeans\_recipe %>%   
 step\_scale(all\_numeric()) %>%  
 step\_center(all\_numeric())   
  
truck\_step1 = prep(truck\_step1, trucks) #prepares the recipe  
  
truck\_cleaned = bake(truck\_step1, trucks) #applies the recipe and yields a data frame

### Task 3

set.seed(64)  
clusts =   
 tibble(k = 1:2) %>%  
 mutate(  
 kclust = map(k, ~kmeans(truck\_cleaned, .x)),  
 tidied = map(kclust, tidy),  
 glanced = map(kclust, glance),  
 augmented = map(kclust, augment, truck\_cleaned)  
 )  
  
clusts

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

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

p1 =   
 ggplot(assignments, aes(x = Distance, y = Speeding)) +  
 geom\_point(aes(color = .cluster), alpha = 0.8)  
p1

The clusters depicted above are the closest cluster. I agree that the split between these two groups is clear and decisive.

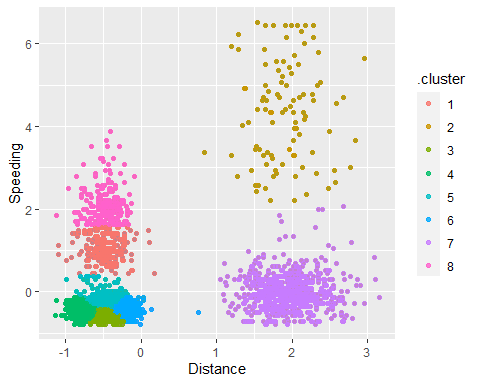
### Task 4

set.seed(412)  
clusts1 =   
 tibble(k = 1:8) %>%  
 mutate(  
 kclust = map(k, ~kmeans(truck\_cleaned, .x)),  
 tidied = map(kclust, tidy),  
 glanced = map(kclust, glance),  
 augmented = map(kclust, augment, truck\_cleaned)  
 )  
  
clusts1

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

clusters1 =   
 clusts1 %>%  
 unnest(cols = c(tidied))  
  
assignments1 =   
 clusts1 %>%   
 unnest(cols = c(augmented))  
  
clusterings1 =   
 clusts1 %>%  
 unnest(cols = c(glanced))

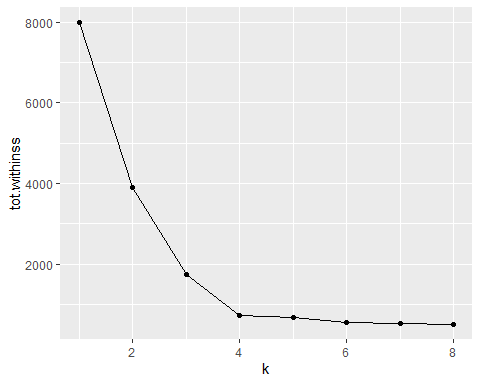
p2 =   
 ggplot(assignments1, aes(x = Distance, y = Speeding)) +  
 geom\_point(aes(color = .cluster), alpha = 0.8)  
p2



I would say the first graph because some of the clusters at the bottom of the chart with the smaller distances are very close and almost indistinguishable.

### Task 5

ggplot(clusterings1, aes(k, tot.withinss)) +  
 geom\_line() +  
 geom\_point()



Four appears to be the best because that is the bend in the elbow. Also, I think that it seems like it would fit the data in a better manner.

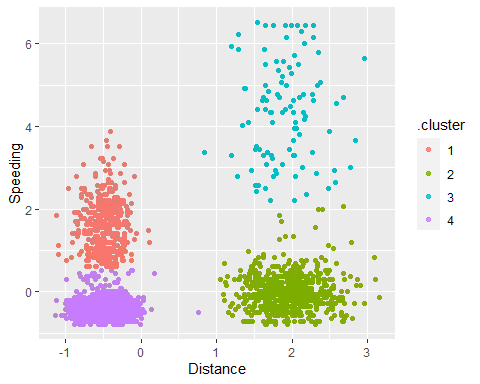
### Task 6

set.seed(64)  
clusts3 =   
 tibble(k = 1:4) %>%  
 mutate(  
 kclust = map(k, ~kmeans(truck\_cleaned, .x)),  
 tidied = map(kclust, tidy),  
 glanced = map(kclust, glance),  
 augmented = map(kclust, augment, truck\_cleaned)  
 )  
  
clusts3

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

clusters3 =   
 clusts3 %>%  
 unnest(cols = c(tidied))  
  
assignments3 =   
 clusts3 %>%   
 unnest(cols = c(augmented))  
  
clusterings3 =   
 clusts3 %>%  
 unnest(cols = c(glanced))

p3 =   
 ggplot(assignments3, aes(x = Distance, y = Speeding)) +  
 geom\_point(aes(color = .cluster), alpha = 0.8)  
p3



I anticipated that this is the best fit for the data. It is a bit more scattered for high distance and high speeding, but it is still clustered more than the dense group at the lower end of speeding.