# BAN 502 Module 6 Assignment 1

## Kristian Marlowe

### Clustering

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

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

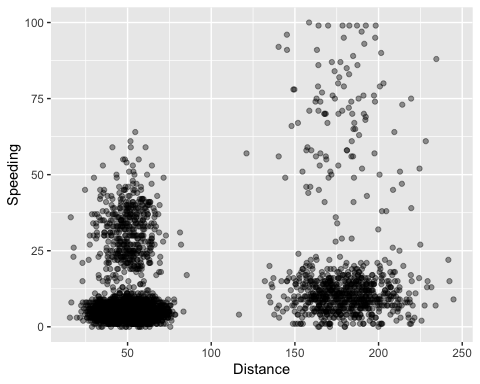
summary(trucks)

## 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

Task 1 will have the relationship between distance and speed plotted.

ggplot(trucks, aes(Distance, Speeding)) +  
 geom\_point(alpha = 0.4)



In this graph, we can see about 4 natural clusters. We can see one from a distance of around 0-100 and a speed of about 0 - 15. There is another cluster from distances of about 120 - 250 with speeds from 0 - 25. These are the low-speed clusters. Additionally, there are more high speed clusters with one being from about 0 – 100 distance and about 15- 60 speed. The next one would be from about 120 – 250 on the distance scale and about 25 – 100 on the speed scale.

## Task 2

Task 2 seeks to create a new data frame “trucks\_cleaned” that contains the scaled and centered variables of “Distance” and “Speeding”.

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

summary(trucks\_cleaned)

## 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

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 3

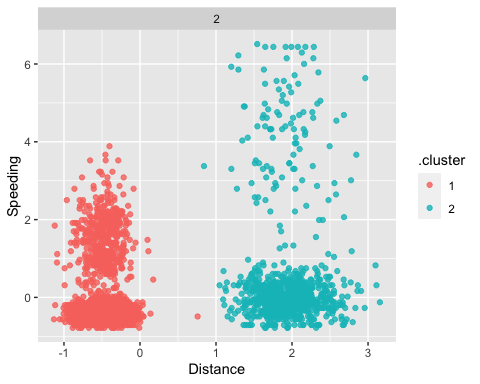
In task 3, the k-means clustering with two clusters will be created, visualized, and commented on.

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]>

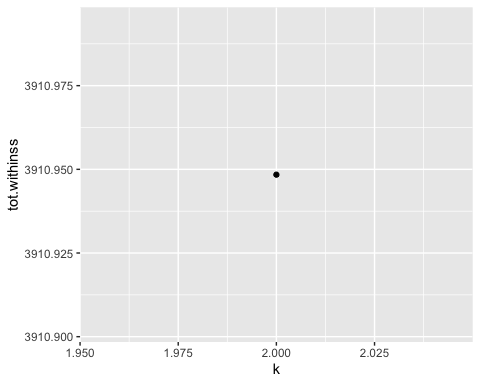
clusters =   
 clusts %>%  
 unnest(cols = c(tidied))  
  
assignments =   
 clusts %>%   
 unnest(cols = c(augmented))  
  
clusterings =   
 clusts %>%  
 unnest(cols = c(glanced))

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



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

## geom\_path: Each group consists of only one observation. Do you need to adjust  
## the group aesthetic?



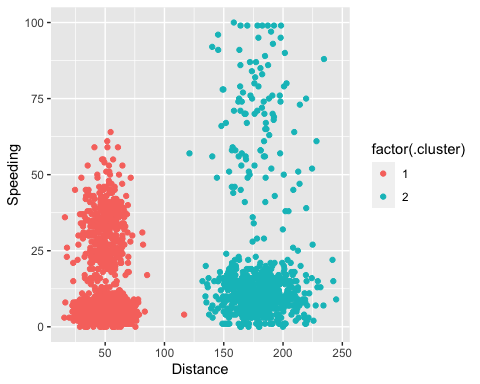
As expected, this point is at 2 since we set the k value to only be equal to 2.

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()  
## .. )

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



From the cluster plot we can see the 2 distance clusters - one from the left side in red and one from the right side in blue.The distance in cluster 1 spans from 0 - ~120 with speeds from 0 - ~70. The second cluster has distances from ~125 - 250 with speeds from 0- 100. In my opinion there could be more clusters broken up to more accurately show the separation of speeds especially in the second cluster. There are two points that seem to be on the borderline of being in either cluster based on distance in the ~125 range, but could be in different clusters due to the difference in speeds with one point being at ~7 and the other being at ~60.

## Task 4

Task 4 will create a visualization on how the clusters appear from values of k from 1 to 8.

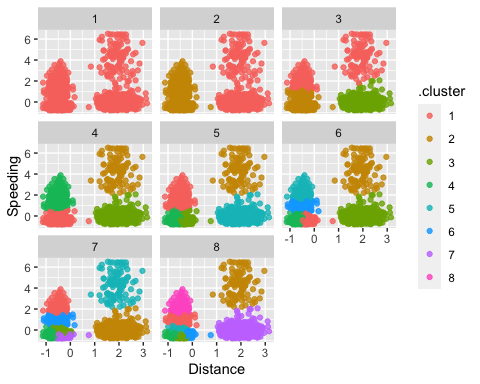
Perform k-means clustering with a pre-specified number of clusters. We use the scaled and dummied data frame.

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))

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

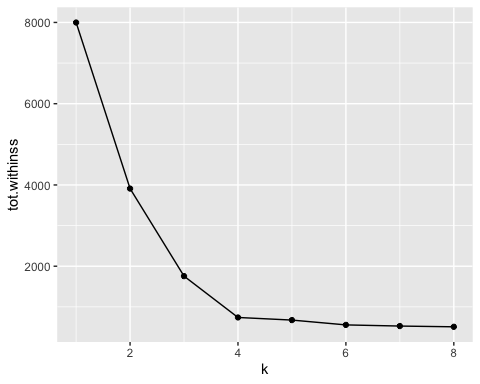


Based on the graph above, the value of k that could best represent the data would be 4. This value to me allows the breakdown of the 2 separate groups on the left and right to be broken up into high and low. This fits the data more appropriately without points seeming to overlap or over fit. In k values 1-3, these don’t seem to group the data as well without some overlap or questionable groupings. In the k values of 5-8 I think these overfit the data and there are some groups that are separated too much that takes away from some of the overall trends.

## Task 5

In task 5, the goal is to create a plot of k versus within cluster sum of squares to find the best value of k.

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



Based on the graph above, the value of k that could best represent the data would be 4. This value is found in the ‘elbow’ of the curve.

## Task 6

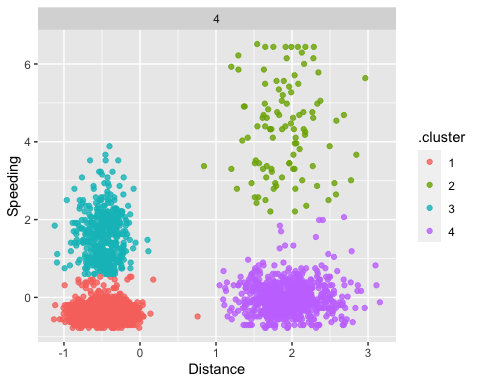
Task 6 seeks to repeat task 4 based on the k-value of 4 identified previously.

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]>

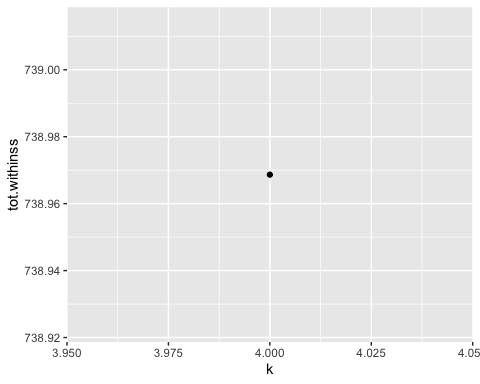
clusters =   
 clusts %>%  
 unnest(cols = c(tidied))  
  
assignments =   
 clusts %>%   
 unnest(cols = c(augmented))  
  
clusterings =   
 clusts %>%  
 unnest(cols = c(glanced))

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



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

## geom\_path: Each group consists of only one observation. Do you need to adjust  
## the group aesthetic?



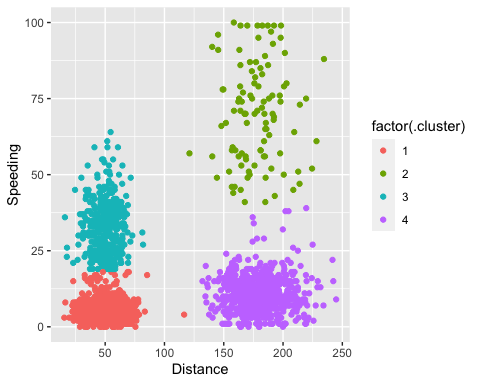
As expected, this value is at 4 since we specified the k value to only be 4.

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()  
## .. )

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



The clustering of the data into 4 clusters seems to appropriately model the data. We can see clear distinctions in the speeds and distances. Cluster 1 hasshorter distances from around 0 to mainly 100 with a point extending to the 125 range and has slower speeds from 0 to about 20. Cluster 2 has longer distances from about 125 to 250 with higher speeds from about 45 to 100. In cluster 3 we have the shorter distances (from 0 to about 80) with higher speeds (from about 20 to 60). Cluster 4 has the higher distances (about 125 to 250) with the slower speeds of about 0 to 45. The points have a pretty close knit range and we are able to more accurately place points in a group that fits them more appropriately without overfitting.