# Module 6: Assignment 1 – Unsupervised Learning Assignment

## BAN 502, Predictive Analytics

### Je’Kolby Worthy

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.0.5 ✓ dplyr 1.0.3  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.0

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

library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 0.1.2 ──

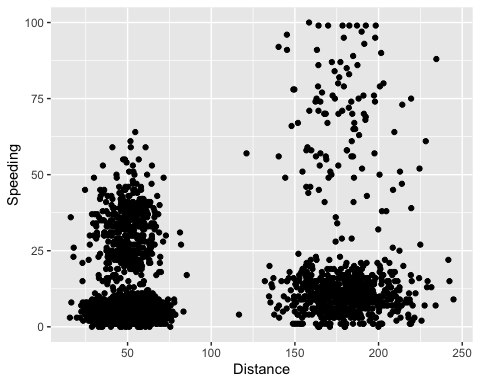
## ✓ broom 0.7.3 ✓ recipes 0.1.15  
## ✓ dials 0.0.9 ✓ rsample 0.0.8   
## ✓ infer 0.5.4 ✓ tune 0.1.2   
## ✓ modeldata 0.1.0 ✓ workflows 0.2.1   
## ✓ parsnip 0.1.5 ✓ yardstick 0.0.7

## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## x scales::discard() masks purrr::discard()  
## x dplyr::filter() masks stats::filter()  
## x recipes::fixed() masks stringr::fixed()  
## x dplyr::lag() masks stats::lag()  
## x yardstick::spec() masks readr::spec()  
## x recipes::step() masks stats::step()

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

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

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

 While looking at the relationship between Distance and Speeding, it seems as if Drivers who drive at least 50 miles tend to go within the 25-50mph category compared to drivers who drive long distance seem to go at a slower a slower rate. Yes, there appears to be a natural clustering of drivers that are speeding.

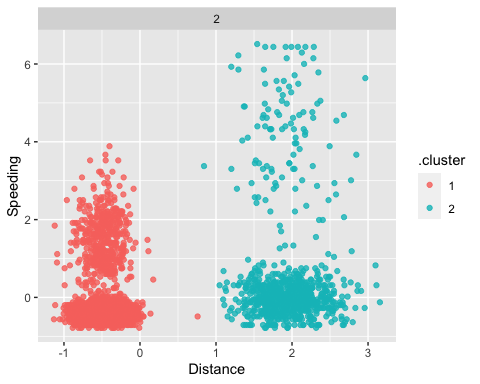
kmeans\_recipe = recipe(~ Speeding + Distance, trucks)  
  
trucks\_dummy = kmeans\_recipe %>%  
 step\_scale(all\_numeric()) %>%  
 step\_center(all\_numeric())  
  
trucks\_dummy = prep(trucks\_dummy, trucks)  
trucks\_cleaned = bake(trucks\_dummy, trucks)

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



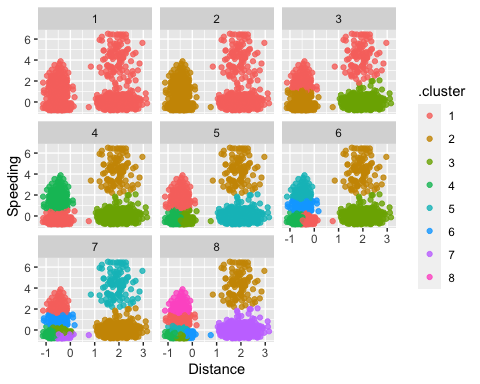
There are no outliers in the clusters, which are equally distributed between 1 and 2.

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

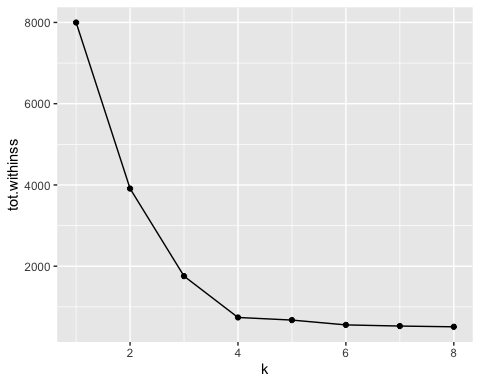
## # A tibble: 8 x 5  
## k kclust tidied glanced augmented   
## <int> <list> <list> <list> <list>   
## 1 1 <kmeans> <tibble [1 × 5]> <tibble [1 × 4]> <tibble [4,000 × 3]>  
## 2 2 <kmeans> <tibble [2 × 5]> <tibble [1 × 4]> <tibble [4,000 × 3]>  
## 3 3 <kmeans> <tibble [3 × 5]> <tibble [1 × 4]> <tibble [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 = 0.8) + facet\_wrap(~ k)  
p2



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



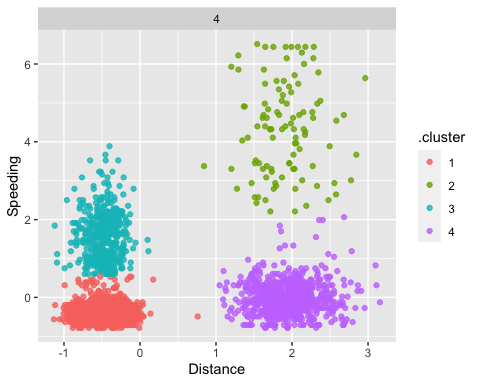
Since this is where you see the bend in the curve, 4 tends to be the best value.

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))

p3 =   
 ggplot(assignments, aes(x = Distance, y = Speeding)) +   
 geom\_point(aes(color = .cluster), alpha = 0.8) + facet\_wrap(~ k)  
p3



This cluster is superior to all others because it explicitly divides the clusters into four groups without including any outlier variables.