# Module 6 Assignment 1

## Underwood, Katie

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.4 v dplyr 1.0.2  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.4.0 v 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 --

## v broom 0.7.2 v recipes 0.1.15  
## v dials 0.0.9 v rsample 0.0.8   
## v infer 0.5.4 v tune 0.1.2   
## v modeldata 0.1.0 v workflows 0.2.1   
## v parsnip 0.1.5 v 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()

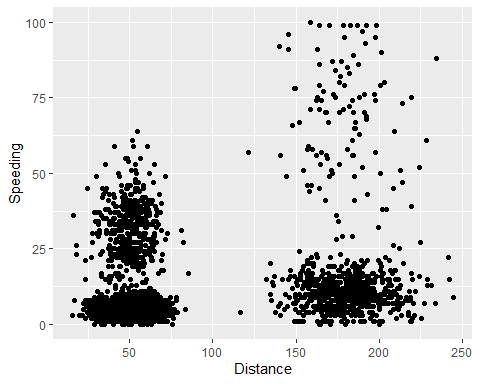
trucks <- read\_csv("trucks.csv")

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

### Task 1

There does seem to be some natural grouping of the data, Distance between 80 and 120 (roughly) doesn’t seem to have data points.

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

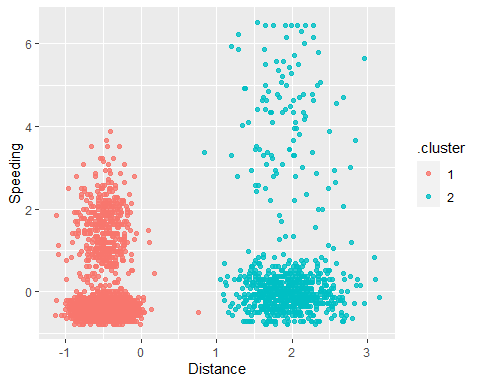


### Task 2

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

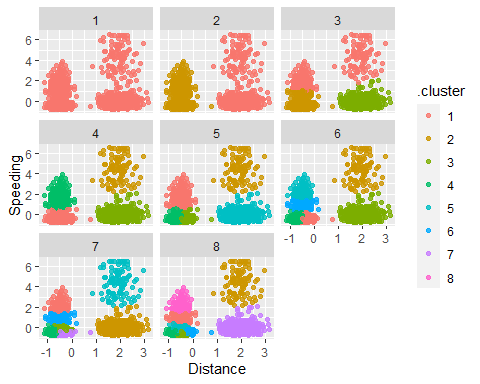
set.seed(64)  
clusts =  
 tibble(k= 2) %>%  
 mutate(  
 kclust = map(k, ~kmeans(trucks\_cleaned, .x)),  
 tidied = map(kclust, tidy),  
 glanced = map(kclust, glance),  
 augmented = map(kclust, augment, trucks\_cleaned)  
 )  
  
clusters =  
 clusts %>%  
 unnest(cols = c(tidied))  
  
assignments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



### Task 4

2 clusters seems to be appropriate for this data set. If for some reason there was a desire to have more clusters, 3 or 4 would also be a viable option based on how the charts look.

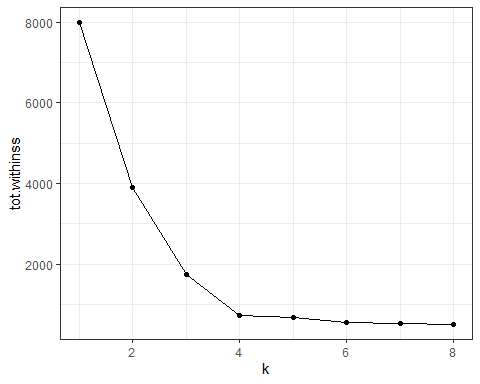
set.seed(412)  
clusts =  
 tibble(k= 1:8) %>%  
 mutate(  
 kclust = map(k, ~kmeans(trucks\_cleaned, .x)),  
 tidied = map(kclust, tidy),  
 glanced = map(kclust, glance),  
 augmented = map(kclust, augment, trucks\_cleaned)  
 )  
  
clusters =  
 clusts %>%  
 unnest(cols = c(tidied))  
  
assignments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



### Task 5

Based on this visual it looks like 4 clusters may be the optimal choice for this data set.

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



### Task 6

Resulting clusters make sense to me. The clusters are split into “high-high”, “high-low”, “low-high”, “low-low” combinations of Speeding and Distance.

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)  
 )  
  
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

