Mod6\_Assign1-Quiz

Grace Williams

2023-06-22

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 1.1.0 ──  
## ✔ broom 1.0.5 ✔ rsample 1.1.1  
## ✔ dials 1.2.0 ✔ tune 1.1.1  
## ✔ infer 1.0.4 ✔ workflows 1.1.3  
## ✔ modeldata 1.1.0 ✔ workflowsets 1.0.1  
## ✔ parsnip 1.1.0 ✔ yardstick 1.2.0  
## ✔ recipes 1.0.6

## Warning: package 'broom' was built under R version 4.3.1

## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## ✖ scales::discard() masks purrr::discard()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ recipes::fixed() masks stringr::fixed()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ yardstick::spec() masks readr::spec()  
## ✖ recipes::step() masks stats::step()  
## • Learn how to get started at https://www.tidymodels.org/start/

library(readr)  
library(cluster)  
library(factoextra)

## Warning: package 'factoextra' was built under R version 4.3.1

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

trucks <- read\_csv("trucks-1.csv")

## Rows: 4000 Columns: 3  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## dbl (3): Driver\_ID, Distance, Speeding  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

na.omit(trucks)

## # A tibble: 4,000 × 3  
## Driver\_ID Distance Speeding  
## <dbl> <dbl> <dbl>  
## 1 3423311935 71.2 28  
## 2 3423313212 52.5 25  
## 3 3423313724 64.5 27  
## 4 3423311373 55.7 22  
## 5 3423310999 54.6 25  
## 6 3423313857 41.9 10  
## 7 3423312432 58.6 20  
## 8 3423311434 52.0 8  
## 9 3423311328 31.2 34  
## 10 3423312488 44.3 19  
## # ℹ 3,990 more rows

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

trucks\_scaled = scale(trucks)  
summary(trucks\_scaled)

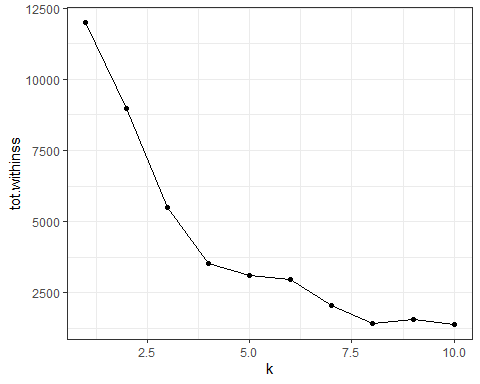
## Driver\_ID Distance Speeding   
## Min. :-1.7314 Min. :-1.1319 Min. :-0.7821   
## 1st Qu.:-0.8657 1st Qu.:-0.5759 1st Qu.:-0.4903   
## Median : 0.0000 Median :-0.4248 Median :-0.3444   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.8657 3rd Qu.:-0.1947 3rd Qu.:-0.1255   
## Max. : 1.7314 Max. : 3.1560 Max. : 6.5127

set.seed(1234)  
clusts =   
 tibble(k = 1:10) %>% #try from 1 to 10 clusters  
 mutate(  
 kclust = map(k, ~kmeans(trucks\_scaled, .x)),  
 tidied = map(kclust, tidy),  
 glanced = map(kclust, glance),  
 augmented = map(kclust, augment, trucks\_scaled)  
 )  
  
clusts

## # A tibble: 10 × 5  
## k kclust tidied glanced augmented   
## <int> <list> <list> <list> <list>   
## 1 1 <kmeans> <tibble [1 × 6]> <tibble [1 × 4]> <tibble [4,000 × 4]>  
## 2 2 <kmeans> <tibble [2 × 6]> <tibble [1 × 4]> <tibble [4,000 × 4]>  
## 3 3 <kmeans> <tibble [3 × 6]> <tibble [1 × 4]> <tibble [4,000 × 4]>  
## 4 4 <kmeans> <tibble [4 × 6]> <tibble [1 × 4]> <tibble [4,000 × 4]>  
## 5 5 <kmeans> <tibble [5 × 6]> <tibble [1 × 4]> <tibble [4,000 × 4]>  
## 6 6 <kmeans> <tibble [6 × 6]> <tibble [1 × 4]> <tibble [4,000 × 4]>  
## 7 7 <kmeans> <tibble [7 × 6]> <tibble [1 × 4]> <tibble [4,000 × 4]>  
## 8 8 <kmeans> <tibble [8 × 6]> <tibble [1 × 4]> <tibble [4,000 × 4]>  
## 9 9 <kmeans> <tibble [9 × 6]> <tibble [1 × 4]> <tibble [4,000 × 4]>  
## 10 10 <kmeans> <tibble [10 × 6]> <tibble [1 × 4]> <tibble [4,000 × 4]>

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

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



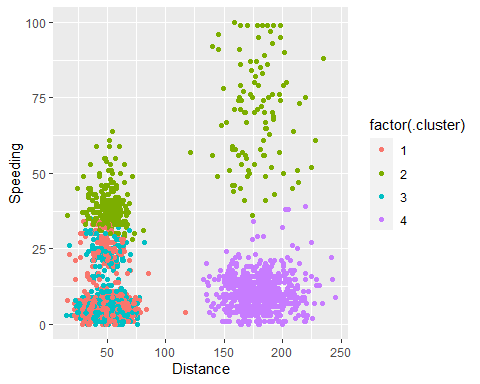
cust\_clust = kmeans(trucks\_scaled, centers = 4)  
cust\_clust

## K-means clustering with 4 clusters of sizes 1522, 327, 1457, 694  
##   
## Cluster means:  
## Driver\_ID Distance Speeding  
## 1 0.86802507 -0.4861001 -0.29880377  
## 2 0.01878405 0.2963155 2.78654635  
## 3 -0.87423847 -0.4887583 -0.30533176  
## 4 -0.07710531 1.9525505 -0.01664689  
##   
## Clustering vector:  
## [1] 2 1 1 3 3 1 3 3 2 1 2 2 1 3 2 1 2 3 3 1 3 2 3 1 2 2 3 3 1 2 2 2 2 2 3 2 3  
## [38] 2 2 2 2 1 3 2 3 3 3 2 2 3 1 3 1 2 2 2 2 3 3 2 3 3 1 2 3 2 2 2 2 2 1 3 2 2  
## [75] 1 2 2 3 2 2 1 3 3 1 1 3 3 1 2 1 3 3 2 3 2 1 2 1 2 3 3 3 3 2 1 2 2 3 3 1 2  
## [112] 3 3 3 1 1 2 3 3 2 3 2 2 3 2 1 1 1 2 3 2 2 2 1 2 2 2 1 1 3 1 2 2 2 1 2 3 2  
## [149] 1 1 2 2 1 2 3 2 3 3 3 2 2 1 3 2 1 1 2 2 3 1 3 1 1 3 2 2 2 3 2 3 3 2 2 1 2  
## [186] 1 2 1 3 2 3 1 3 1 1 2 3 2 2 2 2 2 2 1 3 2 3 2 1 1 1 2 1 3 1 1 3 1 3 2 2 1  
## [223] 2 3 2 1 2 1 3 1 2 1 2 2 1 1 3 2 3 1 2 2 2 2 2 1 3 3 3 1 2 2 2 3 1 1 2 2 3  
## [260] 2 2 2 1 3 2 3 2 2 3 3 2 2 1 2 2 1 3 2 1 1 2 2 1 1 1 2 1 1 1 1 2 2 2 3 2 3  
## [297] 2 3 2 2 3 2 2 3 3 2 1 1 1 1 3 2 1 2 1 2 2 1 2 2 1 2 3 3 1 3 2 2 3 2 2 2 2  
## [334] 2 2 3 2 1 2 2 3 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 3 2 2 1 2 2 3 2 1 2 2  
## [371] 2 2 3 1 2 3 2 2 3 3 1 2 3 3 2 1 2 1 1 2 1 3 3 3 2 1 2 2 2 1 3 2 2 3 2 2 3  
## [408] 2 3 3 2 1 1 2 1 1 1 3 2 1 2 3 1 1 2 1 1 3 2 2 3 1 3 3 1 1 1 1 1 1 1 1 1 1  
## [445] 1 3 2 2 2 1 2 2 3 2 3 2 1 1 1 2 2 2 3 2 2 2 3 2 1 2 1 3 2 2 1 2 2 2 1 2 3  
## [482] 3 3 1 3 1 3 1 3 3 3 3 3 1 3 3 3 3 3 3 1 1 1 1 1 1 1 3 1 3 3 3 1 3 3 3 1 3  
## [519] 1 3 1 1 3 1 3 3 1 3 3 3 1 3 3 1 1 1 3 3 3 1 1 1 1 3 3 1 1 3 3 3 3 3 1 1 3  
## [556] 3 3 3 3 3 3 1 3 1 1 1 3 1 1 1 3 1 3 3 1 3 3 3 1 3 1 1 3 1 3 1 1 1 3 3 1 1  
## [593] 3 1 1 3 1 1 3 1 1 3 3 3 1 1 3 1 1 3 3 3 1 1 3 3 3 1 1 1 3 1 3 1 3 1 1 1 3  
## [630] 3 3 3 1 1 3 3 3 3 1 3 1 1 1 1 3 3 3 1 3 3 1 1 3 3 1 1 3 1 3 1 1 3 1 3 3 1  
## [667] 3 1 1 3 3 1 1 1 1 3 3 3 1 1 3 3 3 3 3 3 3 1 3 1 3 3 3 3 1 3 3 1 3 1 3 3 1  
## [704] 3 1 1 3 3 3 3 3 3 3 3 1 3 1 3 3 1 1 1 3 3 1 1 3 3 1 3 1 1 1 3 3 1 3 3 1 1  
## [741] 1 3 1 3 1 3 1 1 1 3 3 3 3 1 3 1 3 1 3 3 3 3 1 1 1 1 3 1 1 3 1 3 3 3 1 3 1  
## [778] 3 3 1 1 3 3 3 3 1 3 1 1 1 1 3 3 1 1 3 3 3 3 1 3 3 1 1 3 1 3 1 3 1 1 3 1 3  
## [815] 1 1 1 3 1 3 1 1 1 3 3 1 1 1 3 1 1 3 3 3 3 1 3 1 1 1 1 3 3 3 1 3 1 1 1 3 1  
## [852] 3 1 1 1 3 1 3 1 1 1 1 3 3 3 3 1 3 3 1 3 1 1 3 1 1 1 1 3 1 3 1 1 3 3 3 3 1  
## [889] 1 1 1 3 1 3 1 3 3 1 1 3 1 1 1 3 3 3 1 3 3 1 1 1 3 1 3 3 3 1 3 1 1 3 3 1 1  
## [926] 1 3 3 3 3 1 3 3 3 1 3 1 3 1 1 3 1 3 3 1 3 1 3 1 1 3 1 3 3 3 3 3 3 1 1 1 1  
## [963] 1 1 3 1 3 3 1 1 1 1 3 1 1 3 3 3 3 1 1 3 1 3 1 3 3 1 1 3 3 3 3 3 1 1 1 1 3  
## [1000] 1 3 1 3 3 3 3 3 1 1 1 1 1 3 1 1 3 3 1 1 1 1 3 1 3 1 1 3 3 3 3 3 3 1 1 3 3  
## [1037] 1 3 1 3 1 3 3 3 1 1 3 1 1 3 3 3 1 3 1 3 3 1 1 1 3 1 3 1 1 3 1 1 1 3 1 1 1  
## [1074] 1 1 1 1 1 3 1 3 1 3 1 3 1 1 1 1 1 1 1 1 1 3 3 1 1 3 3 3 3 1 1 3 3 3 1 1 1  
## [1111] 3 1 3 1 1 3 1 3 1 3 3 3 3 1 3 1 1 1 3 1 1 3 1 3 1 1 3 1 1 1 1 3 1 1 3 3 3  
## [1148] 3 1 3 1 1 1 3 3 1 3 3 3 3 3 1 1 3 1 1 1 3 1 1 1 3 1 3 3 3 3 1 3 1 3 1 3 1  
## [1185] 1 1 1 3 1 3 3 3 1 3 1 1 3 1 3 3 3 3 3 1 1 1 3 1 1 3 3 1 1 3 1 1 3 3 1 3 3  
## [1222] 3 1 3 1 1 1 1 1 1 3 3 1 1 1 1 3 1 3 1 1 3 3 1 1 3 1 1 3 1 1 1 3 3 1 3 3 1  
## [1259] 1 1 1 3 3 3 3 3 3 3 1 3 1 3 3 1 3 3 3 1 1 1 1 3 3 1 1 3 3 3 3 3 3 1 3 1 1  
## [1296] 1 1 3 3 1 3 3 1 3 1 3 1 1 3 1 3 1 1 3 1 3 3 3 1 3 1 3 1 1 1 1 3 3 1 1 1 3  
## [1333] 1 3 1 3 1 3 1 1 1 3 1 3 1 1 3 3 3 1 3 3 1 3 1 3 1 1 3 1 1 3 1 1 1 3 1 1 3  
## [1370] 1 1 1 1 1 3 1 3 3 3 1 3 3 3 1 1 3 1 1 3 1 3 3 3 3 3 3 1 3 3 3 1 3 1 3 3 3  
## [1407] 1 3 1 1 1 3 1 3 1 3 3 1 3 3 3 1 3 3 3 3 3 3 3 1 1 1 3 1 3 3 3 3 3 1 1 3 1  
## [1444] 1 1 3 3 3 3 3 1 1 3 1 1 3 1 3 1 1 1 1 1 3 3 1 1 3 1 3 3 3 3 1 1 3 1 3 1 3  
## [1481] 1 3 1 1 3 1 3 3 1 1 1 1 1 1 3 1 3 1 3 3 1 1 1 1 3 3 3 1 1 1 3 1 1 3 3 3 1  
## [1518] 3 1 1 3 1 1 3 1 1 1 3 3 3 1 1 3 1 1 3 3 1 1 1 3 1 3 1 1 3 3 1 1 1 1 1 1 3  
## [1555] 1 1 1 3 1 1 1 1 3 1 1 3 3 3 3 3 3 1 3 1 1 1 3 1 3 3 1 1 3 1 3 3 1 1 1 3 1  
## [1592] 3 1 3 1 3 1 1 3 1 3 1 1 1 3 1 1 3 1 1 1 1 3 1 1 3 3 3 1 1 3 1 3 1 1 3 1 1  
## [1629] 3 3 1 3 1 3 3 1 3 1 1 1 3 1 1 1 3 3 1 1 1 3 1 3 3 3 1 3 1 3 1 3 1 3 1 1 3  
## [1666] 3 1 1 3 3 1 3 3 3 3 1 3 3 1 3 3 3 3 3 3 1 1 3 3 1 1 1 1 1 1 3 1 1 1 3 1 1  
## [1703] 3 1 1 3 1 1 1 3 3 1 3 1 1 3 3 3 3 1 1 1 3 1 3 3 1 1 3 1 3 3 1 1 1 1 3 1 3  
## [1740] 1 1 3 1 3 1 3 1 3 3 3 1 3 3 1 1 3 1 3 3 1 1 3 1 1 1 1 1 1 3 3 1 1 3 1 3 1  
## [1777] 1 1 1 1 3 1 1 1 1 1 3 3 1 1 1 3 3 3 1 3 3 3 1 1 1 1 3 3 1 1 3 3 1 3 3 1 1  
## [1814] 1 1 3 3 1 3 3 1 1 1 1 1 3 3 3 1 3 3 3 1 3 3 3 3 3 3 1 1 3 1 3 3 3 1 3 3 1  
## [1851] 1 1 3 1 1 3 3 3 1 1 3 3 3 1 1 1 1 1 3 1 3 3 1 3 1 3 1 3 3 3 1 3 3 3 1 3 3  
## [1888] 1 1 1 1 1 1 1 3 1 3 1 1 1 3 1 1 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 1 3 1 3  
## [1925] 3 3 1 3 1 1 3 1 1 1 3 3 3 1 1 3 3 1 1 3 1 3 3 1 1 3 3 3 1 1 1 3 3 1 1 1 3  
## [1962] 1 3 1 3 1 3 3 3 1 1 3 3 3 1 3 1 3 1 1 3 1 3 3 1 1 3 3 3 1 1 1 1 1 1 1 3 3  
## [1999] 1 3 1 3 1 1 1 1 3 3 3 3 3 3 3 1 3 3 1 1 1 3 1 3 3 1 1 1 3 1 1 3 3 3 1 1 1  
## [2036] 1 3 1 3 1 1 1 3 1 3 3 1 3 1 1 1 1 3 3 1 3 3 3 3 3 1 3 3 1 3 3 3 3 1 1 3 1  
## [2073] 1 3 3 3 1 1 1 1 1 1 3 1 3 1 3 1 3 3 1 3 3 1 3 1 1 3 3 3 1 3 1 3 3 3 3 3 1  
## [2110] 3 1 3 1 1 1 3 1 1 1 1 1 3 3 3 3 1 1 3 3 3 3 1 1 1 1 3 3 1 1 3 1 1 3 1 1 1  
## [2147] 1 1 1 3 3 1 1 3 1 3 1 1 1 3 3 1 3 3 3 1 1 1 1 1 3 1 3 1 1 1 1 1 1 3 3 1 1  
## [2184] 3 1 3 1 1 3 1 1 3 1 3 1 1 1 1 1 1 3 1 3 1 3 1 3 1 1 3 1 1 1 3 3 3 3 3 1 3  
## [2221] 3 1 1 3 1 1 1 1 3 3 1 1 1 3 3 1 3 3 3 3 1 3 1 3 1 1 1 3 1 3 3 1 1 3 3 1 3  
## [2258] 1 3 3 3 3 3 3 1 3 3 1 1 3 3 3 1 1 1 1 1 1 3 3 1 1 3 1 3 1 3 3 1 3 1 3 1 1  
## [2295] 1 3 1 1 1 3 1 3 1 1 3 3 3 3 3 3 3 1 3 3 1 3 1 1 3 3 3 3 1 3 1 1 1 1 3 3 3  
## [2332] 1 1 3 3 3 3 1 3 1 1 3 3 3 1 3 1 1 1 3 3 1 3 1 1 3 1 1 3 3 3 3 3 3 1 3 1 3  
## [2369] 3 1 1 3 1 3 1 1 3 1 1 1 3 1 3 1 1 1 3 3 1 1 1 3 1 1 1 3 1 1 1 1 3 1 3 1 1  
## [2406] 1 1 1 1 1 3 1 3 1 1 3 3 1 1 3 3 3 1 1 1 3 1 3 1 3 1 1 1 3 3 1 1 1 1 3 3 3  
## [2443] 3 1 1 3 3 3 3 1 1 3 3 1 1 3 3 3 3 1 3 1 3 3 3 3 1 3 3 3 1 1 3 1 1 1 3 1 3  
## [2480] 1 3 3 1 3 3 1 3 1 3 1 1 3 1 1 3 3 3 1 1 3 1 1 3 3 3 3 1 3 1 3 1 1 3 3 1 1  
## [2517] 3 3 1 3 1 1 3 1 3 1 3 1 3 1 1 1 3 1 1 3 1 1 1 3 1 1 1 1 1 1 1 3 1 3 3 3 3  
## [2554] 3 3 1 3 1 1 3 1 3 3 1 1 1 1 3 3 1 3 1 3 3 1 1 3 1 1 1 1 3 1 1 3 3 1 1 1 3  
## [2591] 1 1 3 1 1 3 3 3 3 1 3 3 3 1 3 3 3 1 1 3 1 1 1 1 3 1 1 1 1 1 3 1 1 1 3 3 1  
## [2628] 3 1 3 3 3 1 3 3 3 1 1 3 3 3 3 1 1 3 3 3 3 1 3 3 3 1 3 1 1 1 3 3 3 3 1 3 1  
## [2665] 3 1 3 1 1 1 1 3 1 1 1 3 3 1 1 3 3 1 1 3 3 1 1 3 1 3 3 3 3 3 1 3 1 1 1 3 1  
## [2702] 3 1 1 3 3 3 1 1 1 3 1 3 1 3 1 3 3 3 1 1 3 3 1 3 3 3 1 3 3 1 3 1 1 3 3 1 3  
## [2739] 3 3 3 1 3 3 3 3 3 3 1 1 3 1 1 3 1 3 3 1 1 3 1 1 3 1 3 1 1 3 3 3 1 1 1 3 3  
## [2776] 1 3 1 3 3 3 3 1 1 3 1 3 1 3 1 1 3 1 3 3 3 1 1 3 3 1 1 1 1 1 3 1 3 3 1 3 1  
## [2813] 1 1 1 1 3 3 3 1 3 1 3 3 1 1 1 3 3 1 3 1 1 3 1 1 1 1 3 1 1 3 1 1 3 1 1 1 3  
## [2850] 3 3 3 3 1 3 3 1 3 3 3 1 3 3 1 1 1 3 1 1 3 3 3 1 1 1 3 1 1 3 1 1 3 3 1 1 3  
## [2887] 1 1 3 1 3 3 1 3 1 3 3 3 3 1 3 3 1 1 1 1 1 1 3 1 3 3 3 1 3 3 1 1 3 1 1 1 3  
## [2924] 3 1 3 3 3 1 3 1 3 1 3 1 1 3 1 1 1 1 1 1 1 3 3 3 3 3 1 3 3 3 3 3 3 3 1 1 3  
## [2961] 3 1 1 3 3 1 3 3 1 3 1 1 3 3 1 1 1 1 1 1 1 1 1 3 1 1 3 3 3 1 3 3 3 1 3 3 3  
## [2998] 1 3 3 1 1 1 1 1 1 1 3 3 1 1 1 1 1 3 3 3 1 1 3 3 3 1 1 1 1 3 3 3 3 3 3 3 1  
## [3035] 1 1 1 1 3 3 3 3 1 1 3 3 1 1 3 3 1 1 3 1 3 1 1 3 1 1 3 3 1 1 3 1 3 3 1 1 3  
## [3072] 1 3 1 1 1 1 3 1 1 3 1 3 3 3 1 1 1 1 1 1 1 1 3 1 3 3 3 3 3 1 3 3 3 1 1 3 3  
## [3109] 3 1 1 3 3 1 3 1 1 3 3 1 3 1 1 3 3 1 3 3 3 1 3 1 3 1 1 1 1 1 1 3 1 3 1 3 3  
## [3146] 1 1 3 3 1 1 3 1 1 3 1 1 1 3 3 3 1 3 1 3 3 3 3 3 1 3 3 3 3 1 3 1 1 3 1 3 3  
## [3183] 1 1 3 3 3 3 3 1 1 3 1 3 1 1 1 1 1 1 2 2 2 2 2 2 4 2 2 2 2 2 2 4 2 4 2 2 2  
## [3220] 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 4 2 4 2 2 2 2 2 2  
## [3257] 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 4 2 2 2 2 2 2 2 4 2 2 2 4  
## [3294] 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 4 2 2 2 2 2 4 4 4 4 4 4 4 4 4 4  
## [3331] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3368] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3405] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3442] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3479] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3516] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3553] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3590] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3627] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3664] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3701] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3738] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3775] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3812] 4 4 4 4 4 4 4 4 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3849] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3886] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3923] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3960] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## [3997] 4 4 4 4  
##   
## Within cluster sum of squares by cluster:  
## [1] 713.9785 1288.1896 647.0885 889.8846  
## (between\_SS / total\_SS = 70.5 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

trucks <- augment(cust\_clust, trucks)  
  
head(trucks)

## # A tibble: 6 × 4  
## Driver\_ID Distance Speeding .cluster  
## <dbl> <dbl> <dbl> <fct>   
## 1 3423311935 71.2 28 2   
## 2 3423313212 52.5 25 1   
## 3 3423313724 64.5 27 1   
## 4 3423311373 55.7 22 3   
## 5 3423310999 54.6 25 3   
## 6 3423313857 41.9 10 1

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



# Questions:

# 1. Which characteristics (select all that apply) of the relationship between Distance and Speeding seem most apparent?

## \*a. There appears to be more speeding among the drivers with smaller Distances

## \*b. The data points are arranged in what appear to be four clusters

## \*c. Longer distance drivers appear more likely to speed

## \*d. There are no well-defined clusters of data points

### b and c

# 2. What is the maximum value (to four decimal places) of the Distance variable in the scaled dataset?

### 3.1559

# 3. Which statement best describes the resulting clusters?

## \*a. Drivers with shorter distances are in one cluster and those with longer distances are in another

## \*b. Drivers with a higher proportion of speeding are in one cluster and those with a lower proportion of speeding are in another

## \*c. Neither of these statements apply to the resulting clusters

### a

## 4. Which value of k appears to be most appropriate for this data?

### 4

## 5. What number of clusters appears to be ideal based on this plot?

### 4

## 6. Which statements (select all that apply) appear to be most apparent about the clusters created in this question?

## \*a. One cluster is composed of short distance drivers with a low proportion of speeding.

## \*b. One cluster is composed of long distance drivers with a high proportion of speeding.

## \*c. One cluster is composed of long distance drivers with a low proportion of speeding.

## \*d. One cluster is composed of short distance drivers with a high proportion of speeding.

### All of the above