jorge’s rmarkdown

Jorge Reyes

2024-02-28

Getting all of the data to join into one.

```r
data1 <- read.csv("Criminal_Offenses_On_campus.csv") |>
  mutate(unique_id = paste0(OPEID, "_", Campus.ID)) |>
  rename_with(~ paste0(.x,"_all_campus"), recycle0 = TRUE) |>
  rename(Survey.year = Survey.year_all_campus, unique_id = unique_id_all_campus)

data2 <- read.csv("Criminal_Offenses_On_campus_Student_Housing_Facilities.csv") |>
  mutate(unique_id = paste0(OPEID, "_", Campus.ID)) |>
  rename_with(~ paste0(.x,"_student_housing"), recycle0 = TRUE) |>
  rename(Survey.year = Survey.year_student_housing, unique_id = unique_id_student_housing)

data3 <- read.csv("Criminal_Offenses_Noncampus.csv") |>
  mutate(unique_id = paste0(OPEID, "_", Campus.ID)) |>
  rename_with(~ paste0(.x, "_crim_offense_noncampus"), recycle0 = TRUE) |>
  rename(Survey.year = Survey.year_crim_offense_noncampus, unique_id = unique_id_crim_offense_noncampus)

data4 <- read.csv("Criminal_Offenses_Public_property.csv") |>
   mutate(unique_id = paste0(OPEID, "_", Campus.ID)) |>
  rename_with(~ paste0(.x, "_crim_offense_public"), recycle0 = TRUE) |>
  rename(Survey.year = Survey.year_crim_offense_public, unique_id = unique_id_crim_offense_public)

data5 <- read.csv("Arrests_On_campus.csv") |>
  mutate(unique_id = paste0(OPEID, "_", Campus.ID)) |>
  rename_with(~ paste0(.x, "_arrests_campus"), recycle0 = TRUE) |>
  rename(Survey.year = Survey.year_arrests_campus, unique_id = unique_id_arrests_campus)

data6 <- read.csv("Arrests_On_campus_Student_Housing_Facilities.csv") |>
  mutate(unique_id = paste0(OPEID, "_", Campus.ID)) |>
  rename_with(~ paste0(.x, "_arrests_stuhousing"), recycle0 = TRUE) |>
  rename(Survey.year = Survey.year_arrests_stuhousing, unique_id = unique_id_arrests_stuhousing)

data7 <- read.csv("Arrests_Noncampus.csv") |>
  mutate(unique_id = paste0(OPEID, "_", Campus.ID)) |>
  rename_with(~ paste0(.x, "_arrests_noncampus"), recycle0 = TRUE) |>
  rename(Survey.year = Survey.year_arrests_noncampus, unique_id = unique_id_arrests_noncampus)

data8 <- read.csv("Arrests_Public_Property.csv") |>
  mutate(unique_id = paste0(OPEID, "_", Campus.ID)) |>
  rename_with(~ paste0(.x, "_arrests_public"), recycle0 = TRUE) |>
  rename(Survey.year = Survey.year_arrests_public, unique_id = unique_id_arrests_public)

data9 <- read.csv("Disciplinary_Actions_On_campus.csv") |>
  mutate(unique_id = paste0(OPEID, "_", Campus.ID)) |>
  rename_with(~ paste0(.x, "_disciplinary_campus"), recycle0 = TRUE) |>
  rename(Survey.year = Survey.year_disciplinary_campus, unique_id = unique_id_disciplinary_campus)

data10 <- read.csv("Disciplinary_Actions_Student_Housing_Facilities.csv") |>
  mutate(unique_id = paste0(OPEID, "_", Campus.ID)) |>
  rename_with(~ paste0(.x, "_disciplinary_housing"), recycle0 = TRUE) |>
  rename(Survey.year = Survey.year_disciplinary_housing, unique_id = unique_id_disciplinary_housing)

data11 <- read.csv("Disciplinary_Actions_Noncampus.csv") |>
  mutate(unique_id = paste0(OPEID, "_", Campus.ID)) |>
  rename_with(~ paste0(.x, "_disciplinary_noncampus"), recycle0 = TRUE) |>
  rename(Survey.year = Survey.year_disciplinary_noncampus, unique_id = unique_id_disciplinary_noncampus)

data12 <- read.csv("Disciplinary_Actions_Public_Property.csv") |>
  mutate(unique_id = paste0(OPEID, "_", Campus.ID)) |>
  rename_with(~ paste0(.x, "_disciplinary_public"), recycle0 = TRUE) |>
  rename(Survey.year = Survey.year_disciplinary_public, unique_id = unique_id_disciplinary_public)

# This is our datasets being joined into one
dataset <- data1 |> left_join(data2) |>
  left_join(data3) |>
  left_join(data4) |>
  left_join(data5) |>
  left_join(data6) |>
  left_join(data7) |>
  left_join(data8) |>
  left_join(data9) |>
  left_join(data10) |>
  left_join(data11) |>
  left_join(data12)
```

```
## Joining with ‘by = join_by(Survey.year, unique_id)‘
## Joining with ‘by = join_by(Survey.year, unique_id)‘
## Joining with ‘by = join_by(Survey.year, unique_id)‘
## Joining with ‘by = join_by(Survey.year, unique_id)‘
## Joining with ‘by = join_by(Survey.year, unique_id)‘
## Joining with ‘by = join_by(Survey.year, unique_id)‘
## Joining with ‘by = join_by(Survey.year, unique_id)‘
## Joining with ‘by = join_by(Survey.year, unique_id)‘
## Joining with ‘by = join_by(Survey.year, unique_id)‘
## Joining with ‘by = join_by(Survey.year, unique_id)‘
## Joining with ‘by = join_by(Survey.year, unique_id)‘
```

```
#remove NAs
dataset[is.na(dataset)] <- 0

#remove repeated columns (like unitid repeating for each xcel file)
#(3/4/24) just fixed some problems w this
```

```
cols_to_remove <- c("Unitid_student_housing", "Institution.name_student_housing", "OPEID_student_housing


## had to change this dataset name before removing the campuses ##

cleaned <- dataset[, !names(dataset) %in% cols_to_remove]
```

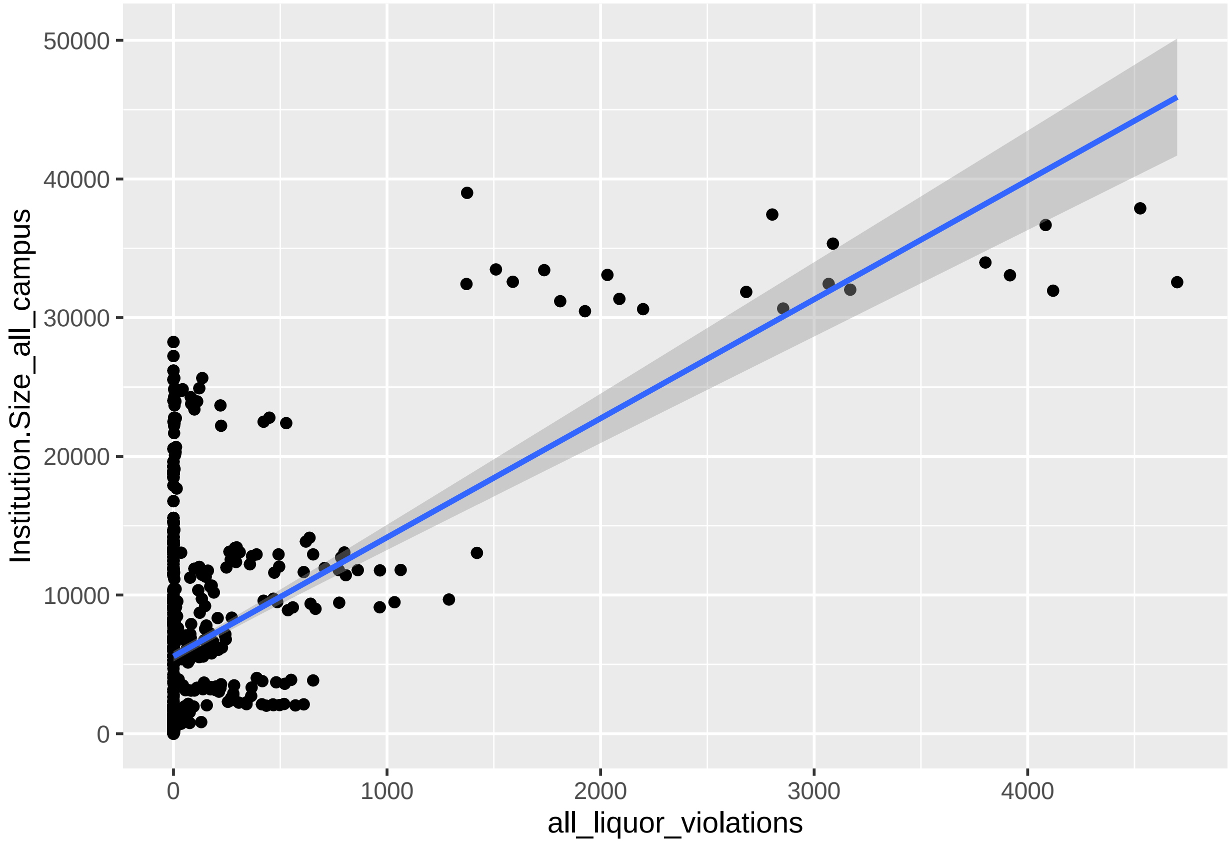
remove campuses

Removes campuses outside of Colorado.

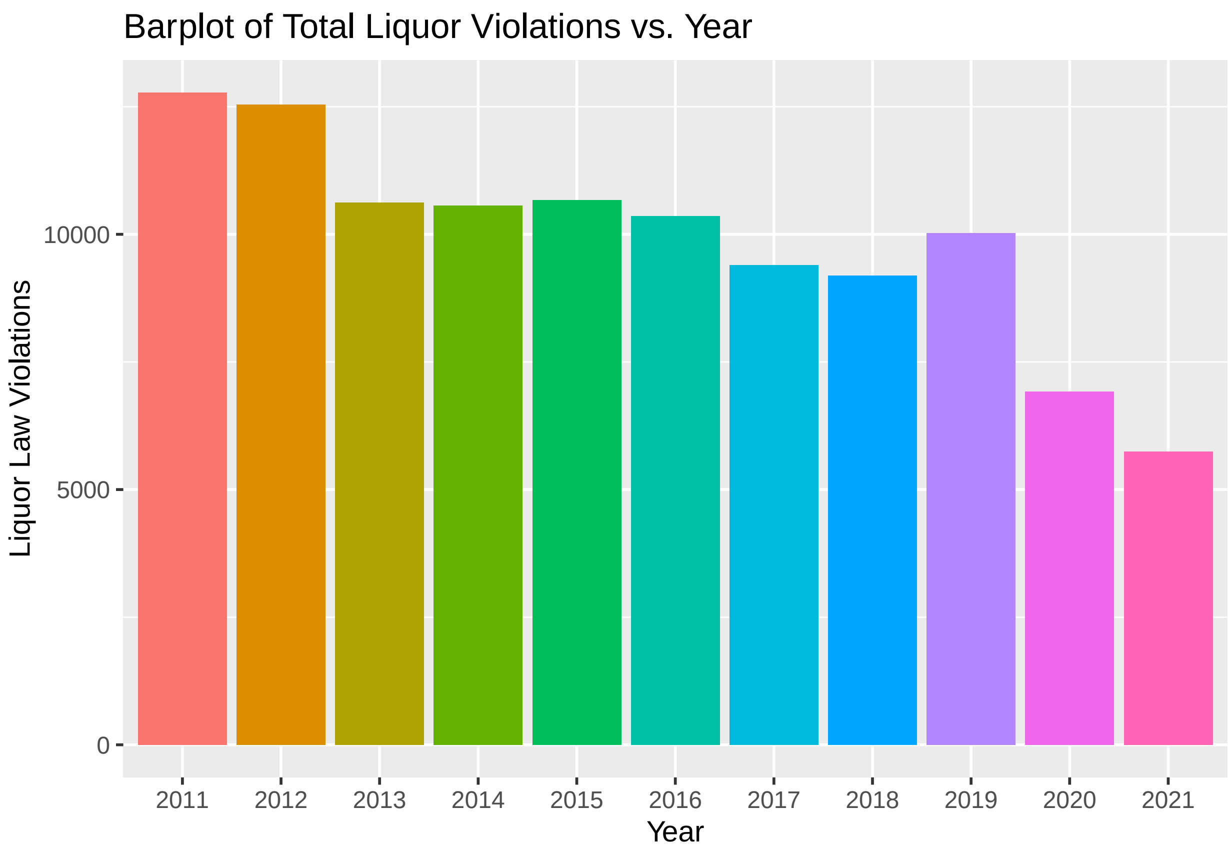
```
to_remove1 <- c("Jacksonville", "San Diego", "Memphis", "Dunnam", "Ft. Drum", "San Luis Obispo", "Syracu

#check vector length
#length(to_remove1)

matches <- unique(grep(paste(to_remove1,collapse="|"),
                        cleaned$Campus.Name_all_campus, value=TRUE))
cleaned_1 <- cleaned |> filter(!Campus.Name_all_campus %in% matches)

to_remove2 <- c("Albuquerque", "Wiesbaden", "Beale", "Gateway", "Ocala Metropolitan Campus", "Baton Roug

#length(to_remove2)

matches <- unique(grep(paste(to_remove2,collapse="|"),
                        cleaned_1$Campus.Name_all_campus, value=TRUE))
cleaned_2 <- cleaned_1 |> filter(!Campus.Name_all_campus %in% matches)

to_remove3 <- c("Webster University St. Louis-Main Campus", "Space Coast", "Fort Worth", "San Francisco

#length(to_remove3)

matches <- unique(grep(paste(to_remove3,collapse="|"),
                        cleaned_2$Campus.Name_all_campus, value=TRUE))
cleaned_data <- cleaned_2 |> filter(!Campus.Name_all_campus %in% matches)


# take a look
#head(cleaned_data)
```

```
#new column combining liquor law violations across disciplinary, arrests and location (public, stuhousi
cleaned_data$all_liquor_violations <- cleaned_data$Liquor.law.violations_arrests_campus + cleaned_data$L

numeric_data <- select(cleaned_data, where(is.numeric))

# figure margins too large
#pairs(numeric_data)

ggplot(cleaned_data, aes(y=Institution.Size_all_campus, x=all_liquor_violations)) +
  geom_point() +
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
#cleaned_data$all_liquor_violations[16]

year_factor <- as.factor(cleaned_data$Survey.year)

#ggplot(cleaned_data, aes(x=year_factor, y=all_liquor_violations)) +
 # geom_bar(stat= "identity", aes(fill=year_factor)) +
  #xlab("Year") +
  #ylab("Liquor Law Violations") +
  #ggtitle("Barplot of Total Liquor Violations v. Year")

ggplot(cleaned_data, aes(x = year_factor, y = all_liquor_violations, fill = year_factor)) +
  geom_bar(stat = "identity") +
  labs(x = "Year", y = "Liquor Law Violations", fill = "Year") +
  ggtitle("Barplot of Total Liquor Violations vs. Year") +
  theme(legend.position = "none")
```



Tests

```
mean(cleaned_data$Negligent.manslaughter_all_campus)
```

```
## [1] 0
```

```
mean(cleaned_data$Sex.offenses...Forcible_all_campus)
```

```
## [1] 0.1454261
```

```
mean(cleaned_data$Rape_all_campus)
```

```
## [1] 0.5402658
```

```
mean(cleaned_data$Fondling_all_campus)
```

```
## [1] 0.354183
```

```r
mean(cleaned_data$Sex.offenses...Non.forcible_all_campus)
```

```
## [1] 0.0007818608
```

```r
mean(cleaned_data$Incest_all_campus)
```

```
## [1] 0
```

```r
mean(cleaned_data$Statutory.rape_all_campus)
```

```
## [1] 0.002345582
```

```r
mean(cleaned_data$Robbery_all_campus)
```

```
## [1] 0.129007
```

```r
mean(cleaned_data$Burglary_all_campus)
```

```
## [1] 1.628616
```

```r
mean(cleaned_data$Motor.vehicle.theft_all_campus)
```

```
## [1] 0.8350274
```

```r
mean(cleaned_data$Arson_all_campus)
```

```
## [1] 0.1196247
```

```r
library(dplyr)
library(knitr)

# Sample data creation (assuming 'cleaned' is your data frame)
means <- round(c(mean(cleaned_data$Negligent.manslaughter_all_campus),
                 mean(cleaned_data$Sex.offenses...Forcible_all_campus),
                 mean(cleaned_data$Rape_all_campus),
                 mean(cleaned_data$Fondling_all_campus),
                 mean(cleaned_data$Sex.offenses...Non.forcible_all_campus),
                 mean(cleaned_data$Incest_all_campus),
                 mean(cleaned_data$Statutory.rape_all_campus),
                 mean(cleaned_data$Robbery_all_campus),
                 mean(cleaned_data$Burglary_all_campus),
                 mean(cleaned_data$Motor.vehicle.theft_all_campus),
                 mean(cleaned_data$Arson_all_campus)), 3)

sds <- round(c(
  sd(cleaned_data$Negligent.manslaughter_all_campus),
  sd(cleaned_data$Sex.offenses...Forcible_all_campus),
  sd(cleaned_data$Rape_all_campus),
  sd(cleaned_data$Fondling_all_campus),
  sd(cleaned_data$Sex.offenses...Non.forcible_all_campus),
  sd(cleaned_data$Incest_all_campus),
  sd(cleaned_data$Statutory.rape_all_campus),
  sd(cleaned_data$Robbery_all_campus),
  sd(cleaned_data$Burglary_all_campus),
  sd(cleaned_data$Motor.vehicle.theft_all_campus),
  sd(cleaned_data$Arson_all_campus)
), 3)

# Creating data frame
summary_df <- data.frame(
  Variable  = c("Negligent Manslaughter", "Sex Offenses (Forcible)", "Rape",
                "Fondling", "Sex Offenses (Non-forcible)", "Incest",
                "Statutory Rape", "Robbery", "Burglary", "Motor Vehicle Theft",
                "Arson"),
  Mean = means,
  StandardDeviation = sds
)

# Sorting the data frame by Mean in descending order
sorted_summary_df <- summary_df %>%
  arrange(desc(Mean), desc(StandardDeviation))

# Creating the kable
knitr::kable(sorted_summary_df, caption = "Average Values of Different Campus Offenses",
             col.names = c("Variables", "Average", "Standard Deviation"))
```

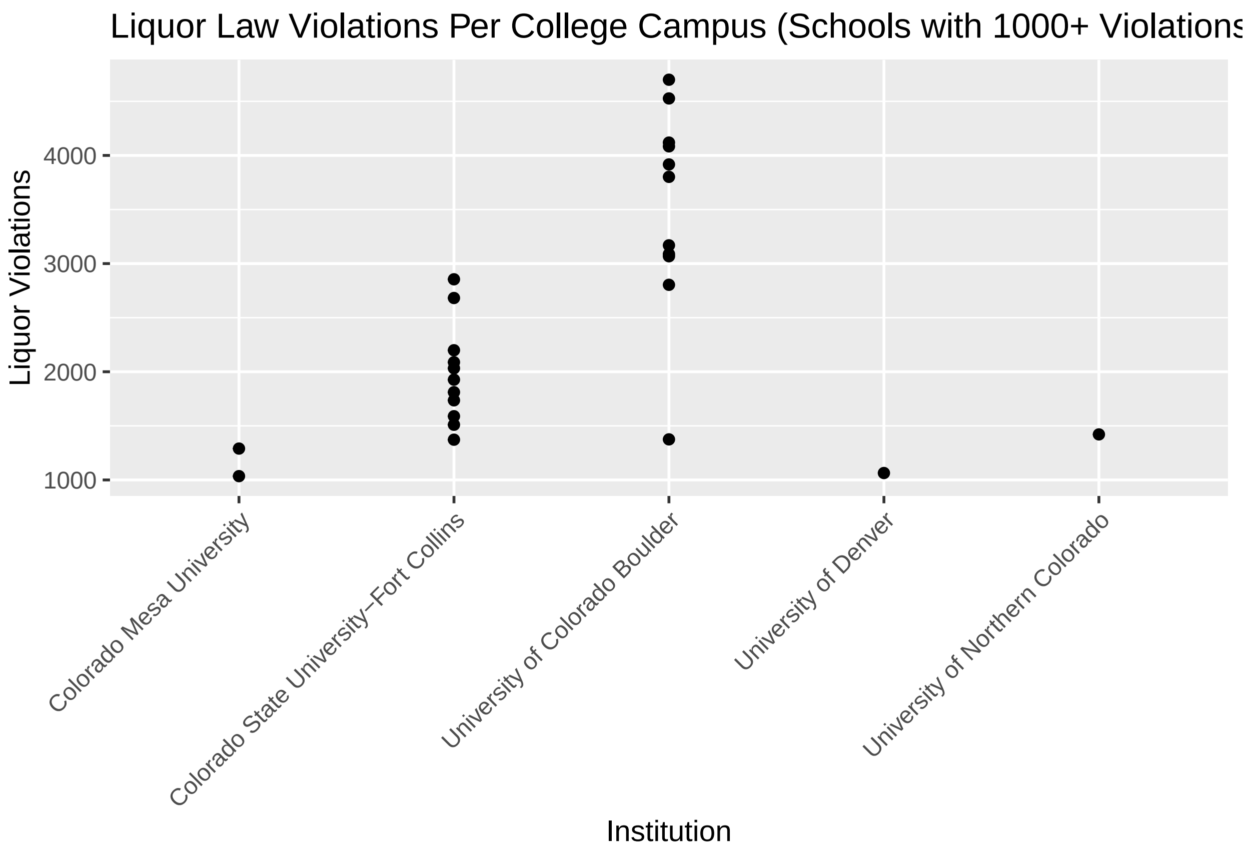
Table 1: Average Values of Different Campus Offenses

Variables	Average	Standard Deviation
Burglary	1.629	5.381
Motor Vehicle Theft	0.835	3.291
Rape	0.540	2.145
Fondling	0.354	1.476
Sex Offenses (Forcible)	0.145	1.006
Robbery	0.129	0.565
Arson	0.120	0.662
Statutory Rape	0.002	0.048
Sex Offenses (Non-forcible)	0.001	0.028
Negligent Manslaughter	0.000	0.000
Incest	0.000	0.000

EDA, Jorge's Part

```r
#test <- cleaned_data |> mutate(liquor_violations_capita = Liquor.law.violations_disciplinary_campus/1e
#cleaned_data |> filter(Liquor.law.violations_disciplinary_campus > 25) |> group_by(Institution.name_al
 # summarize(all_offense_capita = sum(all_liquor_violations)/Institution.Size_all_campus) |>
```

```
cleaned_data |> filter(all_liquor_violations > 1000) |>
 ggplot() +
  geom_point(aes(x = Institution.name_all_campus, y = all_liquor_violations),
             color = "black") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Liquor Law Violations Per College Campus (Schools with 1000+ Violations)") +
  xlab("Institution") + ylab("Liquor Violations")
```



```
set.seed(4242)

## split cleaned data into 25/75
smp_size <- floor(0.75 * nrow(cleaned_data))

train_split <- sample(seq_len(nrow(cleaned_data)), size = smp_size)

# create train = 75% and test = 25% set
train <- cleaned_data[train_split,] |> as_tibble() |> mutate(train = TRUE)
test <- cleaned_data[-train_split,] |> as_tibble() |> mutate(train = FALSE)


## check split to ensure nothing got screwed up

# create df of training data means and sd of each column
train_means_sd <- sapply(train[,c(7:20, 22:86)],
                         function(x) c(mean(x, na.rm = TRUE),
                                       sd(x, na.rm=TRUE)),
                         simplify = FALSE) |> bind_rows()
# transpose so table is legible
ttrain_means_sd <- t(train_means_sd)
# create kable table
#knitr::kable(ttrain_means_sd, digits = 5, caption = "Training Data, metrics to compare to test", col.n

# create df of testing data means and sd of each column
test_means_sd <- sapply(test[,c(7:20, 22:86)],
                         function(x) c(mean(x, na.rm = TRUE),
                                       sd(x, na.rm=TRUE)),
                         simplify = FALSE) |> bind_rows()
ttest_means_sd <- t(test_means_sd)
#knitr::kable(ttest_means_sd, digits = 5, caption = "Test Data, metrics to compare to training", col.nam


## kable tables for hw 5

train_means <- round(c(mean(train$Negligent.manslaughter_all_campus),
           mean(train$Sex.offenses...Forcible_all_campus),
           mean(train$Rape_all_campus),
           mean(train$Fondling_all_campus),
           mean(train$Sex.offenses...Non.forcible_all_campus),
           mean(train$Incest_all_campus),
           mean(train$Statutory.rape_all_campus),
           mean(train$Robbery_all_campus),
           mean(train$Burglary_all_campus),
           mean(train$Motor.vehicle.theft_all_campus),
           mean(train$Arson_all_campus)), 3)

train_sds <- round(c(
  sd(train$Negligent.manslaughter_all_campus),
  sd(train$Sex.offenses...Forcible_all_campus),
  sd(train$Rape_all_campus),
  sd(train$Fondling_all_campus),
  sd(train$Sex.offenses...Non.forcible_all_campus),
  sd(train$Incest_all_campus),
  sd(train$Statutory.rape_all_campus),
  sd(train$Robbery_all_campus),
  sd(train$Burglary_all_campus),
  sd(train$Motor.vehicle.theft_all_campus),
  sd(train$Arson_all_campus)
), 3)

train_pres <- data.frame(
  Variable  = c("Negligent Manslaughter", "Sex Offenses (Forcible)", "Rape",
               "Fondling", "Sex Offenses (Non-forcible)", "Incest",
               "Statutory Rape", "Robbery", "Burglary", "Motor Vehicle Theft",
               "Arson"),
  Mean = train_means,
  StandardDeviation = train_sds
)
```

```
knitr::kable(train_pres, caption = "Training Data", col.names = c("Variable", "Mean", "SD"))
```

Table 2: Training Data

Variable	Mean	SD
Negligent Manslaughter	0.000	0.000
Sex Offenses (Forcible)	0.131	0.988
Rape	0.514	2.041
Fondling	0.332	1.362
Sex Offenses (Non-forcible)	0.000	0.000
Incest	0.000	0.000
Statutory Rape	0.002	0.046
Robbery	0.137	0.581
Burglary	1.555	5.217
Motor Vehicle Theft	0.826	3.259
Arson	0.103	0.639

```
test_means <- round(c(mean(test$Negligent.manslaughter_all_campus),
           mean(test$Sex.offenses...Forcible_all_campus),
           mean(test$Rape_all_campus),
           mean(test$Fondling_all_campus),
           mean(test$Sex.offenses...Non.forcible_all_campus),
           mean(test$Incest_all_campus),
           mean(test$Statutory.rape_all_campus),
           mean(test$Robbery_all_campus),
           mean(test$Burglary_all_campus),
           mean(test$Motor.vehicle.theft_all_campus),
           mean(test$Arson_all_campus)), 3)

test_sds <- round(c(
  sd(test$Negligent.manslaughter_all_campus),
  sd(test$Sex.offenses...Forcible_all_campus),
  sd(test$Rape_all_campus),
  sd(test$Fondling_all_campus),
  sd(test$Sex.offenses...Non.forcible_all_campus),
  sd(test$Incest_all_campus),
  sd(test$Statutory.rape_all_campus),
  sd(test$Robbery_all_campus),
  sd(test$Burglary_all_campus),
  sd(test$Motor.vehicle.theft_all_campus),
  sd(test$Arson_all_campus)
), 3)

test_pres <- data.frame(
  Variable  = c("Negligent Manslaughter", "Sex Offenses (Forcible)", "Rape",
             "Fondling", "Sex Offenses (Non-forcible)", "Incest",
             "Statutory Rape", "Robbery", "Burglary", "Motor Vehicle Theft",
             "Arson"),
  Mean = test_means,
  StandardDeviation = test_sds
)
```

```
knitr::kable(test_pres, caption = "Test Data", col.names = c("Variable", "Mean", "SD"))
```

Table 3: Test Data

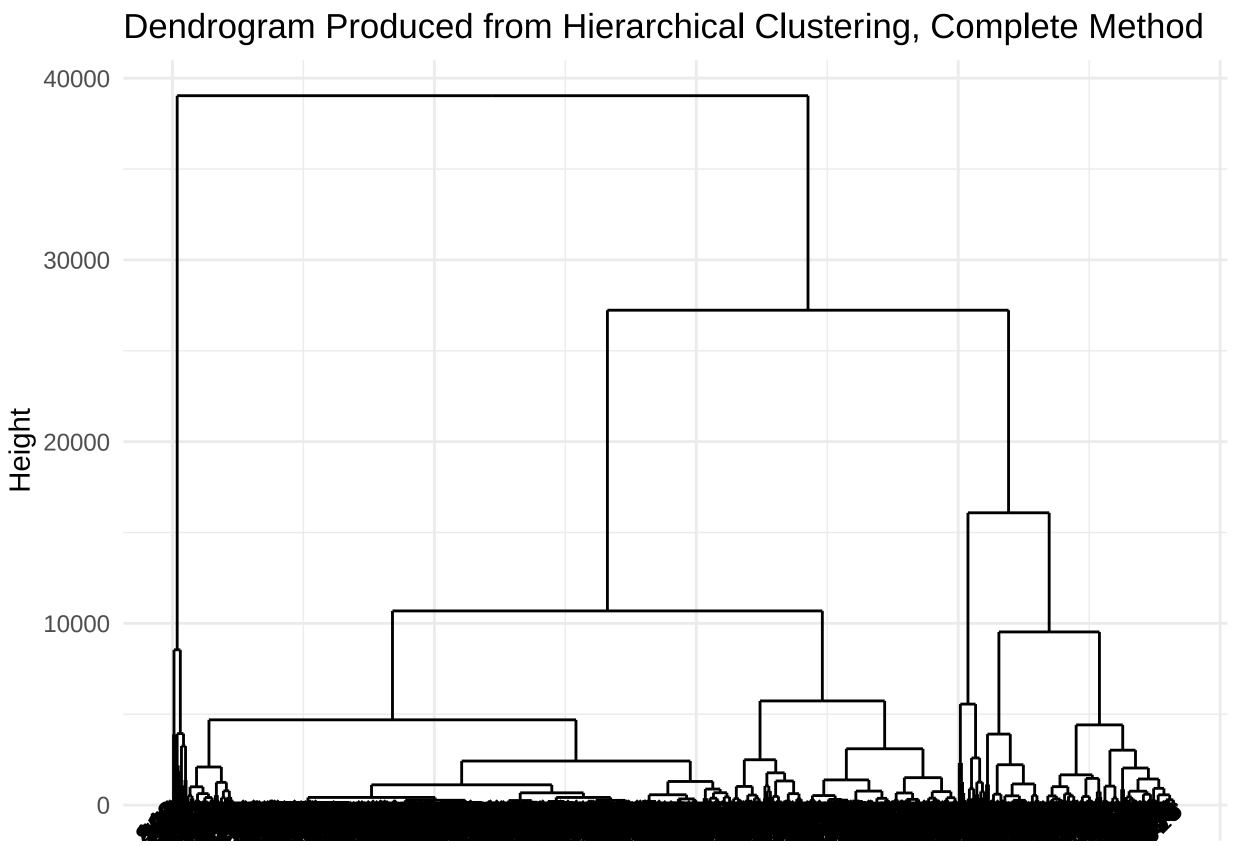
Variable	Mean	SD
Negligent Manslaughter	0.000	0.000
Sex Offenses (Forcible)	0.188	1.058
Rape	0.619	2.431
Fondling	0.422	1.774
Sex Offenses (Non-forcible)	0.003	0.056
Incest	0.000	0.000
Statutory Rape	0.003	0.056
Robbery	0.106	0.514
Burglary	1.850	5.850
Motor Vehicle Theft	0.863	3.390
Arson	0.169	0.728

Clustering method

I will use hierarchical clustering because it doesn't require a choice of K.

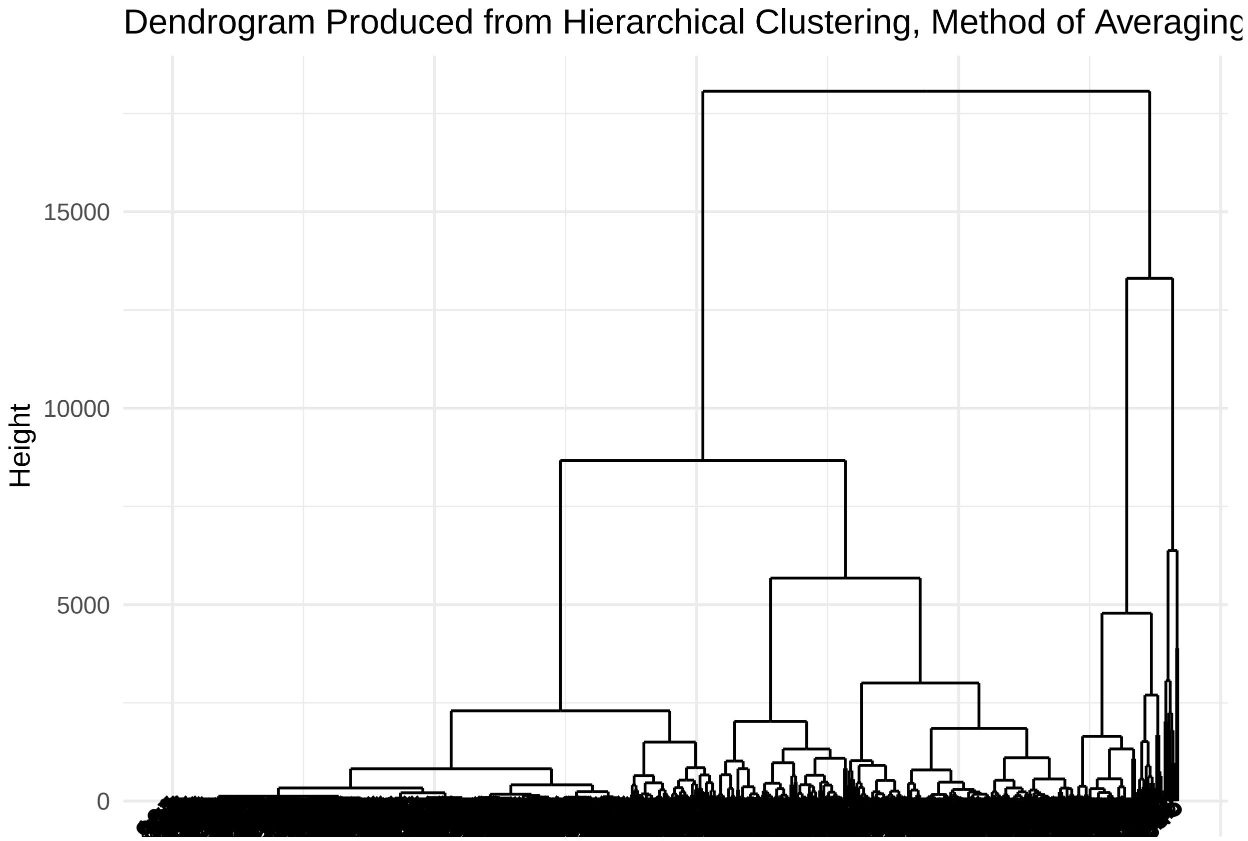
Method here is 'complete'

```
train_num <- train |> as_tibble() |> select(-where(is.character))

# Remove all the columns I do not want
train_num <- train_num[, !names(train_num) %in% c('Unitid_all_campus', 'OPEID_all_campus', 'Campus.ID_al

# Performing clustering
set.seed(432)
dist_matrix <- dist(train_num, method = "euclidean")
h_clus_complete <- hclust(dist_matrix, method = "complete")

## Plotting
dend_data <- dendro_data(h_clus_complete)
label_names <- dend_data$labels
label_names$h <- 0


ggplot() +
  geom_segment(data = dend_data$segments, aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_text(data = label_names, aes(x = x, y = h, label = label), hjust = 1, angle = 45) +
  theme_minimal() +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank()) +
  labs(y = "Height") +
  ggtitle("Dendrogram Produced from Hierarchical Clustering, Complete Method")
```



Method here is 'average'

```
# Performing clustering
set.seed(432)
dist_matrix <- dist(train_num, method = "euclidean")
h_clus_avg <- hclust(dist_matrix, method = "average")

## Plotting
dend_data1 <- dendro_data(h_clus_avg)
label_names1 <- dend_data1$labels
label_names1$h <- 0


ggplot() +
  geom_segment(data = dend_data1$segments, aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_text(data = label_names1, aes(x = x, y = h, label = label), hjust = 1, angle = 45) +
  theme_minimal() +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank()) +
  labs(y = "Height") +
  ggtitle("Dendrogram Produced from Hierarchical Clustering, Method of Averaging")
```



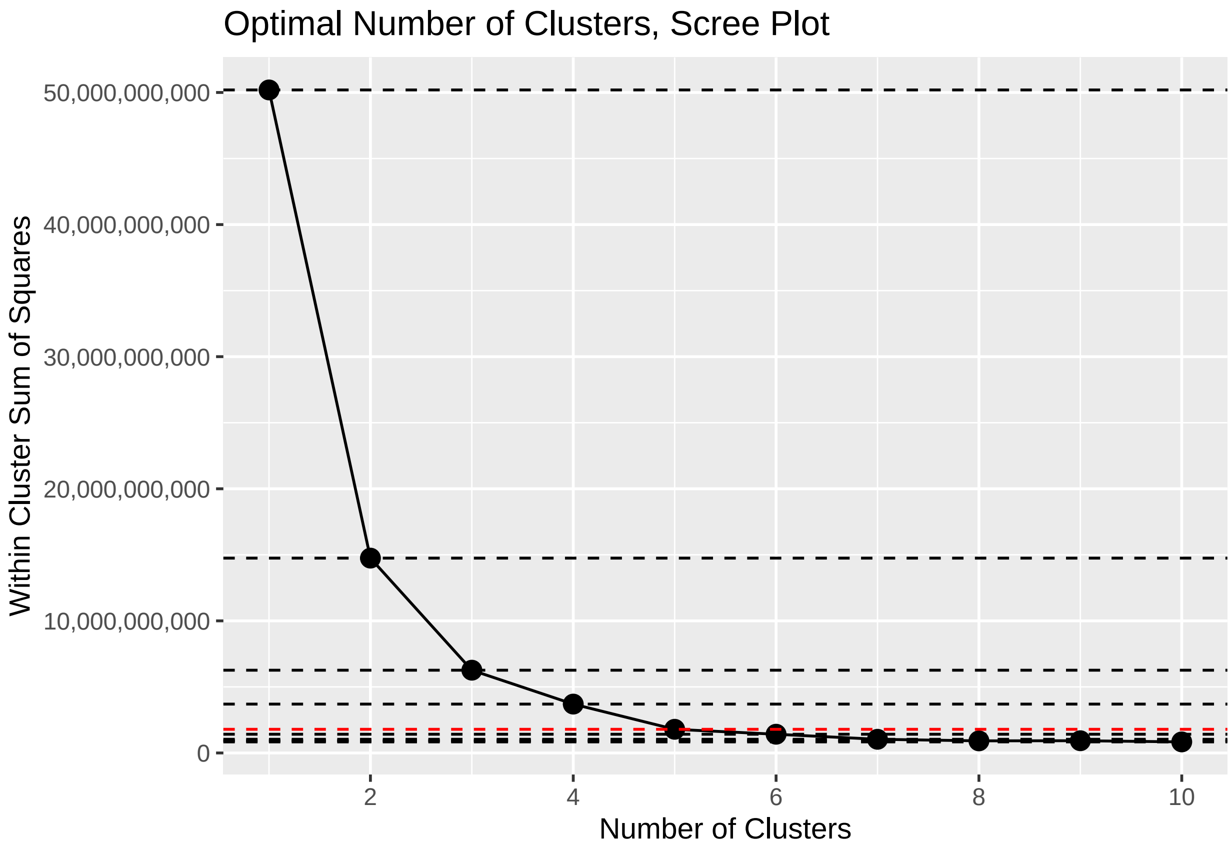
will try K means clustering, getting too many clusters from hierarchical clustering

```
alc_2cols1 <- train_num[ , c("all_liquor_violations", "Institution.Size_all_campus")]
set.seed(421)
km.out <- kmeans(alc_2cols1, centers = 3, nstart = 20)
km.out
```

```
## K-means clustering with 3 clusters of sizes 536, 97, 326
##
## Cluster means:
##   all_liquor_violations Institution.Size_all_campus
## 1              20.76866                    976.4328
## 2             380.19588                  22326.0103
## 3              71.97546                   9736.0583
##
## Clustering vector:
##   [1] 1 1 1 3 1 1 1 1 1 1 2 2 1 1 1 1 2 3 1 3 1 3 3 3 1 1 3 1 1 3 3 1 3 3 3 3 1
##  [38] 3 1 1 1 1 3 1 1 1 1 1 3 1 3 1 1 3 3 1 1 1 3 3 1 3 1 1 1 1 3 1 1 1 1 1 1 3
##  [75] 1 1 1 1 3 2 2 1 3 3 3 2 3 1 3 3 1 1 1 3 3 1 2 1 3 1 3 1 2 3 3 1 1 2 2 3 3
## [112] 3 1 2 1 3 1 1 1 3 3 3 1 1 1 3 1 3 3 3 1 1 1 1 3 2 1 2 1 1 1 2 1 2 3 1 3 1
## [149] 3 1 1 3 1 1 3 1 1 1 3 3 3 3 1 1 1 1 1 2 1 1 3 1 1 1 3 3 2 1 2 3 1 1 1 1 3
## [186] 2 1 3 1 1 3 1 3 1 1 1 1 1 3 1 1 1 2 1 2 2 1 3 3 3 3 1 3 3 1 3 1 1 1 3 1 1
## [223] 1 1 3 2 3 3 3 1 2 1 3 1 1 2 1 1 3 1 1 1 1 1 3 3 1 1 1 1 3 3 3 1 3 1 1 1 3
## [260] 3 3 1 1 1 1 1 1 2 3 1 1 1 1 1 3 3 3 1 1 3 1 1 1 3 1 1 3 1 3 1 1 3 1 1 2 1
## [297] 1 1 1 1 1 3 1 1 1 1 3 3 1 1 1 2 1 3 3 1 1 1 3 1 2 3 3 3 1 3 2 1 1 1 1 1 1
## [334] 1 1 2 1 3 1 3 3 3 1 3 1 1 3 2 1 2 1 3 1 1 1 1 2 1 1 1 1 3 1 1 3 3 1 1 3 3
## [371] 1 1 3 1 1 3 1 3 1 1 1 2 1 3 1 3 1 1 3 1 1 2 3 3 1 1 2 1 1 2 3 1 3 3 1 3 1
## [408] 1 1 3 3 3 2 1 1 3 3 3 1 1 1 2 2 2 1 1 1 1 1 3 1 1 3 2 1 3 2 3 3 3 3 1 1 1
## [445] 1 1 1 2 1 3 3 1 1 3 3 3 1 2 3 1 1 1 1 3 1 3 3 3 2 1 3 2 1 3 3 1 2 3 3 1 1
## [482] 3 1 3 1 1 1 3 1 1 1 1 1 1 3 3 1 3 3 1 3 1 1 1 3 1 3 1 3 1 3 3 1 1 1 3 1 1
## [519] 3 1 1 3 3 1 1 1 3 1 3 1 1 3 3 1 1 3 1 1 1 3 1 3 2 3 3 3 3 1 1 3 2 1 1 1 3
## [556] 1 3 1 3 3 2 1 1 2 3 3 3 1 1 1 1 3 1 2 1 1 1 1 1 1 1 3 1 1 1 1 1 1 3 1 2 3
## [593] 1 1 1 3 2 1 1 3 1 3 2 3 1 1 3 1 3 2 1 3 1 3 3 1 3 1 1 3 3 3 1 1 3 2 1 1 1
## [630] 1 1 1 2 1 3 1 2 3 3 3 3 1 1 3 1 3 3 3 1 1 3 3 3 1 2 2 2 1 1 1 1 1 1 3 3 1
## [667] 1 1 3 1 1 3 3 1 1 1 3 3 1 1 3 3 1 3 1 1 1 1 1 1 1 3 1 2 1 3 3 2 3 1 1 3 3
## [704] 3 1 1 1 1 1 1 1 3 1 1 3 1 2 1 1 1 2 1 1 1 1 1 3 3 3 3 1 1 1 3 3 1 3 1 2 3
## [741] 3 3 3 3 3 3 1 2 3 1 3 1 1 1 1 1 1 1 3 3 3 2 1 1 1 1 3 1 1 3 3 2 1 3 3 1 3
## [778] 1 3 2 1 2 3 1 3 1 1 3 1 3 1 1 1 2 3 1 3 1 3 1 2 3 1 1 1 2 1 1 1 3 1 1 3 1
## [815] 3 1 3 3 1 1 3 3 3 3 1 1 2 1 1 1 1 1 3 1 1 3 2 2 1 1 1 3 1 2 1 2 1 2 1 2 3
## [852] 3 2 3 1 1 2 3 1 1 2 1 1 1 3 1 3 1 1 1 1 3 1 1 1 1 1 2 1 3 1 1 2 3 3 2 1 1
## [889] 3 1 1 3 1 1 3 3 1 1 1 1 1 3 3 1 1 1 1 1 1 3 1 1 3 3 3 1 3 3 3 1 1 1 3 1 3
## [926] 3 2 3 1 1 1 3 1 1 1 2 1 1 3 1 3 1 1 2 1 2 1 1 3 1 1 1 2 1 2 3 3 3 2
##
## Within cluster sum of squares by cluster:
## [1]  801514625 2520528102 2941684316
##  (between_SS / total_SS =  87.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

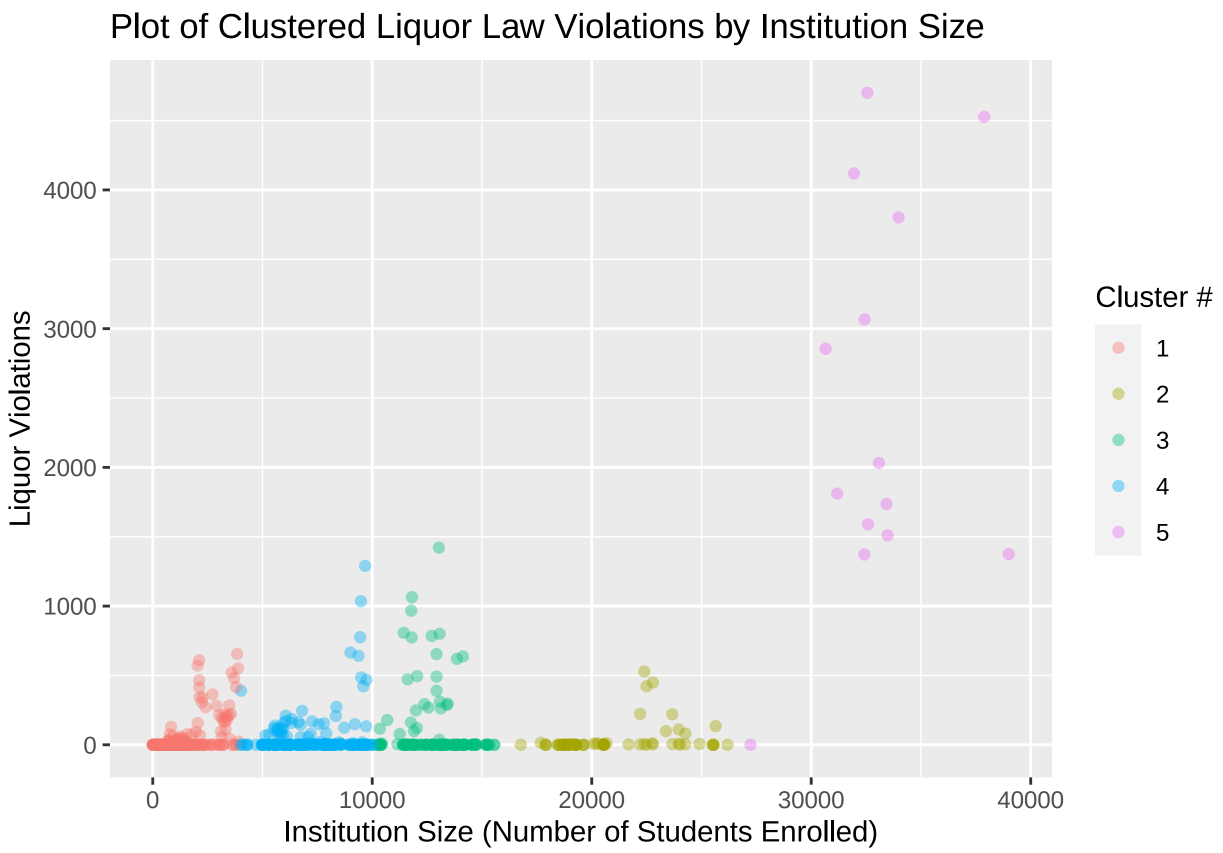
Plotting optimal number of clusters

```r
nclust <- 10
wss <- numeric(nclust)
set.seed(421)
## Looping through different number of clusters
for (i in 1:nclust) {
  km.out <- kmeans(alc_2cols1, centers = i, nstart = 20)
  wss[i] <- km.out$tot.withinss
}

## Plotting
wss_df <- tibble(clusters = 1:nclust, wss = wss)
sc_plot <- ggplot(wss_df, aes(x = clusters, y = wss, group = 1)) +
  geom_point(size = 3) +
  geom_line() +
  scale_x_continuous(breaks = c(2, 4, 6, 8, 10)) +
  scale_y_continuous(labels = scales::comma) +
  xlab("Number of Clusters") +
  ylab("Within Cluster Sum of Squares") +
  ggtitle("Optimal Number of Clusters, Scree Plot")
sc_plot +
  geom_hline(
    yintercept = wss,
    linetype = 'dashed',
    col = c(rep('black',4),'red', rep('black', 5))
  )
```



Now plotting with the optimal number of clusters

```
k <- 5

set.seed(421)
km.out <- kmeans(alc_2cols1, centers = k, nstart = 20)

train_num$cluster_id <- factor(km.out$cluster)
ggplot(train_num, aes(Institution.Size_all_campus, all_liquor_violations, color = cluster_id)) +
    geom_point(alpha = 0.40) +
    xlab("Institution Size (Number of Students Enrolled)") +
    ylab("Liquor Violations") +
  ggtitle("Plot of Clustered Liquor Law Violations by Institution Size") +
  labs(color = "Cluster #")
```



```
idx <- which(km.out$cluster == 5)
train[idx,]$Institution.name_all_campus
```

```
##  [1] "University of Colorado Boulder"
##  [2] "Colorado State University-Fort Collins"
##  [3] "University of Colorado Boulder"
##  [4] "Colorado State University-Fort Collins"
##  [5] "University of Colorado Boulder"
##  [6] "Colorado Technical University-Colorado Springs"
##  [7] "Colorado State University-Fort Collins"
##  [8] "University of Colorado Boulder"
##  [9] "Colorado State University-Fort Collins"
## [10] "University of Colorado Boulder"
## [11] "University of Colorado Boulder"
## [12] "Colorado State University-Fort Collins"
## [13] "Colorado State University-Fort Collins"
## [14] "Colorado State University-Fort Collins"
```

```
idx <- km.out$cluster
idx <- which(km.out$cluster == 5)
train[idx,]
```

```
## # A tibble: 14 x 86
##    Survey.year Unitid_all_campus Institution.name_all_campus    OPEID_all_campus
##          <int>             <int> <chr>                                     <int>
##  1        2014            126614 University of Colorado Boulder           137000
##  2        2021            126818 Colorado State University-For~           135000
##  3        2012            126614 University of Colorado Boulder           137000
##  4        2019            126818 Colorado State University-For~           135000
##  5        2019            126614 University of Colorado Boulder           137000
##  6        2019            126827 Colorado Technical University~          1014800
##  7        2020            126818 Colorado State University-For~           135000
##  8        2011            126614 University of Colorado Boulder           137000
##  9        2013            126818 Colorado State University-For~           135000
## 10        2016            126614 University of Colorado Boulder           137000
## 11        2021            126614 University of Colorado Boulder           137000
## 12        2018            126818 Colorado State University-For~           135000
## 13        2017            126818 Colorado State University-For~           135000
## 14        2012            126818 Colorado State University-For~           135000
## # i 82 more variables: Campus.ID_all_campus <int>,
## #   Campus.Name_all_campus <chr>, Institution.Size_all_campus <dbl>,
## #   Murder.Non.negligent.manslaughter_all_campus <int>,
## #   Negligent.manslaughter_all_campus <int>,
## #   Sex.offenses...Forcible_all_campus <dbl>, Rape_all_campus <dbl>,
## #   Fondling_all_campus <dbl>, Sex.offenses...Non.forcible_all_campus <dbl>,
## #   Incest_all_campus <dbl>, Statutory.rape_all_campus <dbl>, ...
```

```
train[idx,]$Institution.name_all_campus
```

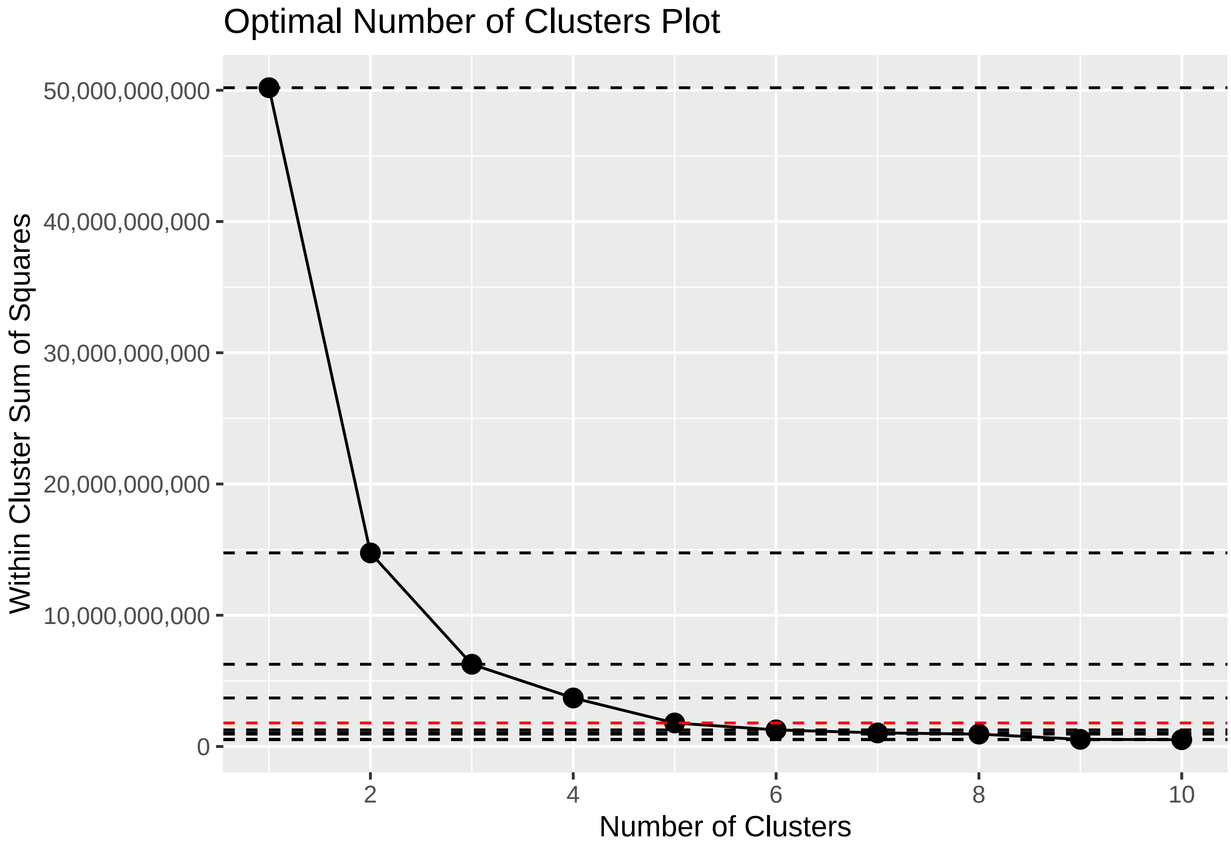
```
##  [1] "University of Colorado Boulder"
##  [2] "Colorado State University-Fort Collins"
##  [3] "University of Colorado Boulder"
##  [4] "Colorado State University-Fort Collins"
##  [5] "University of Colorado Boulder"
##  [6] "Colorado Technical University-Colorado Springs"
##  [7] "Colorado State University-Fort Collins"
##  [8] "University of Colorado Boulder"
##  [9] "Colorado State University-Fort Collins"
## [10] "University of Colorado Boulder"
## [11] "University of Colorado Boulder"
## [12] "Colorado State University-Fort Collins"
## [13] "Colorado State University-Fort Collins"
## [14] "Colorado State University-Fort Collins"
```

Now clustering with 2 predictors.

```
alc_2cols2 <- train_num[ , c("all_liquor_violations", "Survey.year", "Institution.Size_all_campus")]
set.seed(421)
km.out2 <- kmeans(alc_2cols2, centers = 3, nstart = 20)
km.out2
```

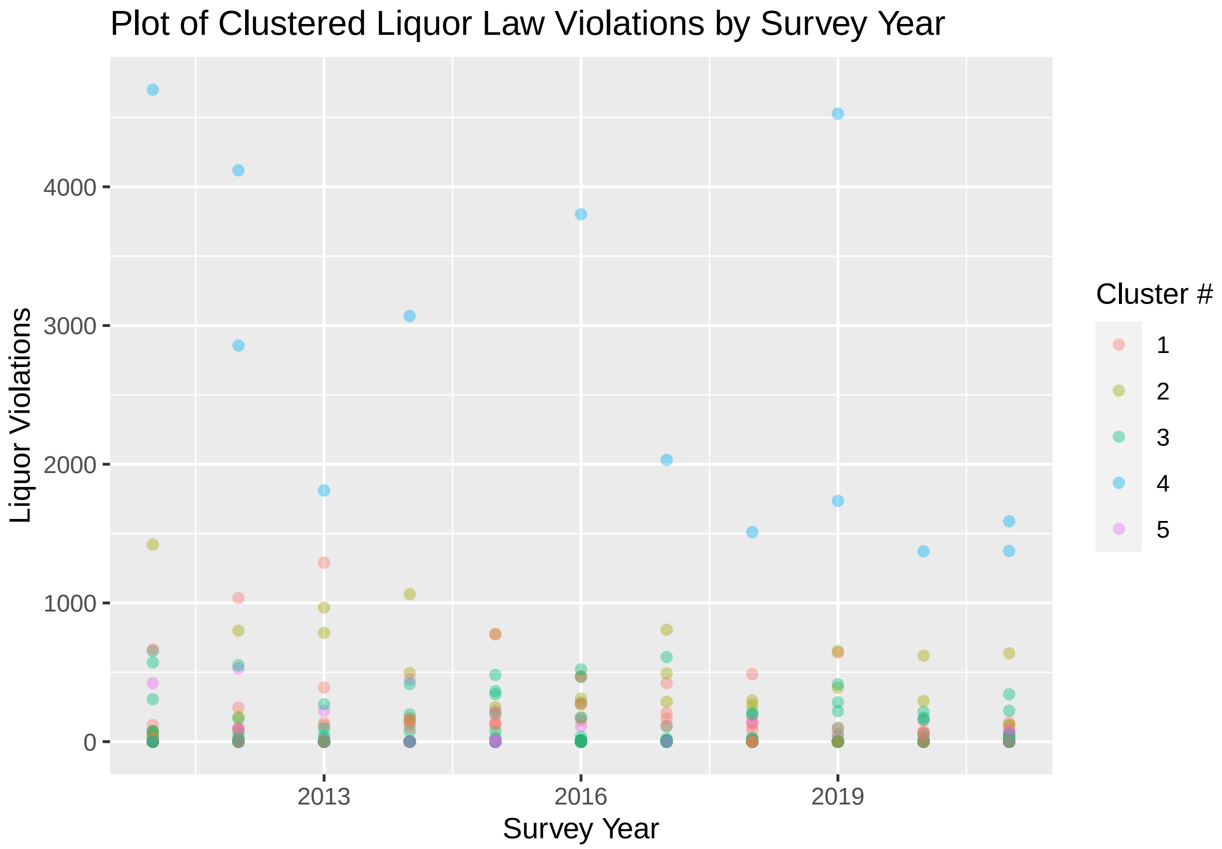
```
## K-means clustering with 3 clusters of sizes 326, 536, 97
##
## Cluster means:
##   all_liquor_violations Survey.year Institution.Size_all_campus
## 1              71.97546    2016.571                   9736.0583
## 2              20.76866    2016.172                    976.4328
## 3             380.19588    2015.753                  22326.0103
##
## Clustering vector:
##   [1] 2 2 2 1 2 2 2 2 2 2 3 3 2 2 2 2 3 1 2 1 2 1 1 1 2 2 1 2 2 1 1 2 1 1 1 1 2
##  [38] 1 2 2 2 2 1 2 2 2 2 2 1 2 1 2 2 1 1 2 2 2 1 1 2 1 2 2 2 2 1 2 2 2 2 2 2 1
##  [75] 2 2 2 2 1 3 3 2 1 1 1 3 1 2 1 1 2 2 2 1 1 2 3 2 1 2 1 2 3 1 1 2 2 3 3 1 1
## [112] 1 2 3 2 1 2 2 2 1 1 1 2 2 2 1 2 1 1 1 2 2 2 2 1 3 2 3 2 2 2 3 2 3 1 2 1 2
## [149] 1 2 2 1 2 2 1 2 2 2 1 1 1 1 2 2 2 2 2 3 2 2 1 2 2 2 1 1 3 2 3 1 2 2 2 2 1
## [186] 3 2 1 2 2 1 2 1 2 2 2 2 2 1 2 2 2 3 2 3 3 2 1 1 1 1 2 1 1 2 1 2 2 2 1 2 2
## [223] 2 2 1 3 1 1 1 2 3 2 1 2 2 3 2 2 1 2 2 2 2 2 1 1 2 2 2 2 1 1 1 2 1 2 2 2 1
## [260] 1 1 2 2 2 2 2 2 3 1 2 2 2 2 2 1 1 1 2 2 1 2 2 2 1 2 2 1 2 1 2 2 1 2 2 3 2
## [297] 2 2 2 2 2 1 2 2 2 2 1 1 2 2 2 3 2 1 1 2 2 2 1 2 3 1 1 1 2 1 3 2 2 2 2 2 2
## [334] 2 2 3 2 1 2 1 1 1 2 1 2 2 1 3 2 3 2 1 2 2 2 2 3 2 2 2 2 1 2 2 1 1 2 2 1 1
## [371] 2 2 1 2 2 1 2 1 2 2 2 3 2 1 2 1 2 2 1 2 2 3 1 1 2 2 3 2 2 3 1 2 1 1 2 1 2
## [408] 2 2 1 1 1 3 2 2 1 1 1 2 2 2 3 3 3 2 2 2 2 2 1 2 2 1 3 2 1 3 1 1 1 1 2 2 2
## [445] 2 2 2 3 2 1 1 2 2 1 1 1 2 3 1 2 2 2 2 1 2 1 1 1 3 2 1 3 2 1 1 2 3 1 1 2 2
## [482] 1 2 1 2 2 2 1 2 2 2 2 2 2 1 1 2 1 1 2 1 2 2 2 1 2 1 2 1 2 1 1 2 2 2 1 2 2
## [519] 1 2 2 1 1 2 2 2 1 2 1 2 2 1 1 2 2 1 2 2 2 1 2 1 3 1 1 1 1 2 2 1 3 2 2 2 1
## [556] 2 1 2 1 1 3 2 2 3 1 1 1 2 2 2 2 1 2 3 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 3 1
## [593] 2 2 2 1 3 2 2 1 2 1 3 1 2 2 1 2 1 3 2 1 2 1 1 2 1 2 2 1 1 1 2 2 1 3 2 2 2
## [630] 2 2 2 3 2 1 2 3 1 1 1 1 2 2 1 2 1 1 1 2 2 1 1 1 2 3 3 3 2 2 2 2 2 2 1 1 2
## [667] 2 2 1 2 2 1 1 2 2 2 1 1 2 2 1 1 2 1 2 2 2 2 2 2 2 1 2 3 2 1 1 3 1 2 2 1 1
## [704] 1 2 2 2 2 2 2 2 1 2 2 1 2 3 2 2 2 3 2 2 2 2 2 1 1 1 1 2 2 2 1 1 2 1 2 3 1
## [741] 1 1 1 1 1 1 2 3 1 2 1 2 2 2 2 2 2 2 1 1 1 3 2 2 2 2 1 2 2 1 1 3 2 1 1 2 1
## [778] 2 1 3 2 3 1 2 1 2 2 1 2 1 2 2 2 3 1 2 1 2 1 2 3 1 2 2 2 3 2 2 2 1 2 2 1 2
## [815] 1 2 1 1 2 2 1 1 1 1 2 2 3 2 2 2 2 2 1 2 2 1 3 3 2 2 2 1 2 3 2 3 2 3 2 3 1
## [852] 1 3 1 2 2 3 1 2 2 3 2 2 2 1 2 1 2 2 2 2 1 2 2 2 2 2 3 2 1 2 2 3 1 1 3 2 2
## [889] 1 2 2 1 2 2 1 1 2 2 2 2 2 1 1 2 2 2 2 2 2 1 2 2 1 1 1 2 1 1 1 2 2 2 1 2 1
## [926] 1 3 1 2 2 2 1 2 2 2 3 2 2 1 2 1 2 2 3 2 3 2 2 1 2 2 2 3 2 3 1 1 1 3
##
## Within cluster sum of squares by cluster:
## [1] 2941687388  801520125 2520529082
##  (between_SS / total_SS =  87.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
nclust <- 10
wss <- numeric(nclust)
set.seed(421)
## Looping through different number of clusters
for (i in 1:nclust) {
  km.out2 <- kmeans(alc_2cols2, centers = i, nstart = 20)
  wss[i] <- km.out2$tot.withinss
}

## Plotting
wss_df2 <- tibble(clusters = 1:nclust, wss = wss)
sc_plot2 <- ggplot(wss_df2, aes(x = clusters, y = wss, group = 1)) +
  geom_point(size = 3) +
  geom_line() +
  scale_x_continuous(breaks = c(2, 4, 6, 8, 10)) +
  scale_y_continuous(labels = scales::comma) +
  xlab("Number of Clusters") +
  ylab("Within Cluster Sum of Squares") +
  ggtitle("Optimal Number of Clusters Plot")
sc_plot2 +
  geom_hline(
    yintercept = wss,
    linetype = 'dashed',
    col = c(rep('black',4),'red', rep('black', 5))
  )
```

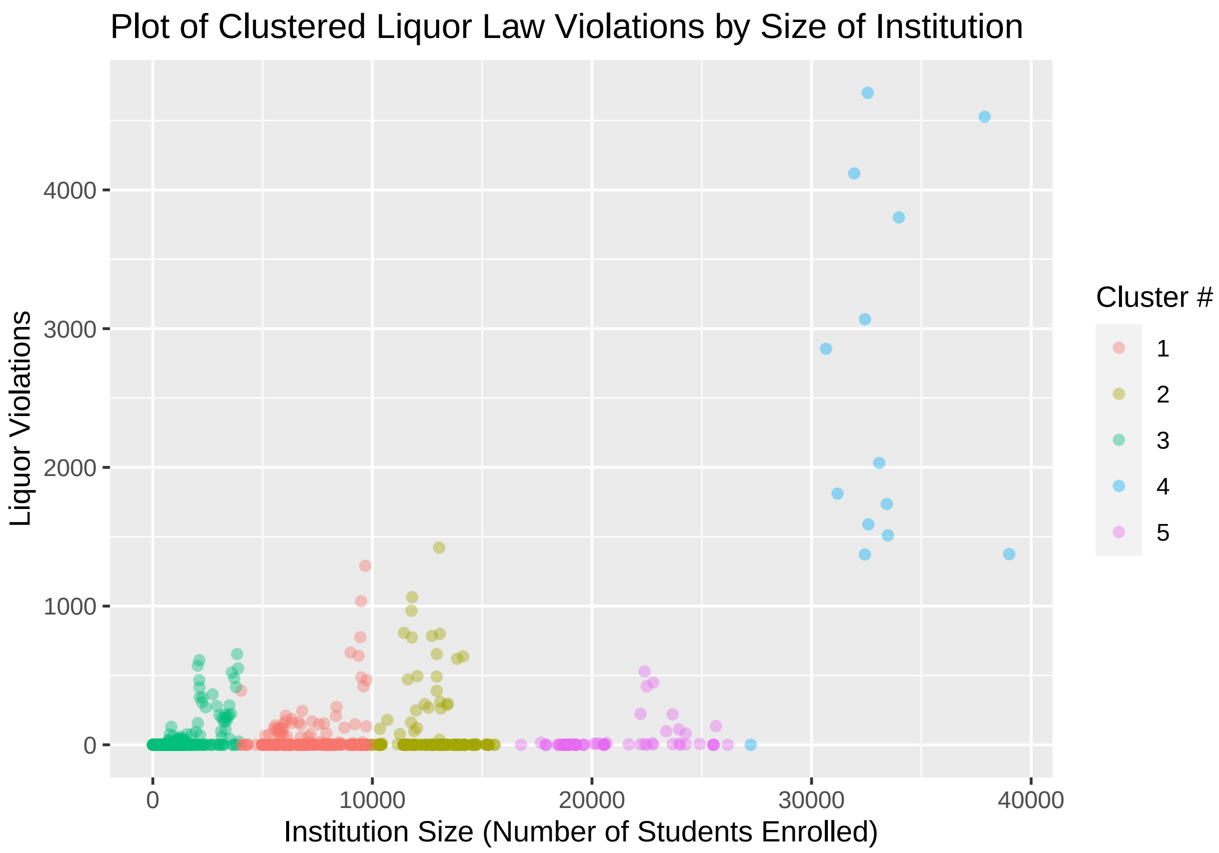


```
k <- 5

set.seed(421)
km.out2 <- kmeans(alc_2cols2, centers = k, nstart = 20)

train_num$cluster_id2 <- factor(km.out2$cluster)
p1 <- ggplot(train_num, aes(Survey.year, all_liquor_violations, color = cluster_id2)) +
    geom_point(alpha = 0.40) +
    xlab("Survey Year") +
    ylab("Liquor Violations") +
  ggtitle("Plot of Clustered Liquor Law Violations by Survey Year") +
  labs(color = "Cluster #")
```

```
train_num$cluster_id2 <- factor(km.out2$cluster)
p2 <- ggplot(train_num, aes(Institution.Size_all_campus, all_liquor_violations, color = cluster_id2)) +
    geom_point(alpha = 0.40) +
    xlab("Institution Size (Number of Students Enrolled)") +
    ylab("Liquor Violations") +
  ggtitle("Plot of Clustered Liquor Law Violations by Size of Institution") +
  labs(color = "Cluster #")
p1
```



```
p2
```



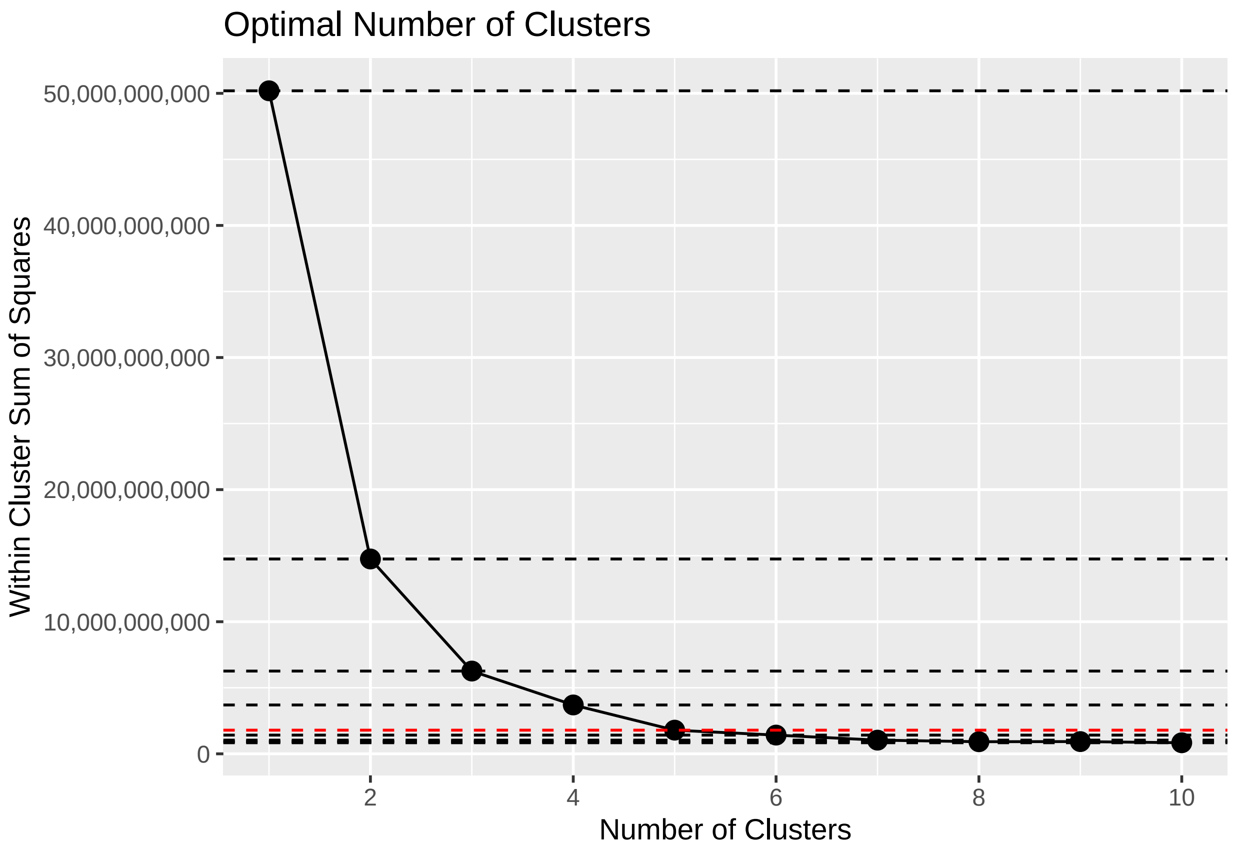
```
#grid.arrange(p1, p2, ncol = 2)
```

Clustering with our new dataset

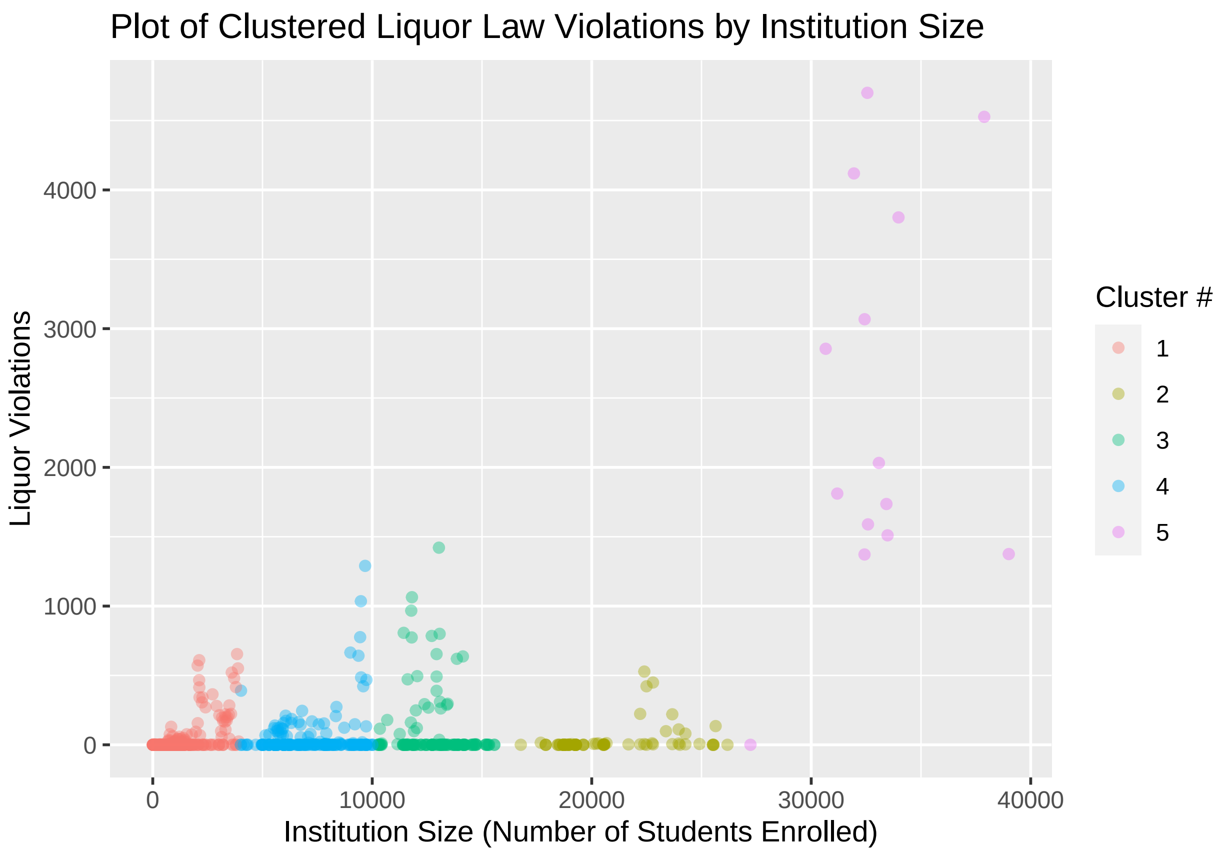
```
alc_2cols1 <- train[ , c("all_liquor_violations", "Institution.Size_all_campus")]
set.seed(421)
km.out <- kmeans(alc_2cols1, centers = 3, nstart = 20)
km.out
```

```
## K-means clustering with 3 clusters of sizes 536, 97, 326
##
## Cluster means:
##   all_liquor_violations Institution.Size_all_campus
## 1              20.76866                    976.4328
## 2             380.19588                  22326.0103
## 3              71.97546                   9736.0583
##
## Clustering vector:
##   [1] 1 1 1 3 1 1 1 1 1 1 2 2 1 1 1 1 2 3 1 3 1 3 3 3 1 1 3 1 1 3 3 1 3 3 3 3 1
##  [38] 3 1 1 1 1 3 1 1 1 1 1 3 1 3 1 1 3 3 1 1 1 3 3 1 3 1 1 1 1 3 1 1 1 1 1 1 3
##  [75] 1 1 1 1 3 2 2 1 3 3 3 2 3 1 3 3 1 1 1 3 3 1 2 1 3 1 3 1 2 3 3 1 1 2 2 3 3
## [112] 3 1 2 1 3 1 1 1 3 3 3 1 1 1 3 1 3 3 3 1 1 1 1 3 2 1 2 1 1 1 2 1 2 3 1 3 1
## [149] 3 1 1 3 1 1 3 1 1 1 3 3 3 3 1 1 1 1 1 2 1 1 3 1 1 1 3 3 2 1 2 3 1 1 1 1 3
## [186] 2 1 3 1 1 3 1 3 1 1 1 1 1 3 1 1 1 2 1 2 2 1 3 3 3 3 1 3 3 1 3 1 1 1 3 1 1
## [223] 1 1 3 2 3 3 3 1 2 1 3 1 1 2 1 1 3 1 1 1 1 1 3 3 1 1 1 1 3 3 3 1 3 1 1 1 3
## [260] 3 3 1 1 1 1 1 1 2 3 1 1 1 1 1 3 3 3 1 1 3 1 1 1 3 1 1 3 1 3 1 1 3 1 1 2 1
## [297] 1 1 1 1 1 3 1 1 1 1 3 3 1 1 1 2 1 3 3 1 1 1 3 1 2 3 3 3 1 3 2 1 1 1 1 1 1
## [334] 1 1 2 1 3 1 3 3 3 1 3 1 1 3 2 1 2 1 3 1 1 1 1 2 1 1 1 1 3 1 1 3 3 1 1 3 3
## [371] 1 1 3 1 1 3 1 3 1 1 1 2 1 3 1 3 1 1 3 1 1 2 3 3 1 1 2 1 1 2 3 1 3 3 1 3 1
## [408] 1 1 3 3 3 2 1 1 3 3 3 1 1 1 2 2 2 1 1 1 1 1 3 1 1 3 2 1 3 2 3 3 3 3 1 1 1
## [445] 1 1 1 2 1 3 3 1 1 3 3 3 1 2 3 1 1 1 1 3 1 3 3 3 2 1 3 2 1 3 3 1 2 3 3 1 1
## [482] 3 1 3 1 1 1 3 1 1 1 1 1 1 3 3 1 3 3 1 3 1 1 1 3 1 3 1 3 1 3 3 1 1 1 3 1 1
## [519] 3 1 1 3 3 1 1 1 3 1 3 1 1 3 3 1 1 3 1 1 1 3 1 3 2 3 3 3 3 1 1 3 2 1 1 1 3
## [556] 1 3 1 3 3 2 1 1 2 3 3 3 1 1 1 1 3 1 2 1 1 1 1 1 1 1 3 1 1 1 1 1 1 3 1 2 3
## [593] 1 1 1 3 2 1 1 3 1 3 2 3 1 1 3 1 3 2 1 3 1 3 3 1 3 1 1 3 3 3 1 1 3 2 1 1 1
## [630] 1 1 1 2 1 3 1 2 3 3 3 3 1 1 3 1 3 3 3 1 1 3 3 3 1 2 2 2 1 1 1 1 1 1 3 3 1
## [667] 1 1 3 1 1 3 3 1 1 1 3 3 1 1 3 3 1 3 1 1 1 1 1 1 1 3 1 2 1 3 3 2 3 1 1 3 3
## [704] 3 1 1 1 1 1 1 1 3 1 1 3 1 2 1 1 1 2 1 1 1 1 1 3 3 3 3 1 1 1 3 3 1 3 1 2 3
## [741] 3 3 3 3 3 3 1 2 3 1 3 1 1 1 1 1 1 1 3 3 3 2 1 1 1 1 3 1 1 3 3 2 1 3 3 1 3
## [778] 1 3 2 1 2 3 1 3 1 1 3 1 3 1 1 1 2 3 1 3 1 3 1 2 3 1 1 1 2 1 1 1 3 1 1 3 1
## [815] 3 1 3 3 1 1 3 3 3 3 1 1 2 1 1 1 1 1 3 1 1 3 2 2 1 1 1 3 1 2 1 2 1 2 1 2 3
## [852] 3 2 3 1 1 2 3 1 1 2 1 1 1 3 1 3 1 1 1 1 3 1 1 1 1 1 2 1 3 1 1 2 3 3 2 1 1
## [889] 3 1 1 3 1 1 3 3 1 1 1 1 1 3 3 1 1 1 1 1 1 3 1 1 3 3 3 1 3 3 3 1 1 1 3 1 3
## [926] 3 2 3 1 1 1 3 1 1 1 2 1 1 3 1 3 1 1 2 1 2 1 1 3 1 1 1 2 1 2 3 3 3 2
##
## Within cluster sum of squares by cluster:
## [1]  801514625 2520528102 2941684316
##  (between_SS / total_SS =  87.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
nclust <- 10
wss <- numeric(nclust)
set.seed(421)
## Looping through different number of clusters
for (i in 1:nclust) {
  km.out <- kmeans(alc_2cols1, centers = i, nstart = 20)
  wss[i] <- km.out$tot.withinss
}

## Plotting
wss_df <- tibble(clusters = 1:nclust, wss = wss)
sc_plot <- ggplot(wss_df, aes(x = clusters, y = wss, group = 1)) +
  geom_point(size = 3) +
  geom_line() +
  scale_x_continuous(breaks = c(2, 4, 6, 8, 10)) +
  scale_y_continuous(labels = scales::comma) +
  xlab("Number of Clusters") +
  ylab("Within Cluster Sum of Squares") +
  ggtitle("Optimal Number of Clusters")
sc_plot +
  geom_hline(
    yintercept = wss,
    linetype = 'dashed',
    col = c(rep('black',4),'red', rep('black', 5))
  )
```



```
k <- 5

set.seed(421)
km.out <- kmeans(alc_2cols1, centers = k, nstart = 20)

train_num$cluster_id <- factor(km.out$cluster)
ggplot(train_num, aes(Institution.Size_all_campus, all_liquor_violations, color = cluster_id)) +
    geom_point(alpha = 0.40) +
    xlab("Institution Size (Number of Students Enrolled)") +
    ylab("Liquor Violations") +
  ggtitle("Plot of Clustered Liquor Law Violations by Institution Size") +
  labs(color = "Cluster #")
```



Most Recent Attempt

```r
library(cluster)
test_num <- test |> as_tibble() |> select(-where(is.character))
test_num <- test_num[, !names(test_num) %in% c('Unitid_all_campus', 'OPEID_all_campus', 'Campus.ID_all_c
cols_test <- train_num[ , c("all_liquor_violations", "Survey.year", "Institution.Size_all_campus")]

kmTEST <- kmeans(cols_test, centers = k, nstart = 20)

sil <- silhouette(kmTEST$cluster, dist(train_num))
```

```
## Warning in dist(train_num): NAs introduced by coercion
```

```r
library(dplyr)
data_frame <- data.frame(sil_width = sil[, "sil_width"],
                         cluster = sil[, "cluster"])
avg_sil_scores_by_cluster <- data_frame %>%
  group_by(cluster) %>%
  summarise(avg_silhouette = mean(sil_width))
print(avg_sil_scores_by_cluster)
```

```
## # A tibble: 5 x 2
##   cluster avg_silhouette
##     <dbl>          <dbl>
## 1       1          0.670
## 2       2          0.836
## 3       3          0.658
## 4       4          0.582
## 5       5          0.643
```

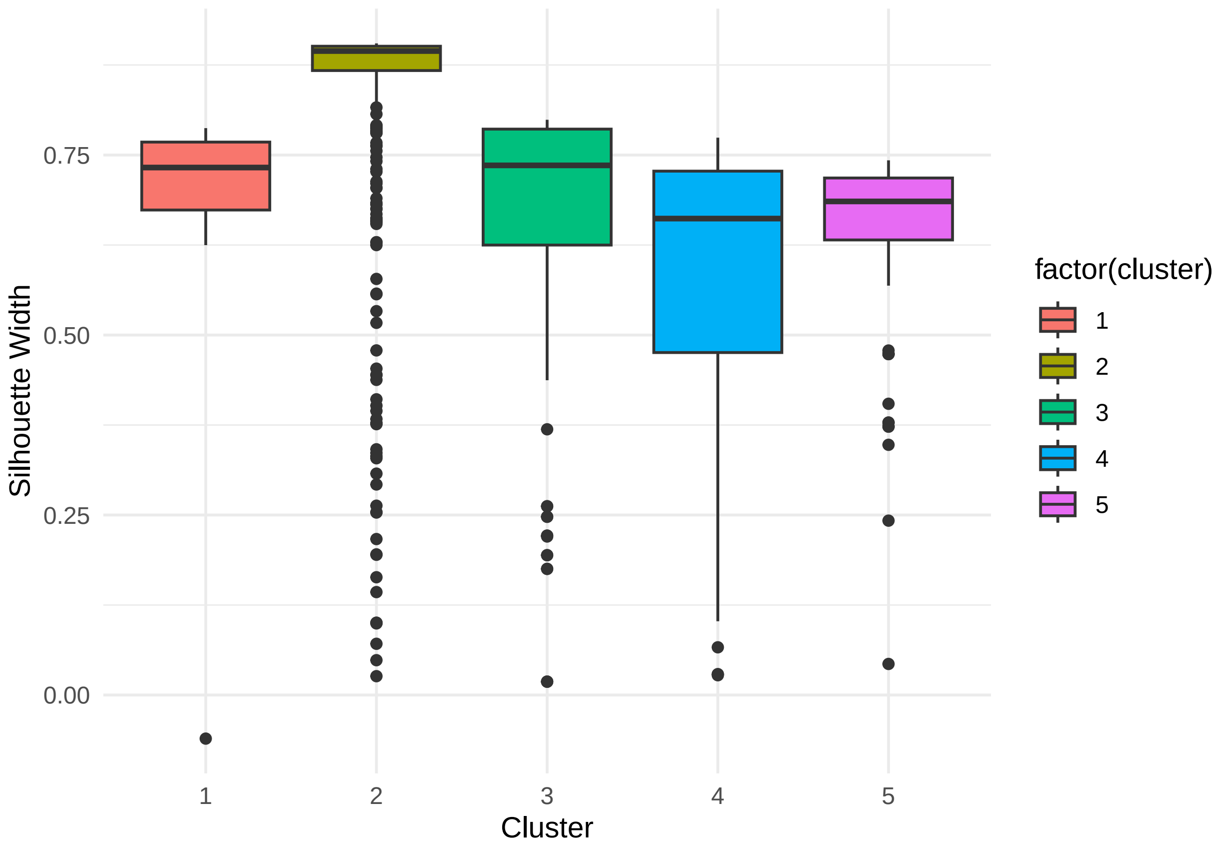
```
kable(avg_sil_scores_by_cluster, format = "markdown",
      col.names =c("Cluster", "Average Silhouette Score"),
        caption = "Average Silhouette Scores by Cluster")
```

Table 4: Average Silhouette Scores by Cluster

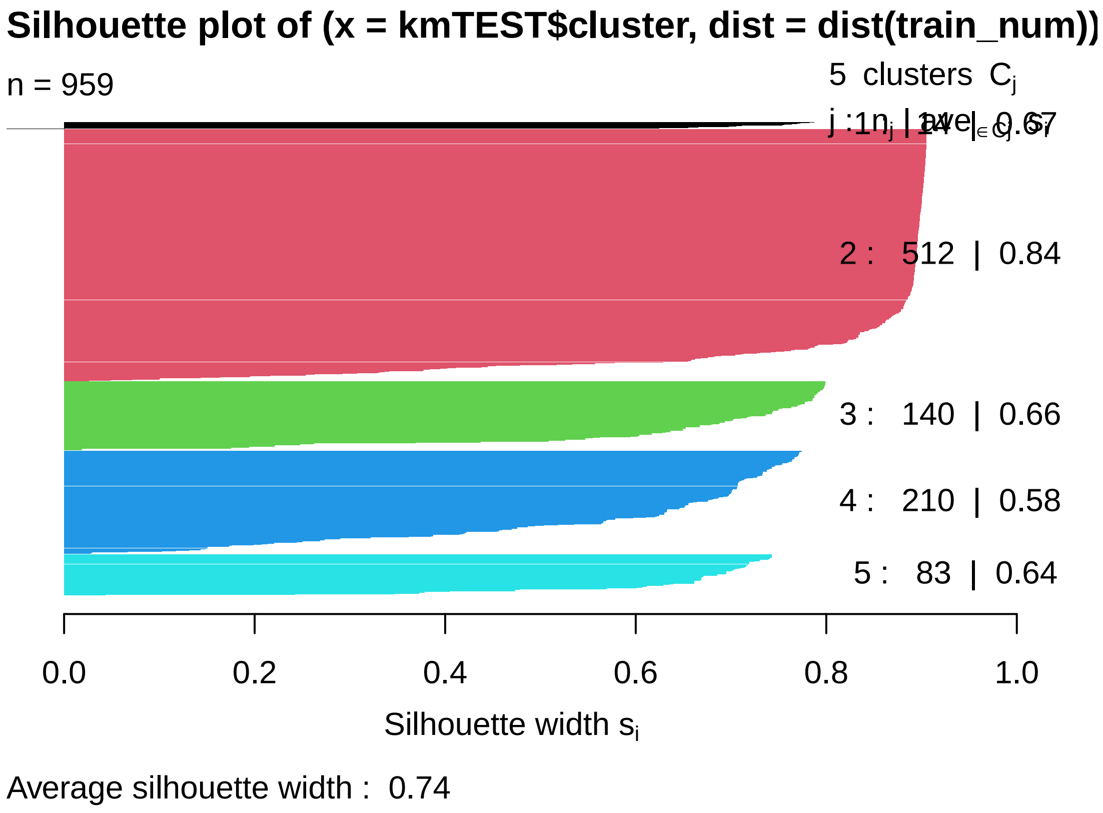
Cluster	Average Silhouette Score
1	0.6702670
2	0.8359748
3	0.6577862
4	0.5818711
5	0.6429205

x

```
ggplot(data_frame, aes(x = factor(cluster), y = sil_width, fill = factor(cluster))) +
  geom_boxplot() +
  labs(x = "Cluster", y = "Silhouette Width") +
  theme_minimal()
```



```r
plot(sil, col = 1:5, border = NA)
```



3D Plot !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```
#library(plotly)

#fig <- plot_ly(train_num, x = ~Institution.Size_all_campus, y = ~Survey.year, z = ~all_liquor_violatio
#fig <- fig %>% layout(title = "Cluster Plot of Institution Size and Survey Year",
          #            scene = list(xaxis = list(title = "Institution Size"),
           #                        yaxis = list(title = "Survey Year"),
            #                       zaxis = list(title = "Liquor Law Violations")))
#fig
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.