Final-Project\_Arham\_v5.R

arham

2023-12-11

#=========================================================================#  
# libraries   
#=========================================================================#  
  
# libary   
#install.packages("cluster")  
#install.packages("factoextra")   
  
# Load necessary libraries  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(factoextra)

## Loading required package: ggplot2

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

library(purrr)  
library(ggplot2)  
library(reshape2)  
library(stargazer)

##   
## Please cite as:

## Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.

## R package version 5.2.3. https://CRAN.R-project.org/package=stargazer

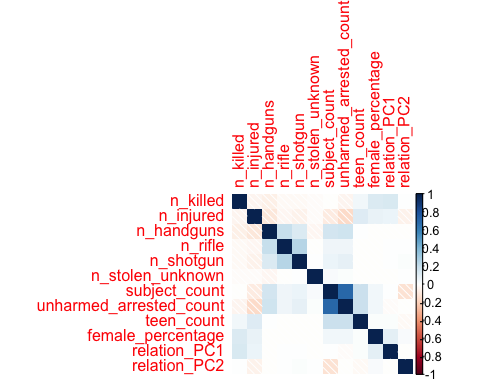
library(cluster)  
library(factoextra)  
library(cluster)  
library(dplyr)  
library(stargazer)  
library(ggplot2)  
library(ggfortify)  
  
  
  
#=========================================================================#  
# read and set project filters   
#=========================================================================#  
  
file\_path <- "/Users/arham/Downloads/02. MVS/Final Project/Dataset 1 — Gun violence.csv"  
gun <- read.csv(file\_path)  
  
gun <- subset(gun, n\_guns\_involved != 0) # filter to gun incidents ,i.e., n\_guns\_involved >1  
# filter to relevant columns  
gun <- gun[, c("incident\_id", "date", "state", "city\_or\_county", "latitude", "longitude", "n\_killed", "n\_injured", "congressional\_district", "gun\_stolen", "gun\_type", "incident\_characteristics",  
 "n\_guns\_involved", "notes", "participant\_age",  
 "participant\_age\_group", "participant\_gender",   
 "participant\_relationship","participant\_status", "participant\_type")]  
  
#### Prelimary filter to columns which are not repeated or non-redundant  
gun <- na.omit(gun)  
  
#=========================================================================#  
# read and set project filters   
#=========================================================================#  
  
##============================ 1. Stolen Guns  
  
  
# count number of occurences of stolen of gun\_stolen and record in new column (a cell has values like :0::Unknown||1::Unknown )  
gun$stolen\_count <- sapply(gun$gun\_stolen, function(x) length(strsplit(x, split = "\\|\\|")[[1]]))  
# remove gun\_stolen column  
gun <- gun[, !(names(gun) %in% "gun\_stolen")]  
  
##============================ 2. gun\_types  
  
  
  
# First, convert all the text in the "gun\_type" column to lowercase  
gun$gun\_type <- tolower(gun$gun\_type)  
# Count the number of occurrences of specific keywords in the "gun\_type" column, split using "||"  
gun$n\_handguns <- sapply(gun$gun\_type, function(x) length(grep("handgun", strsplit(x, split = "\\|\\|")[[1]])))  
gun$n\_auto <- sapply(gun$gun\_type, function(x) length(grep("auto", strsplit(x, split = "\\|\\|")[[1]])))  
gun$n\_mm <- sapply(gun$gun\_type, function(x) length(grep("mm", strsplit(x, split = "\\|\\|")[[1]])))  
gun$n\_spl <- sapply(gun$gun\_type, function(x) length(grep("spl", strsplit(x, split = "\\|\\|")[[1]])))  
gun$n\_mag <- sapply(gun$gun\_type, function(x) length(grep("mag", strsplit(x, split = "\\|\\|")[[1]])))  
# Sum up counts and remove unnecessary columns  
gun$n\_handguns <- gun$n\_handguns + gun$n\_auto + gun$n\_mm + gun$n\_spl + gun$n\_mag   
gun <- gun[, !(names(gun) %in% c("n\_auto", "n\_mm", "n\_spl", "n\_mag", "n\_win"))]  
# Count occurrences of 'win' and 'rifle', adjust counts, sum them up, and remove unnecessary columns  
gun$n\_win <- sapply(gun$gun\_type, function(x) length(grep("win", strsplit(x, split = "\\|\\|")[[1]])))  
  
gun$n\_rifle <- sapply(gun$gun\_type, function(x) length(grep("rifle", strsplit(x, split = "\\|\\|")[[1]])))  
gun$n\_rifle <- gun$n\_rifle + gun$n\_win  
gun <- gun[, !(names(gun) %in% c("n\_win"))]  
# Count occurrences of 'gauge' and 'shotgun', adjust counts, sum them up, and remove unnecessary columns  
gun$n\_gauge <- sapply(gun$gun\_type, function(x) length(grep("gauge", strsplit(x, split = "\\|\\|")[[1]])))  
gun$n\_shotgun <- sapply(gun$gun\_type, function(x) length(grep("shotgun", strsplit(x, split = "\\|\\|")[[1]])))  
gun$n\_shotgun <- gun$n\_shotgun + gun$n\_gauge  
gun <- gun[, !(names(gun) %in% c("n\_gauge"))]  
# Count the number of occurrences of "||", add 1 to it, and subtract counts of shotguns, rifles, and handguns  
gun$n\_other <- sapply(gun$gun\_type, function(x) sum(gregexpr("\\|\\|", x)[[1]] > 0))  
gun$n\_other <- gun$n\_other - gun$n\_shotgun - gun$n\_rifle - gun$n\_handguns + 1  
# remove gun\_type column  
gun <- gun[, !(names(gun) %in% "gun\_type")]  
  
  
##============================ 3. number of suspects, victims, and total people involved  
# count number of Subject-Suspect, Victim, and Total people involved  
gun$subject\_count <- sapply(gun$participant\_type, function(x) length(grep("Subject-Suspect", strsplit(x, split = "\\|\\|")[[1]])))  
gun$victim\_count <- sapply(gun$participant\_type, function(x) length(grep("Victim", strsplit(x, split = "\\|\\|")[[1]])))  
gun$total\_count <- sapply(gun$participant\_type, function(x) length(strsplit(x, split = "\\|\\|")[[1]]))  
#remove participant\_type column  
gun <- gun[, !(names(gun) %in% "participant\_type")]  
  
  
##============================ 4. number of injured, killed, unharmed arrested, and unharmed  
# from participant\_status column count number of Injured, Killed, and 'Unharmed, Arrested', and 'Unharmed'  
gun$injured\_count <- sapply(gun$participant\_status, function(x) length(grep("Injured", strsplit(x, split = "\\|\\|")[[1]])))  
gun$killed\_count <- sapply(gun$participant\_status, function(x) length(grep("Killed", strsplit(x, split = "\\|\\|")[[1]])))  
gun$unharmed\_arrested\_count <- sapply(gun$participant\_status, function(x) length(grep("Unharmed, Arrested", strsplit(x, split = "\\|\\|")[[1]])))  
gun$unharmed\_count <- sapply(gun$participant\_status, function(x) length(grep("Unharmed", strsplit(x, split = "\\|\\|")[[1]])))  
  
  
# from participant age group count number of Adult 18+ and Teen 12-17, child 0-11  
gun$adult\_count <- sapply(gun$participant\_age\_group, function(x) length(grep("Adult 18+", strsplit(x, split = "\\|\\|")[[1]])))  
gun$teen\_count <- sapply(gun$participant\_age\_group, function(x) length(grep("Teen 12-17", strsplit(x, split = "\\|\\|")[[1]])))  
gun$child\_count <- sapply(gun$participant\_age\_group, function(x) length(grep("Child 0-11", strsplit(x, split = "\\|\\|")[[1]])))  
# remove participant\_age\_group column  
gun <- gun[, !(names(gun) %in% "participant\_age\_group")]  
  
  
##============================ 5. gender ratio  
  
calculate\_female\_percentage <- function(participant\_gender) {  
 genders <- strsplit(participant\_gender, "\\|\\|")[[1]]  
 total\_participants <- length(genders)  
 female\_count <- sum(grepl("Female", genders))  
   
 if (total\_participants > 0) {  
 return((female\_count / total\_participants) \* 100)  
 } else {  
 return(NA)  
 }  
}  
  
##============================ 6. relations between participants  
gun$participant\_relationship <- tolower(gun$participant\_relationship)  
# Family  
# Random victims  
# Aquaintance  
# Significant Others  
# Armed Robbery  
# Gang  
# Mass Shooting  
# Knows victims  
# Co-worker  
# Neighbor  
# Friends  
# Home Invasion  
# Does Not Know Victim  
  
# New column relation\_family = if family is present in participant\_relationship column, then 1 else 0  
gun <- gun %>% mutate(relation\_family = ifelse(grepl("family", participant\_relationship, ignore.case = TRUE), 1, 0))  
# New column relation\_random\_victims = if random victims is present in participant\_relationship column, then 1 else 0  
gun <- gun %>% mutate(relation\_random\_victims = ifelse(grepl("random victims", participant\_relationship, ignore.case = TRUE), 1, 0))  
# New column relation\_aquaintance = if aquaintance is present in participant\_relationship column, then 1 else 0  
gun <- gun %>% mutate(relation\_aquaintance = ifelse(grepl("aquaintance", participant\_relationship, ignore.case = TRUE), 1, 0))  
# New column relation\_significant\_others = if significant others is present in participant\_relationship column, then 1 else 0  
gun <- gun %>% mutate(relation\_significant\_others = ifelse(grepl("significant others", participant\_relationship, ignore.case = TRUE), 1, 0))  
# New column relation\_armed\_robbery = if armed robbery is present in participant\_relationship column, then 1 else 0  
gun <- gun %>% mutate(relation\_armed\_robbery = ifelse(grepl("armed robbery", participant\_relationship, ignore.case = TRUE), 1, 0))  
# New column relation\_gang = if gang is present in participant\_relationship column, then 1 else 0  
gun <- gun %>% mutate(relation\_gang = ifelse(grepl("gang", participant\_relationship, ignore.case = TRUE), 1, 0))  
# New column relation\_mass\_shooting = if mass shooting is present in participant\_relationship column, then 1 else 0  
gun <- gun %>% mutate(relation\_mass\_shooting = ifelse(grepl("mass shooting", participant\_relationship, ignore.case = TRUE), 1, 0))  
# New column relation\_knows\_victims = if knows victims is present in participant\_relationship column, then 1 else 0  
gun <- gun %>% mutate(relation\_knows\_victims = ifelse(grepl("knows victims", participant\_relationship, ignore.case = TRUE), 1, 0))  
# New column relation\_co\_worker = if co-worker is present in participant\_relationship column, then 1 else 0  
gun <- gun %>% mutate(relation\_co\_worker = ifelse(grepl("co-worker", participant\_relationship, ignore.case = TRUE), 1, 0))  
# New column relation\_neighbor = if neighbor is present in participant\_relationship column, then 1 else 0  
gun <- gun %>% mutate(relation\_neighbor = ifelse(grepl("neighbor", participant\_relationship, ignore.case = TRUE), 1, 0))  
# New column relation\_friends = if friends is present in participant\_relationship column, then 1 else 0  
gun <- gun %>% mutate(relation\_friends = ifelse(grepl("friends", participant\_relationship, ignore.case = TRUE), 1, 0))  
# New column relation\_home\_invasion = if home invasion is present in participant\_relationship column, then 1 else 0  
gun <- gun %>% mutate(relation\_home\_invasion = ifelse(grepl("home invasion", participant\_relationship, ignore.case = TRUE), 1, 0))  
# New column relation\_does\_not\_know\_victim = if does not know victim is present in participant\_relationship column, then 1 else 0  
gun <- gun %>% mutate(relation\_does\_not\_know\_victim = ifelse(grepl("does not know victim", participant\_relationship, ignore.case = TRUE), 1, 0))  
# remove participant\_relationship column  
gun <- gun[, !(names(gun) %in% "participant\_relationship")]  
  
  
  
  
  
  
  
  
  
  
#=========================================================================#  
# More Feature Engineering  
#=========================================================================#  
  
# Apply the function to create a new column for female percentage  
gun <- gun %>%  
 mutate(female\_percentage = sapply(participant\_gender, calculate\_female\_percentage))  
# remove participant\_status column  
gun <- gun[, !(names(gun) %in% "participant\_status")]  
  
  
# save as gun\_preprocessed\_v1.csv  
write.csv(gun, file = "gun\_preprocessed\_vF.csv", row.names = FALSE)  
#gun <- read.csv("gun\_preprocessed\_vF.csv")  
  
tempo = gun   
  
selected\_columns <- c(  
 "child\_count",  
 "relation\_family",  
 "relation\_random\_victims",  
 "relation\_aquaintance",  
 "relation\_significant\_others",  
 "relation\_armed\_robbery",  
 "relation\_gang",  
 "relation\_mass\_shooting",  
 "relation\_knows\_victims",  
 "relation\_co\_worker",  
 "relation\_neighbor",  
 "relation\_friends",  
 "relation\_home\_invasion",  
 "relation\_does\_not\_know\_victim"  
)  
  
gun\_pca\_result <- prcomp(gun[, selected\_columns])  
principal\_components <- as.data.frame(gun\_pca\_result$x[, 1:2])  
names(principal\_components) <- c("relation\_PC1", "relation\_PC2")  
gun <- cbind(gun[, -which(names(gun) %in% selected\_columns)], principal\_components)  
gun <- gun[, !(names(gun) %in% selected\_columns)]  
  
gun <- gun[, !(names(gun) %in% c("participant\_age","congressional\_district", "state", "city\_or\_county", "latitude", "longitude"))]  
exclude\_columns <- c('incident\_id', 'date', 'notes', 'incident\_characteristics', 'gun\_type', 'participant\_relationship', 'location\_description',  
 'participant\_gender', 'adult\_count', 'gun\_stolen','unharmed\_count','stolen\_count', 'victim\_count', 'total\_count', 'injured\_count', 'killed\_count','n\_guns\_involved')  
X <- gun[, !(names(gun) %in% exclude\_columns)]  
  
  
names(gun)

## [1] "incident\_id" "date"   
## [3] "n\_killed" "n\_injured"   
## [5] "incident\_characteristics" "n\_guns\_involved"   
## [7] "notes" "participant\_gender"   
## [9] "stolen\_count" "n\_handguns"   
## [11] "n\_rifle" "n\_shotgun"   
## [13] "n\_other" "subject\_count"   
## [15] "victim\_count" "total\_count"   
## [17] "injured\_count" "killed\_count"   
## [19] "unharmed\_arrested\_count" "unharmed\_count"   
## [21] "adult\_count" "teen\_count"   
## [23] "female\_percentage" "relation\_PC1"   
## [25] "relation\_PC2"

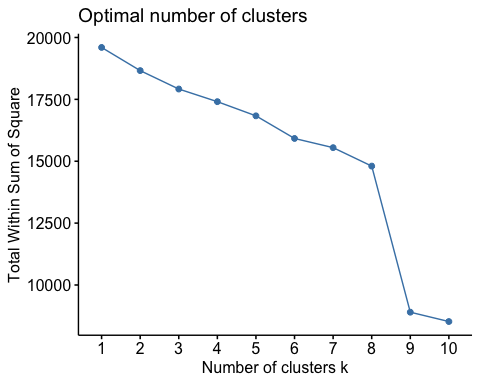
#rename n\_other to n\_stolen\_unkown  
names(X)[names(X) == 'n\_other'] <- 'n\_stolen\_unknown'  
  
# X[is.infinite(X)] <- 0  
X[is.na(X)] <- 0  
OG <-X   
X <- scale(X)  
X <- as.data.frame(X)  
W <- X  
Z <- X  
  
  
######### correlation  
library(corrplot)

## corrplot 0.92 loaded

corr <- cor(X)  
corrplot(corr, method = "shade")



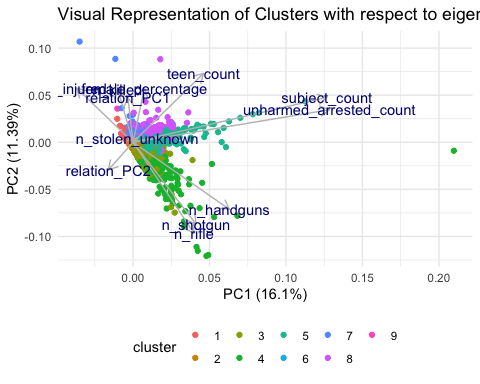
# ========================================================================#  
## K Means Clustering  
# ========================================================================#  
  
W <- X  
  
#================= What should be the k?   
  
## support 1  
set.seed(123) # for reproducibility  
sample\_indices <- sample(nrow(W), 1000) # adjust the size as needed  
subset\_W <- W[sample\_indices, ]  
par(pty = "m")  
fviz\_nbclust(subset\_W, pam, method = "wss")



## support 2  
#calculate gap statistic based on number of clusters  
##gap\_stat <- clusGap(subset\_W,  
 #FUN = pam,  
 #K.max = 10, #max clusters to consider  
 #B = 50) #total bootstrapped iterations  
  
###plot number of clusters vs. gap statistic  
##viz\_gap\_stat(gap\_stat)  
  
# k = 9  
  
  
#================= apply k means for 9 clusters  
  
  
set.seed(123)  
kmeans\_W <- kmeans(W, centers = 9, nstart = 25)  
kmeans\_W$cluster <- as.factor(kmeans\_W$cluster)  
# add cluster column to W  
W$cluster <- as.factor(kmeans\_W$cluster)  
OG$cluster <- as.factor(kmeans\_W$cluster)  
  
summary\_by\_cluster <- OG %>%  
 group\_by(cluster) %>%  
 summarise(  
 size = n(),  
 mean\_n\_killed = round(mean(n\_killed),2),  
 mean\_n\_injured = round(mean(n\_injured),2),  
 mean\_n\_handguns = round(mean(n\_handguns),2),  
 mean\_n\_rifle = round(mean(n\_rifle),2),  
 mean\_n\_shotgun = round(mean(n\_shotgun),2),  
 mean\_n\_stolen\_unknown = round(mean(n\_stolen\_unknown),2),  
 mean\_subject\_count = round(mean(subject\_count),2),  
 mean\_unharmed\_arrested\_count = round(mean(unharmed\_arrested\_count),2),  
 mean\_teen\_count = round(mean(teen\_count),2),  
 mean\_female\_percentage = round(mean(female\_percentage),2),  
 mean\_relation\_PC1 = round(mean(as.numeric(relation\_PC1)),2),  
 mean\_relation\_PC2 = round(mean(as.numeric(relation\_PC2)),2)  
 )  
  
  
  
summary\_by\_cluster = data.frame(summary\_by\_cluster)  
summary\_by\_cluster = t(summary\_by\_cluster)  
colnames(summary\_by\_cluster) <- summary\_by\_cluster[1, ]  
summary\_by\_cluster <- summary\_by\_cluster[-1, ]  
  
  
stargazer(summary\_by\_cluster, title = "Cluster Profiles",type = "html", digits = 2)

##   
## <table style="text-align:center"><caption><strong>Cluster Profiles</strong></caption>  
## <tr><td colspan="10" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"></td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr>  
## <tr><td colspan="10" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left">size</td><td>35588</td><td>60564</td><td>4084</td><td>217</td><td>8309</td><td>2904</td><td>1987</td><td>2396</td><td>12242</td></tr>  
## <tr><td style="text-align:left">mean\_n\_killed</td><td>0.06</td><td>0.29</td><td>0.16</td><td>0.01</td><td>0.18</td><td>0.23</td><td>0.57</td><td>0.32</td><td>0.58</td></tr>  
## <tr><td style="text-align:left">mean\_n\_injured</td><td>1.22</td><td>0.01</td><td>0.21</td><td>0.01</td><td>0.22</td><td>0.68</td><td>0.85</td><td>0.80</td><td>0.48</td></tr>  
## <tr><td style="text-align:left">mean\_n\_handguns</td><td>0.09</td><td>0.33</td><td>0.45</td><td>10.47</td><td>0.61</td><td>0.35</td><td>0.26</td><td>0.35</td><td>0.22</td></tr>  
## <tr><td style="text-align:left">mean\_n\_rifle</td><td>0.01</td><td>0.03</td><td>0.19</td><td>5.85</td><td>0.06</td><td>0.02</td><td>0.03</td><td>0.03</td><td>0.02</td></tr>  
## <tr><td style="text-align:left">mean\_n\_shotgun</td><td>0.00</td><td>0.00</td><td>1.18</td><td>1.25</td><td>0.01</td><td>0.02</td><td>0.04</td><td>0.04</td><td>0.00</td></tr>  
## <tr><td style="text-align:left">mean\_n\_stolen\_unknown</td><td>0.94</td><td>0.95</td><td>0.44</td><td>36.79</td><td>1.38</td><td>0.84</td><td>0.77</td><td>0.94</td><td>0.84</td></tr>  
## <tr><td style="text-align:left">mean\_subject\_count</td><td>0.55</td><td>0.72</td><td>1.11</td><td>1.71</td><td>2.82</td><td>1.70</td><td>0.87</td><td>2.06</td><td>0.84</td></tr>  
## <tr><td style="text-align:left">mean\_unharmed\_arrested\_count</td><td>0.12</td><td>0.39</td><td>0.74</td><td>1.23</td><td>2.39</td><td>0.49</td><td>0.33</td><td>1.46</td><td>0.39</td></tr>  
## <tr><td style="text-align:left">mean\_teen\_count</td><td>0.08</td><td>0.05</td><td>0.06</td><td>0.05</td><td>0.13</td><td>0.15</td><td>0.12</td><td>2.42</td><td>0.07</td></tr>  
## <tr><td style="text-align:left">mean\_female\_percentage</td><td>1.98</td><td>0.21</td><td>7.95</td><td>4.71</td><td>11.92</td><td>7.89</td><td>24.68</td><td>9.76</td><td>67.77</td></tr>  
## <tr><td style="text-align:left">mean\_relation\_PC1</td><td>-0.02</td><td>-0.02</td><td>-0.01</td><td>-0.02</td><td>-0.02</td><td>-0.10</td><td>1.17</td><td>-0.01</td><td>0.00</td></tr>  
## <tr><td style="text-align:left">mean\_relation\_PC2</td><td>0.02</td><td>0.02</td><td>0.03</td><td>0.02</td><td>0.02</td><td>-0.97</td><td>-0.08</td><td>0.01</td><td>0.04</td></tr>  
## <tr><td colspan="10" style="border-bottom: 1px solid black"></td></tr></table>

# pca on W excluding cluster column  
pca <- prcomp(W[, -ncol(W)], scale = TRUE)  
# autoplot(pca, data = W[, -1], colour = 'cluster', loadings = TRUE, loadings.label = TRUE) +  
# theme\_minimal() +  
# theme(legend.position = 'bottom') +  
# ggtitle("PCA Colored by Cluster")  
  
  
# autoplot clusters  
autoplot(pca, data = W[, -1], colour = 'cluster', loadings = TRUE,loadings.label = TRUE,loadings.color = 'grey', loadings.label.color = 'navyblue') +  
 theme\_minimal() +  
 theme(legend.position = 'bottom') +  
 ggtitle("Visual Representation of Clusters with respect to eigen vectors")



# # autoplot clusters  
# autoplot(pca, data = W[, -1], colour = 'cluster', loadings = TRUE, loadings.label = TRUE,loadings.color = 'black', loadings.label.color = 'black') +  
# theme\_minimal() +  
# theme(legend.position = 'bottom') +  
# ggtitle("PCA Colored by Cluster")  
#   
# #autoplot(pca, data = W[, -1], loadings = TRUE, loadings.label = TRUE) +  
# theme\_minimal() +  
# theme(legend.position = 'bottom') +  
# ggtitle("Crime Scene Incident Split")  
  
  
# add cluster names to OG by incident id  
tempo$cluster\_kmeans <- W$cluster  
  
guns\_f = tempo   
  
centroid\_col\_list <- c(  
 "n\_killed",  
 "n\_injured",  
 "n\_guns\_involved",  
 "stolen\_count",  
 "n\_handguns",  
 "n\_rifle",  
 "n\_shotgun",  
 "n\_other",  
 "subject\_count",  
 "victim\_count",  
 "total\_count",  
 "injured\_count",  
 "killed\_count",  
 "unharmed\_arrested\_count",  
 "unharmed\_count",  
 "adult\_count",  
 "teen\_count",  
 "child\_count",  
 "relation\_family",  
 "relation\_random\_victims",  
 "relation\_aquaintance",  
 "relation\_significant\_others",  
 "relation\_armed\_robbery",  
 "relation\_gang",  
 "relation\_mass\_shooting",  
 "relation\_knows\_victims",  
 "relation\_co\_worker",  
 "relation\_neighbor",  
 "relation\_friends",  
 "relation\_home\_invasion",  
 "relation\_does\_not\_know\_victim",  
 "female\_percentage",  
 "cluster\_kmeans")  
  
# filter guns\_f to only columns in centroid\_col\_list   
guns\_f1 <- guns\_f[, centroid\_col\_list]  
  
guns\_f1[is.na(guns\_f1)] <- 0  
  
# convert all columns to numeric  
guns\_f1 <- sapply(guns\_f1, as.numeric)  
  
guns\_f1 = data.frame(guns\_f1)  
  
summary\_by\_cluster\_f1 <- guns\_f1 %>% group\_by(cluster\_kmeans) %>% summarise(  
 size = n(),  
 mean\_n\_killed = round(mean(n\_killed), 2),  
 mean\_n\_injured = round(mean(n\_injured), 2),  
 mean\_n\_guns\_involved = round(mean(n\_guns\_involved), 2),  
 mean\_stolen\_count = round(mean(stolen\_count), 2),  
 mean\_n\_handguns = round(mean(n\_handguns), 2),  
 mean\_n\_rifle = round(mean(n\_rifle), 2),  
 mean\_n\_shotgun = round(mean(n\_shotgun), 2),  
 mean\_n\_other = round(mean(n\_other), 2),  
 mean\_subject\_count = round(mean(subject\_count), 2),  
 mean\_victim\_count = round(mean(victim\_count), 2),  
 mean\_total\_count = round(mean(total\_count), 2),  
 mean\_injured\_count = round(mean(injured\_count), 2),  
 mean\_killed\_count = round(mean(killed\_count), 2),  
 mean\_unharmed\_arrested\_count = round(mean(unharmed\_arrested\_count), 2),  
 mean\_unharmed\_count = round(mean(unharmed\_count), 2),  
 mean\_adult\_count = round(mean(adult\_count), 2),  
 mean\_teen\_count = round(mean(teen\_count), 2),  
 mean\_child\_count = round(mean(child\_count), 2),  
 mean\_relation\_family = round(mean(relation\_family), 2),  
 mean\_relation\_random\_victims = round(mean(relation\_random\_victims), 2),  
 mean\_relation\_aquaintance = round(mean(relation\_aquaintance), 2),  
 mean\_relation\_significant\_others = round(mean(relation\_significant\_others), 2),  
 mean\_relation\_armed\_robbery = round(mean(relation\_armed\_robbery), 2),  
 mean\_relation\_gang = round(mean(relation\_gang), 2),  
 mean\_relation\_mass\_shooting = round(mean(relation\_mass\_shooting), 2),  
 mean\_relation\_knows\_victims = round(mean(relation\_knows\_victims), 2),  
 mean\_relation\_co\_worker = round(mean(relation\_co\_worker), 2),  
 mean\_relation\_neighbor = round(mean(relation\_neighbor), 2),  
 mean\_relation\_friends = round(mean(relation\_friends), 2),  
 mean\_relation\_home\_invasion = round(mean(relation\_home\_invasion), 2)  
)  
summary\_by\_cluster\_f1 = t(summary\_by\_cluster\_f1)  
  
colnames(summary\_by\_cluster\_f1) <- summary\_by\_cluster\_f1[1, ]  
summary\_by\_cluster\_f1 <- summary\_by\_cluster\_f1[-1, ]  
summary\_by\_cluster\_f1 <- summary\_by\_cluster\_f1[-nrow(summary\_by\_cluster\_f1), ]  
summary\_by\_cluster\_f1 <- round(summary\_by\_cluster\_f1, 2)  
  
# stargazer summary\_by\_cluster\_f1  
stargazer(summary\_by\_cluster\_f1, title = "Cluster Profiles",type = "text",column.sep.width = "5pt", digits = 2)

##   
## Cluster Profiles  
## =========================================================================================  
## 1 2 3 4 5 6 7 8 9   
## -----------------------------------------------------------------------------------------  
## size 35,588 60,564 4,084 217 8,309 2,904 1,987 2,396 12,242  
## mean\_n\_killed 0.06 0.29 0.16 0.01 0.18 0.23 0.57 0.32 0.58   
## mean\_n\_injured 1.22 0.01 0.21 0.01 0.22 0.68 0.85 0.80 0.48   
## mean\_n\_guns\_involved 1.04 1.32 2.27 54.37 2.06 1.22 1.10 1.36 1.09   
## mean\_stolen\_count 1.04 1.32 2.27 54.37 2.06 1.22 1.10 1.36 1.09   
## mean\_n\_handguns 0.09 0.33 0.45 10.47 0.61 0.35 0.26 0.35 0.22   
## mean\_n\_rifle 0.01 0.03 0.19 5.85 0.06 0.02 0.03 0.03 0.02   
## mean\_n\_shotgun 0 0 1.18 1.25 0.01 0.02 0.04 0.04 0   
## mean\_n\_other 0.94 0.95 0.44 36.79 1.38 0.84 0.77 0.94 0.84   
## mean\_subject\_count 0.55 0.72 1.11 1.71 2.82 1.70 0.87 2.06 0.84   
## mean\_victim\_count 1.20 0.32 0.40 0.04 0.46 1.01 1.59 1.18 1.06   
## mean\_total\_count 1.75 1.03 1.51 1.76 3.27 2.71 2.45 3.24 1.89   
## mean\_injured\_count 1.22 0.01 0.21 0.01 0.22 0.68 0.85 0.80 0.48   
## mean\_killed\_count 0.06 0.29 0.16 0.01 0.19 0.23 0.57 0.32 0.58   
## mean\_unharmed\_arrested\_count 0.12 0.39 0.74 1.23 2.39 0.49 0.33 1.46 0.39   
## mean\_unharmed\_count 0.42 0.64 1.03 1.46 2.72 1.65 0.92 1.90 0.73   
## mean\_adult\_count 1.32 0.86 1.31 1.49 2.74 2.11 1.03 0.71 1.68   
## mean\_teen\_count 0.08 0.05 0.06 0.05 0.13 0.15 0.12 2.42 0.07   
## mean\_child\_count 0 0 0 0 0 0 1.20 0.01 0   
## mean\_relation\_family 0.01 0.01 0.04 0.01 0 0 0.20 0.03 0.06   
## mean\_relation\_random\_victims 0 0 0 0 0 0 0 0 0   
## mean\_relation\_aquaintance 0.01 0 0.01 0 0.01 0 0 0.01 0.01   
## mean\_relation\_significant\_others 0 0 0.03 0 0 0 0.02 0.01 0.15   
## mean\_relation\_armed\_robbery 0 0 0 0 0 1 0 0.01 0   
## mean\_relation\_gang 0 0 0 0 0.01 0 0 0 0   
## mean\_relation\_mass\_shooting 0 0 0 0 0 0 0 0 0   
## mean\_relation\_knows\_victims 0 0 0 0 0 0 0 0 0   
## mean\_relation\_co\_worker 0 0 0 0 0 0 0 0 0   
## mean\_relation\_neighbor 0 0 0.02 0.01 0 0 0 0 0.01   
## mean\_relation\_friends 0 0 0.01 0 0 0 0.01 0.04 0.01   
## -----------------------------------------------------------------------------------------

# save tempo to csv  
#write.csv(tempo, file = "Final\_Clustering.csv", row.names = FALSE)  
  
# # read tempo  
# tempo <- read.csv("Final\_Clustering.csv")  
  
guns\_clusters = tempo  
# replace blank with 0  
guns\_clusters[is.na(guns\_clusters)] <- 0

## Section 5 - Appendix and the lens of an analyst

## 1 cluster Centroids  
cols = list(names(guns\_clusters))  
  
View(cols)  
  
centroid\_col\_list <- c(  
 "n\_killed",  
 "n\_injured",  
 "n\_guns\_involved",  
 "stolen\_count",  
 "n\_handguns",  
 "n\_rifle",  
 "n\_shotgun",  
 "n\_other",  
 "subject\_count",  
 "victim\_count",  
 "total\_count",  
 "injured\_count",  
 "killed\_count",  
 "unharmed\_arrested\_count",  
 "unharmed\_count",  
 "adult\_count",  
 "teen\_count",  
 "child\_count",  
 "relation\_family",  
 "relation\_random\_victims",  
 "relation\_aquaintance",  
 "relation\_significant\_others",  
 "relation\_armed\_robbery",  
 "relation\_gang",  
 "relation\_mass\_shooting",  
 "relation\_knows\_victims",  
 "relation\_co\_worker",  
 "relation\_neighbor",  
 "relation\_friends",  
 "relation\_home\_invasion",  
 "relation\_does\_not\_know\_victim",  
 "female\_percentage",  
 "cluster\_kmeans")  
  
  
guns\_clusters\_subset = guns\_clusters[, centroid\_col\_list]  
  
View(guns\_clusters\_subset)  
# show column data types  
sapply(guns\_clusters\_subset, class)

## n\_killed n\_injured   
## "integer" "integer"   
## n\_guns\_involved stolen\_count   
## "integer" "integer"   
## n\_handguns n\_rifle   
## "integer" "integer"   
## n\_shotgun n\_other   
## "integer" "numeric"   
## subject\_count victim\_count   
## "integer" "integer"   
## total\_count injured\_count   
## "integer" "integer"   
## killed\_count unharmed\_arrested\_count   
## "integer" "integer"   
## unharmed\_count adult\_count   
## "integer" "integer"   
## teen\_count child\_count   
## "integer" "integer"   
## relation\_family relation\_random\_victims   
## "numeric" "numeric"   
## relation\_aquaintance relation\_significant\_others   
## "numeric" "numeric"   
## relation\_armed\_robbery relation\_gang   
## "numeric" "numeric"   
## relation\_mass\_shooting relation\_knows\_victims   
## "numeric" "numeric"   
## relation\_co\_worker relation\_neighbor   
## "numeric" "numeric"   
## relation\_friends relation\_home\_invasion   
## "numeric" "numeric"   
## relation\_does\_not\_know\_victim female\_percentage   
## "numeric" "numeric"   
## cluster\_kmeans   
## "factor"

# convert all columns to numeric  
guns\_clusters\_subset <- sapply(guns\_clusters\_subset, as.numeric)  
guns\_clusters\_subset = data.frame(guns\_clusters\_subset)  
#create median table with cluster\_kmeans in columns, and all rest variables in rows  
median\_table = aggregate(guns\_clusters\_subset, list(guns\_clusters\_subset$cluster\_kmeans), mean)  
  
#Transpose table(median\_table)  
median\_table = t(median\_table)  
View(median\_table)  
  
  
# use cluster\_kmeans as column names  
colnames(median\_table) <- median\_table[1,]  
# remove first row  
median\_table <- median\_table[-1,]  
# remove last row  
median\_table <- median\_table[-nrow(median\_table),]  
# remove killed\_count row  
median\_table <- median\_table[-which(rownames(median\_table) == "killed\_count"),]  
# Round the median\_table to 2 decimal places  
rounded\_median\_table <- round(median\_table, 2)  
  
  
  
###########  
  
library(tidyr)

##   
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':  
##   
## smiths

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ lubridate 1.9.3 ✔ tibble 3.2.1  
## ✔ readr 2.1.4

## ── 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

# convert date to date format  
guns\_f$date <- as.Date(guns\_f$date)  
# create a new column year-month column  
guns\_f$year\_month <- format(guns\_f$date, "%Y-%m")  
  
  
# Create a table of the number of incidents per cluster per year-month  
incidents\_per\_cluster\_per\_year\_month <- guns\_f %>%  
 group\_by(cluster\_kmeans, year\_month) %>%  
 summarise(n = n())

## `summarise()` has grouped output by 'cluster\_kmeans'. You can override using  
## the `.groups` argument.

print(incidents\_per\_cluster\_per\_year\_month)

## # A tibble: 499 × 3  
## # Groups: cluster\_kmeans [9]  
## cluster\_kmeans year\_month n  
## <fct> <chr> <int>  
## 1 1 2013-01 4  
## 2 1 2013-03 2  
## 3 1 2013-04 1  
## 4 1 2013-05 7  
## 5 1 2013-06 3  
## 6 1 2013-07 6  
## 7 1 2013-08 4  
## 8 1 2013-09 6  
## 9 1 2013-10 3  
## 10 1 2013-11 3  
## # ℹ 489 more rows

library(tidyverse)  
  
df\_pivoted <- incidents\_per\_cluster\_per\_year\_month %>%  
 pivot\_wider(names\_from = cluster\_kmeans, values\_from = n)  
  
# If you want to fill missing values with 0  
df\_pivoted[is.na(df\_pivoted)] <- 0  
  
# Print the pivoted data frame  
print(df\_pivoted)

## # A tibble: 63 × 10  
## year\_month `1` `2` `3` `4` `5` `6` `7` `8` `9`  
## <chr> <int> <int> <int> <int> <int> <int> <int> <int> <int>  
## 1 2013-01 4 0 1 0 0 0 1 0 1  
## 2 2013-03 2 0 1 0 1 0 0 2 1  
## 3 2013-04 1 0 1 0 1 0 2 1 1  
## 4 2013-05 7 1 0 0 0 1 1 1 2  
## 5 2013-06 3 0 0 0 1 0 0 3 1  
## 6 2013-07 6 2 0 0 0 0 0 2 2  
## 7 2013-08 4 2 1 0 0 0 1 0 3  
## 8 2013-09 6 0 0 0 0 0 1 1 2  
## 9 2013-10 3 1 1 0 0 0 0 0 1  
## 10 2013-11 3 0 0 0 0 0 0 0 2  
## # ℹ 53 more rows

df\_pivoted = data.frame(df\_pivoted)