Project proposal

Team 17

1. Introduction

The issue of mass shootings and the relationship between gun control laws and incidents of violence have sparked debates across the United States. In this study, we aim to investigate the impact of different categories of laws on the occurrence of mass shootings. Specifically, we focus on the restriction on individuals with a history of drug and alcohol abuse from purchasing firearms. By examining these factors, we contribute valuable insights to policy discussions and strategies for mitigating mass shootings, ultimately enhancing public safety. However, conducting comprehensive analyses on mass shootings is challenging due to the limited number of observations and the rarity of these events. Additionally, mass shootings are influenced by multiple factors, which can complicate establishing definitive relationships with laws. Comparing the effectiveness of gun control laws across states is also challenging due to changing legislation, interpretation, and enforcement. These complexities affect our understanding of the impact of laws in preventing mass shootings. While prior research suggests that stricter gun control measures can potentially reduce gun violence, the specific impact of legislation on restricting firearm access for individuals with a history of substance abuse and alcohol issues remains understudied. Therefore, our investigation into this topic fills a gap in the existing literature. For our analysis, we will use a Poisson model to examine the general question regarding the impact of different laws on the number of mass shootings. Additionally, we will employ an ARIMAX model for our specific question to predict and compare the actual and predicted values, providing insights into the effectiveness of the laws in question. In summary, this study aims to shed light on the relationship between laws and mass shootings, contribute to policy discussions, and advance our understanding of effective measures to prevent and mitigate such incidents.

2. Data overview

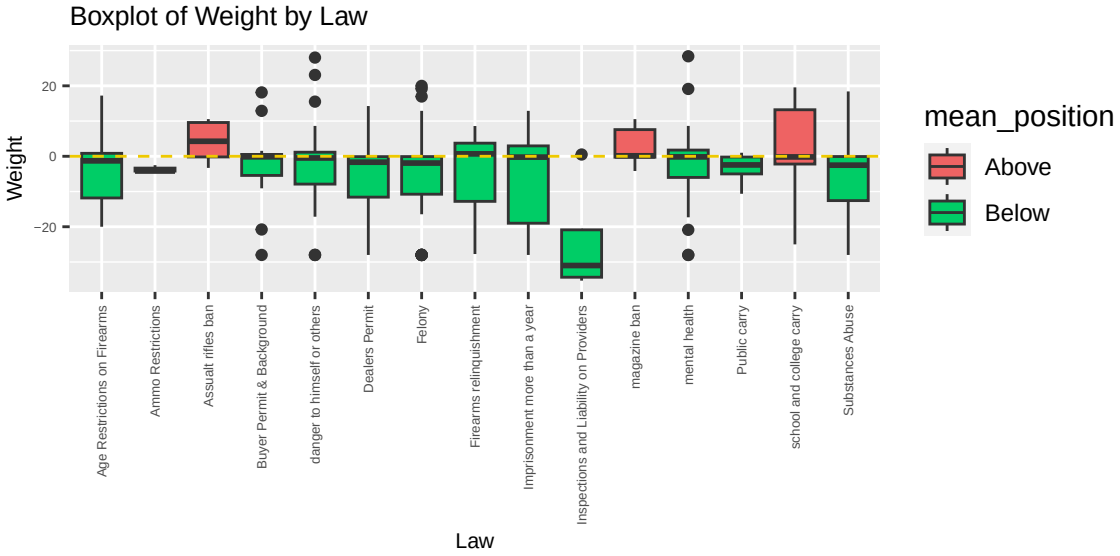
In our research we used two datasets that provides information regarding mass shootings in the USA and laws implemented in each state of the USA in different years. In the Mass Shootings dataset: • Year: The specific year in which the shooting incident took place. (ranging from 1966 - 2020) • State: The state within the USA where the shooting occurred, providing geographical context. • Mass Shooter Info: Details about the individuals involved in the shootings,like age, gender, motive, psychological state, substance abuse. • Number of Injured and Killed - Number of individuals who were injured and killed in each mass shooting event.
In the Laws dataset: • Year: The specific state within the USA where each law was implemented. (ranging from 1991 - 2020) • State: The state within the USA where the laws are. • Laws: Details about the specific laws that were enacted • Law Catagory: Grouping of similar laws based on common themes or objectives. These datasets collectively provide a comprehensive understanding of mass shootings and the associated gun control laws in the USA. By examining the entities and features within these datasets, we can explore the relationships, patterns, and potential influences between mass shootings and the implementation of specific laws, contributing to a deeper understanding of the problem at hand.

3. Methods and results

A. General Question - Understand Law Categories and Predicting Mass Shootings:
We chose to use Poisson regression for our goal because it allows us to predict the amount of mass shootings

not only based on laws that are in place but also considering the passage of time. We have found that most of the laws which the model was built on them, are lowering the number of mass shootings – especially laws that limit people who has prior felonies, dealing with mental health problems and laws of acquiring firearms. We have used this model on 50 states in the USA, for each state we looked at the number of mass shootings that occurred in the state for each year between 1991-2018. Then we used the year 2019 as a test and tried to predict the number of mass shootings in each state this year.In the following plot we can see each prediction:

The X-axis represent the law categories and the Y-axis represents the weights assigned to each law. The weights indicate the impact or importance of each law in relation to the problem being studied. Higher weight suggest that the law has a stronger influence or significance, while lower weight indicate a lesser impact. The plot provides an overview of how different laws contribute to the issue being studied and helps in understanding the relative importance of each law based on its weight.



In addition we have preformed t-test on the averages of each coefficient of each law: Our null hypothesis: true mean is 0 – gun control laws don't affect the amount of mass shootings. Alternative hypothesis: true mean is less than 0 – gun control laws lower the amount of mass shootings.

```
##  One Sample t-test
## t = -3.3907, df = 14, p-value = 0.002197
```

Based on our results we reject the null hypothesis in favor of the alternative hypothesis meaning gun control laws lower the amount of mass shootings.

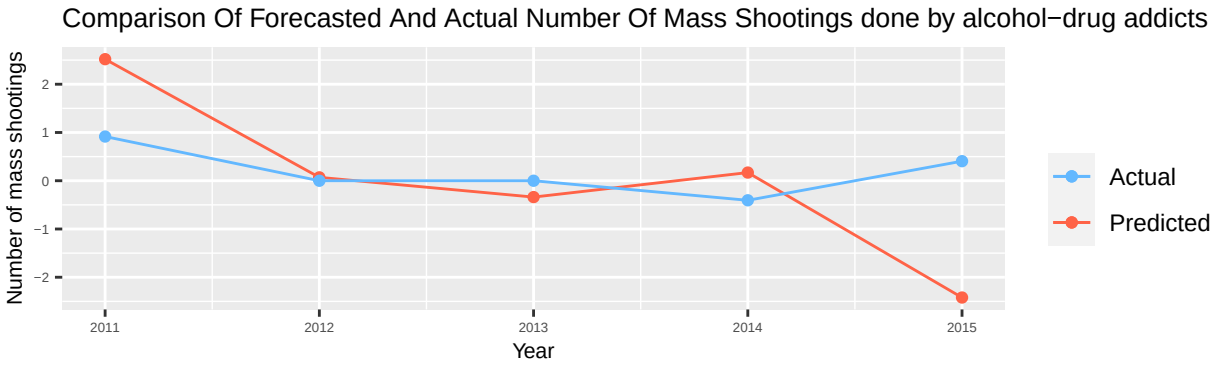
When predicting the number of mass shootings for 2019 for each country in the USA using the weights assigned to each category, we evaluated the performance using a confusion matrix. These metrics provide insights into the model's performance in predicting the number of mass shootings for 2019.

```
## [1] "precision: 0.5 recall: 0.25 accuracy: 0.849056603773585 f1: 0.333333333333333"
```

B. Specific Question: Investigating the Effectiveness of Drugs and Alcohol Laws in Reducing Mass Shootings Committed by Individuals Substance Abuse History

First, we used ARIMAX model. The model was selected based on the assumption that the number of mass shootings exhibits temporal patterns and could be influenced by the laws related to substances abuse. The ARIMAX model is suitable for analyzing time series data and for capturing temporal dependencies. Secondly, to ensure the suitability of the data for the model, we conducted Augmented Dickey-Fuller (ADF) test that checks for the presence of unit roots, which indicate non-stationarity.

By ensuring that the data is stationary we can improve the model's accuracy and reliability. Forecasting and Evaluation - The ARIMAX model was used to forecast the number of mass shootings committed by people with substance abuse and alcohol problems for the years 2011-2015. These forecasted values were compared to the actual values from the test set to assess the model's accuracy.



We used MSE and RMSE to evaluate the model's ability to accurately predict the number of mass shootings based on the laws related to substances abuse. The results indicate that the model has a relatively high MSE and RMSE, suggesting a moderate level of prediction error.

```
## Mean Squared Error (MSE): 2.200689    Root Mean Squared Error (RMSE): 1.483472
## Mean Absolute Error (MAE): 1.082485   R-squared: 0.1091401
```

In addition, we conducted a t-test to determine whether the coefficient of the 'laws' variable obtained from the ARIMA model is statistically different from zero. Based on the results of the t-test, we cannzot reject the null hypothesis that the coefficient of the 'laws' variable is statistically different from zero. This implies that we do not have sufficient evidence to conclude that the 'laws' variable has either a positive or a negative effect on the outcome.

```
## Coefficient estimate: 0.17109   t statistic: 2.70353   P-value: 0.0539
## The coefficient is not statistically significant.
```

4. Limitations and Future Work

The limitations of our approach include a small sample size, which may not adequately represent the overall population and is prone to selection bias if certain incidents or laws are overrepresented or underrepresented. Additionally, our research focuses on a specific region, which limits the generalizability of our findings to locations with different socio-cultural, political, or legal contexts. As mass shootings are complex events influenced by various factors, we acknowledge the need for additional datasets to broaden the scope of our analysis. Obtaining more comprehensive data would enhance the generalizability of our findings and provide a deeper understanding of the topic. Furthermore, exploring other potential variables that impact the relationship between gun control laws and mass shootings would contribute to a more comprehensive analysis. In order to validate our findings, we plan to compare them with studies conducted in different countries or regions with varying levels of gun control legislation. This comparative analysis would shed light on the effectiveness of different approaches to gun control and their impact on mass shootings. During our research, we encountered challenges in adapting the ARIMAX model to our multivariate data. The nature of our data, including multiple variables and collinearity issues, posed obstacles in achieving optimal results. Therefore, with additional time, we will aim to invest efforts in expanding our knowledge and expertise in advanced predictive modeling techniques that are specifically designed for analyzing multivariate data.

Links

Repository, Mass Shootings Database, Laws Database

Source code

```
suppressWarnings(library(knitr))
suppressWarnings(library(tidyverse))
suppressWarnings(library(broom))
suppressWarnings(library(htmltools))
chooseCRANmirror(graphics=FALSE, ind=40) # set the mirror index to your preferred value
#tinytex::reinstall_tinytex()
opts_chunk$set(echo=FALSE) # hide source code in the document


install.packages("tscount")
library(tscount)

library("readxl")
library(tidyverse)
library(ggplot2)
library(forecast)
library(tseries)
library(zoo)
library(dplyr)

mass_shooting <- read_excel("Mass Shooting.xlsx", sheet="Full Database")
mass_shooting$State <- ifelse(mass_shooting$State == "DC", "WA", mass_shooting$State)
mass_shooting$State <- state.name[match(mass_shooting$State, state.abb)]
laws <- read_excel("Laws.xlsx", sheet="fulldataset")


laws_substances  <- laws %>%
  dplyr::select(`state`, `year`,`drugmisdemeanor`,`alctreatment`, `alcoholism`) %>%
  rename(`Year` = `year`) %>%
  rename(`State` = `state`)


ms_substance  <- mass_shooting %>%
  filter(!is.na(`Shooter First Name`)) %>%
  dplyr::select("Case #", "Year", "State", "Known to Police or FBI", "Criminal Record", "Part II Crimes

mss_new_cols <- ms_substance  %>% #table with a column that says if there is substance abuse
  rename(`Substance Crime` = `Part II Crimes`)%>%
 mutate(`Substance Crime` = ifelse(grepl("7|8", `Substance Crime`), 1, 0))%>%
mutate(`Substance Use` = ifelse(`Substance Use` == 0, 0, 1))%>%
    mutate(Substances  = ifelse(`Substance Crime` == 1 | `Substance Use` == 1, 1, 0))


mssc_years <- mss_new_cols %>% # table with relevant years
  filter(Year > 1990 & Year < 2021)

mssy_sum <- mssc_years %>% mutate(
    `Number Killed` = as.numeric(`Number Killed`),
    `Number Injured` = as.numeric(`Number Injured`)) %>%
  group_by(Year)%>%
  summarise(State, Substances,
```

```
    num_vic = sum(`Number Killed`),
    num_inj = sum(`Number Injured`+ `Number Killed`))

#table that says if there is a substances laws in a country

ls_sum <- laws_substances %>%
  group_by(Year, State)%>%
  summarise(
    laws = ifelse(sum(`drugmisdemeanor` + `alctreatment` + `alcoholism`)>0,1,0))

# combines the table by year and state
ms <- inner_join(ls_sum, mssy_sum, by = c("Year", "State"))

total_ms <- ms %>%
  group_by(Year) %>%
  summarise( total_ms = n(), Substances =sum(Substances)) %>%
  mutate(ms_sober = abs(total_ms-Substances)) %>%
  rename(ms_ui = Substances)

ms_by_sub <- ms %>%
  group_by(Year, Substances)%>%
  summarise( total_ms = n(), Substances =sum(Substances))%>%
  mutate(Substances = ifelse(Substances==0,0,1))

ms_by_state <- ms %>%
  group_by(State) %>%
  summarise(total_ms = n(), Substances = sum(Substances)) %>%
  mutate(ms_sober = abs(total_ms-Substances))%>%
  rename(ms_ui = Substances)


ms_by_year <- ms %>%
  group_by(Year) %>%
  summarise(laws = sum(laws))

total <- mssc_years %>%
  group_by(Year,State) %>%
  summarise(total_ms = n())

limited_laws <- laws %>%
  select(state, year, dealer, dealerh, immunity,inspection, liability, statechecks, statechecksh,   uni

cats <- limited_laws %>%
  group_by(year, state)%>%
  summarise(
   `Dealers Permit` = dealerh,
   `Inspections and Liability on Providers`  = ifelse((sum(`immunity` + `inspection`+`liability`))/3>0.
   `Buyer Permit & Background` = ifelse(sum(`statechecks`+  `statechecksh`+ `universal` +`universalh`
   `Age Restrictions on Firearms` = ifelse(sum(`age21handgunsale`+  `age18longgunsale`+ `age21longgunsa
   `Ammo Restrictions` = ifelse(sum(`ammbackground`+    `ammpermit`+    `ammrestrict`)/3>0.6,1,0),
  `Felony` = felony,
   `Imprisonment more than a year` = violentpartial,
  `danger to himself or others` = danger,
```

```
   `mental health` = ifelse(sum(`mentalhealth`+ `invcommitment`+    `invoutpatient`)/3>0.6,1,0),
       `Substances Abuse` = ifelse(sum(`drugmisdemeanor` + `alctreatment` + `alcoholism`)>0,1,0),
   `Public carry` = ifelse(sum(`opencarryh`+`opencarryl`+`opencarrypermith`+`opencarrypermitl`)/4 > 0.4
   `school and college carry` = ifelse(sum(`college`+   `collegeconcealed`+ `elementary`)/3 > 0.65 ,1,0
   `Firearms relinquishment` = relinquishment,
  `Assualt rifles ban` = assault,
  `magazine ban` = ifelse(sum(`magazine`+   `tenroundlimit`+    `magazinepreowned`)/3 > 0.65 ,1,0)
   )

sum_cats <- cats %>%
  group_by(year, state) %>%
  summarise(`Dealers Permit` = sum(`Dealers Permit`),
`Inspections and Liability on Providers` = sum(`Inspections and Liability on Providers`), `Buyer Permit
  left_join(total,by=c("year" = "Year"))
ms_cats <-cats %>% left_join(total, by= c("state" = "State", "year" = "Year"))
ms_cats <- ms_cats %>% mutate(total_ms = ifelse(is.na(total_ms), 0, total_ms))
states <- as.vector(unique(ms_cats$state))
df <- data.frame(
  "Dealers Permit" = c("NULL"),
  Variable = c("Dealers Permit", "Inspections and Liability on Providers", "Buyer Permit & Background",
rlst <- c()
i <- TRUE
col_names <- c("2019")
df2 <- data.frame(matrix(ncol = length(col_names), nrow = 0))
colnames(df2) <- col_names
for (st in states){
  stdf <- ms_cats %>% filter(state == st)
  stdf <- stdf[,-c(1,2)]
  train <- stdf[1:28,]
  test<- stdf[29:30,]
  ts_train <- ts(train$total_ms, start = c(1991, 1), frequency = 1)
  ts_test<-ts(test$total_ms, start = c(2019, 1), frequency = 1)
  #total_ms_ts
  # Fit a time series count model with additional regressors
  train_mat <- as.matrix(train[,-c(16)])
  test_mat <- as.matrix(test[,-c(16)])
  model <- tsglm(ts_train, xreg = train_mat, link = c("log"), distr = c("poisson"))
  pred <- predict(model, n.ahead=2, newxreg=test_mat)
  vpred <- t(as.data.frame(pred$pred[1]))
  vpred <- cbind(st,vpred)
  df2 <- rbind(df2,vpred)
  n_obs <- length(ts_train)
  n_pred <- length(model$coefficients) - 1  # Subtract 1 for the intercept

  # Calculate the RSS
  residuals <- residuals(model)
  rss <- sum(residuals^2)

  # Calculate the TSS
  tss <- sum((ts_train - mean(ts_train))^2)

  # Calculate the R-squared
  r_squared <- 1 - rss/tss
```

```
  rlst <- c(rlst, r_squared)
  df <- t(as.data.frame(coef(model)))
  df[1,1] <- st
  if (i){
    union_df <- df
  }
  else{
    union_df <- rbind(union_df, df)
  }
  i <- FALSE
}

coeffs <- as.data.frame(union_df)
df22 <- df2 %>% mutate(pred = as.numeric(df2$V2)) %>% select(st, pred)
predf <- data.frame(df22$st, df22$pred)
ms_filtered <- ms_cats %>% filter(year == 2019) %>% select(year, state, total_ms)
ms_texas <- ms_cats %>% filter(state == "Texas")
predf <- data.frame(state = df22$st, pred = df22$pred, real = ms_filtered$total_ms)


# Draw Graph
# ggplot(predf, aes(x = state, y = real, group = 1)) +
#   geom_line(color = "blue", size = 1) +
#   geom_point(aes(y = pred), color = "red", size = 3) +
#   labs(x = "States", y = "Number of Shootings", title = "Actual vs Predicted Shootings by State in 20
#   theme_minimal() +
#   theme(axis.text.x = element_text(angle = 90, hjust = 1))
colnames(coeffs)[1] = "State"

all_states = unique(coeffs$State)
all_laws = colnames(coeffs)[2:length(colnames(coeffs))]

box_df <- data.frame(State = character(),
                     Law = character(),
                     Weight = numeric(),
                     stringsAsFactors = FALSE)

for(s in seq(length(all_states))){

  for(l in seq(length(all_laws))){
    val = coeffs %>% filter(State == all_states[s]) %>% select(State, all_laws[[l]]) %>%  pull()
    #del
    if(val != 0){
      if(all_states[s] != "Virginia" || all_laws[l] != "mental health"){
        new_row =  data.frame(State = all_states[s], Law = all_laws[l], Weight = val)
        box_df <- rbind(box_df, new_row)
      }
    }
  }
}

box_df$Weight = as.numeric(box_df$Weight)
```

```r
mean_df = box_df %>%
  group_by(State) %>%
  summarise(mean_weight = mean(as.numeric(Weight)), .groups = "drop")
merged_df = merge(box_df, mean_df, by = "State")

merged_df$mean_position = ifelse(merged_df$mean_weight > 0, "Above", "Below")

count_df <- merged_df %>%
  group_by(mean_position) %>%
  summarise(count = n())

box_cat_df = box_df %>% arrange(Law)

law_mean = box_cat_df %>%
  group_by(Law) %>%
  summarise(mean_weight = mean(as.numeric(Weight)), .groups = "drop")

merged_cat_df = merge(box_cat_df, law_mean, by = "Law")
merged_cat_df$mean_position = ifelse(merged_cat_df$mean_weight > 0, "Above", "Below")

my_ggp <- ggplot(merged_cat_df, aes(x = Law, y = Weight)) +
  geom_boxplot(aes(fill = mean_position)) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gold2") +
  labs(x = "Law", y = "Weight") +
  ggtitle("Boxplot of Weight by Law") +
  scale_fill_manual(values = c("Above" = "indianred2", "Below" = "springgreen3")) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
    theme(axis.text = element_text(size = 5),
        plot.title = element_text(size = 10),
        axis.title.x = element_text(size = 8),
        axis.title.y = element_text(size = 8))

my_ggp
avgs <- c()
cols <- names(coeffs)
cols <- cols[-1]
for (col in cols){
  avg <- as.numeric(coeffs[[col]])
  avgs <- c(avgs,mean(avg))
}
df <- data.frame(law = cols, avg = avgs)

alpha <- 0.05
t_test_results <- t.test(df$avg, alternative = "less", mu = 0)
p_values <- t_test_results$p.value
#t_test_results
output <- capture.output(t_test_results)

lines <- strsplit(output, "\n")

for (line in seq(length(lines))) {
  if(line %in% c(2,5)){
    cat(lines[[line]], "\n")
```

```
  }
}

tp <- 2
fp <- 2
fn <- 6
tn <- 43

ct <- data.frame(p=c(tp,fn), n=c(fp,tn))
precision <- tp/(tp+fp)
recall <- tp/(tp+fn)
accuracy <- (tp+tn)/(tp+tn+fn+fp)
f1 <- (2*precision*recall)/(recall+precision)
output <- paste("precision:", precision, "recall:", recall, "accuracy:", accuracy, "f1:", f1, sep = " ")

print(output)


laws_substances  <- laws %>%
  dplyr::select(`state`, `year`,`drugmisdemeanor`,`alctreatment`, `alcoholism`) %>%
  rename(`Year` = `year`) %>%
  rename(`State` = `state`)


ms_substance  <- mass_shooting %>%
  filter(!is.na(`Shooter First Name`)) %>%
  dplyr::select("Case #", "Year", "State", "Known to Police or FBI", "Criminal Record", "Part II Crimes"

mss_new_cols <- ms_substance  %>% #table with a column that says if there is substance abuse
  rename(`Substance Crime` = `Part II Crimes`)%>%
 mutate(`Substance Crime` = ifelse(grepl("7|8", `Substance Crime`), 1, 0))%>%
mutate(`Substance Use` = ifelse(`Substance Use` == 0, 0, 1))%>%
    mutate(Substances  = ifelse(`Substance Crime` == 1 | `Substance Use` == 1, 1, 0))


mssc_years <- mss_new_cols %>% # table with relevant years
  filter(Year > 1990 & Year < 2021)

mssy_sum <- mssc_years %>% mutate(
    `Number Killed` = as.numeric(`Number Killed`),
    `Number Injured` = as.numeric(`Number Injured`)) %>%
  group_by(Year)%>%
  summarise(State, Substances,
    num_vic = sum(`Number Killed`),
    num_inj = sum(`Number Injured`+ `Number Killed`))

#table that says if there is a substances laws in a country

ls_sum <- laws_substances %>%
  group_by(Year, State)%>%
  summarise(
    laws = ifelse(sum(`drugmisdemeanor` + `alctreatment` + `alcoholism`)>0,1,0))
```

```r
# combines the table by year and state
ms <- inner_join(ls_sum, mssy_sum, by = c("Year", "State"))

total_ms <- ms %>%
  group_by(Year) %>%
  summarise( total_ms = n(), Substances =sum(Substances)) %>%
  mutate(ms_sober = abs(total_ms-Substances)) %>%
  rename(ms_ui = Substances)

ms_by_sub <- ms %>%
  group_by(Year,State, Substances)%>%
  summarise( total_ms = n(), Substances =sum(Substances))%>%
  mutate(Substances = ifelse(Substances==0,0,1))

ms_by_state <- ms %>%
  group_by(State) %>%
  summarise(total_ms = n(), Substances = sum(Substances)) %>%
  mutate(ms_sober = abs(total_ms-Substances))%>%
  rename(ms_ui = Substances)

ms_by_year <- ms %>%
  group_by(Year) %>%
  summarise(laws = sum(laws))

msf <- left_join(ms_by_year, total_ms, by = c("Year")) %>%
  mutate(ms_ui = ms_ui/total_ms) %>%
   dplyr::select(Year, laws, ms_ui)



s_year <- 1991          # start year
e_year <- 2020          # end year

laws_list <- colnames(msf)[2]
stationary_frames <- list()
i <- 1

for (law in laws_list) {
  temp_df <- msf %>%
  select(all_of(law))
  ts_data <- ts(data = temp_df, start = 1990, end = e_year, frequency = 1)
  adf_result <- adf.test(ts_data)
  p_value <- adf_result$p.value

  if (p_value <= 0.05) {
    #print(adf.test(ts_data))
    stationary_frames[[i]] <- ts_data
  } else {
    station <- diff(log(ts_data + 0.0001))
    adf_result <- adf.test(station)
    p_value <- adf_result$p.value

    while (p_value > 0.05) {
```

```
      station <- diff(station)
      adf_result <- adf.test(station)
      p_value <- adf_result$p.value
    }
    stationary_frames[[i]] <- station
    #print(adf.test(ts_data))
  }
}


laws_list <- colnames(msf)[3]
i <- 2

for (law in laws_list) {
  temp_df <- msf %>%
  select(all_of(law))
  ts_data <- ts(data = temp_df, start = 1990, end = e_year, frequency = 1)
  adf_result <- adf.test(ts_data)
  p_value <- adf_result$p.value

  if (p_value <= 0.05) {
    #print(adf.test(ts_data))
    stationary_frames[[i]] <- ts_data
  } else {
    station <- diff(log(ts_data + 0.0001))
    adf_result <- adf.test(station)
    p_value <- adf_result$p.value

    while (p_value > 0.05) {
      station <- diff(station)
      adf_result <- adf.test(station)
      p_value <- adf_result$p.value
    }
    stationary_frames[[i]] <- station
    #print(adf.test(ts_data))

  }

}

# Convert the list of time series to a data frame
stationary_df <- as.data.frame(do.call(cbind, stationary_frames))

# Fill NAs using na.locf
stationary_df <- na.locf(stationary_df, na.rm = FALSE)

# Convert the data frame to a time series
stationary_df <- ts(stationary_df, start = s_year, end = e_year, frequency = 1)

# Set column names
colnames(stationary_df) <- c(colnames(msf)[2], colnames(msf)[3])

cutoff <- 21
```

```r
t_years_range <- c(1991, 1991 + cutoff - 1)  # the years range of the testing df
train <- ts(as.matrix(stationary_df), start = t_years_range[1], end = 2010, frequency = 1)
test <- ts(as.matrix(stationary_df), start = 2011, end = 2015, frequency = 1)

# Your code to fit the ARIMA model and generate the forecast
fit <- auto.arima(train[,"ms_ui"], xreg = train[,"laws"])

# Generate the forecast and assign it to the forecast_values1 variable
forecast_values1 <- forecast(fit, xreg = test[,"laws"])
fv <- as.data.frame(forecast_values1)
fv <- fv[-c(2,3,4,5,6)]
fv <- fv$`Point Forecast`

year <- c(2011:2015)
fvd <-  data.frame(year,fv)


tst1<- as.data.frame(test)
tst1 <- tst1[-c(1)]
tst1 <- c(tst1)
tst <-  data.frame(year,tst1)

library(ggplot2)

# Your code to create the fvd and tst data frames

# Create the plot with dots and lines
final <- ggplot() +
  geom_line(data = fvd, aes(x = year, y = fv, color = "Predicted")) +
  geom_point(data = fvd, aes(x = year, y = fv, color = "Predicted")) +
  geom_line(data = tst, aes(x = year, y = ms_ui, color = "Actual")) +
  geom_point(data = tst, aes(x = year, y = ms_ui, color = "Actual")) +
  xlab("Year") +
  ylab("Number of mass shootings") +
  ggtitle("Comparison Of Forecasted And Actual Number Of Mass Shootings done by alcohol-drug addicts") +
  scale_color_manual(values = c("Predicted" = "tomato1", "Actual" = "steelblue1"),
                     labels = c("Actual","Predicted")) + labs(color = "") +
  theme(axis.text = element_text(size = 5),
        plot.title = element_text(size = 10),
        axis.title.x = element_text(size = 8),
        axis.title.y = element_text(size = 8))

final

# Assuming you have the real values and predicted values as lists
real_values <- fv
predicted_values <- c(tst[-c(1)])


# Convert lists to numeric vectors
real_values <- unlist(real_values)
predicted_values <- unlist(predicted_values)
```

```r
# Calculate mean squared error (MSE)
mse <- mean((real_values - predicted_values)^2)

# Calculate root mean squared error (RMSE)
rmse <- sqrt(mse)

# Calculate mean absolute error (MAE)
mae <- mean(abs(real_values - predicted_values))

# Calculate R-squared
ss_total <- sum((real_values - mean(real_values))^2)
ss_residual <- sum((real_values - predicted_values)^2)
r_squared <- 1 - (ss_residual / ss_total)

# Print the evaluation metrics
cat("Mean Squared Error (MSE):", mse,
    "   Root Mean Squared Error (RMSE):", rmse,
    "\nMean Absolute Error (MAE):", mae,
    "  R-squared:", r_squared, "\n")

fit2 <- auto.arima(stationary_df[,"ms_ui"], xreg = stationary_df[,"laws"])

coe <- coef(fit2)[[2]]
coef_stderr <- sqrt(vcov(fit2)[4])


t_value <- coe / coef_stderr
df <- 4
p_value <- 2 * (1 - pt(abs(t_value), df))

alpha <- 0.05

if (p_value < alpha) {
  ans = "\nThe coefficient is statistically significant."
} else {
  ans = "\nThe coefficient is not statistically significant."
}

cat("Coefficient estimate:", round(coe,5),
    "  t statistic:", round(t_value,5),
    "  P-value:", round(p_value,5),
    ans)
```