TS_crime_Sarima+Intervention

Haoyu Zhang

2024-05-22

```
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(Metrics)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

##
## Attaching package: 'forecast'

## The following object is masked from 'package:Metrics':
##
##     accuracy
```

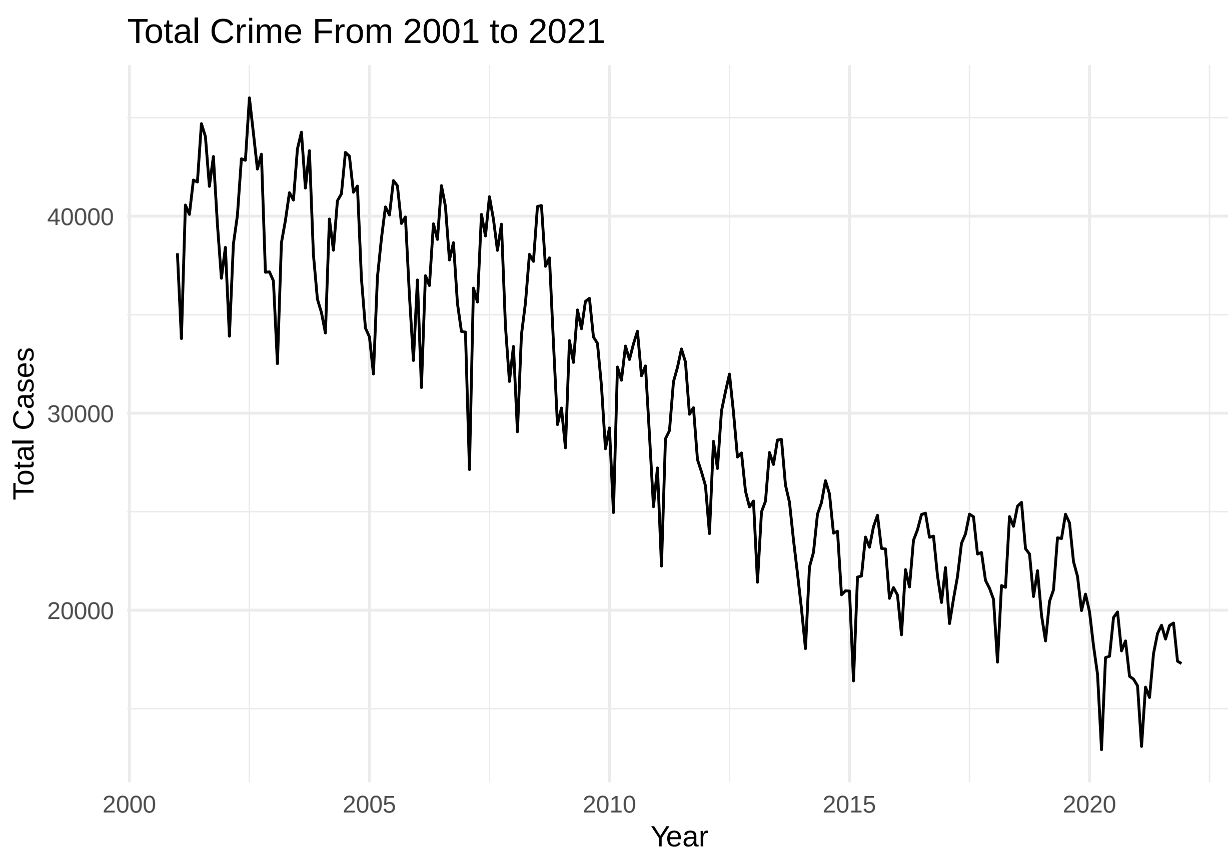
```
library(ggplot2)
library(TSA)
```

```
## Registered S3 methods overwritten by 'TSA':
##   method       from
##   fitted.Arima forecast
##   plot.Arima   forecast

##
## Attaching package: 'TSA'

## The following objects are masked from 'package:stats':
##
##     acf, arima
```

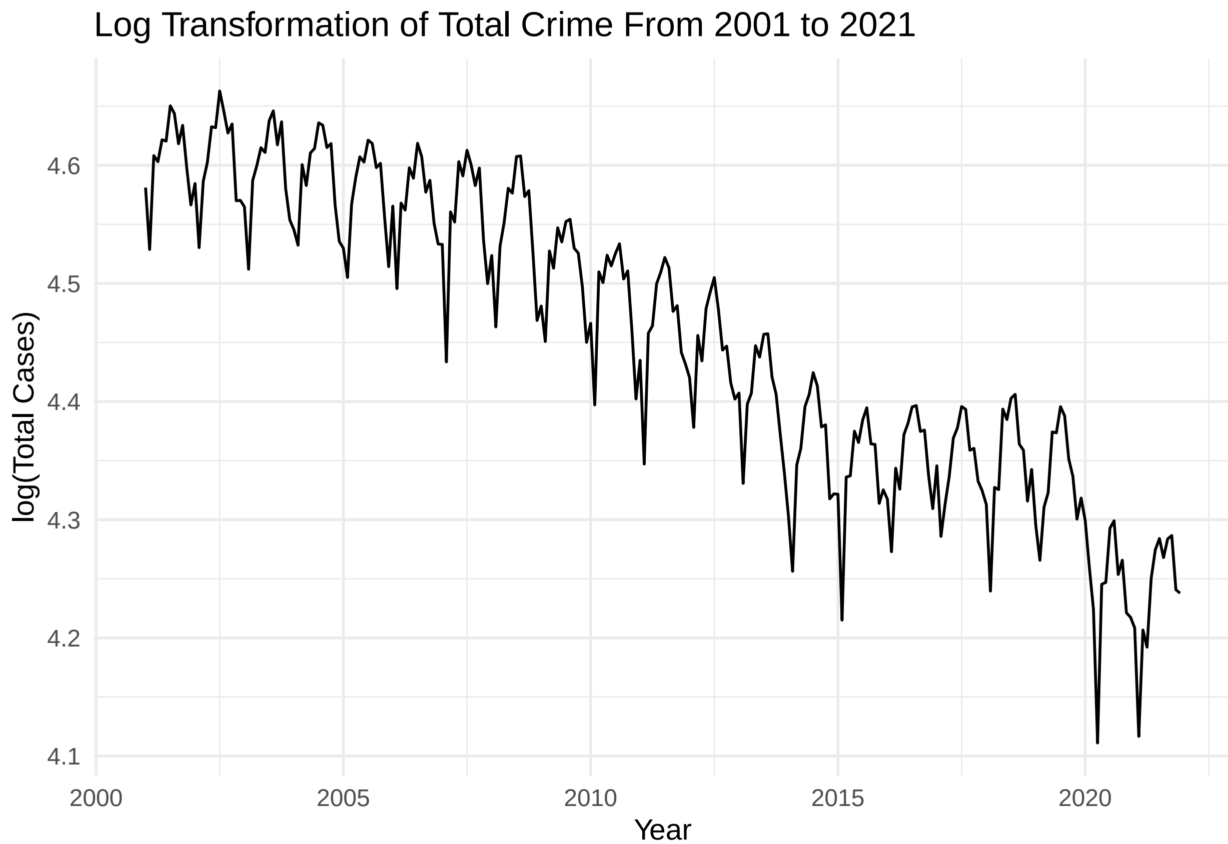
```
## The following object is masked from 'package:utils':
##
##     tar
```

```r
crime_data <- read.csv('crime_data_type.csv')
crime_data <- crime_data[,-1]
crime_data$Date <- as.yearmon(crime_data$Date, "%Y-%m")

# Convert the entire dataset into a time series object
crime_ts <- ts(crime_data[, 2], start = c(2001, 1), frequency = 12)
train_ts <- window(crime_ts, start = c(2001, 1), end = c(2021, 12))
test_ts <- window(crime_ts, start = c(2022, 1), end = c(2022, 12))
```

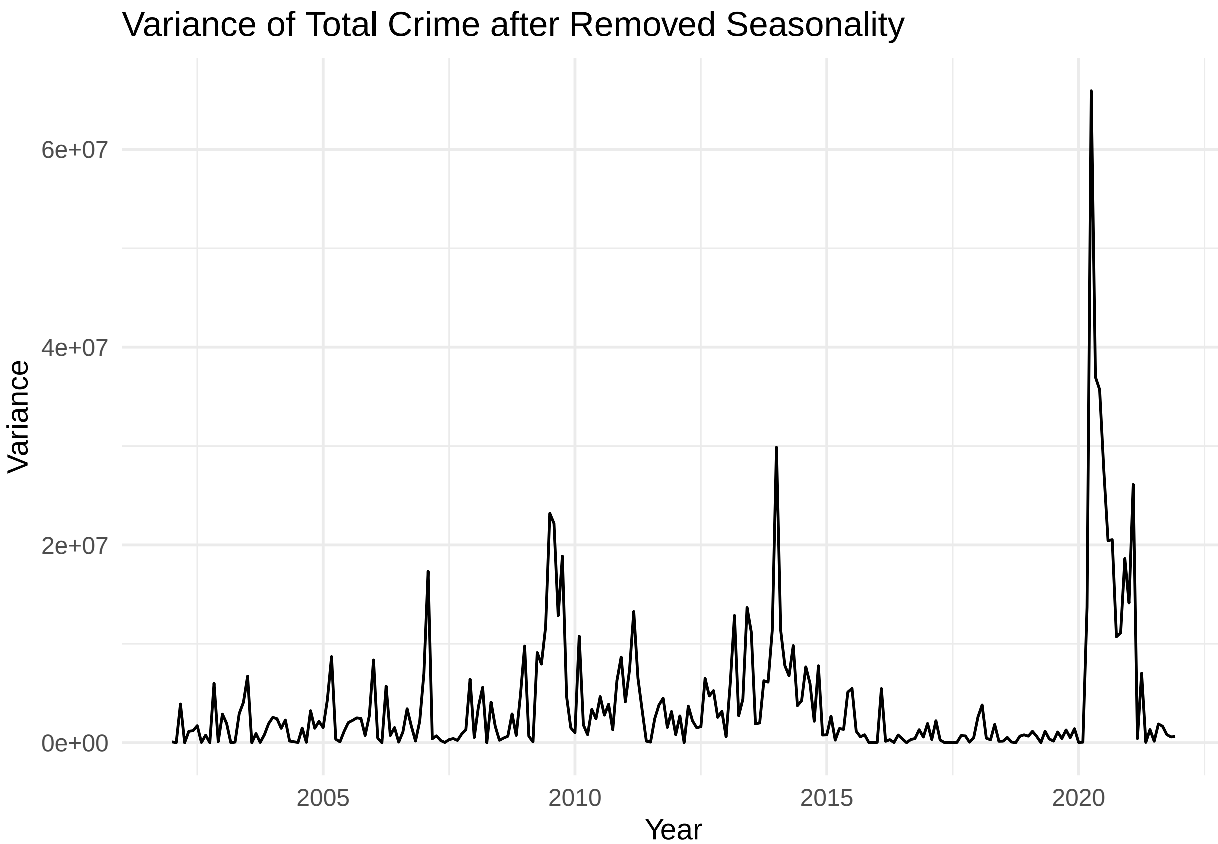
```r
# Plot the forecasts and include training data
autoplot(train_ts) +
  ggtitle("Total Crime From 2001 to 2021") +
  xlab("Year") + ylab("Total Cases") +
  theme_minimal() +
  guides(colour=guide_legend(title="Series"))
```



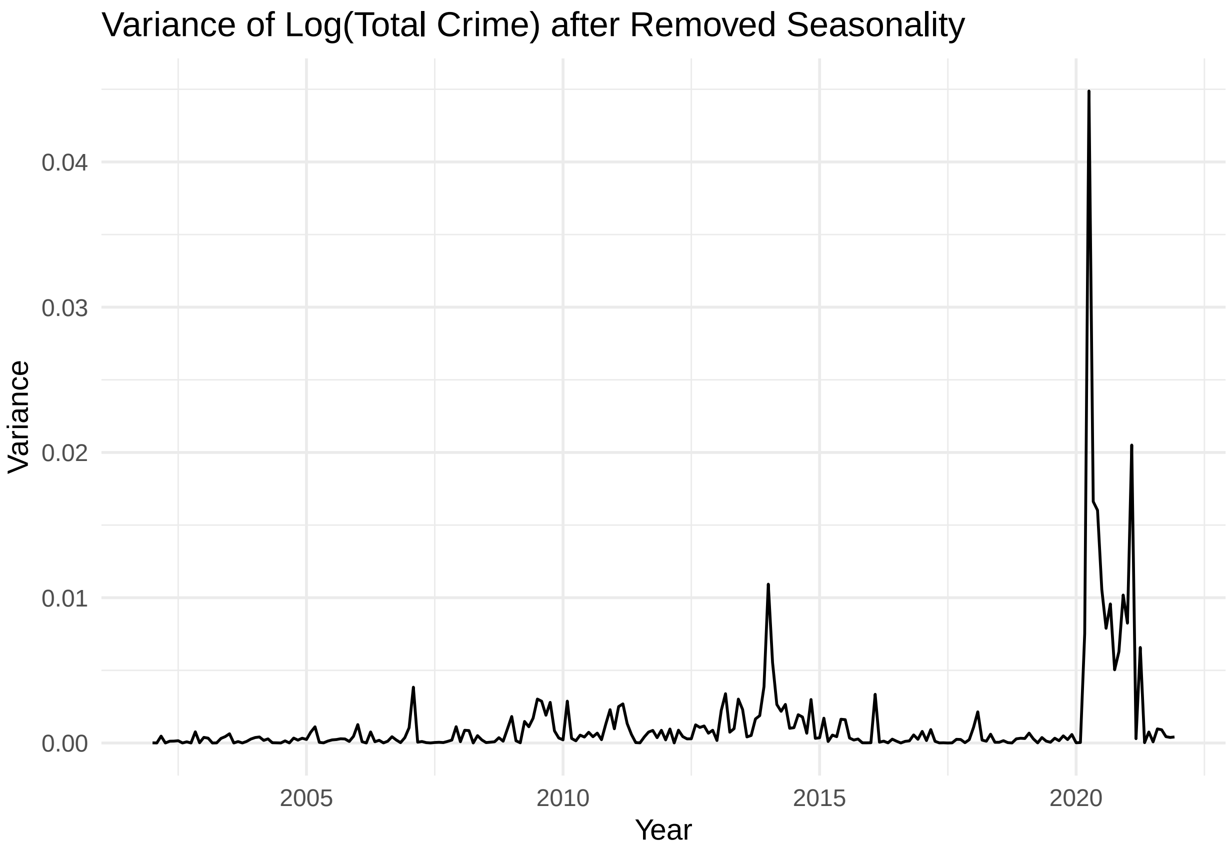

```r
# Plot the forecasts and include training data
autoplot(log10(train_ts)) +
  ggtitle("Log Transformation of Total Crime From 2001 to 2021") +
  xlab("Year") + ylab("log(Total Cases)") +
  theme_minimal() +
  guides(colour=guide_legend(title="Series"))
```




```r
# Plot the forecasts and include training data
autoplot(diff(train_ts, 12)^2) +
  ggtitle("Variance of Total Crime after Removed Seasonality") +
  xlab("Year") + ylab("Variance") +
  theme_minimal() +
  guides(colour=guide_legend(title="Series"))
```



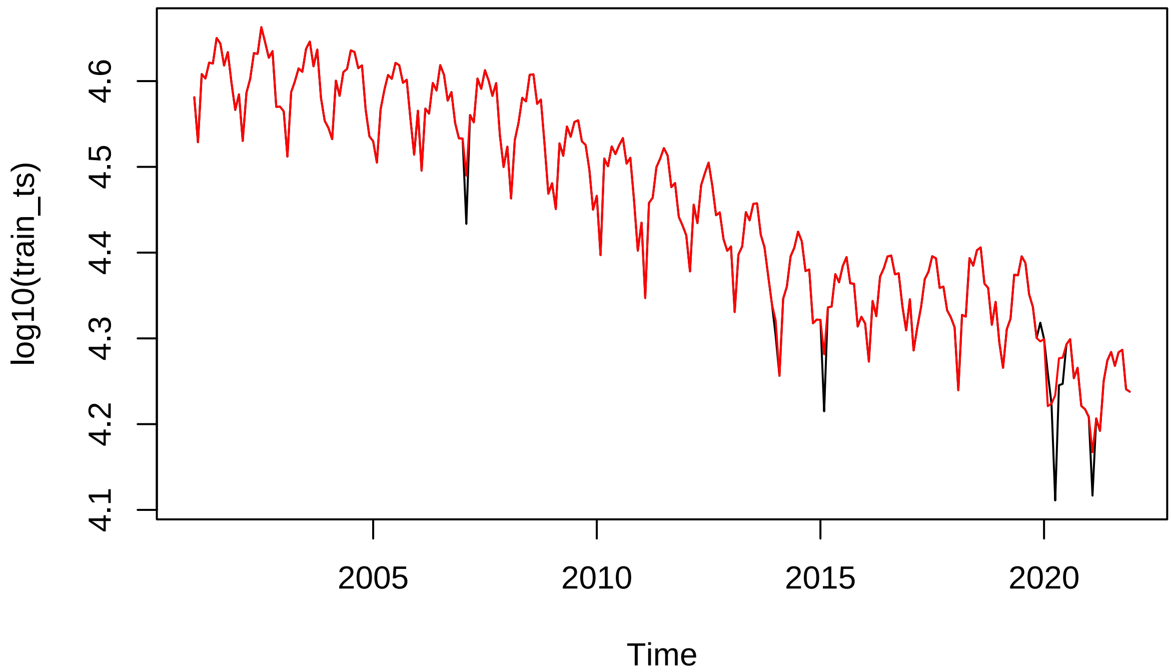

```r
# Plot the forecasts and include training data
autoplot(diff(log10(train_ts), 12)^2) +
  ggtitle("Variance of Log(Total Crime) after Removed Seasonality") +
  xlab("Year") + ylab("Variance") +
  theme_minimal() +
  guides(colour=guide_legend(title="Series"))
```




```
ts.plot(log10(train_ts))
tsoutliers(log10(train_ts))
```

```
## $index
## [1]  74 157 170 228 230 232 233 234 242
##
## $replacements
## [1] 4.489766 4.320709 4.281407 4.296437 4.220917 4.233730 4.276812 4.277601
## [9] 4.167252
```

```
y_clean <- tsclean(log10(train_ts))
lines(y_clean, col = 'red')
```




```
# univariate TS
# SARIMA
model_arima <- auto.arima(log10(train_ts))
summary(model_arima)
```

```
## Series: log10(train_ts)
## ARIMA(1,0,2)(0,1,1)[12] with drift
##
## Coefficients:
##          ar1      ma1      ma2     sma1    drift
##       0.9511  -0.3722  -0.1653  -0.7614  -0.0015
## s.e.  0.0270   0.0697   0.0669   0.0501   0.0003
##
## sigma^2 = 0.0003729:  log likelihood = 604.35
## AIC=-1196.71   AICc=-1196.35   BIC=-1175.83
##
## Training set error measures:
##                        ME      RMSE       MAE         MPE      MAPE      MASE
## Training set 0.0004590722 0.0186482 0.0116813 0.007586505 0.2656976 0.4588717
##                   ACF1
## Training set 0.0102714
```

```
# Placeholder for models and their criteria
possible_models <- list()

# Loop over parameters and drift inclusion
for (p in 1:3) {
  for (q in 1:3) {
    for (P in 0:2) {
      for (Q in 0:2) {
        for (drift in c(TRUE, FALSE)) {
```

```
          tryCatch({
            # Fit the ARIMA model
            model <- Arima(log10(train_ts), order = c(p, 0, q), seasonal = c(P, 1, Q), include.drift = d

            # Extract criteria
            aicc <- model$aicc
            bic <- model$bic

            # Create a model name
            model_name <- paste("SARIMA(", p, ", 0,", q, "), (", P, ", 1,", Q, "), drift=", drift, sep =

            # Store the model's criteria
            possible_models[[model_name]] <- c(aicc, bic)
          }, error = function(e) {
            message("Error with model SARIMA(", p, ", 0,", q, "), (", P, ", 1,", Q, "), drift=", drift,
          })
        }
      }
    }
  }
}
```

```
## Error with model SARIMA(2, 0,2), (2, 1,0), drift=FALSE: non-finite finite-difference value [1]

## Error with model SARIMA(2, 0,2), (2, 1,1), drift=TRUE: initial value in 'vmmin' is not finite

## Error with model SARIMA(3, 0,1), (2, 1,2), drift=FALSE: non-finite finite-difference value [1]

## Error with model SARIMA(3, 0,2), (2, 1,2), drift=TRUE: non-finite finite-difference value [3]
```

```
# Function to find the best model based on a given criterion
find_best_model <- function(models, criterion_index) {
  criteria_values <- sapply(models, `[`, criterion_index)
  best_model_name <- names(models)[which.min(criteria_values)]
  return(best_model_name)
}

# Find the best models based on AICc and BIC
best_aicc_model <- find_best_model(possible_models, 1)
best_bic_model <- find_best_model(possible_models, 2)

print(paste("Best AICc model:", best_aicc_model))
```

```
## [1] "Best AICc model: SARIMA(1, 0,2), (2, 1,1), drift=TRUE"
```

```
print(paste("Best BIC model:", best_bic_model))
```

```
## [1] "Best BIC model: SARIMA(2, 0,1), (0, 1,1), drift=FALSE"
```

```r
# Optionally, you can also print the AICc and BIC values for these best models
print(possible_models[[best_aicc_model]])
```

```
## [1] -1197.972 -1170.750
```

```r
print(possible_models[[best_bic_model]])
```

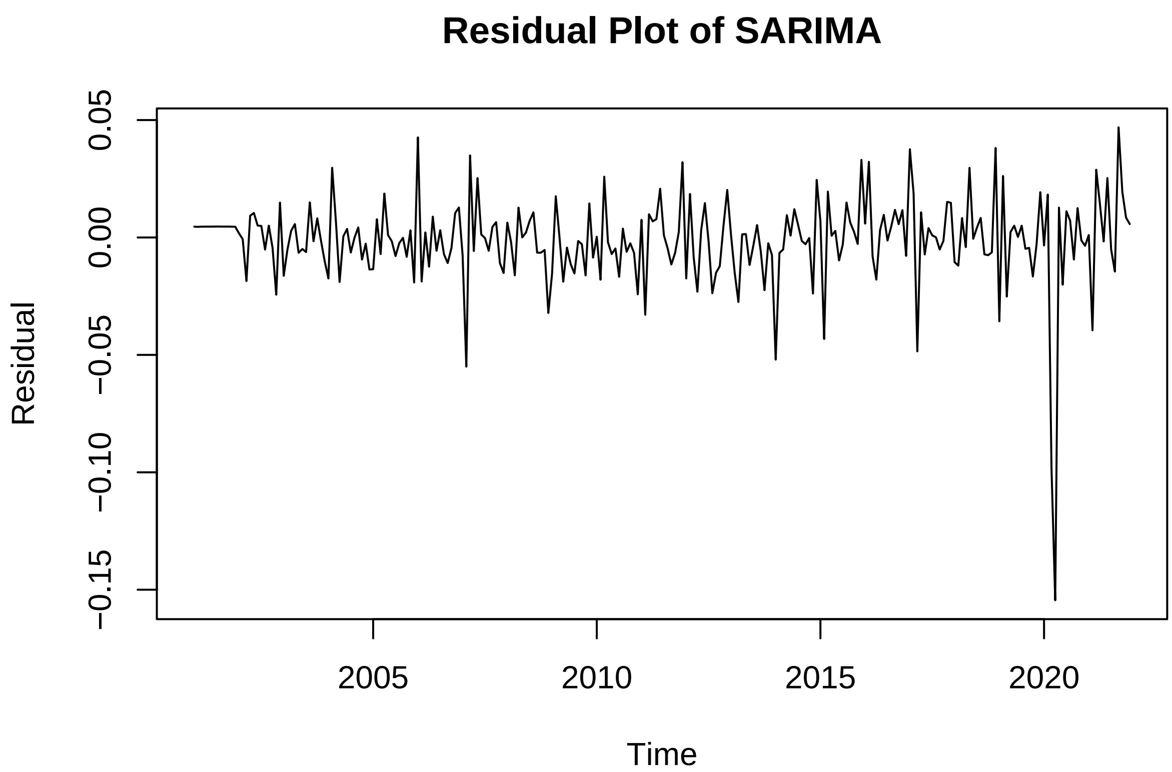
```
## [1] -1193.520 -1176.373
```

I will choose the best BIC model with no drift.

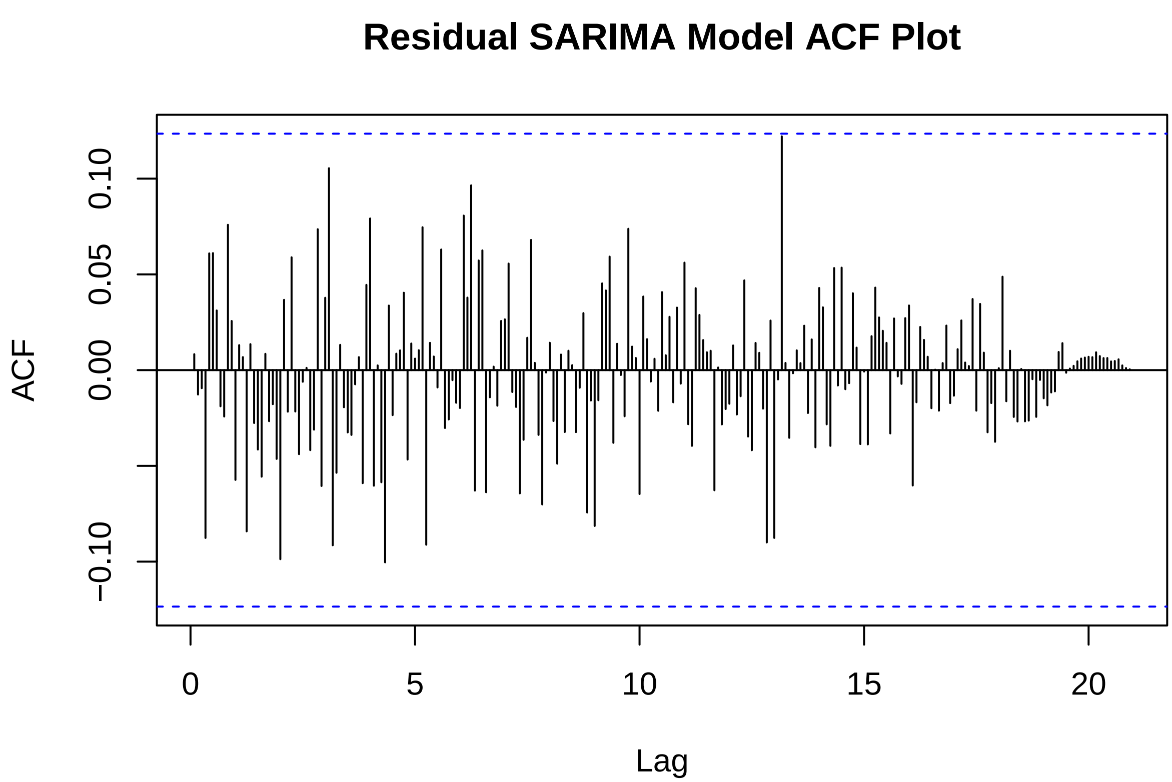
```r
model_arima1 <- Arima(log10(train_ts), order = c(2, 0, 1), seasonal = c(0, 1, 1))
summary(model_arima1)
```

```
## Series: log10(train_ts)
## ARIMA(2,0,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ar2      ma1     sma1
##       1.3448  -0.3480  -0.7483  -0.7636
## s.e.  0.1071   0.1058   0.0750   0.0493
##
## sigma^2 = 0.0003782:  log likelihood = 601.89
## AIC=-1193.78   AICc=-1193.52   BIC=-1176.37
##
## Training set error measures:
##                        ME      RMSE        MAE         MPE      MAPE      MASE
## Training set -0.001289913 0.0188193 0.01168669 -0.03065114 0.2659465 0.4590831
##                     ACF1
## Training set 0.008347155
```

```r
ts.plot(model_arima1$residuals, ylab = 'Residual', main = "Residual Plot of SARIMA")
```



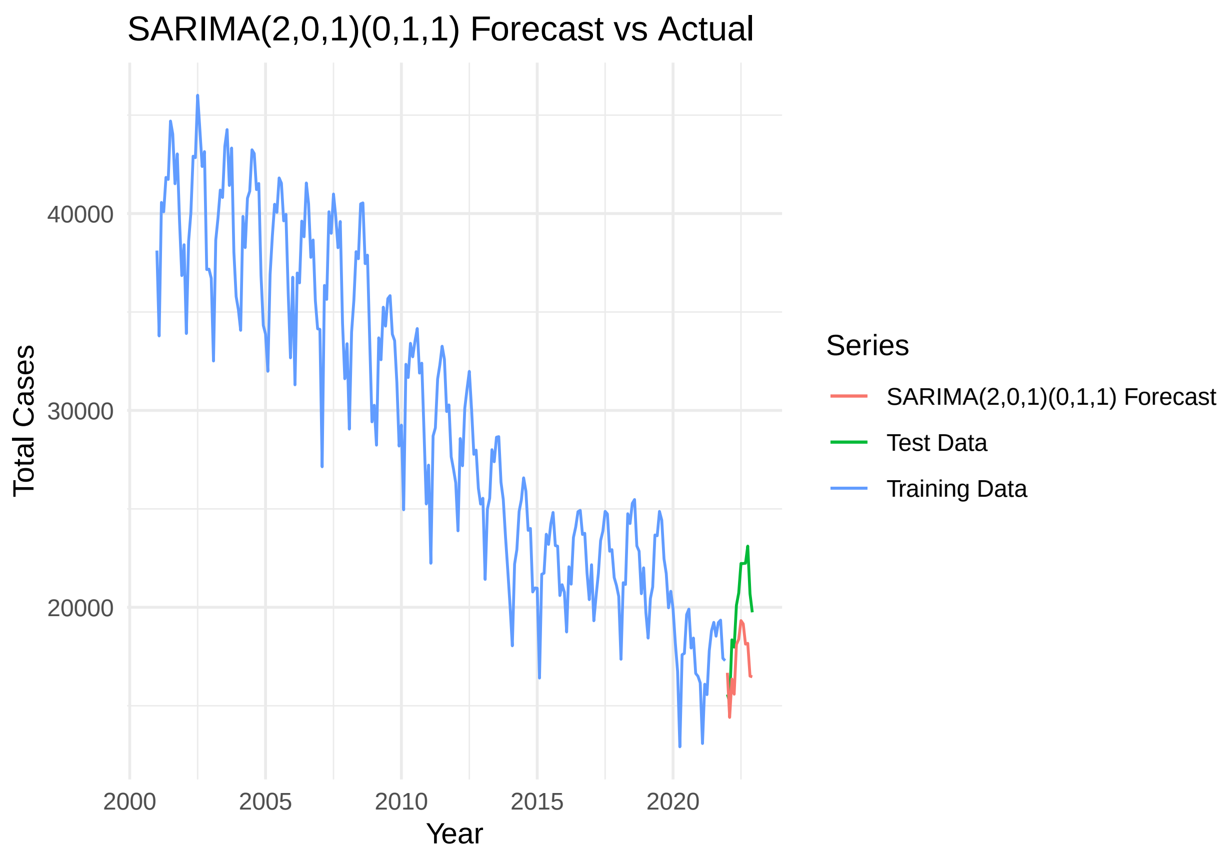

```
acf(model_arima1$residuals, lag = 300, main = "Residual SARIMA Model ACF Plot")
```



There is a huge intervention casued by Covid, then I will fit intervention analysis based on Sarima model.

```r
# Test it
pred1 <- forecast(model_arima1, h = length(test_ts))
original_pred1_arima <- 10^pred1$mean
```

```r
# Plot the forecasts and include training data
autoplot(train_ts, series="Training Data") +
  autolayer(test_ts, series="Test Data") +
  autolayer(original_pred1_arima, series="SARIMA(2,0,1)(0,1,1) Forecast") +
  ggtitle("SARIMA(2,0,1)(0,1,1) Forecast vs Actual") +
  xlab("Year") + ylab("Total Cases") +
  theme_minimal() +
  guides(colour=guide_legend(title="Series"))
```




```r
rmse_sarima <- forecast::accuracy(original_pred1_arima, test_ts)[,2]
smape_sarima <- Metrics::smape(original_pred1_arima, test_ts)
print(paste("RMSE for SARIMA model:", rmse_sarima))
```

```
## [1] "RMSE for SARIMA model: 3004.72283672862"
```

```r
print(paste("sMAPE for SARIMA model:", smape_sarima))
```

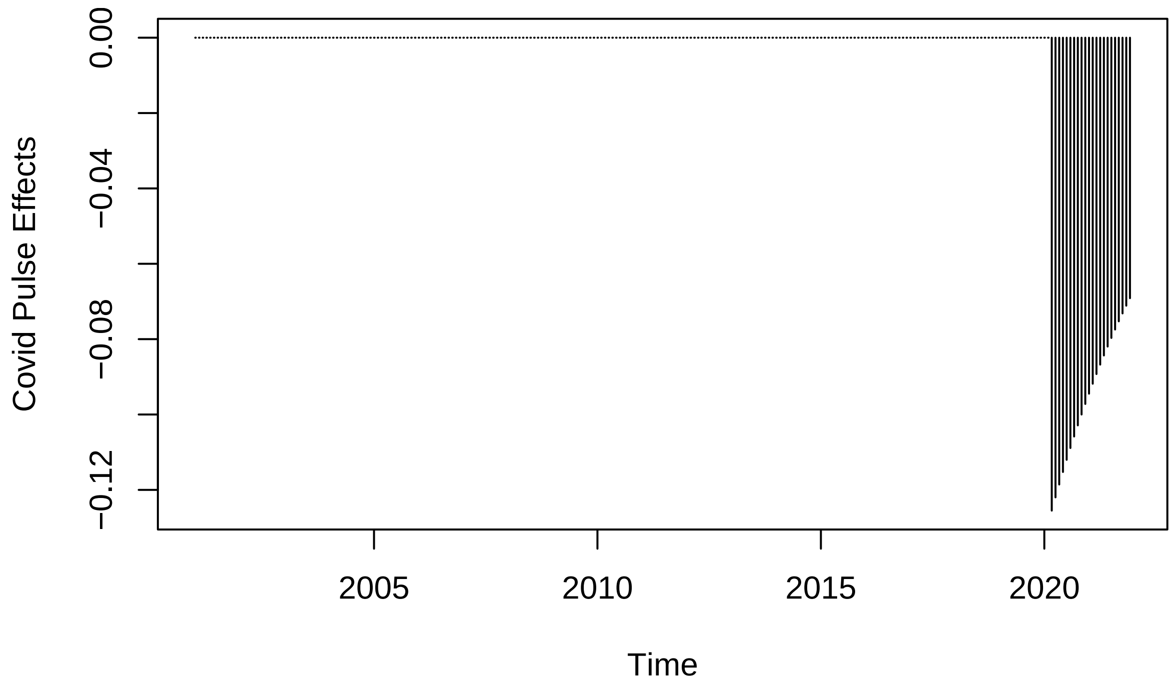
```
## [1] "sMAPE for SARIMA model: 0.145083272849728"
```

Let's try intervention analysis.

```r
crimes.pre.intervention <- window(log10(train_ts), end=c(2020,2))
pre_int <- Arima(crimes.pre.intervention, order = c(2,0,1), seasonal = c(0,1,1))
PCovid <- 1*(seq(train_ts)==231)

crime.mPulse <- arimax(log10(train_ts),order=c(2,0,1),seasonal=list(order=c(0,1,1), period=12), xtransf=
crime.mPulse
```

```
##
## Call:
## arimax(x = log10(train_ts), order = c(2, 0, 1), seasonal = list(order = c(0,
##     1, 1), period = 12), method = "ML", xtransf = data.frame(PCovid), transfer = list(c(1,
##     0)))
##
## Coefficients:
##          ar1      ar2      ma1     sma1  PCovid-AR1  PCovid-MA0
##       1.0831  -0.0866  -0.7465  -0.7504      0.9720     -0.1255
## s.e.  0.0895   0.0887   0.0576   0.0511      0.0155      0.0121
##
## sigma^2 estimated as 0.0002695:  log likelihood = 640.5,  aic = -1269.01
```

```r
plot(ts(filter(PCovid, filter=0.9720, method='recursive', side=1)*(-0.1255), frequency = 12, start=2001)
     type='h',ylab='Covid Pulse Effects')
```




```r
steps.ahead = 12

tf<-filter(1*(seq(1:(length(train_ts) + steps.ahead))==231), filter=0.9720, method='recursive',side=1) *
forecast.arima<-Arima(log10(train_ts), order=c(2,0,1), seasonal = c(0,1,1), xreg=tf[1:(length(tf) - step
forecast.arima
```

```
## Series: log10(train_ts)
```

```
## Regression with ARIMA(2,0,1)(0,1,1)[12] errors
##
## Coefficients:
##          ar1      ar2      ma1     sma1   xreg
##       1.0829  -0.0864  -0.7466  -0.7502  1.000
## s.e.  0.0894   0.0886   0.0575   0.0505  0.096
##
## sigma^2 = 0.0002763:  log likelihood = 640.5
## AIC=-1269.01   AICc=-1268.65   BIC=-1248.12
```

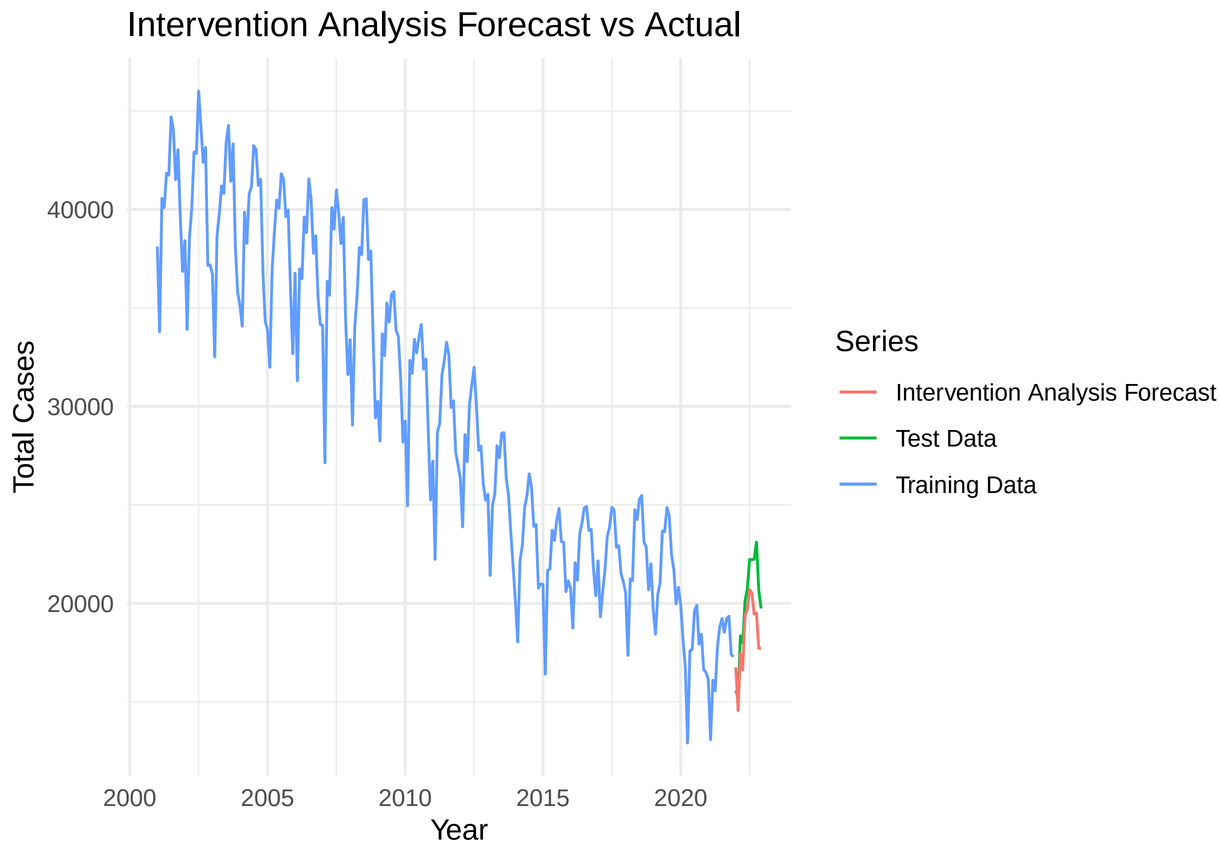
```r
start_idx = length(tf) - steps.ahead + 1
pred_intervention <- predict(forecast.arima, n.ahead = steps.ahead, newxreg=tf[start_idx:length(tf)])

predicted_original_intervention <- 10^pred_intervention$pred

rmse_intervention <- forecast::accuracy(predicted_original_intervention, test_ts)[,2]
smape_intervention <- Metrics::smape(predicted_original_intervention, test_ts)
print(paste("RMSE for intervention model:", rmse_intervention))
```

```
## [1] "RMSE for intervention model: 1934.71206442071"
```

```r
print(paste("sMAPE for intervention model:", smape_intervention))
```

```
## [1] "sMAPE for intervention model: 0.0875244111151827"
```

```r
predicted_original_intervention_ts <- ts(predicted_original_intervention, start = c(2022, 1), frequency

autoplot(train_ts, series="Training Data") +
  autolayer(test_ts, series="Test Data") +
  autolayer(predicted_original_intervention_ts, series="Intervention Analysis Forecast") +
  ggtitle("Intervention Analysis Forecast vs Actual") +
  xlab("Year") + ylab("Total Cases") +
  theme_minimal() +
  guides(colour=guide_legend(title="Series"))
```


Intervention Analysis Forecast vs Actual
Total Cases
40000
30000
20000
2000
2005
2010
2015
2020
Year
Series
Intervention Analysis Forecast
Test Data
Training Data