144_project2

Krish Methi, Andrew Brown, Krithik J

2/18/2024

```
# Clear everything and load libraries
rm(list=ls(all=TRUE))

# Load libraries
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.1.2

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

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.1.2
```

```
library(fpp3)
```

```
## Warning: package 'fpp3' was built under R version 4.1.2

## -- Attaching packages ---------------------------------------------- fpp3 0.5 --

## v tibble      3.2.1     v tsibble     1.1.3
## v dplyr       1.1.2     v tsibbledata 0.4.1
## v tidyr       1.2.1     v feasts      0.3.1
## v lubridate   1.8.0     v fable       0.3.3
## v ggplot2     3.4.4     v fabletools  0.3.4

## Warning: package 'tibble' was built under R version 4.1.2

## Warning: package 'dplyr' was built under R version 4.1.2

## Warning: package 'tidyr' was built under R version 4.1.2

## Warning: package 'tsibble' was built under R version 4.1.2

## Warning: package 'tsibbledata' was built under R version 4.1.2
## Warning: package 'feasts' was built under R version 4.1.2
```

```
## Warning: package 'fable' was built under R version 4.1.2
```

```
## -- Conflicts ------------------------------------------------- fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```
library(tseries)
library(seasonal)
```

```
## Warning: package 'seasonal' was built under R version 4.1.2
```

```
##
## Attaching package: 'seasonal'
```

```
## The following object is masked from 'package:tibble':
##
##     view
```

```
library(fable)
library(stats)
require(graphics)
library(dplyr)
library(tsibble)
library(tsibbledata)
library(vars)
```

```
## Loading required package: MASS
```

```
## Warning: package 'MASS' was built under R version 4.1.2
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## Loading required package: strucchange
```

```
## Warning: package 'strucchange' was built under R version 4.1.2
```

```
## Loading required package: zoo
## Warning: package 'zoo' was built under R version 4.1.2

##
## Attaching package: 'zoo'

## The following object is masked from 'package:tsibble':
##
##     index

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

## Loading required package: sandwich

## Loading required package: urca

## Warning: package 'urca' was built under R version 4.1.2

## Loading required package: lmtest

## Warning: package 'lmtest' was built under R version 4.1.2

##
## Attaching package: 'vars'

## The following object is masked from 'package:fable':
##
##     VAR
```

Part 1: Introduction

The data we are working with for this project is data on monthly imports of goods from January of 2010 to December of 2023 for two states, New Jersey and Georgia. So we have two time series datasets with the same monthly frequency and duration, New Jersey and Georgia. We picked New Jersey and Georgia as they both had relatively similar numbers, and are also two of the more impactful states in the US in terms of economic productivity. Running time series analysis in modeling and predicting these imports can be impactful in gaining economic insight for two powerful economic hubs for the US.

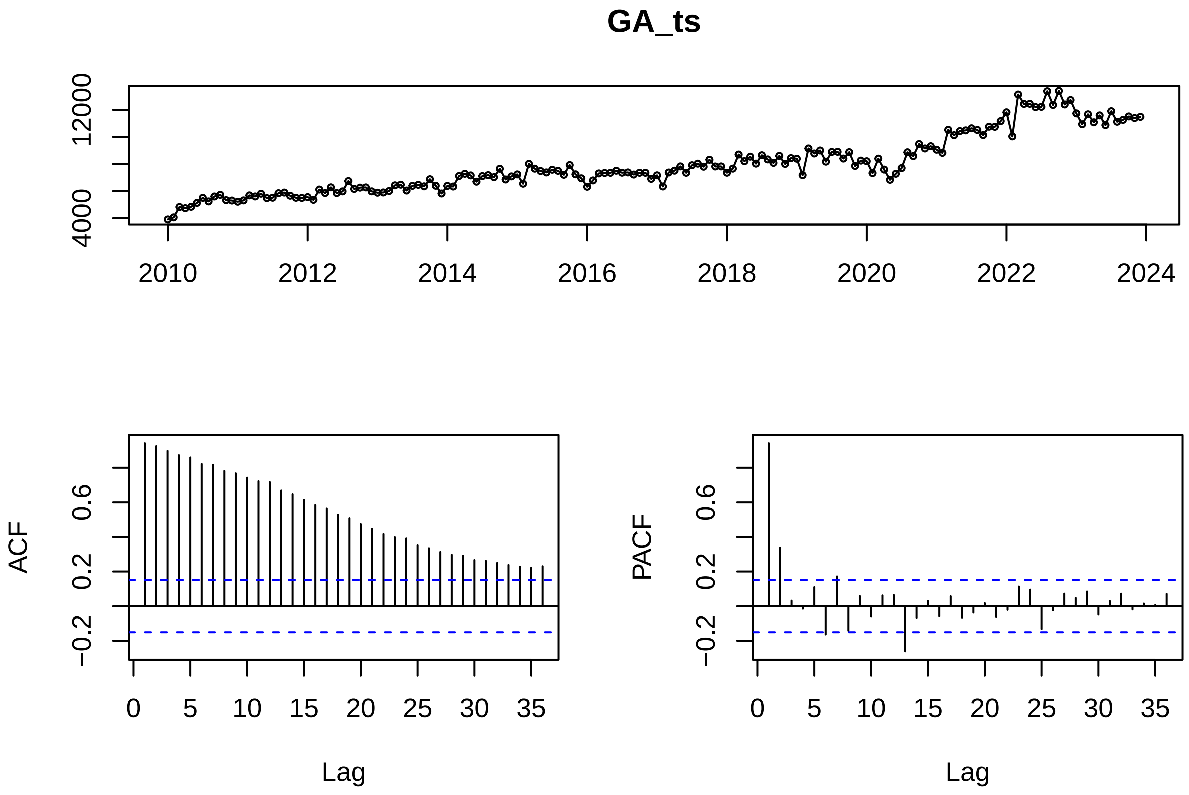
Part 2: Analysis

Load in the Data and Create time series variables

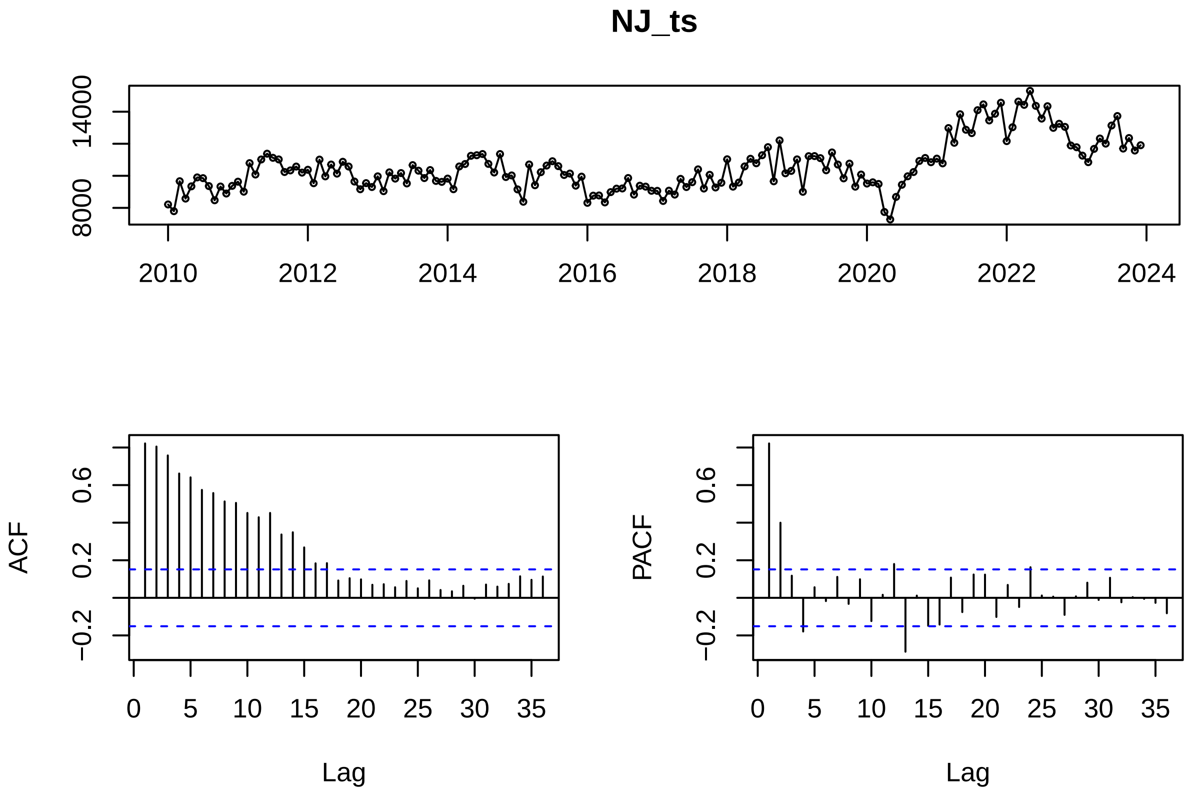
```r
# Load in the data and store them
setwd("~/Downloads")
data1 <- read.csv("IMPTOTNJ.csv")
data2 <- read.csv("IMPTOTGA.csv")
# Create time series variables for Georgia and New Jersey data
GA_ts <- ts(data2$IMPTOTGA, start = 2010, frequency = 12)
NJ_ts <- ts(data1$IMPTOTNJ, start = 2010, frequency = 12)
```

A. Produce time series plot of the data with respective ACF and PACF

```
#Produce time series plot and ACF and PACF using tsdisplay
tsdisplay(GA_ts)
```



```
tsdisplay(NJ_ts)
```

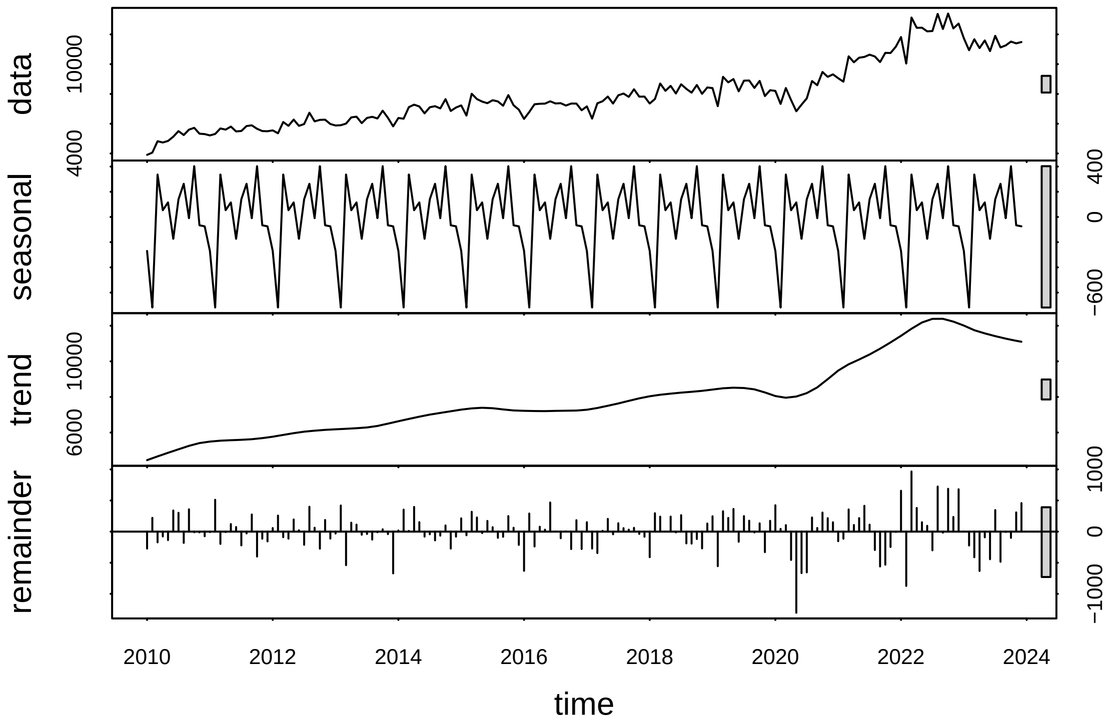


First in terms of the Georgia Data, we can see the upward trend and slight seasonality from the timeplot, and the decay in the ACF and spikes in the PACF suggest an AR process as well for the data. Going back to the trend, discounting the major drop due to the pandemic, the positive trend seems more definitive.

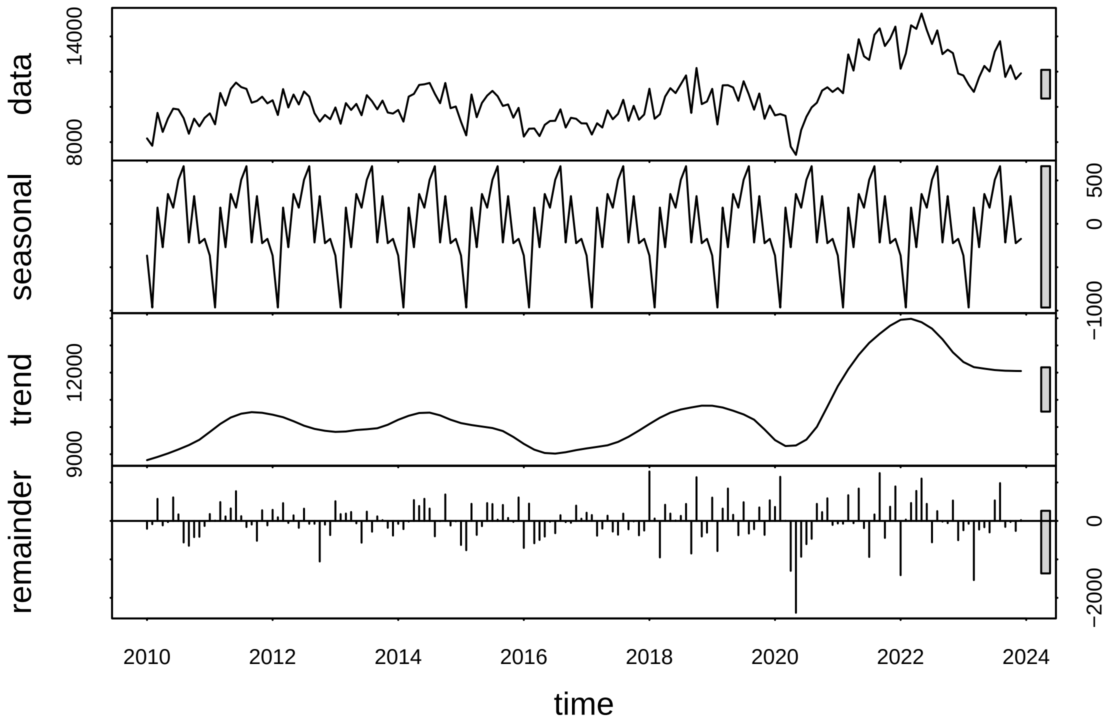
In terms of the New Jersey Data, we see similar aspects. We see a positive trend, although less definitive and more volatile, and we see seasonality as well. We also see similar ACF and PACF plots signifying a potential AR process, however there is faster decay in the ACF in the NJ data than in the GA data which seems accurate with the time series plot.

B. Plot the stl decomposition of the data and discuss results

```
# STL decomposition plot for the Georgia and New Jersey data
plot(stl(GA_ts,s.window="periodic"))
```



```
plot(stl(NJ_ts,s.window="periodic"))
```



Based on these two stl decomposition plots, we can definitely see all three components presented. In terms of the trend, for both GA and NJ we can see a positive trend in imports from 2010 to 2023, with Georgia seeming to be a little bit stronger in the linear aspect but nonetheless a positive trend in both cases.

In terms of the seasonality, again that is very clear in both stl plots, and pretty constant over the course of time as well in terms of amplitude for both plots

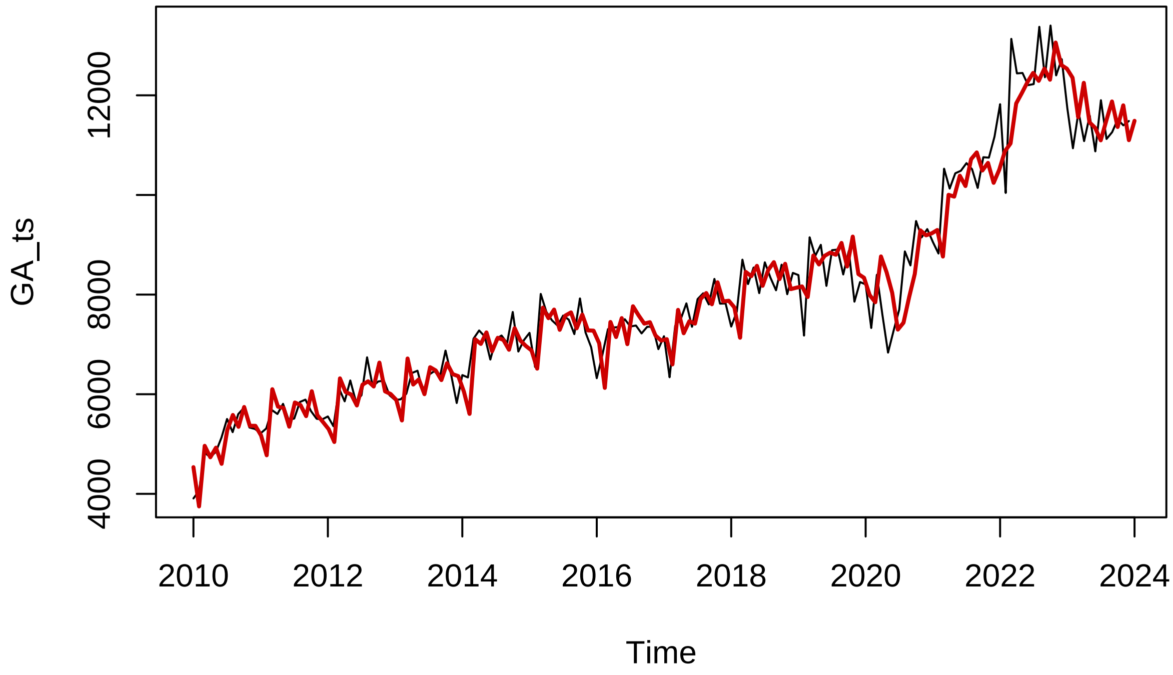
In terms of the cycles, the remainder does seem to exhibit slight patterns indicating cycles in both stl plots. We can also see a clear outlier just after 2020 in both plots which is expected with the pandemic and the economic impact of that.

So overall, we can see all 3 components likely present in both stl plots of the Georgia and New Jersey monthly imports time series.

C. Fit a model to include trend seasonality and cycles and discuss the model

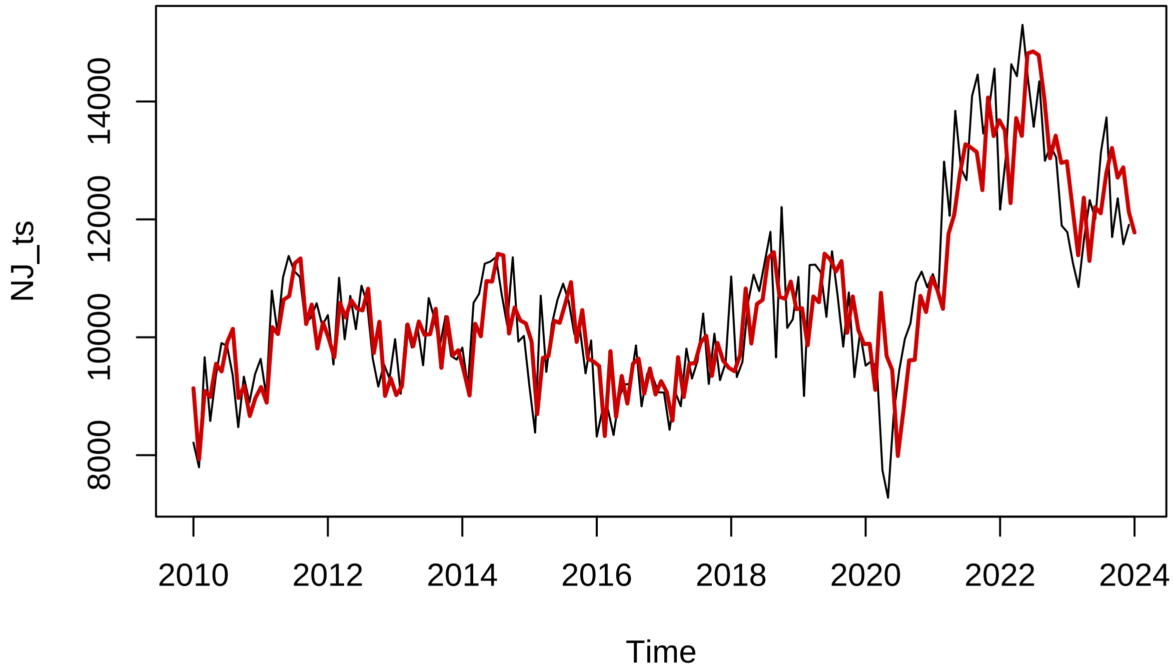
```r
#Quadratic trend fit with seasonal dummies and AR1 for Georgia data
# Initialize t and t^2 and run initial tslm model
t<-seq(2010, 2024,length=length(GA_ts))
t2 = t^2
m1 = tslm(GA_ts ~ t + t2 + season)
# use auto arima to generate component for the cycles, and combine with tslm model to get final fittd v
ARMA_GA=auto.arima(m1$residuals, max.P = 0, max.D = 0, max.Q = 0)
fitted_ARMA_GA <- ARMA_GA$fitted
GA_fitted <- m1$fitted.values + fitted_ARMA_GA
# plot the time series and the fitted values
```

```
plot(GA_ts)
lines(t,GA_fitted,col="red3",lwd=2)
```



For the Georgia time series data, we fit a model with a quadratic trend, monthly seasonal dummies, and an ARMA(2,3) for the cyclical component. The quadratic trend stemmed from the exponential like increase over time from 2010 to 2023 which made it seem like a good fit for the model. The monthly seasonal dummies were placed as specified in the project guidelines. We used auto arima to see what would be a good fit for the cyclical component, and it gave us an arma 2,3 so we used the fitted values from the AR1 fit and added that to the trend and seasonality fitted values to get to our model's fitted values. As we can see, the model does a decent job in terms of how well the fitted values overlay with the original data

```
#Quadratic trend fit with seasonal dummies and MA2 for New Jersey data
# Same process as what we did with Georgia
t<-seq(2010, 2024,length=length(NJ_ts))
t2 = t^2
m2 = tslm(NJ_ts ~ t + t2 + season)
ARMA_NJ=auto.arima(m2$residuals, max.P = 0, max.D = 0, max.Q = 0)
fitted_ARMA_NJ <- ARMA_NJ$fitted
NJ_fitted <- m2$fitted.values + fitted_ARMA_NJ
plot(NJ_ts)
lines(t,NJ_fitted,col="red3",lwd=2)
```

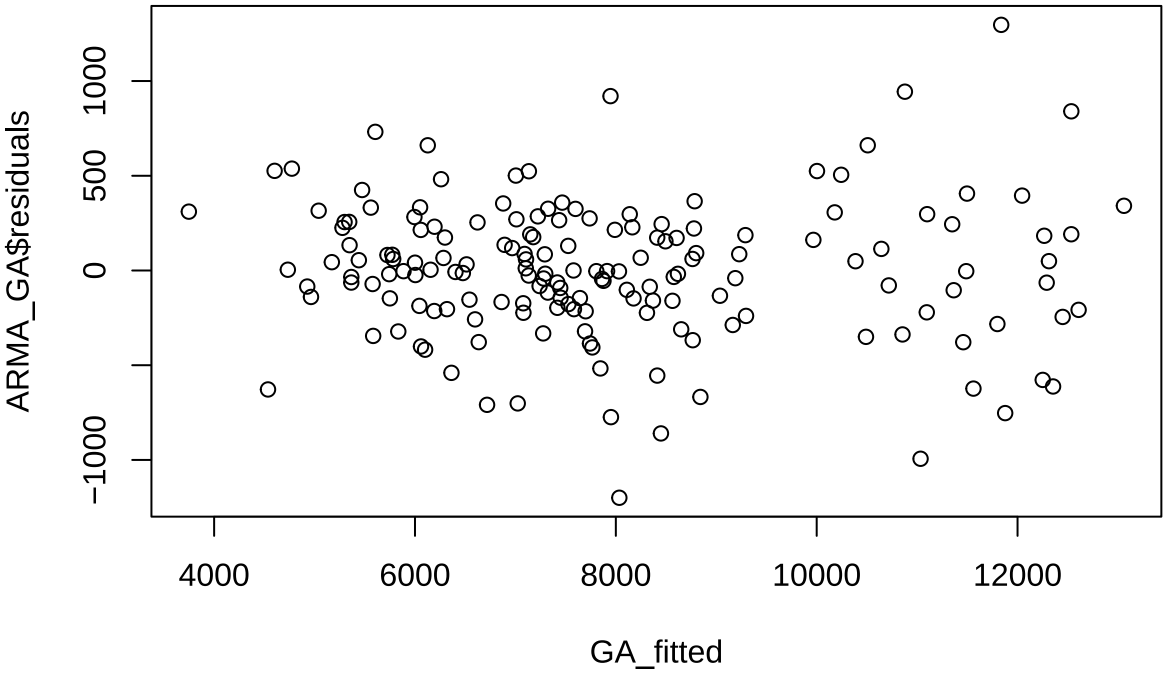


For the New Jersey time series data, similarly we fit a model with a quadratic trend, monthly seasonal dummies, but we did an ARMA(1,4) for the cyclical component. The quadratic trend stemmed from the exponential like increase over time from 2010 to 2023 which made it seem like a good fit for the model. The monthly seasonal dummies were placed as specified in the project guidelines. using auto arima this time gave us an ARMA 1,4 so we used the fitted values from the ARMA 1,4 fit and added that to the trend and seasonality fitted values to get to our model's fitted values. As we can see, the model also does a decent job in terms of how well the fitted values overlay with the original data, discounting the drop due to the pandemic.

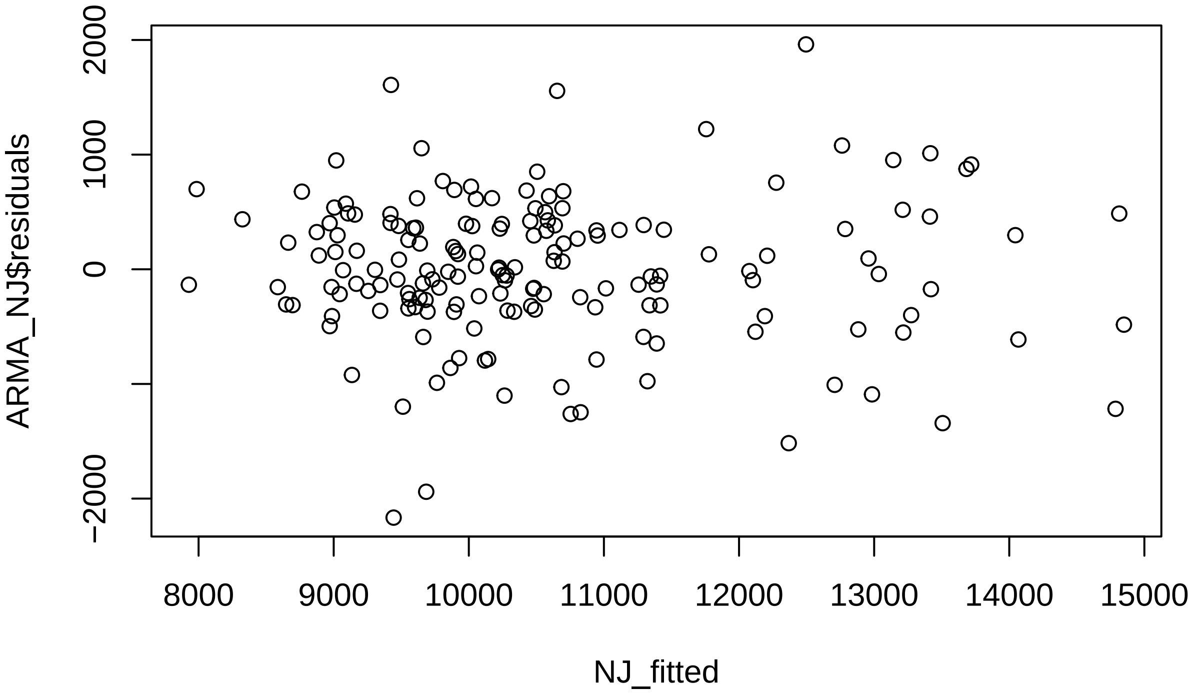
D. Plot the respective residuals vs fitted values

A quick note, in the project guidelines they skip a letter from c to e, however we proceeded as if no letter was skipped which is why we wrote this part as letter D instead of letter E in the guidelines.

```
## Georgia
plot(GA_fitted, ARMA_GA$residuals)
```



```
## New Jersey
plot(NJ_fitted, ARMA_NJ$residuals)
```

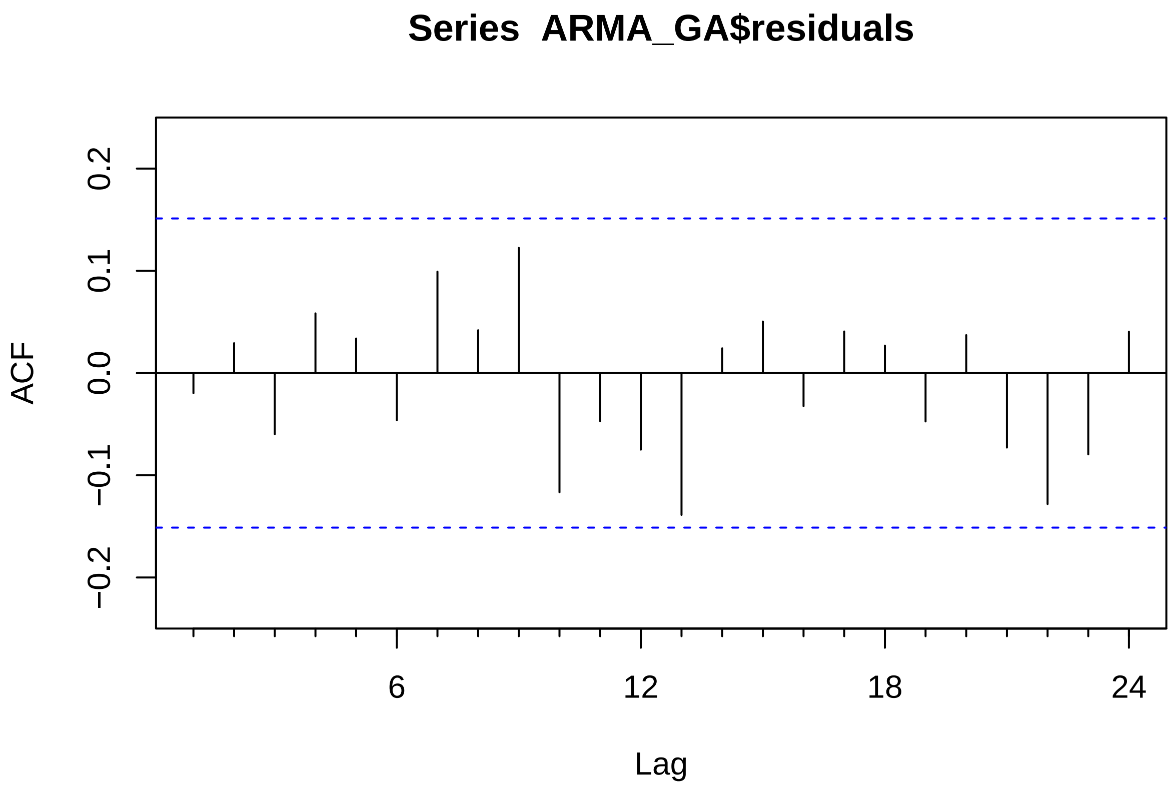


An important thing to note, based on the way we proceeded in the previous code, by having the first model account for the trend and seasonality, and then use the ARMA on the residuals from the first model, we can now operate on the residuals from the 2nd ARMA model as our overall respective residuals.

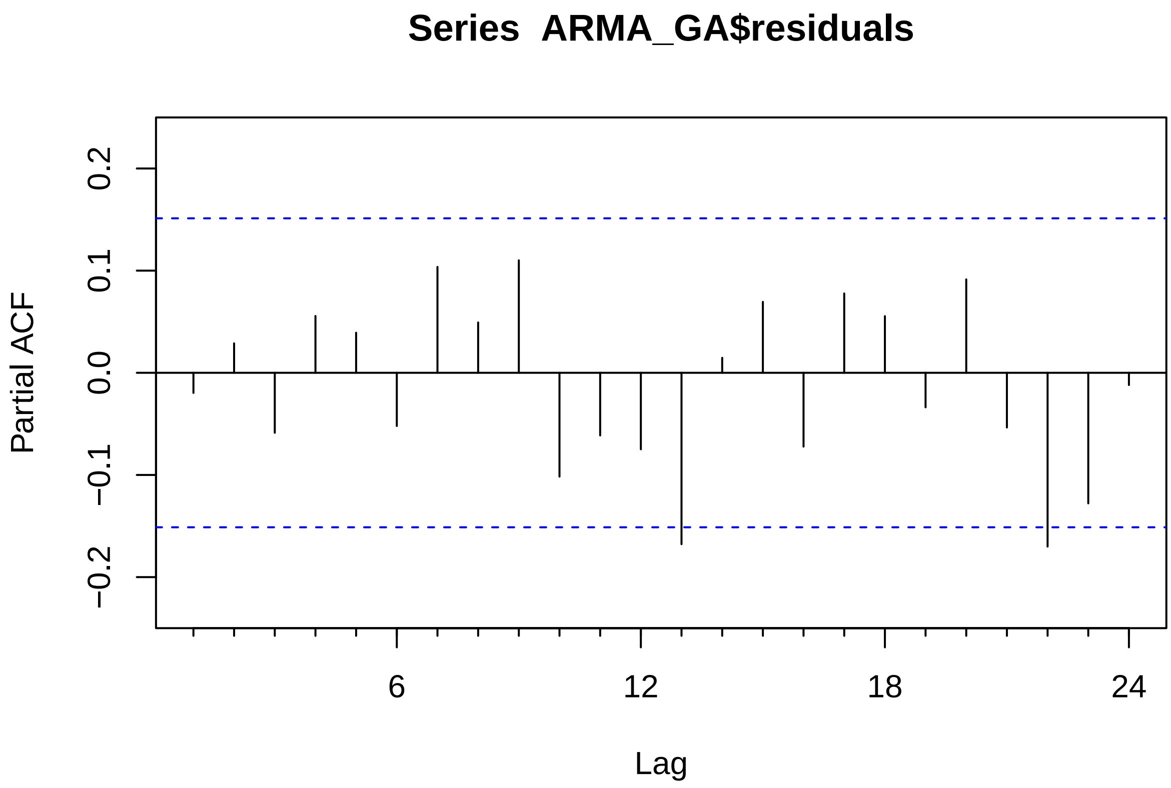
From these plots, we can see that for both Georgia and New Jersey, the residuals are fairly scattered and centered around zero, with some slight outliers across the data. Based on that there are no big patterns, we can say that our models have been pretty successful in capturing the dynamics of the Georgia and New Jersey time series data.

E. Plot the ACF and PACF of the residuals and discuss

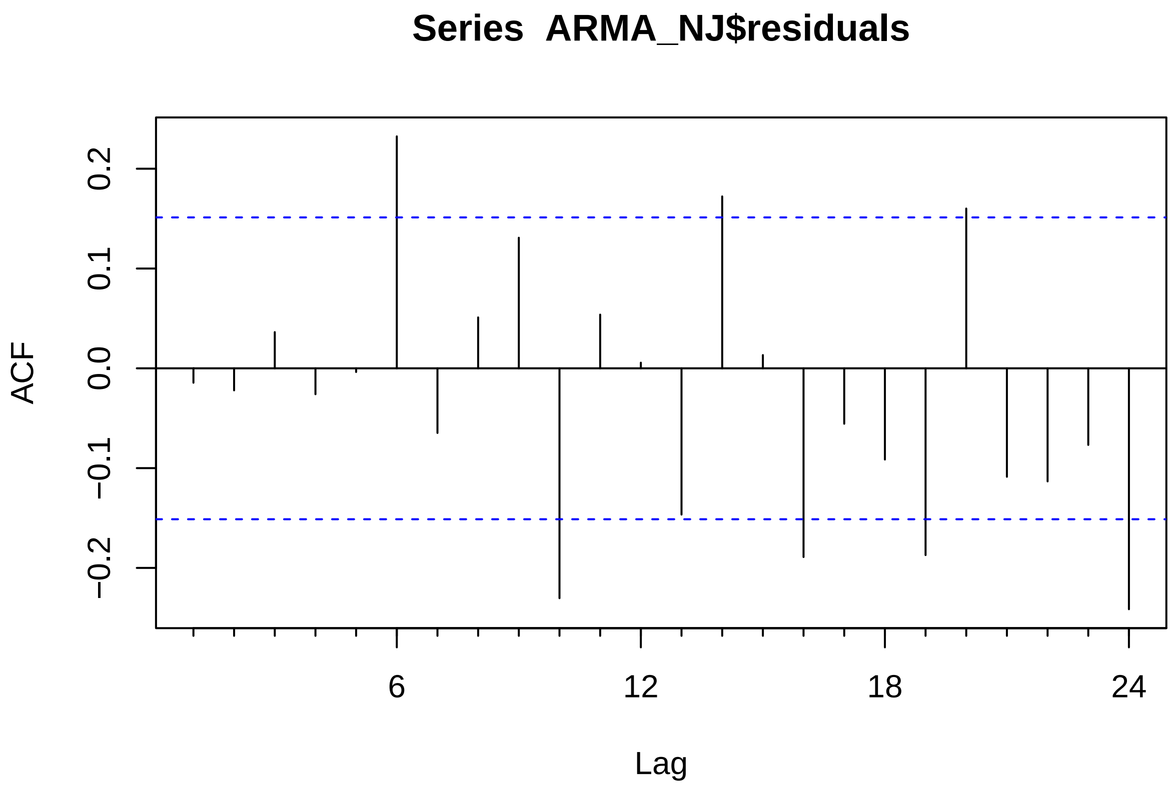
```
## Georgia
Acf(ARMA_GA$residuals)
```



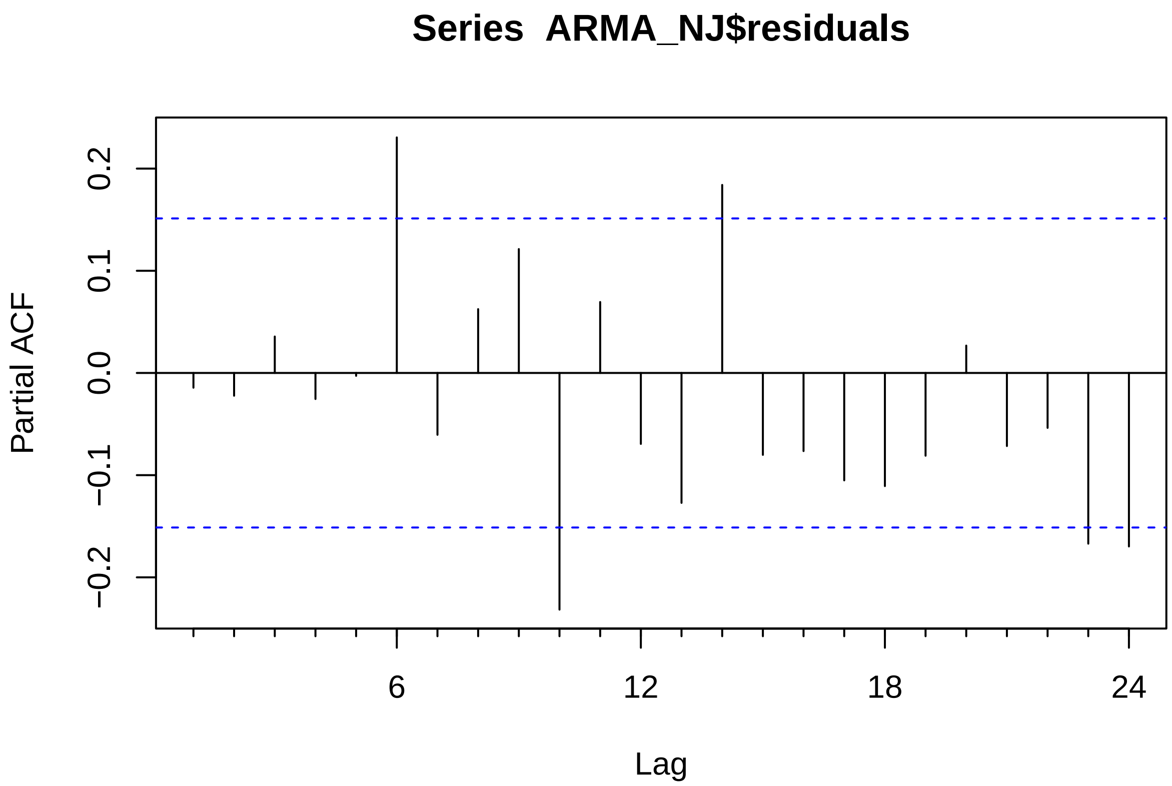
```
Pacf(ARMA_GA$residuals)
```



```
# New Jersey
Acf(ARMA_NJ$residuals)
```



```
Pacf(ARMA_NJ$residuals)
```



First, looking at Georgia, We see hardly any significant spikes in either the ACF or the PACF of the residuals. And even when they do cross the bounds, they only do so by very small increments.

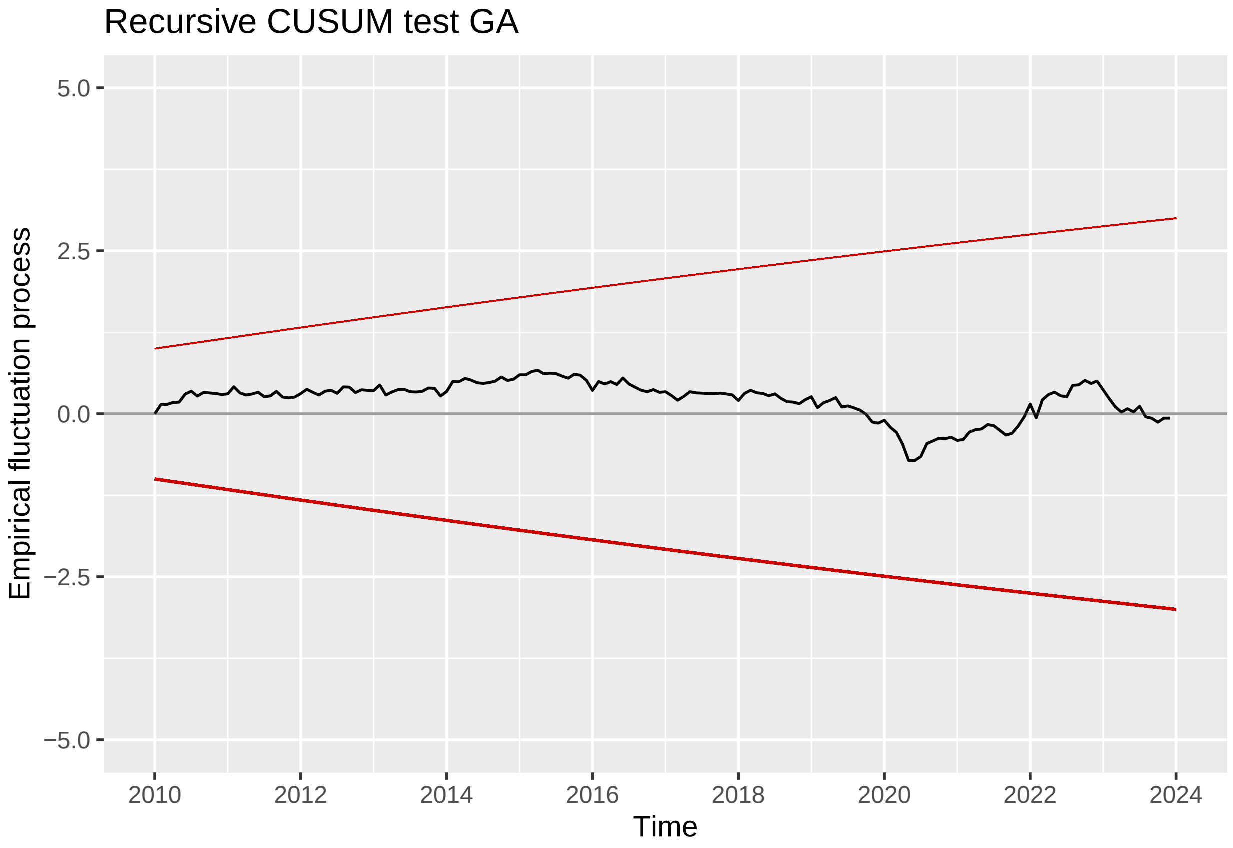
Looking at New Jersey, for the most part we can see that there aren't too many significant spikes in the ACF and PACF. However, we can see that there are more spikes that cross the threshold than in georgia's data. Intuitively this does make some sense as there is more of an MA aspect and more volatility in the New Jersey data that we have observed. But overall, most spikes either stay under the threshold, or barely cross the threshold.

F. Plot the respective CUSUM and interpret the plot

```r
# Georgia
CUSUM_GA <- efp(ARMA_GA$resid ~ 1, type = "Rec-CUSUM")$process %>%
  autoplot() +
  ylim(-5,5) +
  geom_hline(yintercept = 0, color = "grey24", alpha = 0.5) +
  geom_curve(aes(x = 2010, y = 1, xend = 2024, yend = 3), color = "red3",
             lwd = 0.25, curvature = -0.01) +
  geom_curve(aes(x = 2010, y = -1, xend = 2024, yend = -3), color = "red3",
            curvature = 0.01) +
  labs(title = "Recursive CUSUM test GA",
       y = "Empirical fluctuation process", x = "Time") +
  scale_x_continuous(breaks = seq(2010, 2024, by = 2))
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

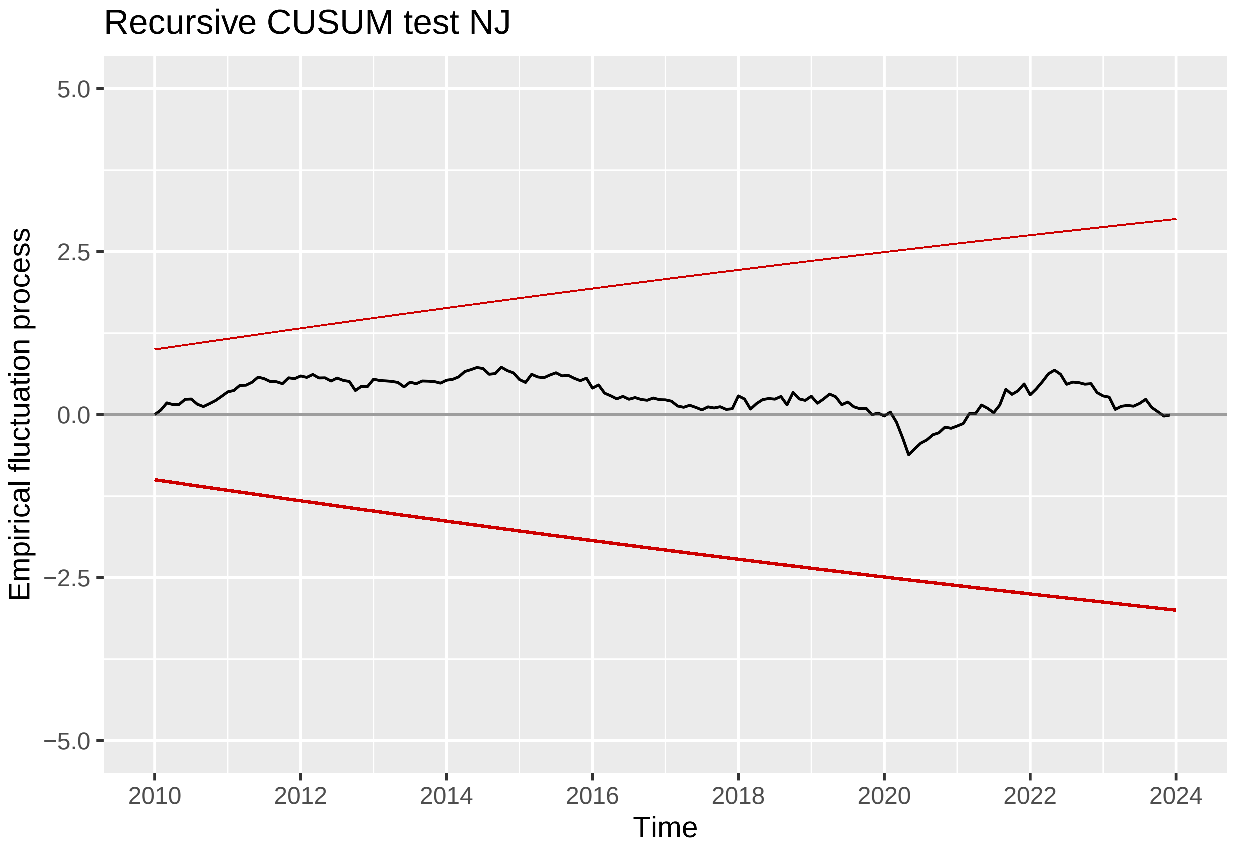
```
CUSUM_GA
```



```
# New Jersey
CUSUM_NJ <- efp(ARMA_NJ$resid ~ 1, type = "Rec-CUSUM")$process %>%
  autoplot() +
  ylim(-5,5) +
  geom_hline(yintercept = 0, color = "grey24", alpha = 0.5) +
  geom_curve(aes(x = 2010, y = 1, xend = 2024, yend = 3), color = "red3",
             lwd = 0.25, curvature = -0.01) +
  geom_curve(aes(x = 2010, y = -1, xend = 2024, yend = -3), color = "red3",
            curvature = 0.01) +
  labs(title = "Recursive CUSUM test NJ",
       y = "Empirical fluctuation process", x = "Time") +
  scale_x_continuous(breaks = seq(2010, 2024, by = 2))
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```
CUSUM_NJ
```



Looking at the CUSUM for Georgia, we can see there is no disparity past fluctuating around 0. There are no fluctuations past the red variance bands. Therefore there are no structural breaks in the model for Georgia.

The same conclusion occurs for New Jersey. Looking at the CUSUM for New Jersey, there is very little break off from 0 for the fluctuations. Again, none of the fluctuations break the red bands, therefore we can say there are no structural breaks in the model for New Jersey.

G. For the model, discuss the associated diagnostic statistics

```
# Summary of the initial model, MAPE/RMSE, and Box test for Georgia
summary(m1)
```

```
##
## Call:
## tslm(formula = GA_ts ~ t + t2 + season)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2737.57  -363.26   -19.58   330.64  2073.05
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.019e+08  1.528e+07   6.670 4.33e-10 ***
## t           -1.015e+05  1.515e+04  -6.701 3.67e-10 ***
## t2           2.528e+01  3.755e+00   6.733 3.10e-10 ***
## season2     -4.410e+02  2.719e+02  -1.622   0.1069
## season3      6.215e+02  2.719e+02   2.286   0.0236 *
## season4      3.403e+02  2.719e+02   1.251   0.2127
## season5      4.034e+02  2.720e+02   1.483   0.1401
## season6      1.077e+02  2.720e+02   0.396   0.6928
## season7      4.140e+02  2.720e+02   1.522   0.1300
## season8      5.270e+02  2.720e+02   1.937   0.0546 .
## season9      2.434e+02  2.721e+02   0.894   0.3725
## season10     6.514e+02  2.721e+02   2.394   0.0179 *
## season11     1.770e+02  2.722e+02   0.650   0.5164
## season12     1.641e+02  2.722e+02   0.603   0.5475
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 719.4 on 154 degrees of freedom
## Multiple R-squared:  0.8955, Adjusted R-squared:  0.8867
## F-statistic: 101.6 on 13 and 154 DF,  p-value: < 2.2e-16
```

```
MAPE(ARMA_GA$residuals, .actual = GA_ts) # MAPE
```

```
## [1] 3.488039
```

```
RMSE(ARMA_GA$residuals, .actual = GA_ts) # RMSE
```

```
## [1] 361.0431
```

```
GA_test <- Box.test(ARMA_GA$resid, lag = 12, type = "Ljung-Box")
GA_test
```

```
##
##  Box-Ljung test
##
## data:  ARMA_GA$resid
## X-squared = 10.645, df = 12, p-value = 0.5595
```

For Georgia data, we can see the model has a R-squared for 0.8955 with an F statistic of 101.6 and a p-value of less than 0.05 as well as in the intercept and slopes. From this, we can conclude that the model is statistically significant. The R-squared number also implies that the model is somewhat of a good fit in explaining variation.

We also calculated the MAPE and RMSE for Georgia. In terms of the MAPE, we got a value of 3.488%, and for the RMSE, we got a value of 361. For the MAPE, 3.488% is relatively low which suggests our model is doing a reasonable job. In terms of the 361 value for the RMSE, comparing that to the scale of the original data, that value is also relatively low therefore overall these statistics show that the model does reasonably well.

A ljung-box test was also performed as seen above. The test had a p value greater than 0.05, meaning that we fail to reject the null hypothesis and conclude that the residuals follow a white-noise pattern. The model then did a good job of capturing the serial correlation in the series.

```
# # Summary of the initial model, MAPE/RMSE, and Box test for New Jersey
summary(m2)
```

```
##
## Call:
## tslm(formula = NJ_ts ~ t + t2 + season)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4197.4  -717.3    59.5   651.8  2873.1
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.460e+08  2.384e+07   6.124 7.29e-09 ***
## t           -1.449e+05  2.363e+04  -6.132 6.98e-09 ***
## t2           3.598e+01  5.859e+00   6.142 6.66e-09 ***
## season2     -5.893e+02  4.243e+02  -1.389   0.1669
## season3      5.708e+02  4.243e+02   1.345   0.1806
## season4      1.200e+02  4.243e+02   0.283   0.7778
## season5      7.393e+02  4.244e+02   1.742   0.0835 .
## season6      5.798e+02  4.244e+02   1.366   0.1739
## season7      8.994e+02  4.244e+02   2.119   0.0357 *
## season8      1.050e+03  4.245e+02   2.473   0.0145 *
## season9      1.641e+02  4.246e+02   0.387   0.6996
## season10     6.941e+02  4.246e+02   1.635   0.1042
## season11     1.463e+02  4.247e+02   0.344   0.7309
## season12     1.959e+02  4.248e+02   0.461   0.6453
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1123 on 154 degrees of freedom
## Multiple R-squared:  0.5243, Adjusted R-squared:  0.4841
## F-statistic: 13.06 on 13 and 154 DF,  p-value: < 2.2e-16
```

```
MAPE(ARMA_NJ$residuals, .actual = NJ_ts) # MAPE
```

```
## [1] 4.527922
```

```
RMSE(ARMA_NJ$residuals, .actual = NJ_ts) # RMSE
```

```
## [1] 623.1835
```

```
NJ_test <- Box.test(ARMA_NJ$resid, lag = 12, type = "Ljung-Box")
NJ_test
```

```
##
##  Box-Ljung test
##
## data:  ARMA_NJ$resid
## X-squared = 24.399, df = 12, p-value = 0.01794
```

For New Jersey Data, there is a different conclusion. The R-squared of the model is only 0.5243, meaning that the model explains the variation comparably worse for New Jersey than the Georgia model for Georgia. The p-value on the f-statistic has a p-value which is still less than 0.05, as well as with the intercept and the slopes having small p-values, from this we can conclude that the model is statistically significant.

With the MAPE and RMSE here, we can see that both values are higher here than with Georgia, at 4.528% and 623 respectively. However, generally, those values are still reasonably low which suggests our model is reasonable as well. However clearly, the model for Georgia does better in terms of the MAPE and RMSE

The ljung-box test provides a different conclusion than scene in the Georgia model. The test has a p-value less than 0.05, meaning that we reject the null hypothesis and conclude that there may exist some residual that is beyond white noise, where a new model can perhaps capture the serial correlation better.

H. Use your model to forecast 12-steps ahead.

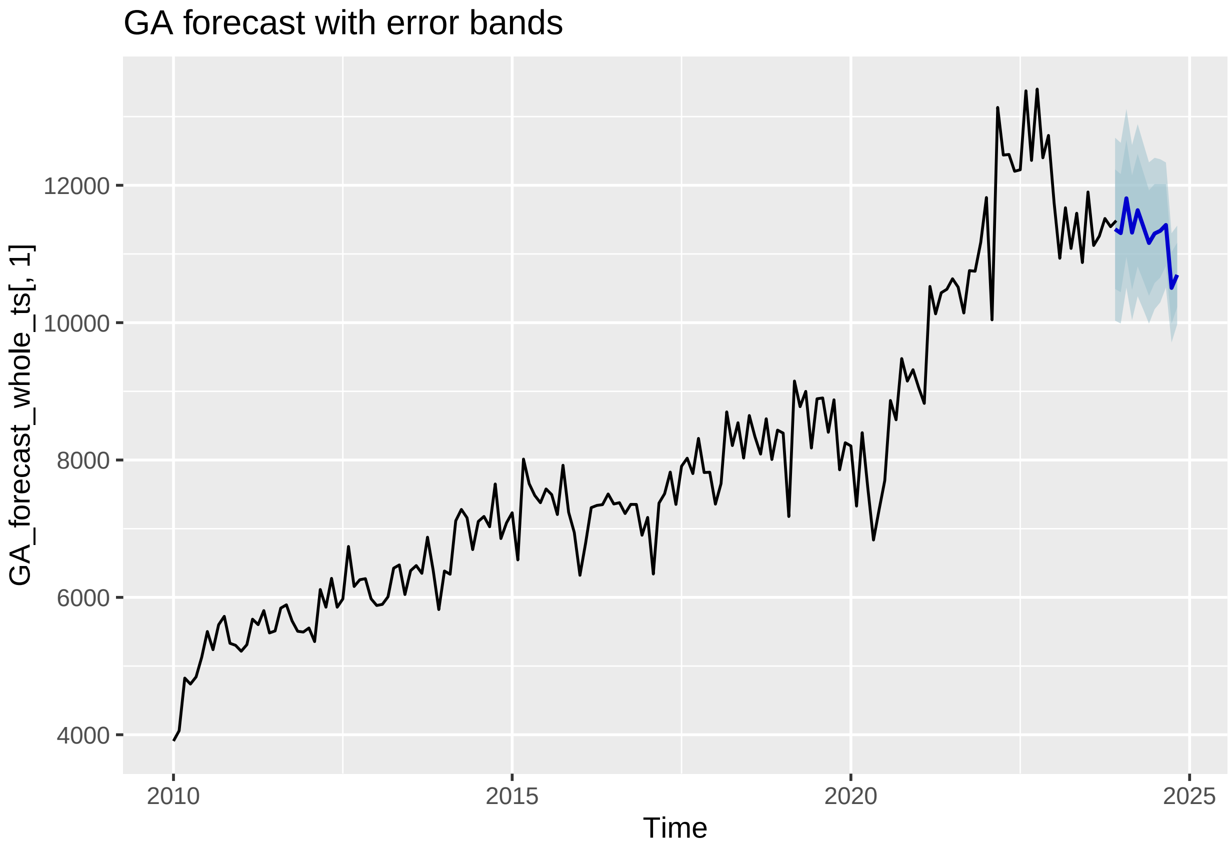
```
# Georgia
GA_forecast_part1 <- as.numeric(forecast(stl(GA_ts,s.window="periodic"), h = 12, method = "naive")[["mea
GA_forecast_part2 <- forecast(ARMA_GA, h = 12)
## Combine the bands and the point forecast
GA_forecast_whole <- bind_cols(Month = seq(as.Date("2024-01-01"), as.Date("2023-12-01"), length.out = 1
                               "estimate" = GA_forecast_part2[["mean"]],
                               GA_forecast_part2[["lower"]],
                               GA_forecast_part2[["upper"]], GA_forecast_part1) %>%
  #Error bands
  rename("upper_95" = "95%...6",
         seasonalhere = "...7") %>%
  mutate_at(vars(estimate:upper_95), function(x) x + GA_forecast_part1) %>%
  dplyr::select(-seasonalhere) %>%
  tsibble(index = Month)
```

```
## New names:
## * ‘80%‘ -> ‘80%...3‘
## * ‘95%‘ -> ‘95%...4‘
## * ‘80%‘ -> ‘80%...5‘
## * ‘95%‘ -> ‘95%...6‘
## * ‘‘ -> ‘...7‘
```

```
GA_forecast_whole_ts <- ts(GA_forecast_whole, start = c(2023.65, 4), frequency = 12)[,-1]
#Plot the forecast
GA_forecast_plot <- autoplot(GA_forecast_whole_ts[,1], color = "blue3") +
  geom_ribbon(aes(ymin = GA_forecast_whole_ts[,3], ymax = GA_forecast_whole_ts[,5]),
              fill = "lightblue3", alpha = 0.5) +
  geom_ribbon(aes(ymin = GA_forecast_whole_ts[,2], ymax = GA_forecast_whole_ts[,4]),
              fill = "lightblue3", alpha = 0.5) +
  geom_line(aes(y = GA_forecast_whole_ts[,1]), color = "blue3", lwd = 0.7) +
  labs(title = "GA forecast with error bands") +
  geom_line(data = GA_ts) + xlim(2010, NA)
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```
GA_forecast_plot
```



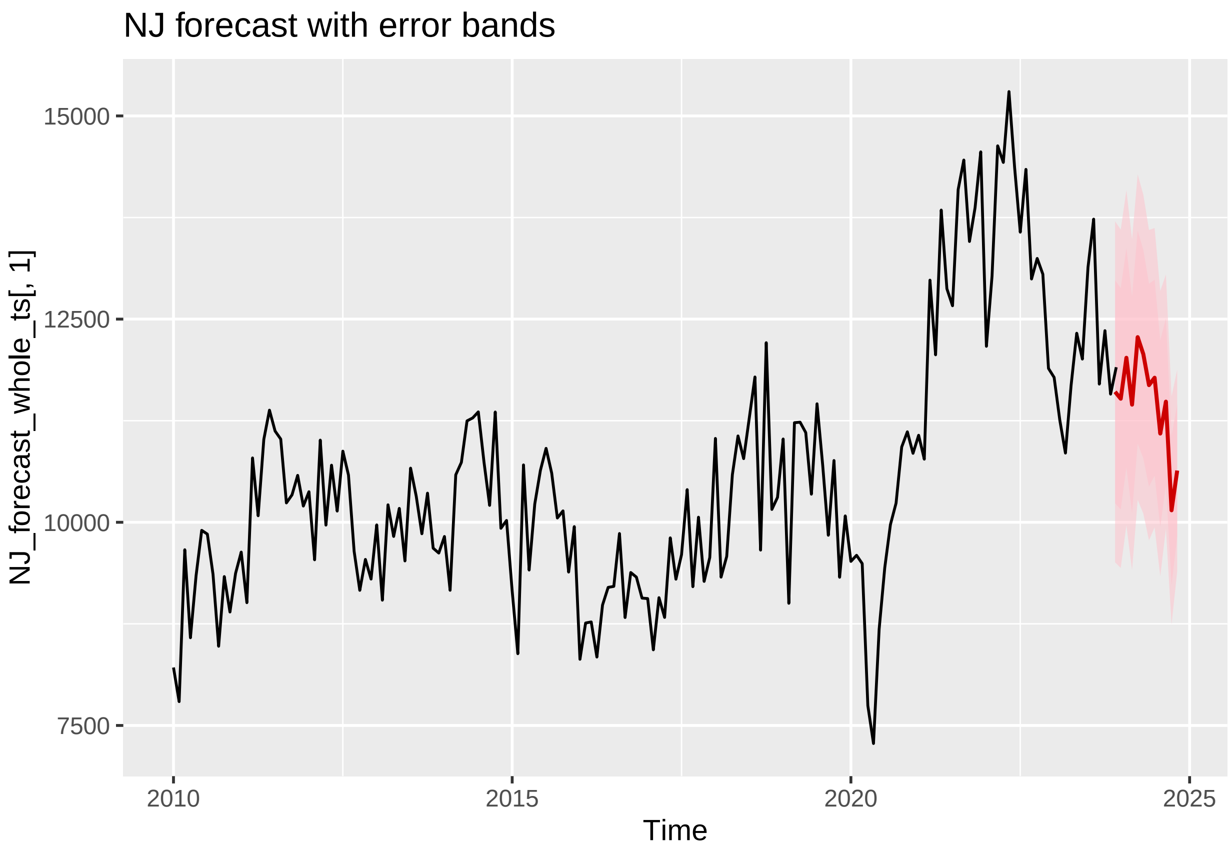
```
# New Jersey
NJ_forecast_part1 <- as.numeric(forecast(stl(NJ_ts,s.window="periodic"), h = 12, method = "naive")[["mea
NJ_forecast_part2 <- forecast(ARMA_NJ, h = 12)
## Combine the bands and the point forecast
NJ_forecast_whole <- bind_cols(Month = seq(as.Date("2024-01-01"), as.Date("2023-12-01"), length.out = 1
                               "estimate" = NJ_forecast_part2[["mean"]],
                               NJ_forecast_part2[["lower"]],
                               NJ_forecast_part2[["upper"]], NJ_forecast_part1) %>%
  ## Error bands
  rename("upper_95" = "95%...6",
         seasonalhere = "...7") %>%
  mutate_at(vars(estimate:upper_95), function(x) x + NJ_forecast_part1) %>%
  dplyr::select(-seasonalhere) %>%
  tsibble(index = Month)
```

```
## New names:
## * ‘80%‘ -> ‘80%...3‘
## * ‘95%‘ -> ‘95%...4‘
## * ‘80%‘ -> ‘80%...5‘
## * ‘95%‘ -> ‘95%...6‘
## * ‘‘ -> ‘...7‘
```

```
NJ_forecast_whole_ts <- ts(NJ_forecast_whole, start = c(2023.65, 4), frequency = 12)[,-1]
# Plot the forecast
NJ_forecast_plot <- autoplot(NJ_forecast_whole_ts[,1], color = "red3") +
  geom_ribbon(aes(ymin = NJ_forecast_whole_ts[,3], ymax = NJ_forecast_whole_ts[,5]),
              fill = "pink", alpha = 0.5) +
  geom_ribbon(aes(ymin = NJ_forecast_whole_ts[,2], ymax = NJ_forecast_whole_ts[,4]),
              fill = "pink", alpha = 0.5) +
  geom_line(aes(y = NJ_forecast_whole_ts[,1]), color = "red3", lwd = 0.7) +
  labs(title = "NJ forecast with error bands") +
  geom_line(data = NJ_ts) + xlim(2010, NA)
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```
NJ_forecast_plot
```



Above are forecasts that start at the end of 2023 and continues till the end of 2024 for both GA and NJ.

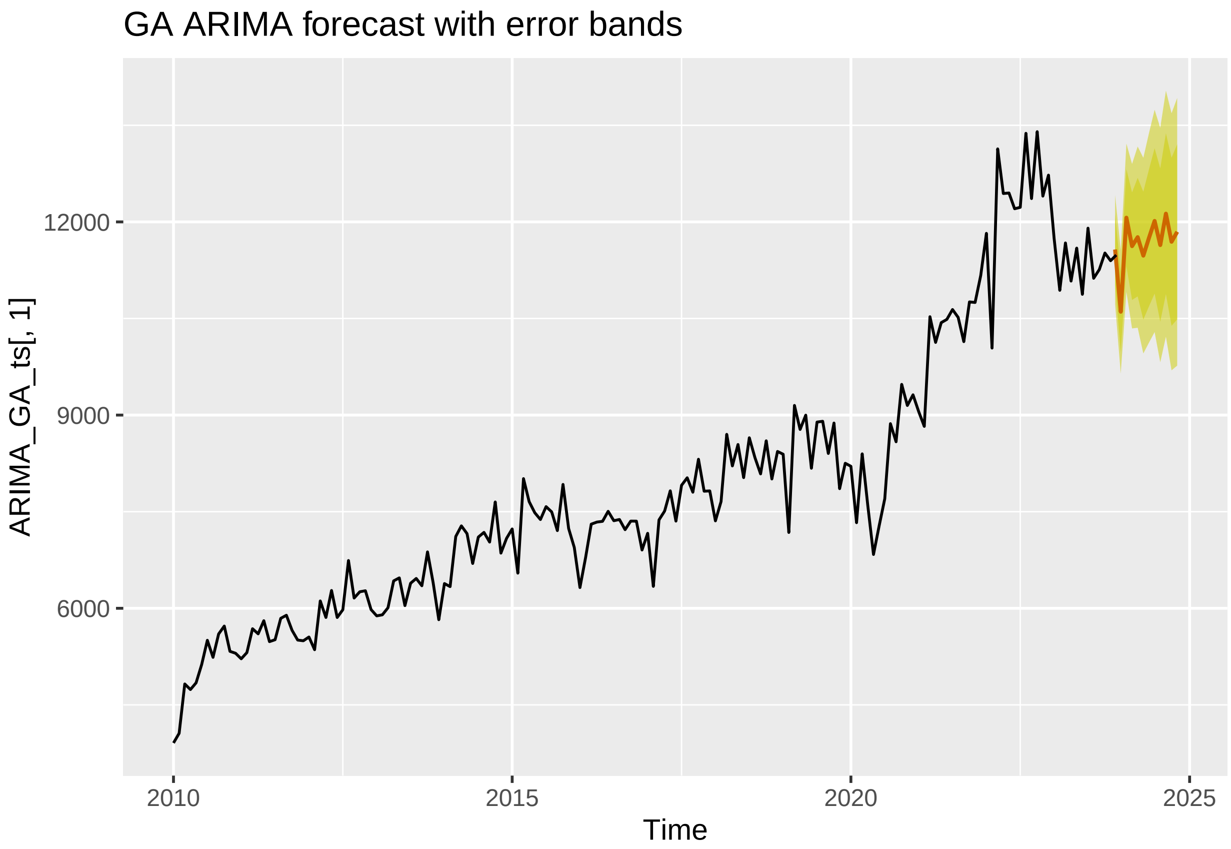
I. Compare your forecast from above to the 12-steps ahead forecasts from auto.arima model. Which model performs best in terms of MAPE?

```
# Georgia
# Initialize arima model and forecast
```

```
ARIMA_GA <- auto.arima(GA_ts)
ARIMA_GA_Sum <- summary(ARIMA_GA)
ARIMA_GA_forecast <-  forecast(ARIMA_GA, h = 12) %>%
  data.frame()
ARIMA_GA_ts <- ts(ARIMA_GA_forecast, start = c(2023.65, 4), freq = 12)
# Plot the forecast
ARIMA_GA_plot <- autoplot(ARIMA_GA_ts[,1], color = "darkorange3") +
  geom_ribbon(aes(ymin = ARIMA_GA_ts[,4], ymax = ARIMA_GA_ts[,5]),
              fill = "yellow3", alpha = 0.5) +
  geom_ribbon(aes(ymin = ARIMA_GA_ts[,2], ymax = ARIMA_GA_ts[,3]),
              fill = "yellow3", alpha = 0.5) +
  geom_line(aes(y = ARIMA_GA_ts[,1]), color = "darkorange3", lwd = 0.7) +
  labs(title = "GA ARIMA forecast with error bands") +
  geom_line(data = GA_ts) + xlim(2010, NA)
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```
ARIMA_GA_plot
```

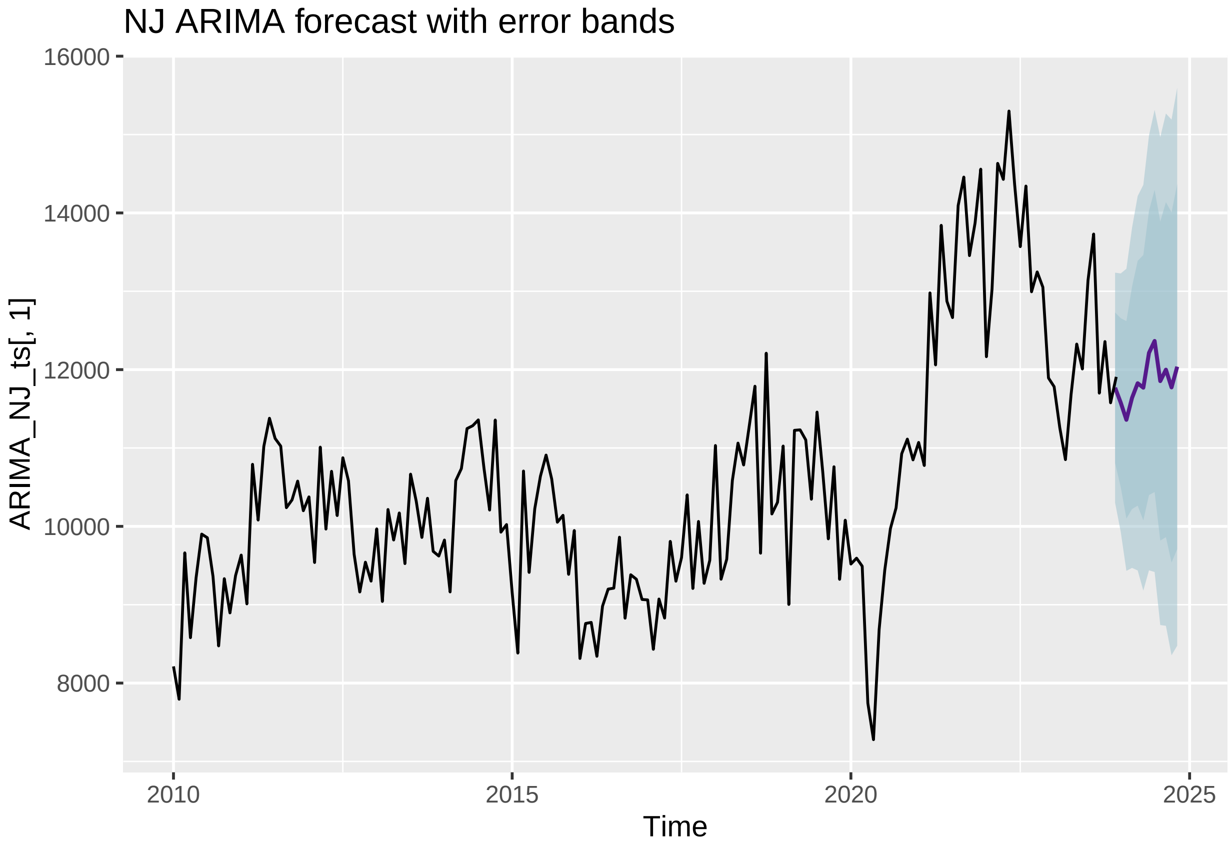


```
# New Jersey
# Initialize arima model and forecast
ARIMA_NJ <- auto.arima(NJ_ts)
ARIMA_NJ_Sum <- summary(ARIMA_NJ)
```

```
ARIMA_NJ_forecast <-  forecast(ARIMA_NJ, h = 12) %>%
  data.frame()
ARIMA_NJ_ts <- ts(ARIMA_NJ_forecast, start = c(2023.65, 4), freq = 12)
# Plot the forecast
ARIMA_NJ_plot <- autoplot(ARIMA_NJ_ts[,1], color = "purple4") +
  geom_ribbon(aes(ymin = ARIMA_NJ_ts[,4], ymax = ARIMA_NJ_ts[,5]),
              fill = "lightblue3", alpha = 0.5) +
  geom_ribbon(aes(ymin = ARIMA_NJ_ts[,2], ymax = ARIMA_NJ_ts[,3]),
              fill = "lightblue3", alpha = 0.5) +
  geom_line(aes(y = ARIMA_NJ_ts[,1]), color = "purple4", lwd = 0.7) +
  labs(title = "NJ ARIMA forecast with error bands") +
  geom_line(data = NJ_ts) + xlim(2010, NA)
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```
ARIMA_NJ_plot
```



```
# Extract the MAPE for both Georgia and New Jersey
GA_MAPE <- MAPE(.resid = ARMA_GA$resid, .actual = GA_ts)
NJ_MAPE <- MAPE(.resid = ARMA_NJ$resid, .actual = NJ_ts)
cat("GA ARMA MAPE:", GA_MAPE, "\n")
```

```
## GA ARMA MAPE: 3.488039
```

```
ARIMA_GA_Sum
```

```
## Series: GA_ts
## ARIMA(1,1,0)(2,0,0)[12]
##
## Coefficients:
##           ar1    sar1    sar2
##       -0.4617  0.2596  0.4051
## s.e.   0.0687  0.0720  0.0814
##
## sigma^2 = 186188:  log likelihood = -1252.22
## AIC=2512.44   AICc=2512.68   BIC=2524.91
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 17.23913 426.3269 328.4758 0.1245975 4.128612 0.4087959
##                      ACF1
## Training set -0.002364147
```

```
cat("NJ ARMA MAPE:", NJ_MAPE, "\n")
```

```
## NJ ARMA MAPE: 4.527922
```

```
ARIMA_NJ_Sum
```

```
## Series: NJ_ts
## ARIMA(0,1,2)(0,0,1)[12]
##
## Coefficients:
##           ma1     ma2    sma1
##       -0.4886  0.1683  0.3285
## s.e.   0.0742  0.0825  0.0695
##
## sigma^2 = 561080:  log likelihood = -1341.6
## AIC=2691.2   AICc=2691.45   BIC=2703.67
##
## Training set error measures:
##                    ME    RMSE      MAE         MPE     MAPE      MASE
## Training set 24.35153 740.082 589.9155 -0.06701548 5.633429 0.5116943
##                     ACF1
## Training set -0.01029137
```

Comparing the Georgia forecasts first, we can see that the Georgia ARIMA model has larger error bands (as seen in yellow) than the previous forecast, there is also a somewhat upwards prediction for the ARIMA forecast compared to the previous forecast that shows a mildly downwards forecast. The MAPE for the GA ARMA model is at 3.488, which is lower than the 4.129 seen in the ARIMA model. The ARIMA has larger magnitudes than the ARMA model as seen here and with the other variables in the summary set. The original model performs better than the ARIMA model for this reason.

Comparing the New Jersey forecasts after, we can see that the New Jersey ARIMA model has larger error bands (as seen in yellow) than the previous forecast, there is also a somewhat flat prediction for the ARIMA forecast compared to the previous forecast that shows a harsh downwards forecast. The MAPE for the NJ

ARMA model is at 4.528, which is lower than the 5.633 seen in the ARIMA model. The ARIMA has larger magnitudes than the ARMA model as seen here and with the other variables in the summary set. The original model performs better than the ARIMA model for this reason.

J. Combine the two forecasts and comment on the MAPE from these forecasts vs. the individual ones.

```
# Georgia
# Combine the arima and original forecasts and compare MAPE
GA_everything <- lm(GA_ts ~ GA_fitted + ARIMA_GA$fitted)
GA_every_resid <- GA_everything$resid
GA_MAPE_everything <- MAPE(.resid = GA_every_resid, .actual = GA_ts)
GA_ARIMA_MAPE <- MAPE(.resid = ARIMA_GA$resid, .actual = GA_ts)

# New Jersey
# Combine the arima and original forecasts and compare MAPE
NJ_everything <- lm(NJ_ts ~ NJ_fitted + ARIMA_NJ$fitted)
NJ_every_resid <- NJ_everything$resid
NJ_MAPE_everything <- MAPE(.resid = NJ_every_resid, .actual = NJ_ts)
NJ_ARIMA_MAPE <- MAPE(.resid = ARIMA_NJ$resid, .actual = NJ_ts)

# Display the MAPE values
cat("GA Combined MAPE:", GA_MAPE_everything ,"\n")
```

```
## GA Combined MAPE: 3.438691
```

```
cat("GA ARMA MAPE", GA_MAPE, "\n")
```

```
## GA ARMA MAPE 3.488039
```

```
cat("GA ARIMA MAPE:", GA_ARIMA_MAPE ,"\n")
```

```
## GA ARIMA MAPE: 4.128612
```

```
cat("NJ Combined MAPE:", NJ_MAPE_everything ,"\n")
```

```
## NJ Combined MAPE: 4.552528
```

```
cat("NJ ARMA MAPE", NJ_MAPE, "\n")
```

```
## NJ ARMA MAPE 4.527922
```

```
cat("NJ ARIMA MAPE:", NJ_ARIMA_MAPE ,"\n")
```

```
## NJ ARIMA MAPE: 5.633429
```

Looking at the Georgia data first, we can see that the MAPE for the combined model has the lowest result. The combined model and the ARMA model both, however, performed significantly better than the ARIMA model.

Looking at the New Jersey data second, we can see that the MAPE for the ARMA model is actually still the lowest result. The combined model is close to the ARMA model, and similar to Georgia, the ARIMA model is significantly higher than the combined or ARMA model in terms of MAPE.

K. Fit an appropriate VAR model using your two variables. Make sure to show the relevant plots and discuss your results from the fit.

```
library(vars)
# Bind the GA and NJ data and run Var select to extract the p
merged_data <- data.frame("NJ" = data1$IMPTOTNJ,"GA" = data2$IMPTOTGA)

VARselect(merged_data, lag.max = 10)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      6      2      2      6
##
## $criteria
##                   1            2            3            4            5
## AIC(n) 2.607649e+01 2.565706e+01 2.564889e+01 2.562005e+01 2.561387e+01
## HQ(n)  2.612372e+01 2.573578e+01 2.575910e+01 2.576174e+01 2.578705e+01
## SC(n)  2.619279e+01 2.585090e+01 2.592026e+01 2.596895e+01 2.604031e+01
## FPE(n) 2.112895e+11 1.389115e+11 1.377920e+11 1.338912e+11 1.330944e+11
##                   6            7            8            9           10
## AIC(n) 2.560935e+01 2.561784e+01 2.564183e+01 2.567318e+01 2.569513e+01
## HQ(n)  2.581402e+01 2.585400e+01 2.590947e+01 2.597231e+01 2.602575e+01
## SC(n)  2.611332e+01 2.619935e+01 2.630087e+01 2.640976e+01 2.650924e+01
## FPE(n) 1.325327e+11 1.337168e+11 1.370348e+11 1.414937e+11 1.447531e+11
```

```
# run model based on p = 6
var_model <- VAR(merged_data, p = 6)
summary(var_model)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: NJ, GA
## Deterministic variables: const
## Sample size: 162
## Log Likelihood: -2507.751
## Roots of the characteristic polynomial:
## 0.9934 0.8369 0.8341 0.8341 0.7325 0.7325 0.7013 0.6789 0.6789 0.6402 0.6319 0.6319
## Call:
## VAR(y = merged_data, p = 6)
##
##
## Estimation results for equation NJ:
## ===================================
## NJ = NJ.l1 + GA.l1 + NJ.l2 + GA.l2 + NJ.l3 + GA.l3 + NJ.l4 + GA.l4 + NJ.l5 + GA.l5 + NJ.l6 + GA.l6 +
##
##         Estimate Std. Error t value Pr(>|t|)
## NJ.l1  4.307e-01  9.393e-02   4.585 9.54e-06 ***
## GA.l1  8.672e-02  1.411e-01   0.615  0.53963
## NJ.l2  3.172e-01  1.008e-01   3.147  0.00199 **
## GA.l2  2.300e-01  1.448e-01   1.588  0.11440
## NJ.l3  1.584e-01  1.025e-01   1.546  0.12418
## GA.l3 -5.393e-02  1.528e-01  -0.353  0.72469
## NJ.l4 -1.645e-01  1.021e-01  -1.610  0.10950
## GA.l4 -1.139e-01  1.527e-01  -0.746  0.45696
## NJ.l5 -2.531e-05  9.974e-02   0.000  0.99980
## GA.l5  2.129e-01  1.475e-01   1.444  0.15093
## NJ.l6  4.877e-02  9.284e-02   0.525  0.60018
## GA.l6 -2.734e-01  1.420e-01  -1.925  0.05609 .
## const  1.500e+03  5.258e+02   2.854  0.00494 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 766.2 on 149 degrees of freedom
## Multiple R-Squared: 0.7743,  Adjusted R-squared: 0.7562
## F-statistic: 42.61 on 12 and 149 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation GA:
## ===================================
## GA = NJ.l1 + GA.l1 + NJ.l2 + GA.l2 + NJ.l3 + GA.l3 + NJ.l4 + GA.l4 + NJ.l5 + GA.l5 + NJ.l6 + GA.l6 +
##
##         Estimate Std. Error t value Pr(>|t|)
## NJ.l1  9.649e-02  6.236e-02   1.547 0.123909
## GA.l1  3.416e-01  9.365e-02   3.647 0.000366 ***
## NJ.l2 -8.269e-03  6.693e-02  -0.124 0.901840
## GA.l2  4.079e-01  9.616e-02   4.242 3.87e-05 ***
## NJ.l3  9.248e-03  6.802e-02   0.136 0.892042
## GA.l3  5.318e-02  1.015e-01   0.524 0.601008
## NJ.l4 -1.113e-02  6.781e-02  -0.164 0.869823
## GA.l4 -2.285e-03  1.014e-01  -0.023 0.982051
## NJ.l5  9.745e-04  6.622e-02   0.015 0.988278
## GA.l5  2.777e-01  9.791e-02   2.837 0.005194 **
## NJ.l6 -7.675e-02  6.164e-02  -1.245 0.215007
## GA.l6 -9.556e-02  9.429e-02  -1.013 0.312474
## const  1.038e+02  3.491e+02   0.297 0.766714
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 508.7 on 149 degrees of freedom
## Multiple R-Squared: 0.9443,  Adjusted R-squared: 0.9398
## F-statistic: 210.5 on 12 and 149 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##        NJ     GA
## NJ 587021 196980
## GA 196980 258746
##
## Correlation matrix of residuals:
##        NJ     GA
## NJ 1.0000 0.5054
## GA 0.5054 1.0000
```

Order 6 seems to be the best option based on the AIC score (it converges to 6 as well, which is a good sign)

When analyzing the results of NJ, we see that only NJ lag 1 and lag 2 along with the constant are significant in predicting NJ's imports. For the results of GA, we see that only GA lag 1, 2, and 5 are significant in predicting GA's imports (There is a very good adjusted R squared of 93.98% for GA). Another good metric to look at here is the correlation matrix of residuals. Here we see that there a 50% correlation of residuals between NJ and GA, implying there is some correlation, but it's not very strong.

```r
# Plot the var model
plot(var_model)
```

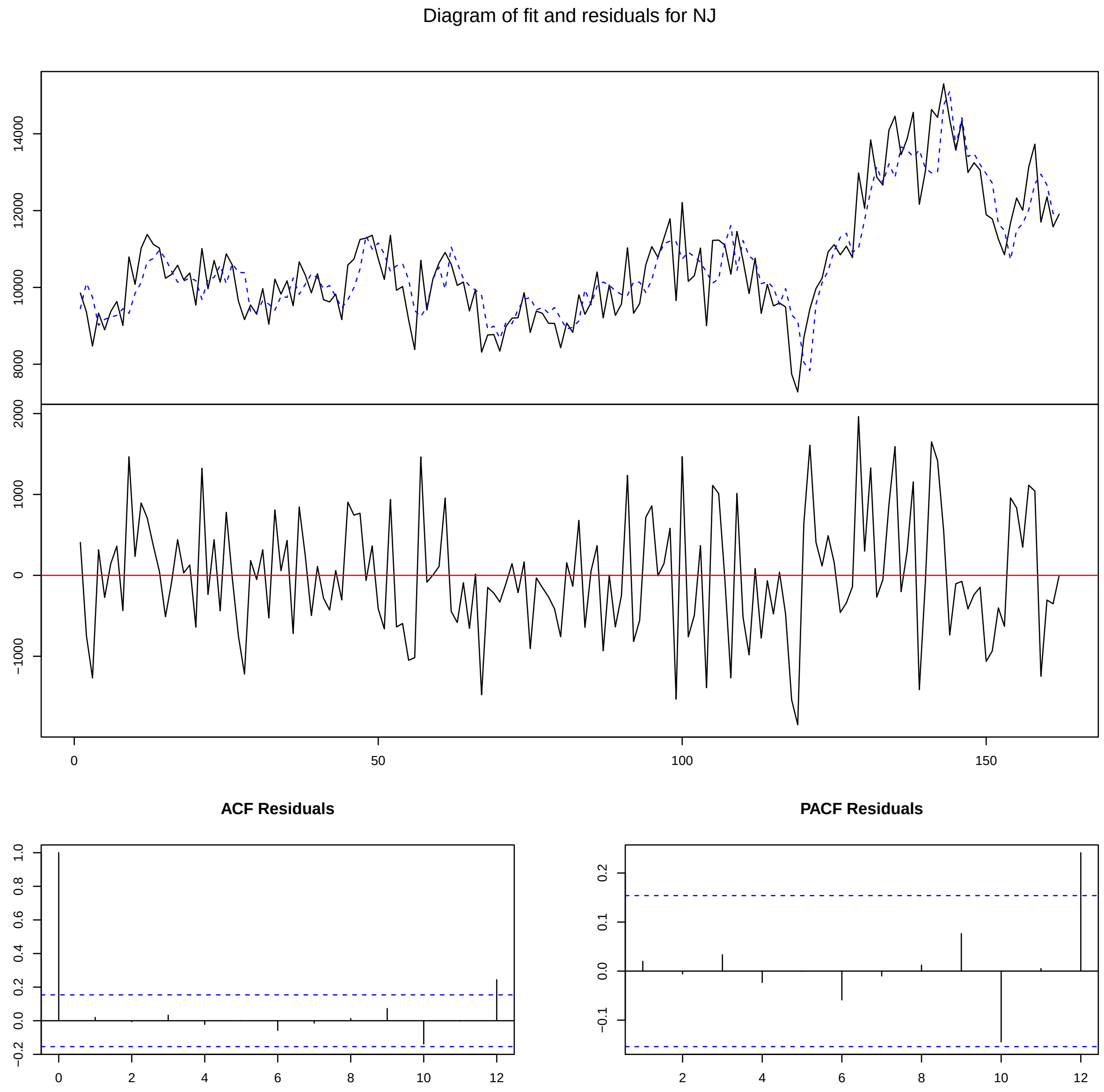
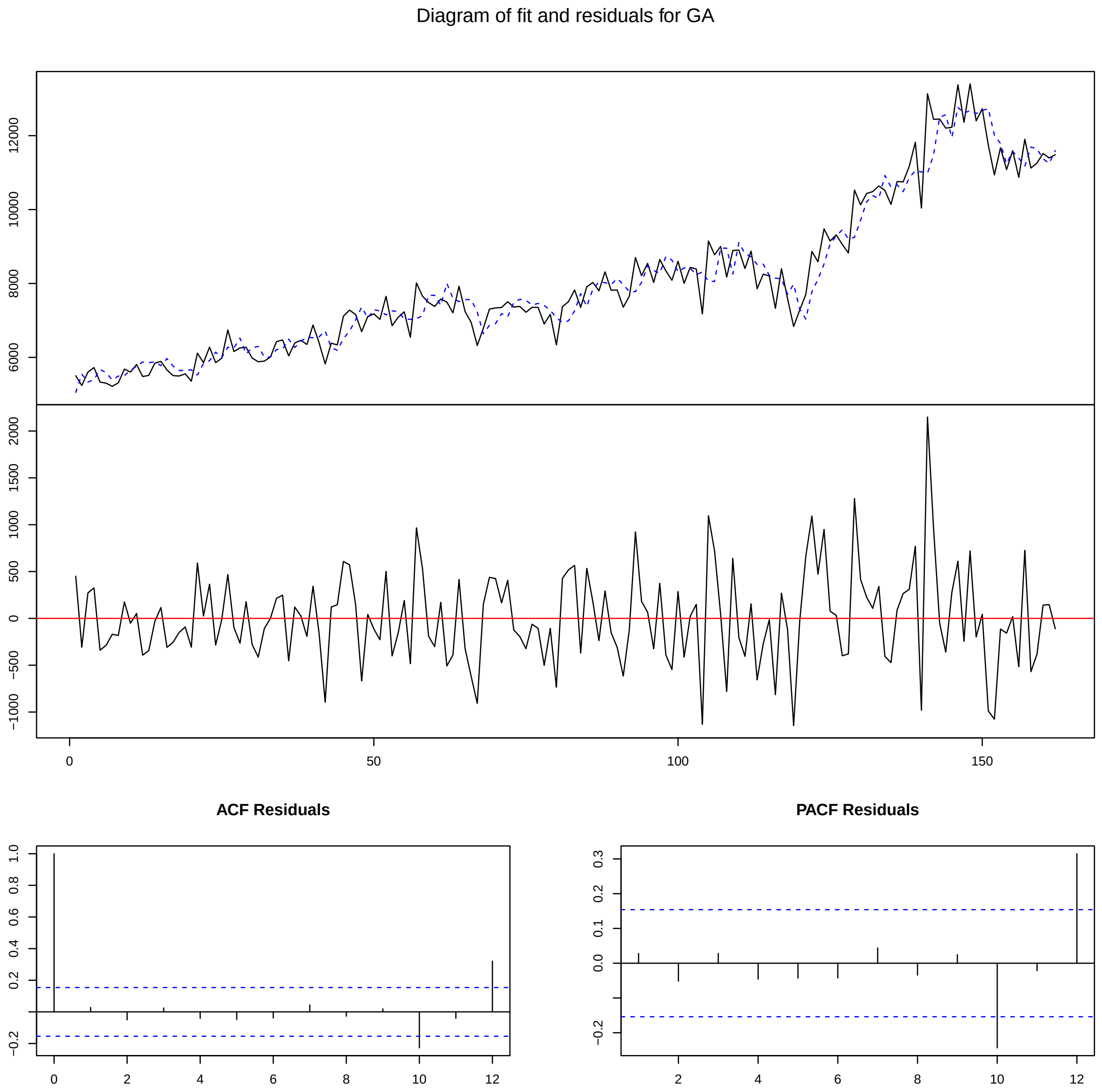


Diagram of fit and residuals for NJ

ACF Residuals

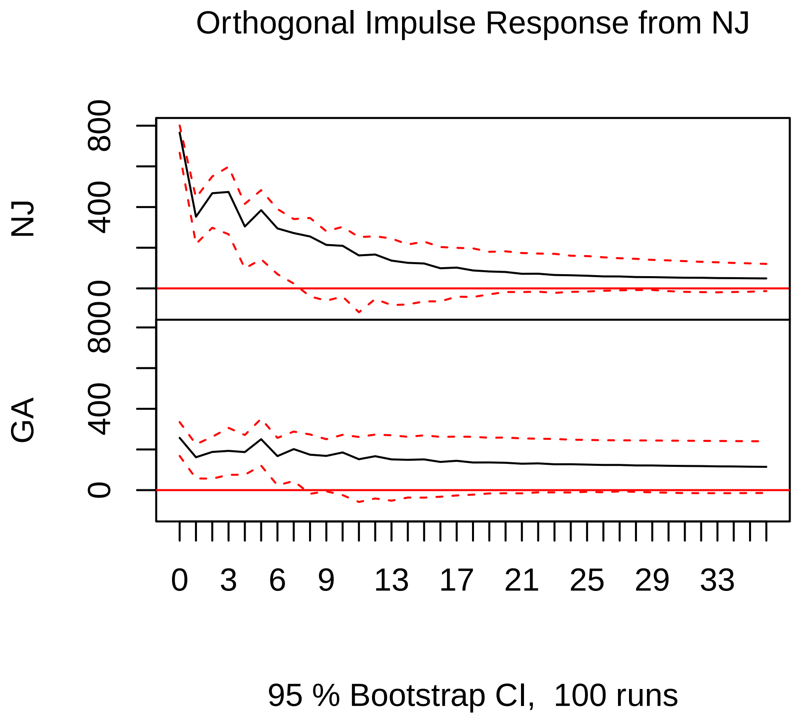
PACF Residuals

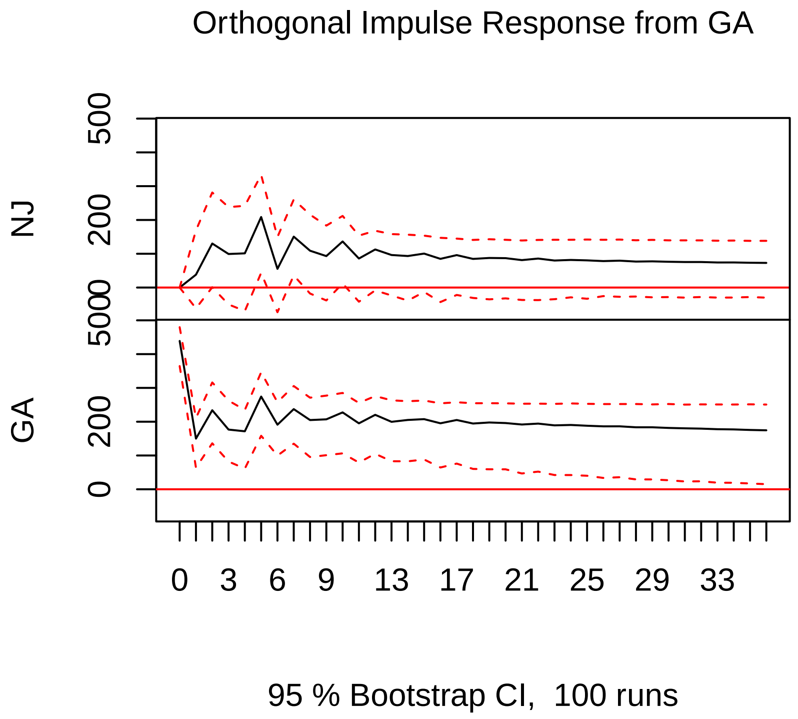


When looking at the top graph for both variables, we see that the plots are fairly similar to their projections. Also, when we look at the residuals, we see that they are mean reverting and don't seem to have any particular pattern. For the ACF residuals, there are no significant spikes in NJ, although there is a significant spike in the PACF residuals at Lag 12. For GA, there are two spikes in the ACF Residuals, but they barely cross the line, so it's not really noteworthy. However, for the PACF residuals, we do see significant spikes at 10 and 12.

L. Compute, plot, and interpret the respective impulse response functions.

```
# Compute and plot the impulse response function from the var model
plot(irf(var_model, n.ahead=36))
```


Orthogonal Impulse Response from NJ
NJ
800
400
0
GA
8000
400
0
0
3
6
9
13
17
21
25
29
33
95 % Bootstrap CI, 100 runs



When looking at the IRF plot for NJ, we see that NJ has a high impact on NJ (this is to be expected), while NJ doesn't really have that big of an impact on GA. It is interesting to note that the further we go along the IRF plot, we see that NJ has a slightly higher impact on GA than it does on itself.

When looking at the IRF plot for GA, we see that early on, it has an noticeable impact on NJ, and as time goes on, it doesn't really have an impact on NJ. As expected, GA consistently has a high impact on GA.

M. Perform a Granger-Causality test on your variables and discuss your results from the test

```
# Perform the granger causality test for both NJ ~ GA and GA ~ NJ
grangertest(NJ ~ GA, data = merged_data, order = 6)
```

```
## Granger causality test
##
## Model 1: NJ ~ Lags(NJ, 1:6) + Lags(GA, 1:6)
## Model 2: NJ ~ Lags(NJ, 1:6)
##   Res.Df Df      F Pr(>F)
## 1    149
## 2    155 -6 1.6977 0.1253
```

```
grangertest(GA ~ NJ, data = merged_data, order = 6)
```

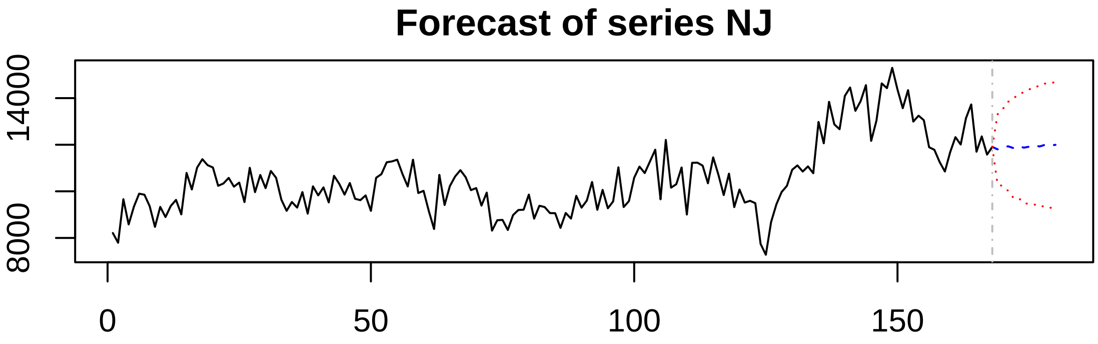
```
## Granger causality test
```

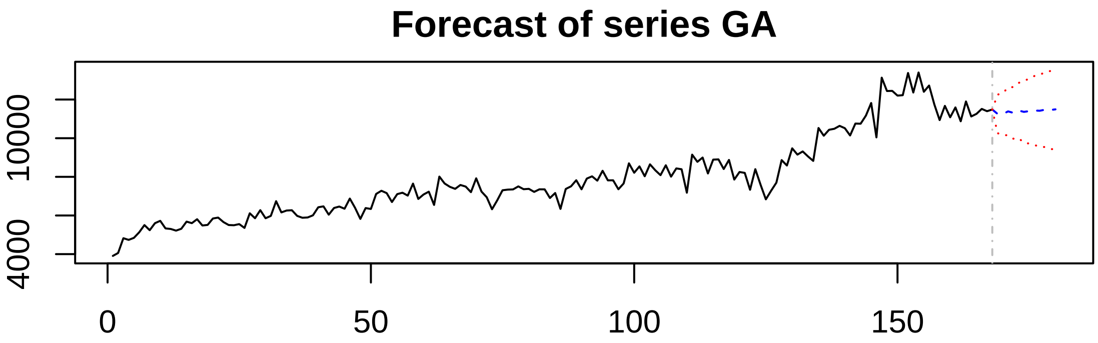
```
##
## Model 1: GA ~ Lags(GA, 1:6) + Lags(NJ, 1:6)
## Model 2: GA ~ Lags(GA, 1:6)
##   Res.Df Df      F Pr(>F)
## 1    149
## 2    155 -6 0.7765 0.5896
```

NJ and GA do not Granger cause each other.

N. Use your VAR model to forecast 12-steps ahead. Your forecast should include the respective error bands. Comment on the differences between the VAR forecast and the other ones obtained using the different methods.

```r
# Compute and plot the 12 step ahead forecast using the var model
library(forecast)
forecast <- predict(var_model, n.ahead = 12)
par(mar = c(4, 4, 2, 2))
plot(forecast)
```





When comparing the forecasts of the VAR model vs the STL model, we see that in both states, the VAR models displays a general upward trend while the STL model displays a general downward trend. This could be due to the lags used or how different models measure different variables. When comparing VAR vs ARMA, we see that they generally seem to give pretty similar forecasts. However, the bands for ARMA seem to be a lot bigger compared to VAR.

Conclusions from the project

Overall, what we did was use multiple models to model original data and forecast two monthly time series datasets, monthly imports from Georgia and New Jersey from January of 2010 to December of 2023. We first fit our own model using a quadratic trend, seasonal dummies and an ARMA model for the cycles, and used that to forecast the data for both states. Then we used auto arima, and lastly a VAR model to come up with a forecast. We also combined both our fitted model to auto arima's model in one comprehensive forecast as well.

When comparing these models, purely comparing the original model with arima, the original model was better for both states we observed. However when we combined the models into one, the combined model worked better for Georgia however the original model still was the best for New Jersey. In the end, in terms of the VAR model, we observed our VAR model gave very similar results to our original model and forecast, but slightly different results to arima's forecast. In terms of Future Work, we could generalize this analysis to other states in the US as well, maybe states that have higher levels of imports, or have some discrepancies in the states we choose. Potentially choosing a state with high imports and a state with low imports and running analysis on how these new circumstances potentially could change the modeling and forecasting. Overall, there is lots more analysis that can be run, as state imports play a big part economically in the world today.

References

Georgia time series data source: https://fred.stlouisfed.org/series/IMPTOTGA

New Jersey time series data source: https://fred.stlouisfed.org/series/IMPTOTNJ