

VAR analysis - Granger-Causality-Initial Comparisons

Katie Schilling - 501130072

07/02/2022

Required tools to be loaded

```
library(dlookr)
```

```
##  
## Attaching package: 'dlookr'  
  
## The following object is masked from 'package:base':  
##  
##     transform
```

```
library(lubridate)
```

```
##  
## Attaching package: 'lubridate'  
  
## The following objects are masked from 'package:base':  
##  
##     date, intersect, setdiff, union
```

```
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(ggpubr)
```

```
## Loading required package: ggplot2
```

```

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:ggpubr':
## 
##     gghistogram

library(funModeling)

## Loading required package: Hmisc

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
## 
##     src, summarize

## The following object is masked from 'package:dlookr':
## 
##     describe

## The following objects are masked from 'package:base':
## 
##     format.pval, units

## funModeling v.1.9.4 :)
## Examples and tutorials at livebook.datascienceheroes.com
## / Now in Spanish: librovivodecienciadedatos.ai

require(ggfortify)

## Loading required package: ggfortify

```

```

## Registered S3 methods overwritten by 'ggfortify':
##   method           from
##   autoplot.Arima    forecast
##   autoplot.acf     forecast
##   autoplot.ar      forecast
##   autoplot.bats    forecast
##   autoplot.decomposed.ts forecast
##   autoplot.ets     forecast
##   autoplot.forecast forecast
##   autoplot.stl     forecast
##   autoplot.ts      forecast
##   fitted.ar       forecast
##   fortify.ts      forecast
##   residuals.ar    forecast

require(tseries)

## Loading required package: tseries

require(MTS)

## Loading required package: MTS

require(vars)

## Loading required package: vars

## Loading required package: MASS

##
## Attaching package: 'MASS'

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

## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

## Loading required package: sandwich

```

```

## Loading required package: urca

## Loading required package: lmtest

##
## Attaching package: 'vars'

## The following object is masked from 'package:MTS':
##      VAR

## The following object is masked from 'package:dlookr':
##      normality

require(fUnitRoots)

## Loading required package: fUnitRoots

## Loading required package: timeDate

##
## Attaching package: 'timeDate'

## The following objects are masked from 'package:dlookr':
##      kurtosis, skewness

## Loading required package: timeSeries

##
## Attaching package: 'timeSeries'

## The following object is masked from 'package:zoo':
##      time<-

## Loading required package: fBasics

##
## Attaching package: 'fUnitRoots'

## The following objects are masked from 'package:urca':
##      punitroot, qunitroot, unitrootTable

require(lattice)

```

Import both Clean datasets

```
Covid_monthly <- read.csv("C:/Users/Katie Schilling/Downloads/covid_monthly_clean.csv")
Vital_Events <- read.csv("C:/Users/Katie Schilling/Downloads/vital_events_clean.csv")
```

Combine the vital events data with the Covid Monthly data

```
Final_dataset <- merge(x=Vital_Events, y=Covid_monthly, all = TRUE)
```

Check the data and ensure data merged properly

```
summary(Final_dataset)
```

```
##      Date        Births      Marriages      Deaths
##  Length:336      Min.   :10020      Min.   : 597      Min.   : 5926
##  Class :character 1st Qu.:11260      1st Qu.: 2596     1st Qu.: 6706
##  Mode  :character Median :11818      Median : 3559     Median : 7326
##                               Mean   :11763      Mean   : 5085     Mean   : 7500
##                               3rd Qu.:12288      3rd Qu.: 7627     3rd Qu.: 8094
##                               Max.   :13398      Max.   :11532     Max.   :11390
##                               NA's    :6          NA's    :6          NA's    :6
##      Stillbirths      Covid
##  Min.   : 0.00      Min.   : 11
##  1st Qu.: 73.00      1st Qu.:22889
##  Median : 90.50      Median :75935
##  Mean   : 90.74      Mean   :95144
##  3rd Qu.:114.00      3rd Qu.:134128
##  Max.   :156.00      Max.   :395815
##  NA's    :6          NA's    :313
```

Change the N/A in the Covid Positive Cases to 0 so that the data is not omitted from the predictions

```
Final_dataset$Covid[is.na(Final_dataset$Covid)] = 0
```

See if there are anymore NA's in the dataset

```
Final_dataset %>% filter_all(any_vars(is.na(.)))
```

```
##      Date Births Marriages Deaths Stillbirths Covid
## 1 2021-07-1     NA      NA     NA       NA 15968
## 2 2021-08-1     NA      NA     NA       NA 67913
## 3 2021-09-1     NA      NA     NA       NA 125560
## 4 2021-10-1     NA      NA     NA       NA 91834
## 5 2021-11-1     NA      NA     NA       NA 75935
## 6 2021-12-1     NA      NA     NA       NA 395815
```

Remove rows with NA as they will skew the results

```
Final_dataset <- na.omit(Final_dataset)
```

Check for NA's to confirm all have been removed

```
Final_dataset %>% filter_all(any_vars(is.na(.)))
```

```
## [1] Date         Births       Marriages    Deaths       Stillbirths Covid
## <0 rows> (or 0-length row.names)
```

```
summary(Final_dataset)
```

```

##          Date           Births        Marriages       Deaths
## Length:330      Min. :10020      Min. : 597      Min. : 5926
## Class :character 1st Qu.:11260     1st Qu.: 2596    1st Qu.: 6706
## Mode  :character   Median :11818      Median : 3559     Median : 7326
##                           Mean  :11763      Mean  : 5085     Mean  : 7500
##                           3rd Qu.:12288     3rd Qu.: 7627    3rd Qu.: 8094
##                           Max. :13398      Max. :11532     Max. :11390
## Stillbirths        Covid
## Min.   : 0.00      Min.   : 0
## 1st Qu.: 73.00     1st Qu.: 0
## Median : 90.50     Median : 0
## Mean   : 90.74     Mean   : 4289
## 3rd Qu.:114.00     3rd Qu.: 0
## Max.   :156.00     Max.   :237308

```

```
Final_dataset$Date <- as.Date(Final_dataset$Date, "%Y-%m-%d")
```

```
glimpse(Final_dataset)
```

Normalize the Data

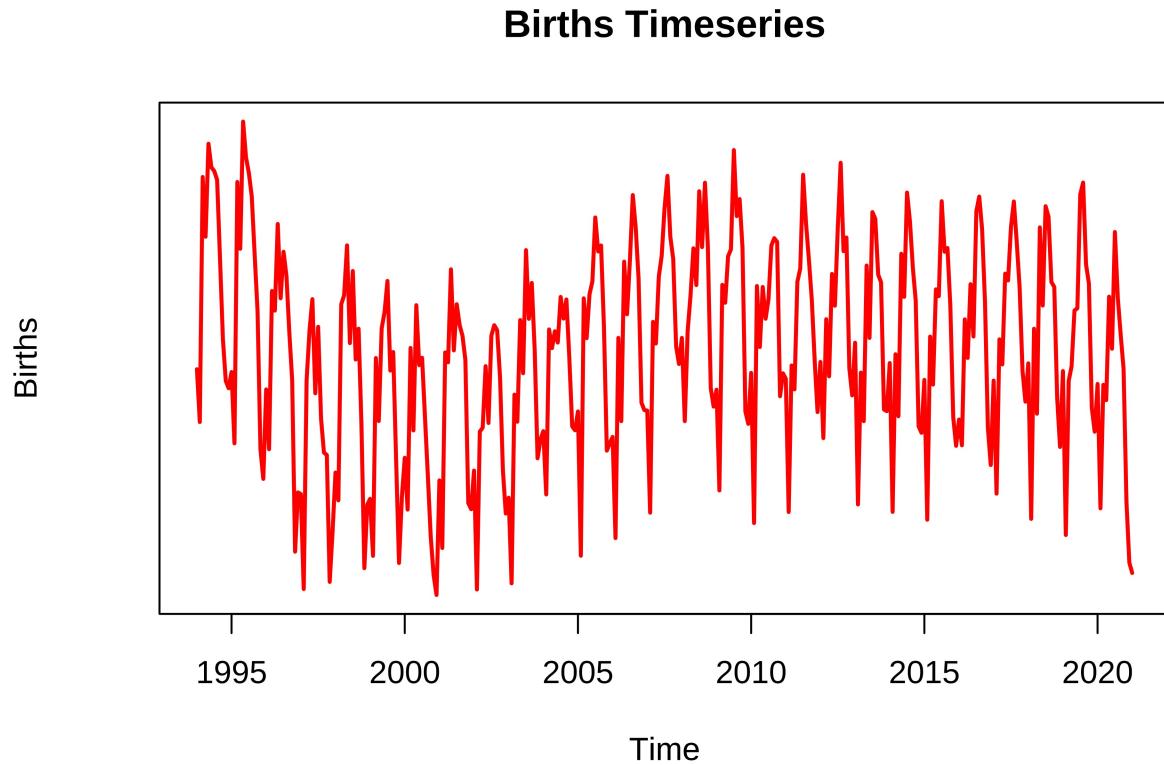
```
Final_dataset_standardized <- Final_dataset %>% mutate_each_(list(~scale(.) %>% as.vector),
                                         vars = c("Births", "Marriages", "Deaths", "Stillbirths", "Covid"))

## Warning: `mutate_each_()` was deprecated in dplyr 0.7.0.
## Please use `across()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

Convert data frame to a time series

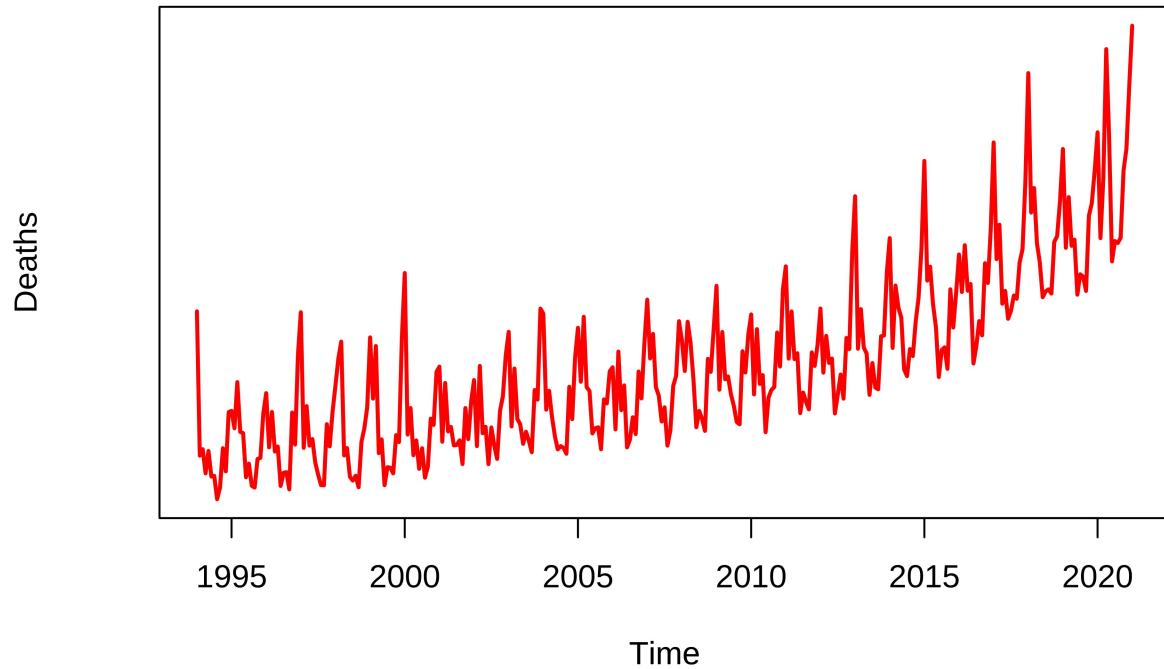
```
Final_dataset_TS <- ts(Final_dataset_standardized[2:6], frequency = 12, start = 1994, end = 2021)
```

```
plot(Final_dataset_TS[,1], ylab="", yaxt="n", col = "red", lwd = 2)
title(main = "Births Timeseries", xlab = "Time", ylab = "Births")
```



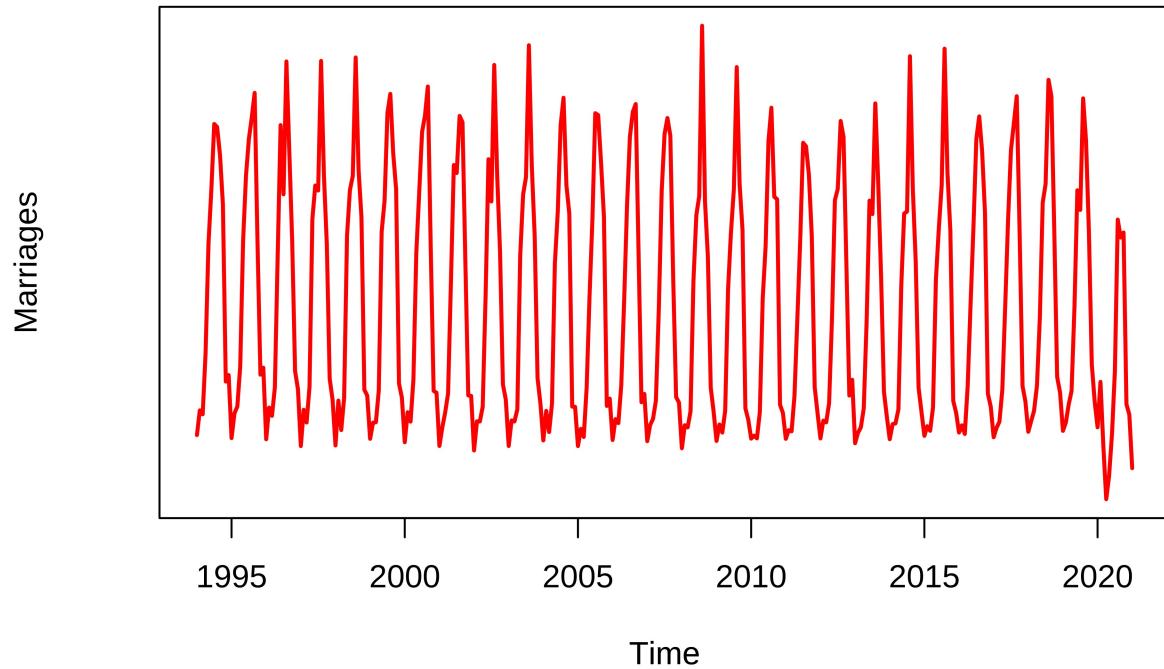
```
plot(Final_dataset_TS[,3], ylab="", yaxt="n", col = "red", lwd = 2)
title(main = "Deaths Timeseries", xlab = "Time", ylab = "Deaths")
```

Deaths Timeseries



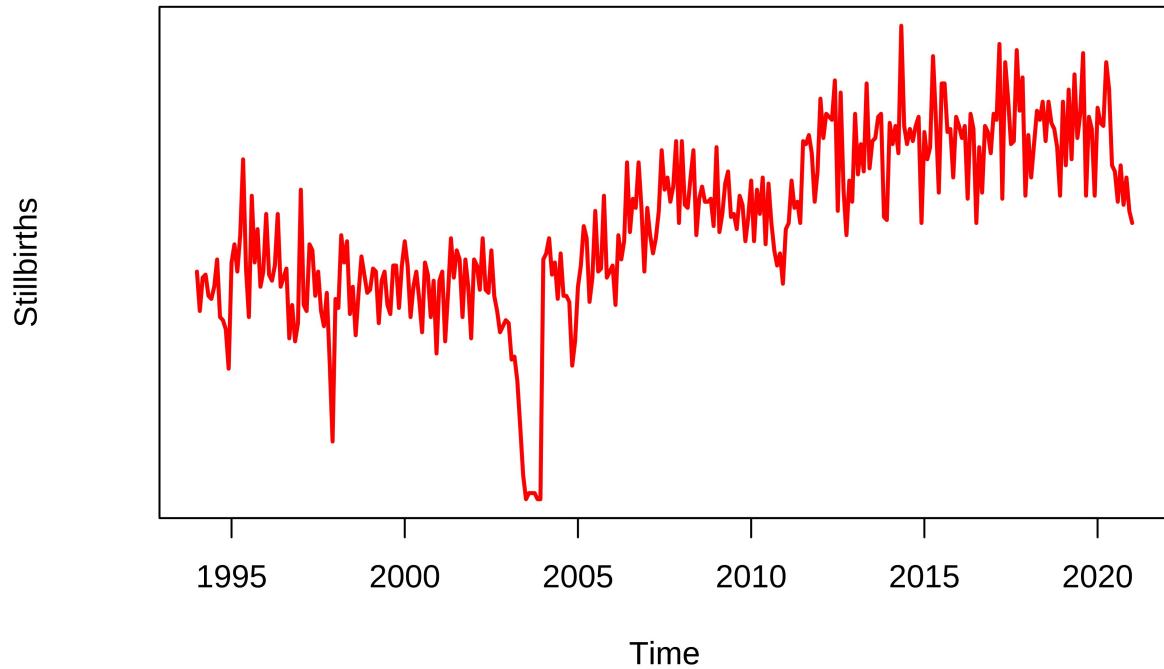
```
plot(Final_dataset_TS[,2], ylab="", yaxt="n", col = "red", lwd = 2)
title(main = "Marriages Timeseries", xlab = "Time", ylab = "Marriages")
```

Marriages Timeseries



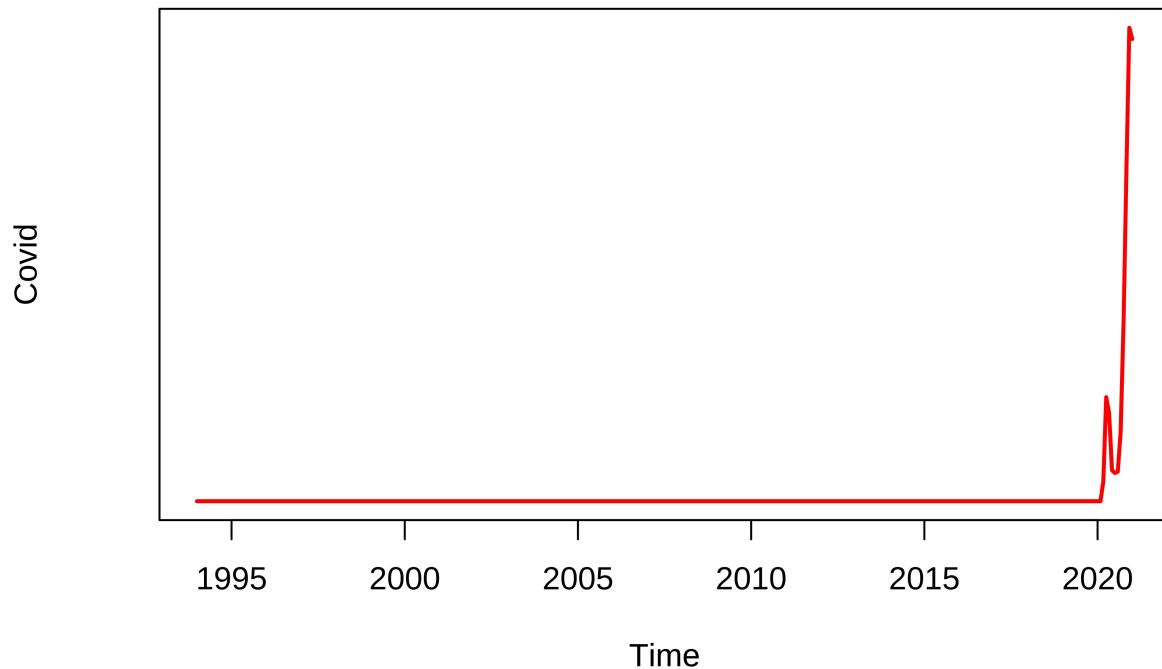
```
plot(Final_dataset_TS[,4], ylab="", yaxt="n", col = "red", lwd = 2)
title(main = "Stillbirths Timeseries", xlab = "Time", ylab = "Stillbirths")
```

Stillbirths Timeseries



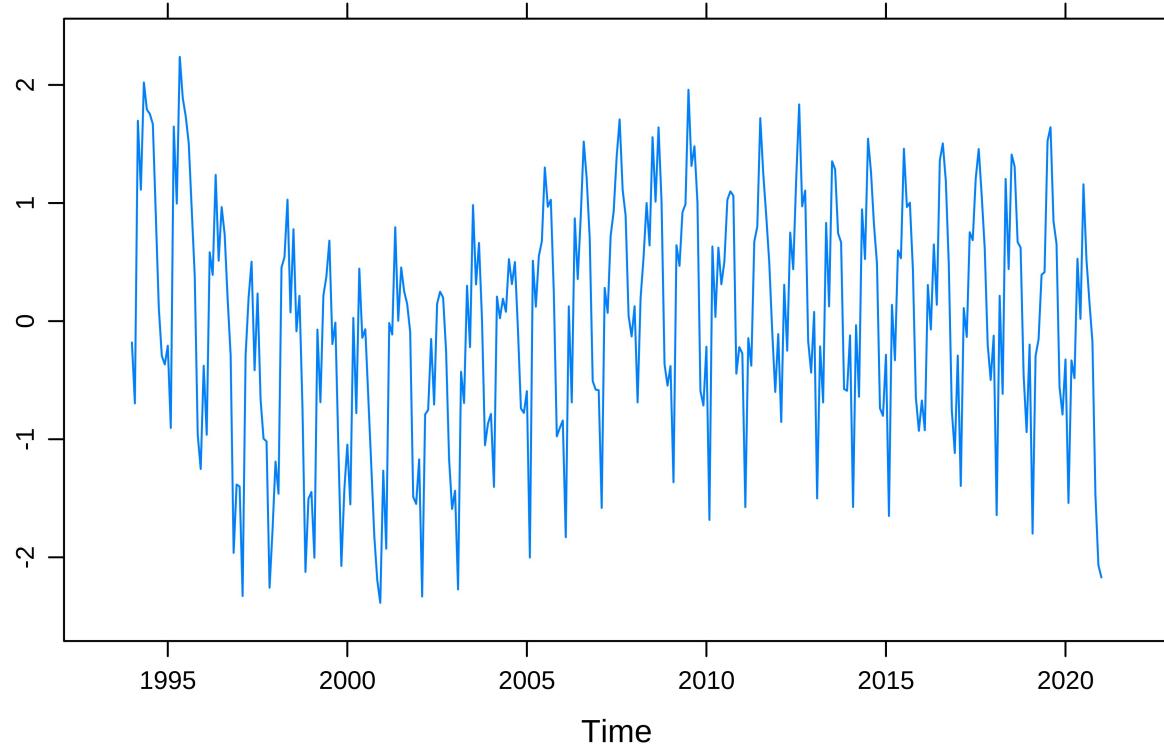
```
plot(Final_dataset_TS[,5], ylab="", yaxt="n", col = "red", lwd = 2)
title(main = "Covid Timeseries", xlab = "Time", ylab = "Covid")
```

Covid Timeseries



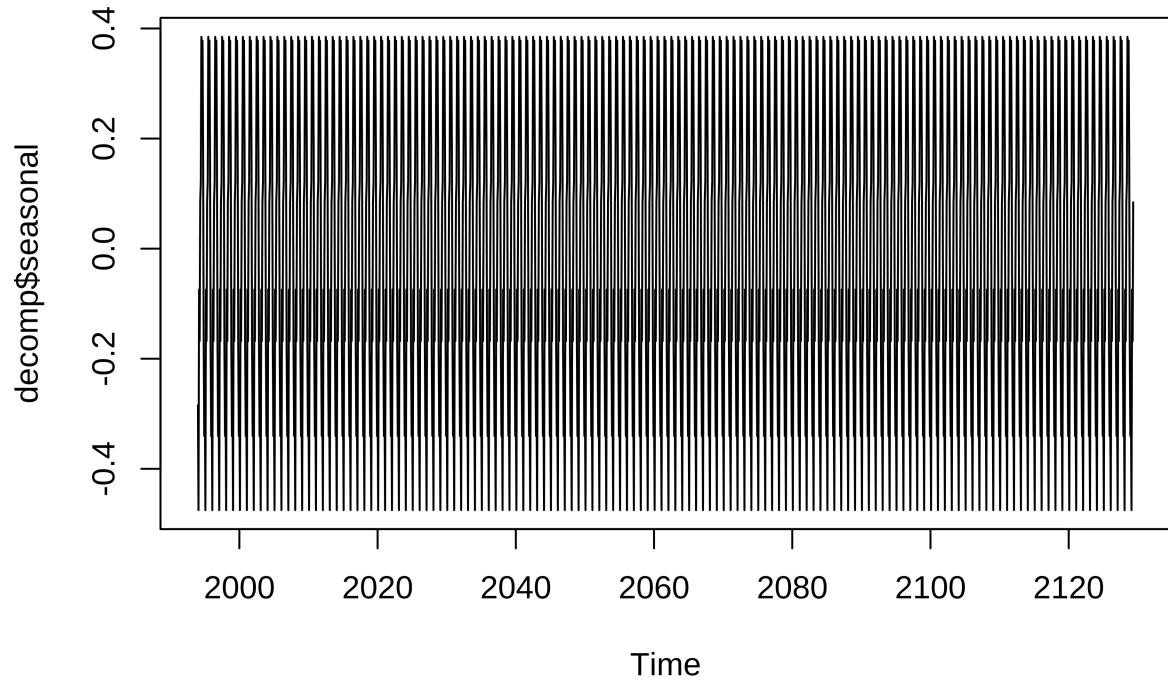
```
##ts.plot(Final_dataset_TS)
```

```
xyplot.ts(Final_dataset_TS[,1])
```



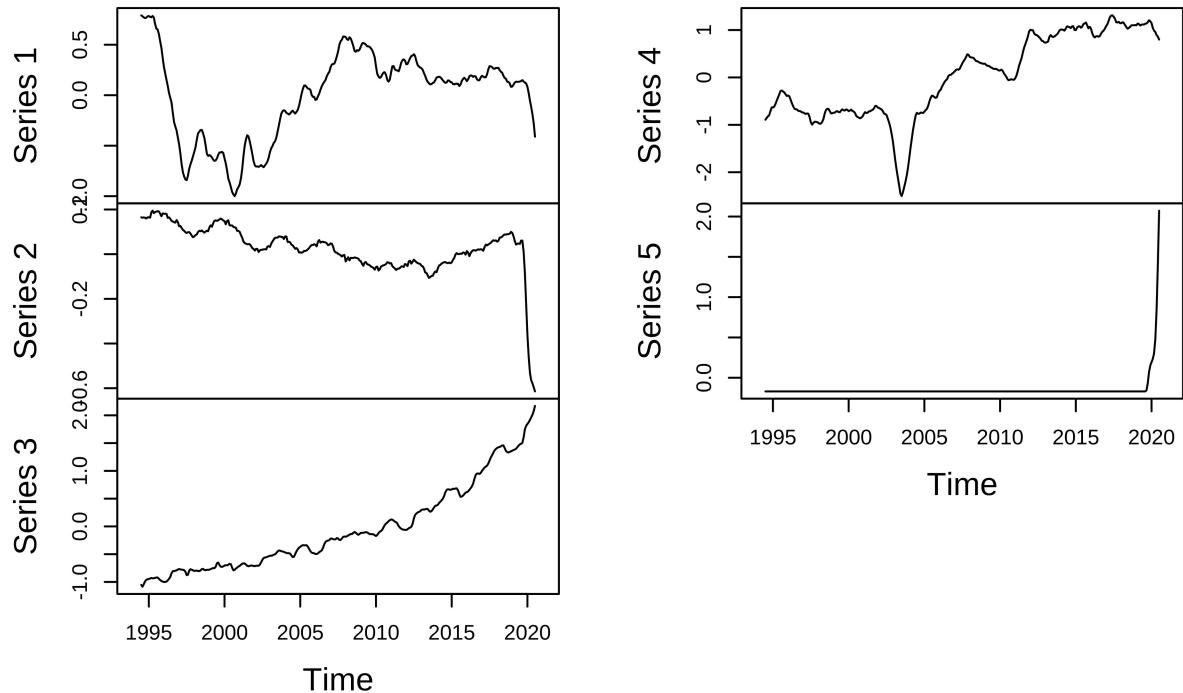
decompose data and display results

```
decomp <- decompose(Final_dataset_TS)
plot(decomp$seasonal)
```



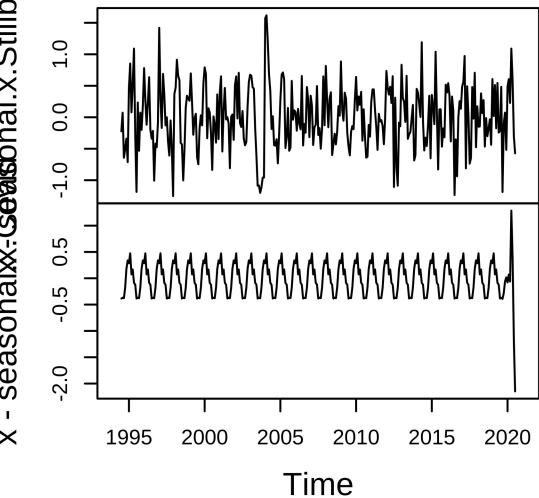
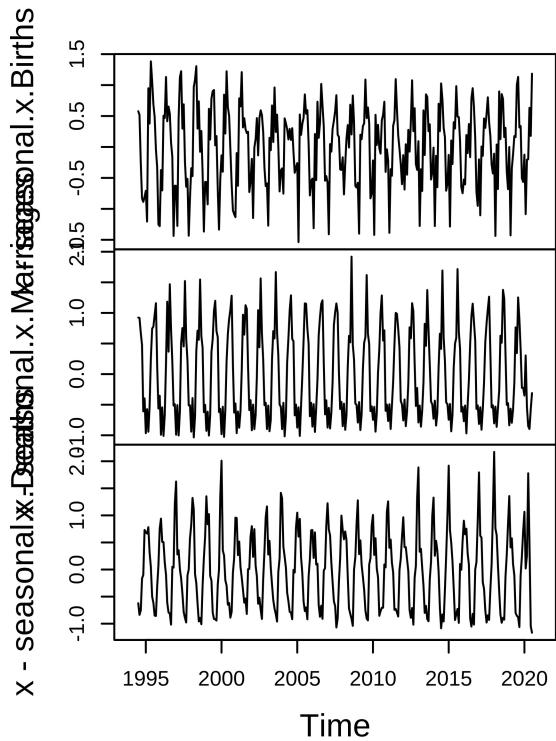
```
plot(decomp$trend)
```

decomp\$trend



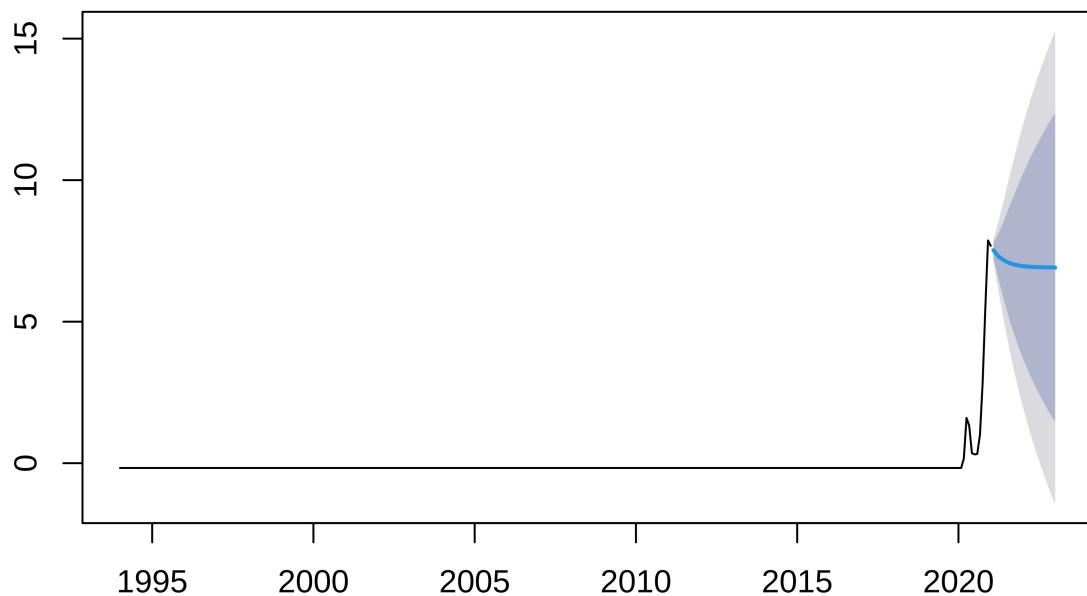
```
plot(decomp$random)
```

decomp\$random



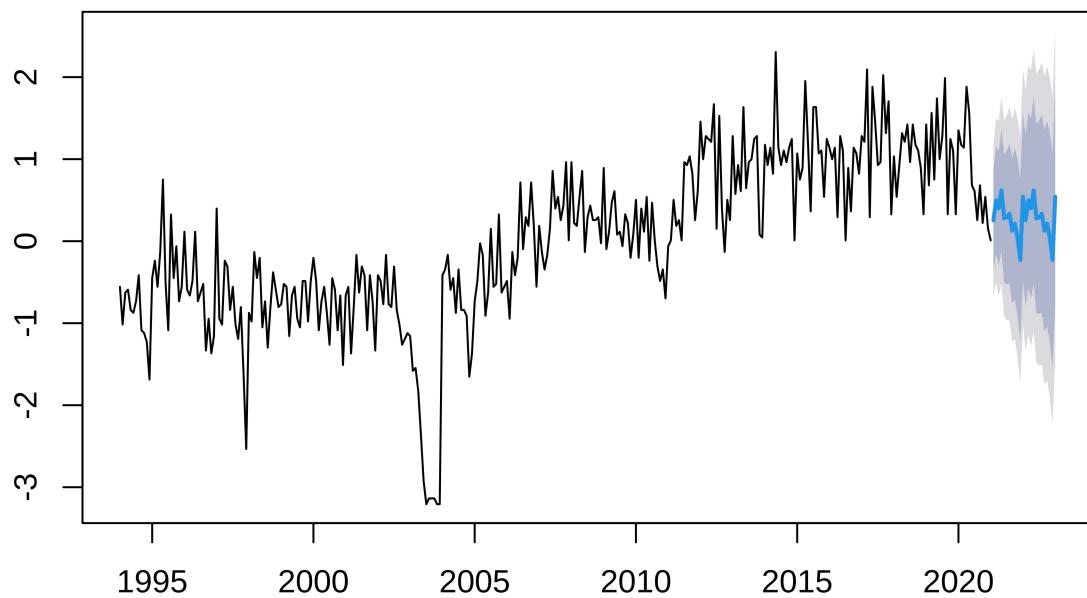
```
Covid <- forecast(Final_dataset_TS[,5])
plot(Covid, main = "Covid Forecast")
```

Covid Forecast



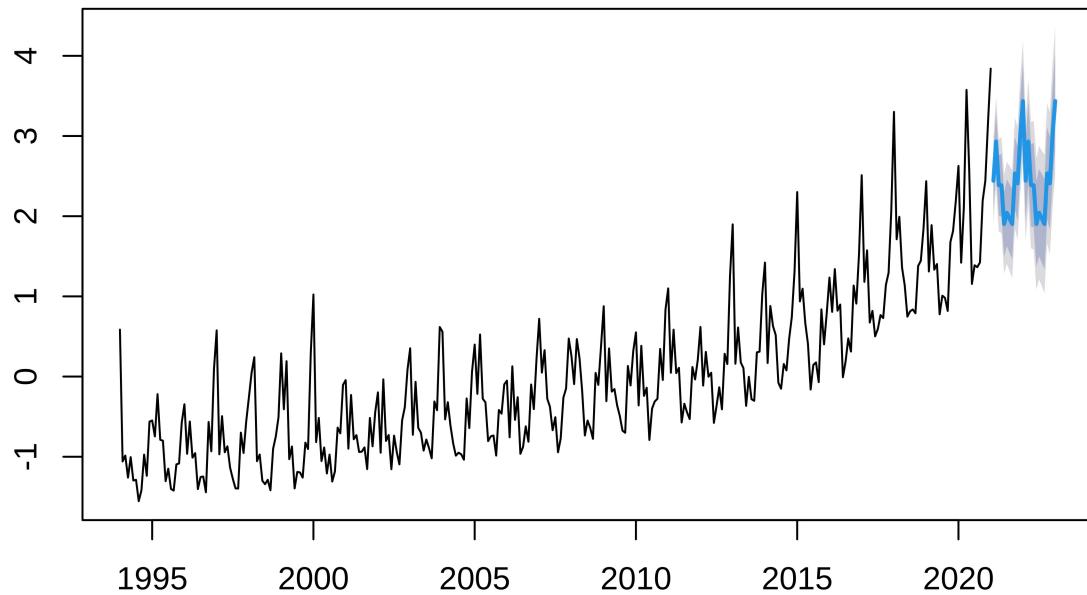
```
Stillbirths <- forecast(Final_dataset_TS[,4])
plot(Stillbirths, main = "Stillbirths Forecast")
```

Stillbirths Forecast



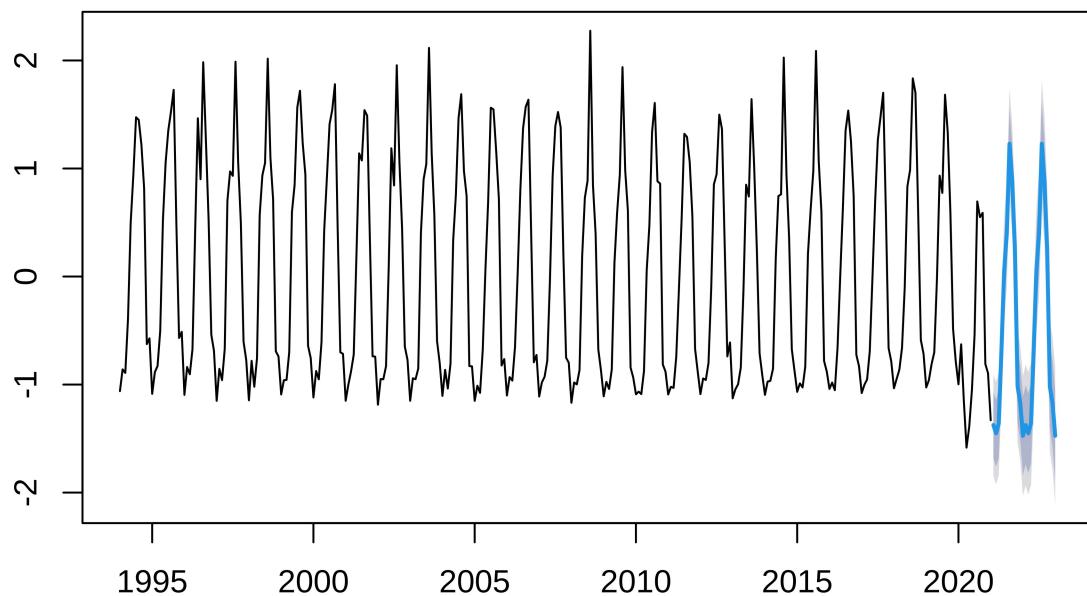
```
Deaths <- forecast(Final_dataset_TS[,3])
plot(Deaths, main = "Deaths Forecast")
```

Deaths Forecast



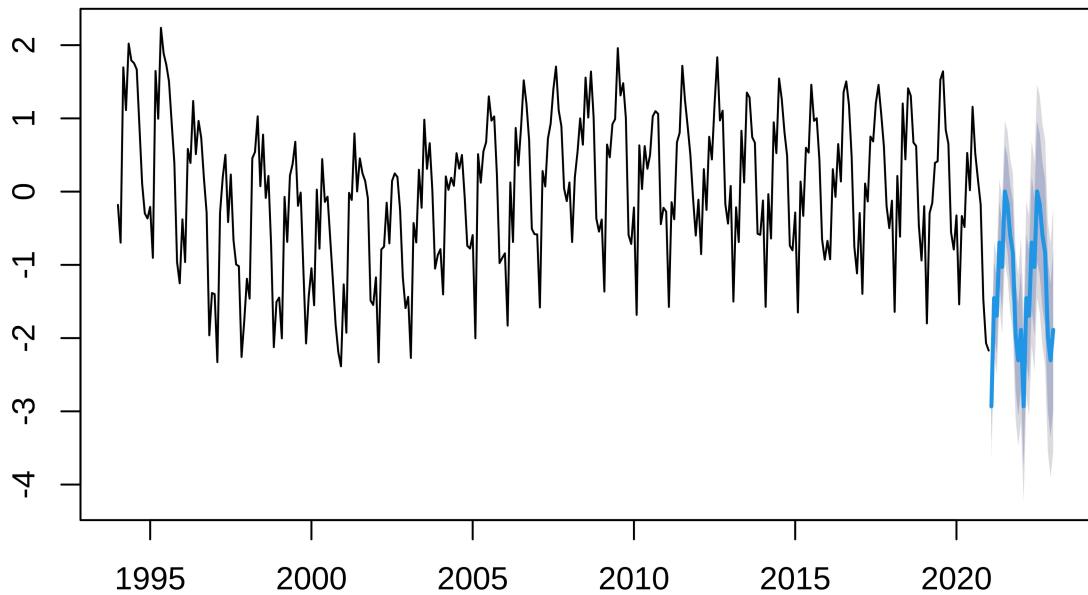
```
Marriages <- forecast(Final_dataset_TS[,2])
plot(Marriages, main = "Marriages Forecast")
```

Marriages Forecast



```
Births <- forecast(Final_dataset_TS[,1])
plot(Births, main = "Births Forecast")
```

Births Forecast



——— VAR time series forecasting for multivariate

```
apply(Final_dataset_TS, 2, adf.test)

## Warning in FUN(newX[, i], ...): p-value smaller than printed p-value
## Warning in FUN(newX[, i], ...): p-value smaller than printed p-value
## Warning in FUN(newX[, i], ...): p-value smaller than printed p-value
## Warning in FUN(newX[, i], ...): p-value greater than printed p-value

## $Births
##
## Augmented Dickey-Fuller Test
##
## data: newX[, i]
## Dickey-Fuller = -6.1948, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
##
## $Marriages
##
## Augmented Dickey-Fuller Test
```