Task3

Khaled Hasan

2024-03-10

Setup

import libraries:

```
library(fpp2)
```

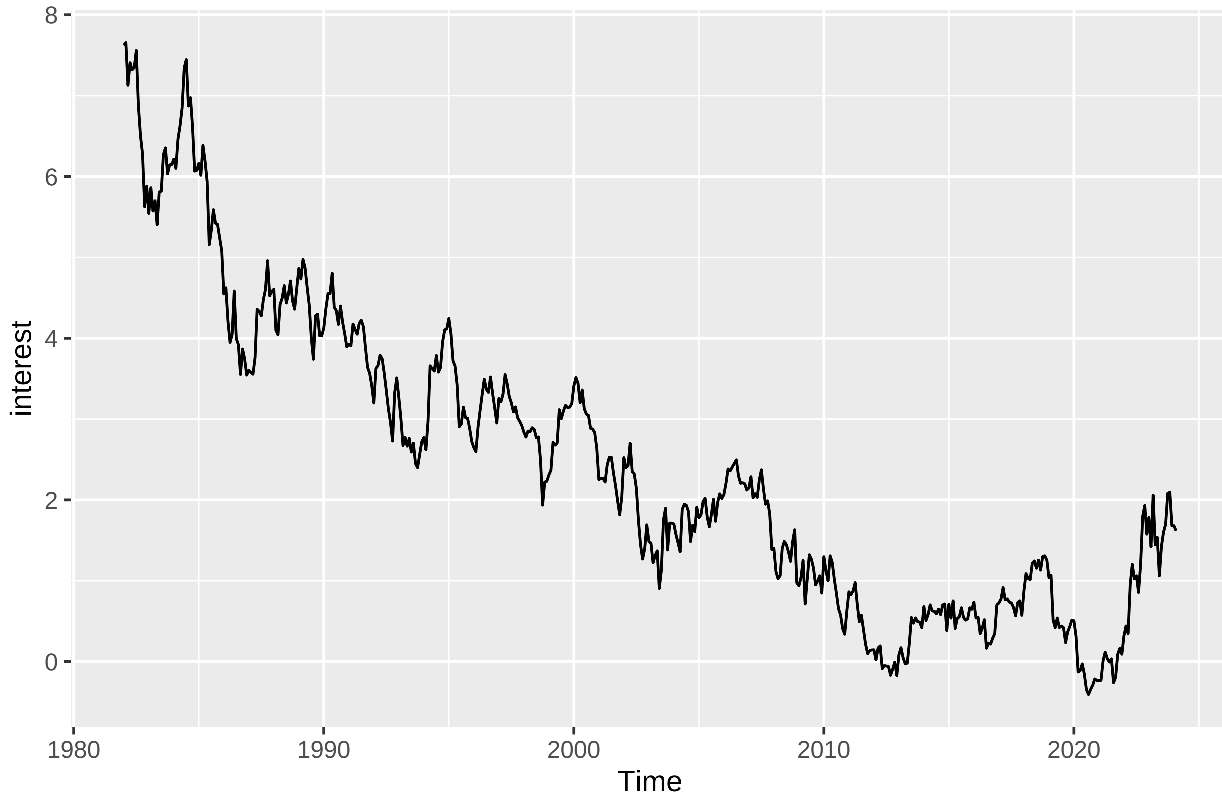
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## -- Attaching packages ---------------------------------------------- fpp2 2.5 --

## v ggplot2   3.4.4      v fma       2.5
## v forecast  8.21.1     v expsmooth 2.3

##
```

```
library(tseries)
library(urca)
```

Import the data:

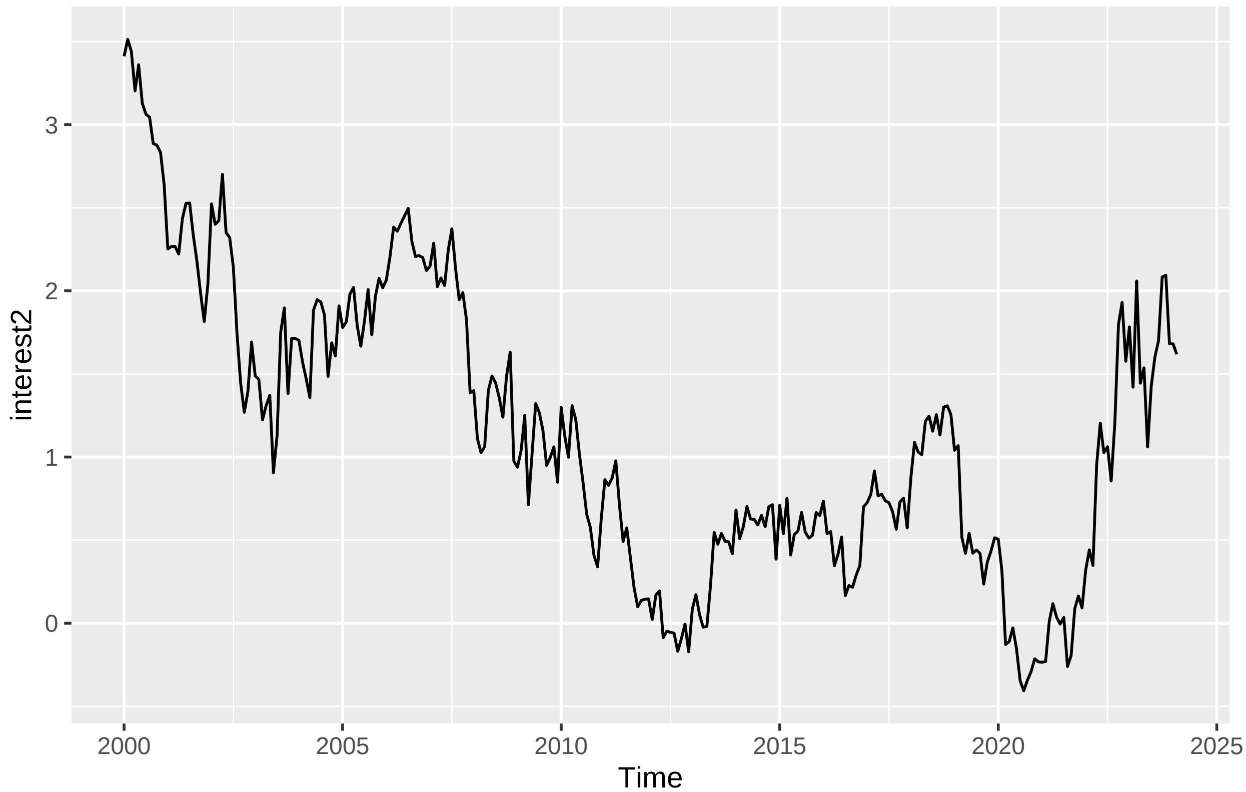
import 10 years real interest rate time series from csv (source:https://fred.stlouisfed.org/graph/?g=1hoLl):

```
REAINTRATREARAT10Y <- read.csv("C:\\Users\\ss\\Downloads\\REAINTRATREARAT10Y.csv")
interest <- ts(REAINTRATREARAT10Y[, "REAINTRATREARAT10Y"], frequency = 12, start = c(1982, 1))
autoplot(interest)
```



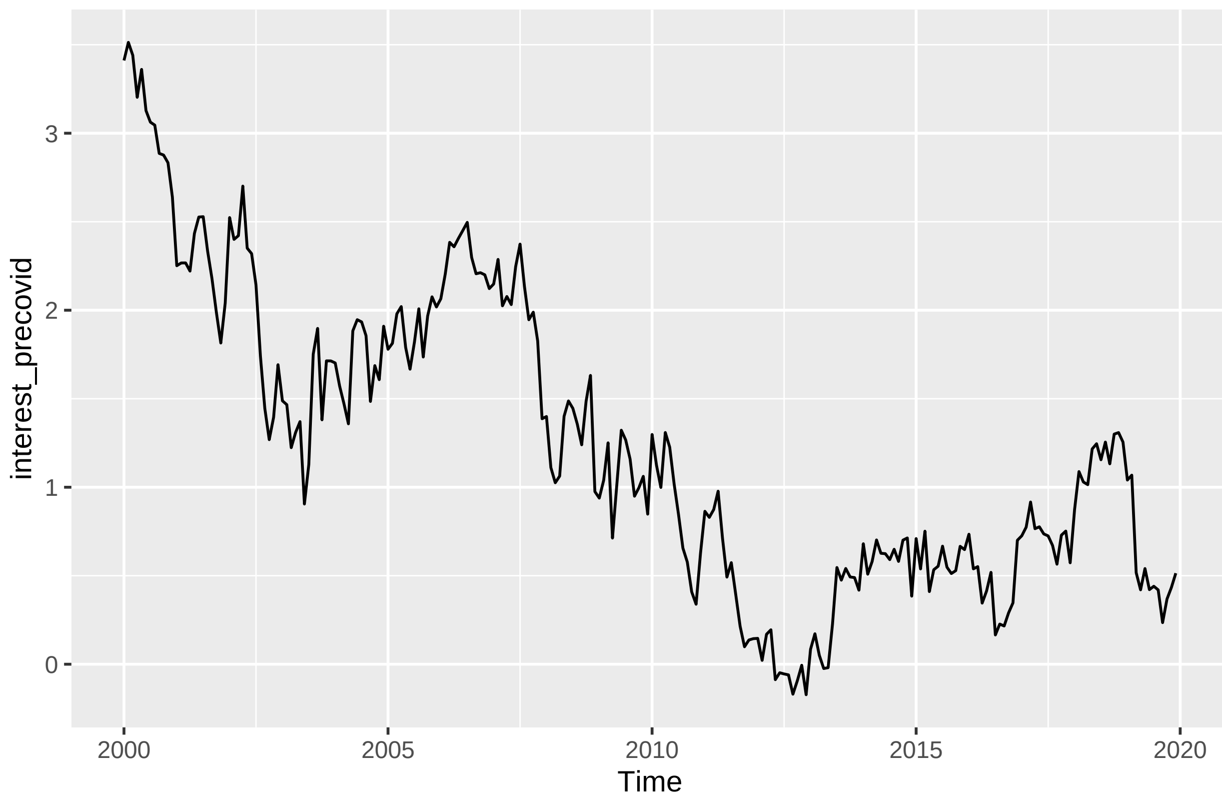
cut a window starting from year 2000

```
interest2 <- window(interest, frequency = 12, start = c(2000, 1))
autoplot(interest2)
```



Excluding the post pandamic era (2020-):

```
interest_precovid = window(interest2, frequency = 12, end=c(2019, 12))

autoplot(interest_precovid)
```



Arima

Fist I will try the auto.arima function over the pre-covid data:

```
fit_arima = auto.arima(interest_precovid, stepwise = FALSE, approximation = FALSE)
summary(fit_arima)
```

```
## Series: interest_precovid
## ARIMA(2,1,0)
##
## Coefficients:
##           ar1      ar2
##       -0.0958  -0.1401
## s.e.   0.0640   0.0639
##
## sigma^2 = 0.03548:  log likelihood = 60.85
## AIC=-115.69   AICc=-115.59   BIC=-105.27
##
## Training set error measures:
##                       ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.01502826 0.1871744 0.1422655 -7.463848 30.84157 0.3389113
##                     ACF1
## Training set -0.00215749
```

This is somewhat different from the results obtained in the case of (interest2) time series where the post covide data was not excluded (the order was determined to be (0, 1, 1) in the latter). Since the data was not determined to be seasonal, I will assume that it is indeed not seasonal (this assumption was further investigated in the prior report).

I will try to apply the auto.arima funciton, this time however, I will use (AIC) rather than the AICc used by the auto.arima in default mode.

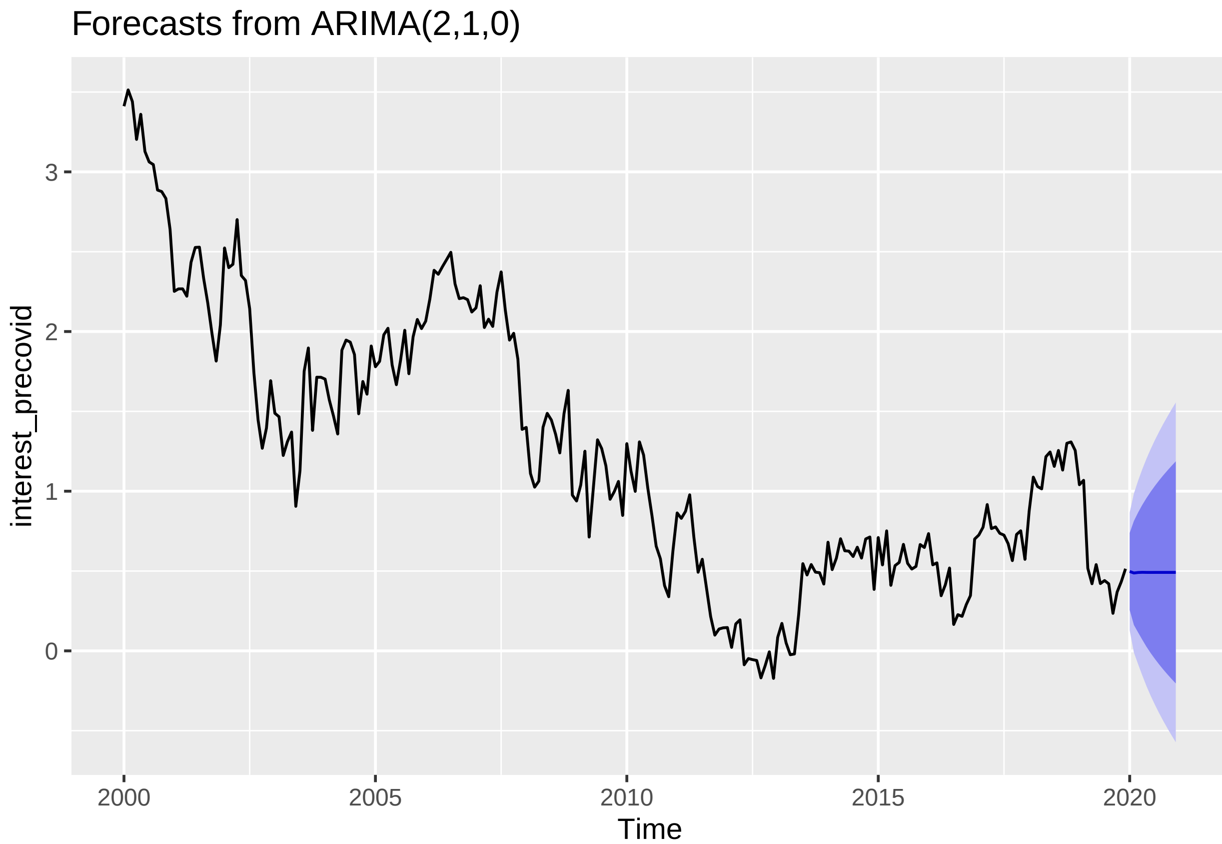
```
fit_arima_aic = auto.arima(interest_precovid, stepwise = FALSE, approximation = FALSE, ic = "aic")
summary(fit_arima_aic)
```

```
## Series: interest_precovid
## ARIMA(2,1,0)
##
## Coefficients:
##           ar1      ar2
##       -0.0958  -0.1401
## s.e.   0.0640   0.0639
##
## sigma^2 = 0.03548:  log likelihood = 60.85
## AIC=-115.69   AICc=-115.59   BIC=-105.27
##
## Training set error measures:
##                       ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.01502826 0.1871744 0.1422655 -7.463848 30.84157 0.3389113
##                     ACF1
## Training set -0.00215749
```

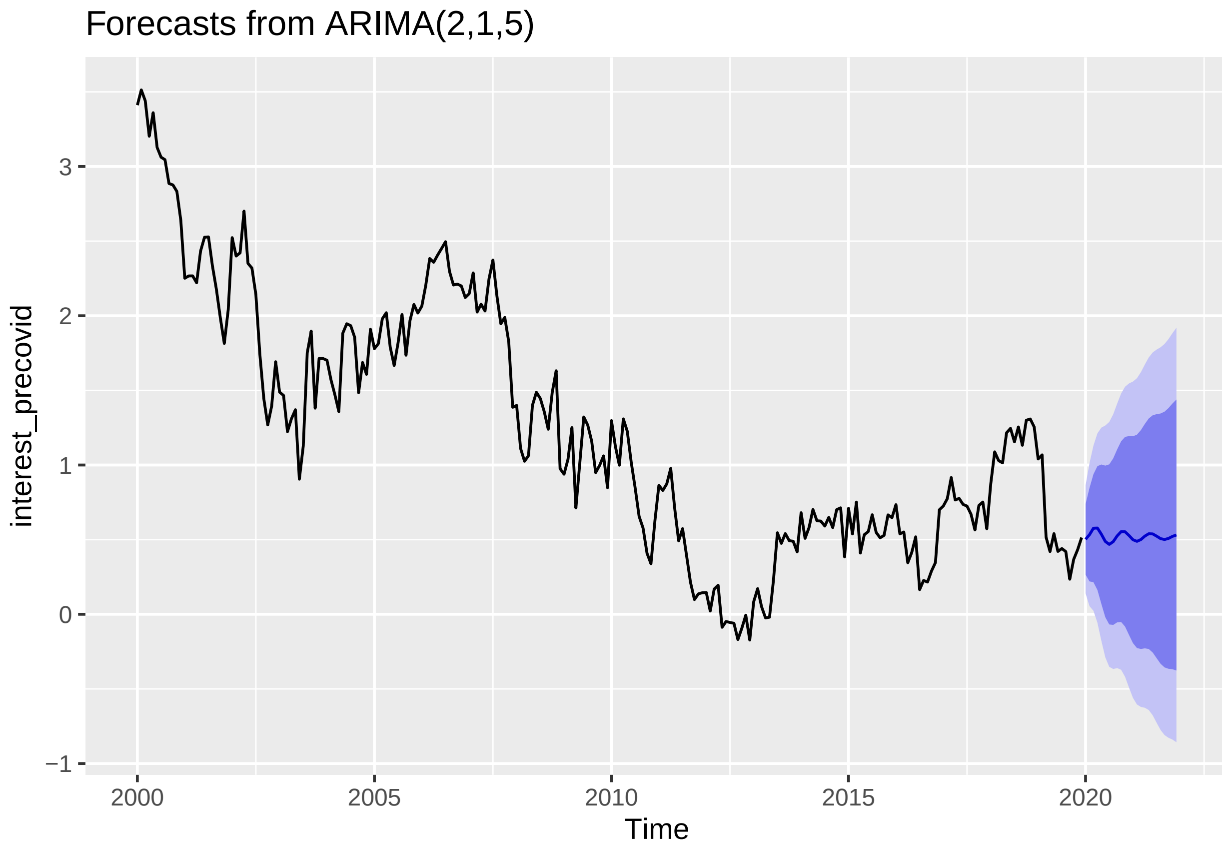
```
minimum = 0
for(i in 0:6){
  for(j in 0:6){
    if(AIC(Arima(interest_precovid, order = c(j, 1, i))) < minimum){
      minimum = AIC(Arima(interest_precovid, order = c(j, 1, i)))
      m = c(j, i)
    }
  }
}
{print(m)
print(minimum)}
```

```
## [1] 2 5
## [1] -118.1047
```

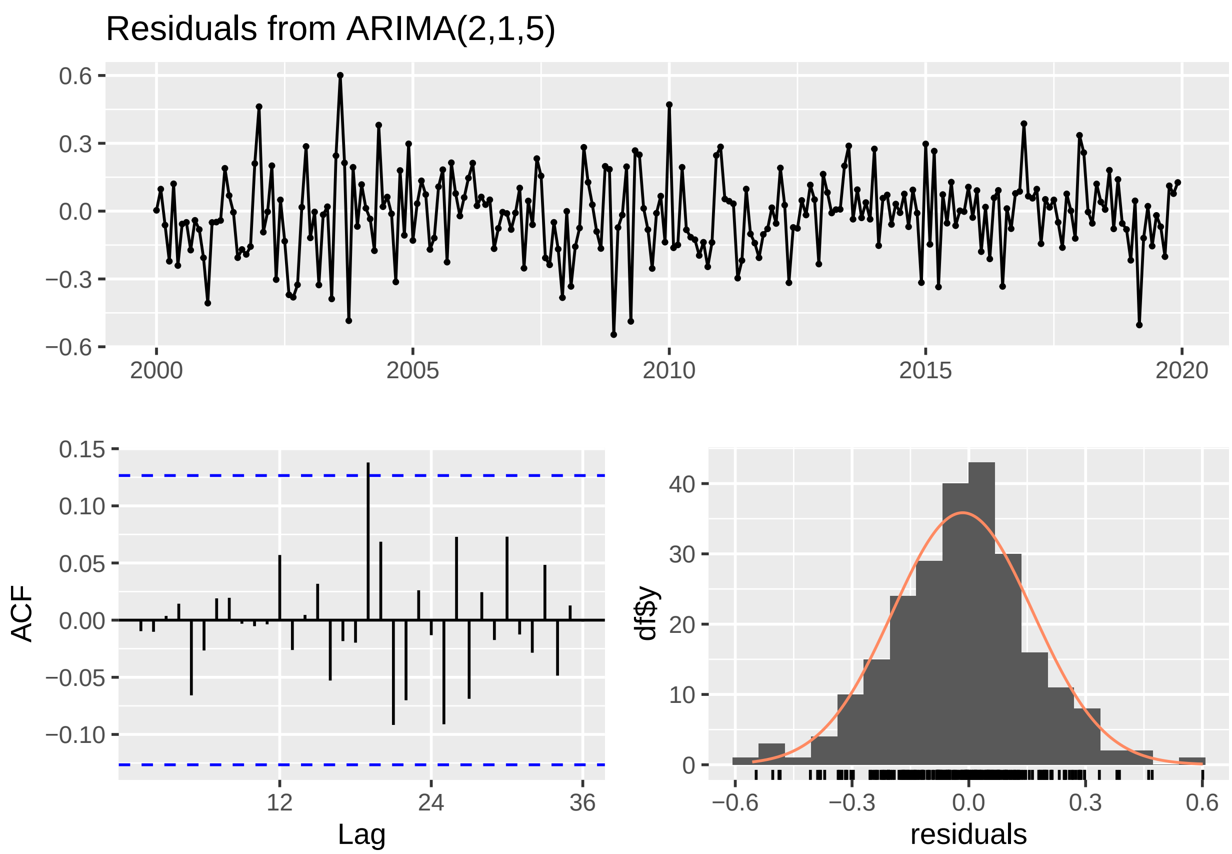
```
fit_arima_aic%>% forecast::forecast(h=12) %>% autoplot()
```




```r
Arima(interest_precovid, c(2, 1, 5))%>% forecast::forecast(h=24) %>% autoplot()
```




```
checkresiduals(Arima(interest_precovid, c(2, 1, 5)))
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,5)
## Q* = 13.738, df = 17, p-value = 0.6855
##
## Model df: 7.   Total lags used: 24
```

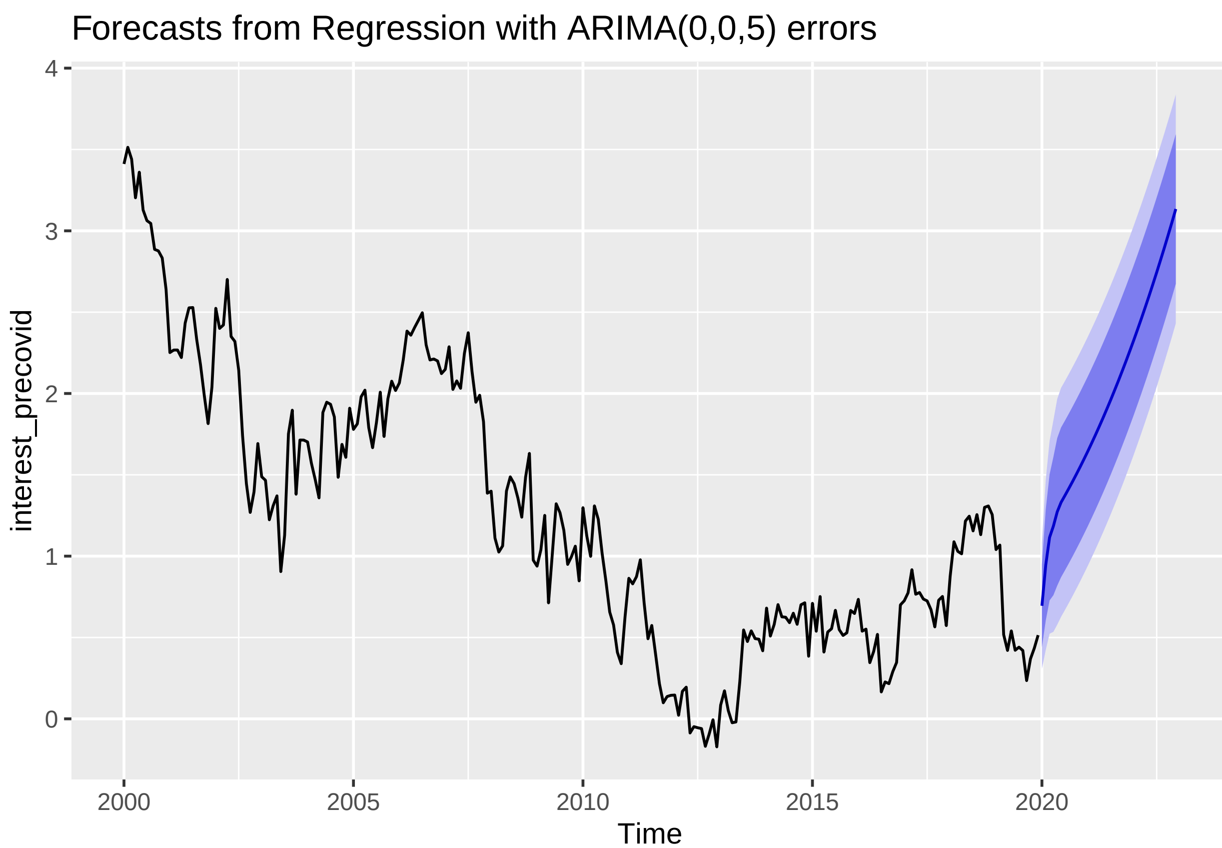
Arima with regression

In the previous tasks, the best Arima model I obtained for the data was Arima(2, 1, 2) with no seasonal parts, this will be further investigated in this report, but including regression this time:

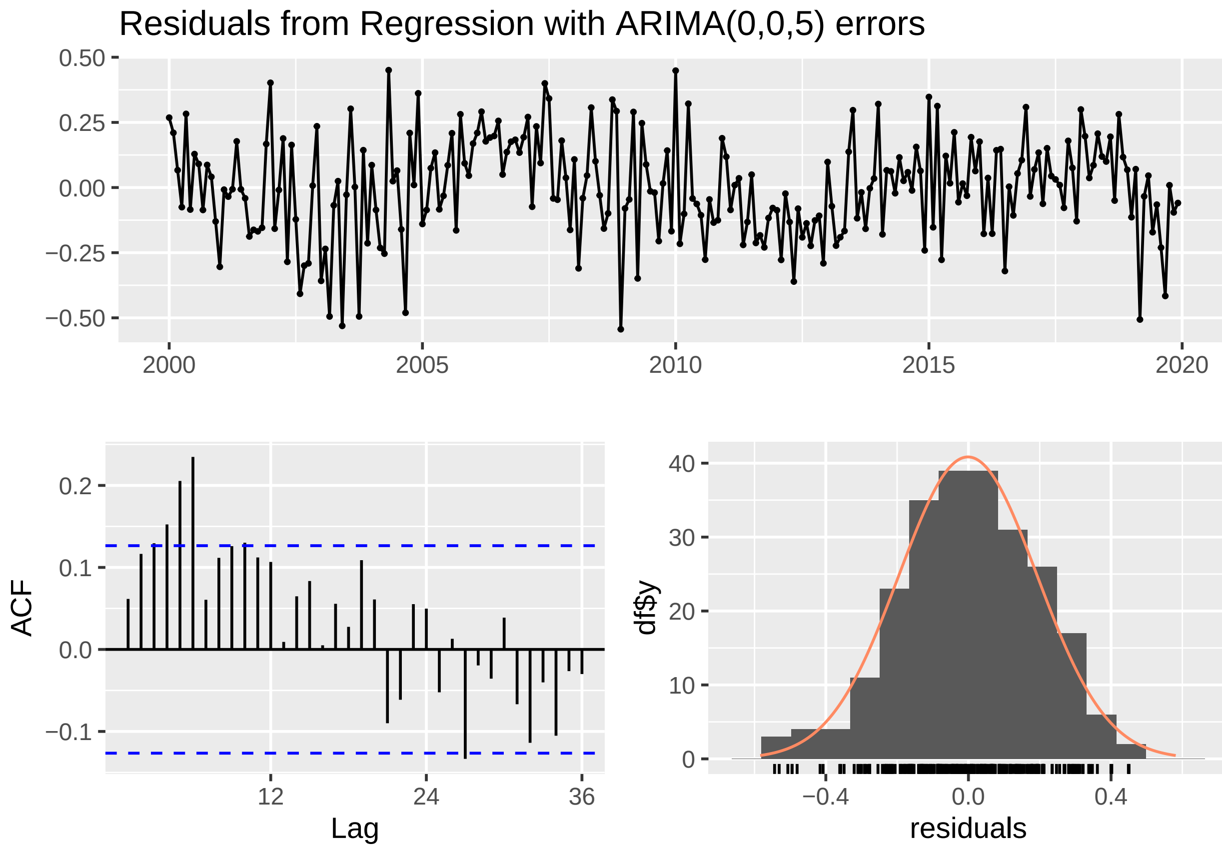
```
# time vector
t <- seq_along(interest_precovid)

# regression vector
treg <- length(t)+1:36
```

4th order polynomial:

```
xreg = cbind(
  t^1, t^2, t^3, t^4
)

fit1 <- auto.arima(interest_precovid, xreg = xreg, approximation = FALSE, stepwise = FALSE, seasonal = '
autoplot(forecast(fit1, xreg = cbind(treg^1, treg^2, treg^3, treg^4)))
```



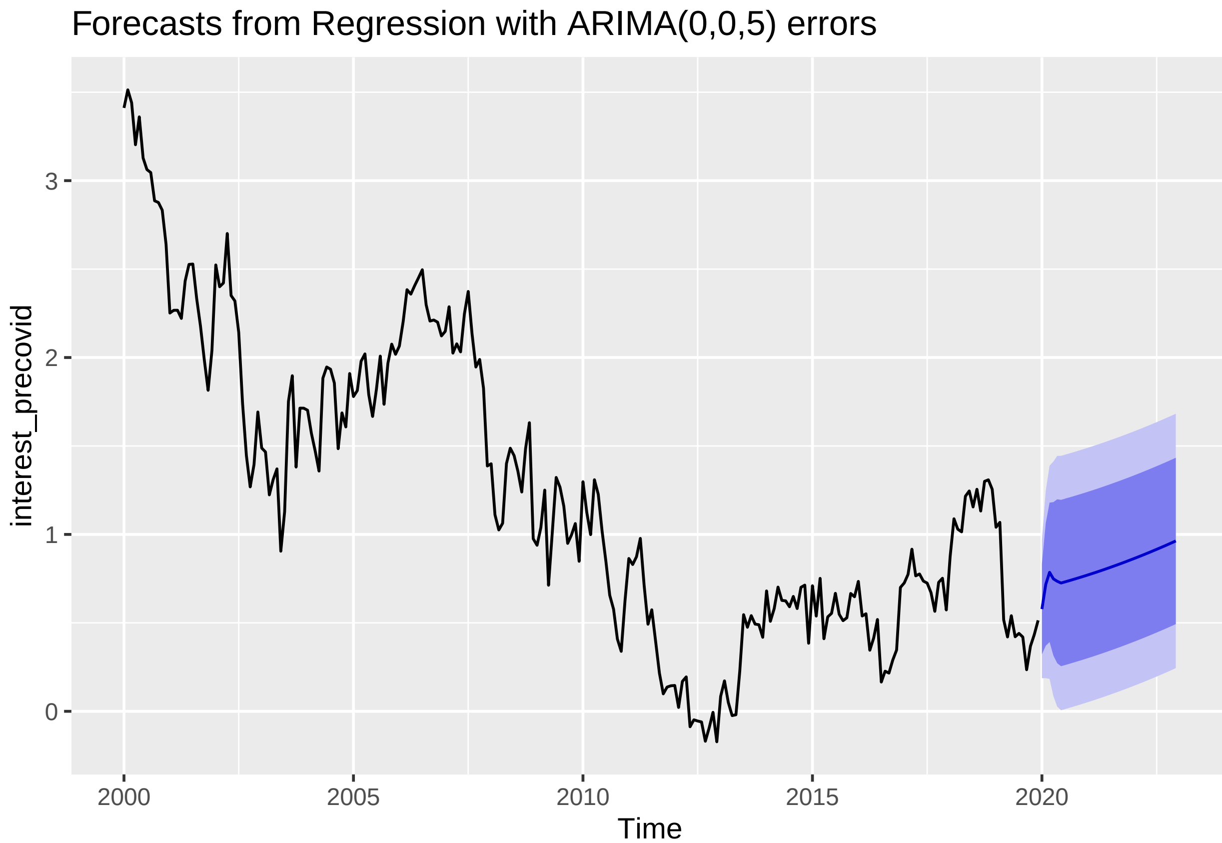

```
checkresiduals(fit1)
```



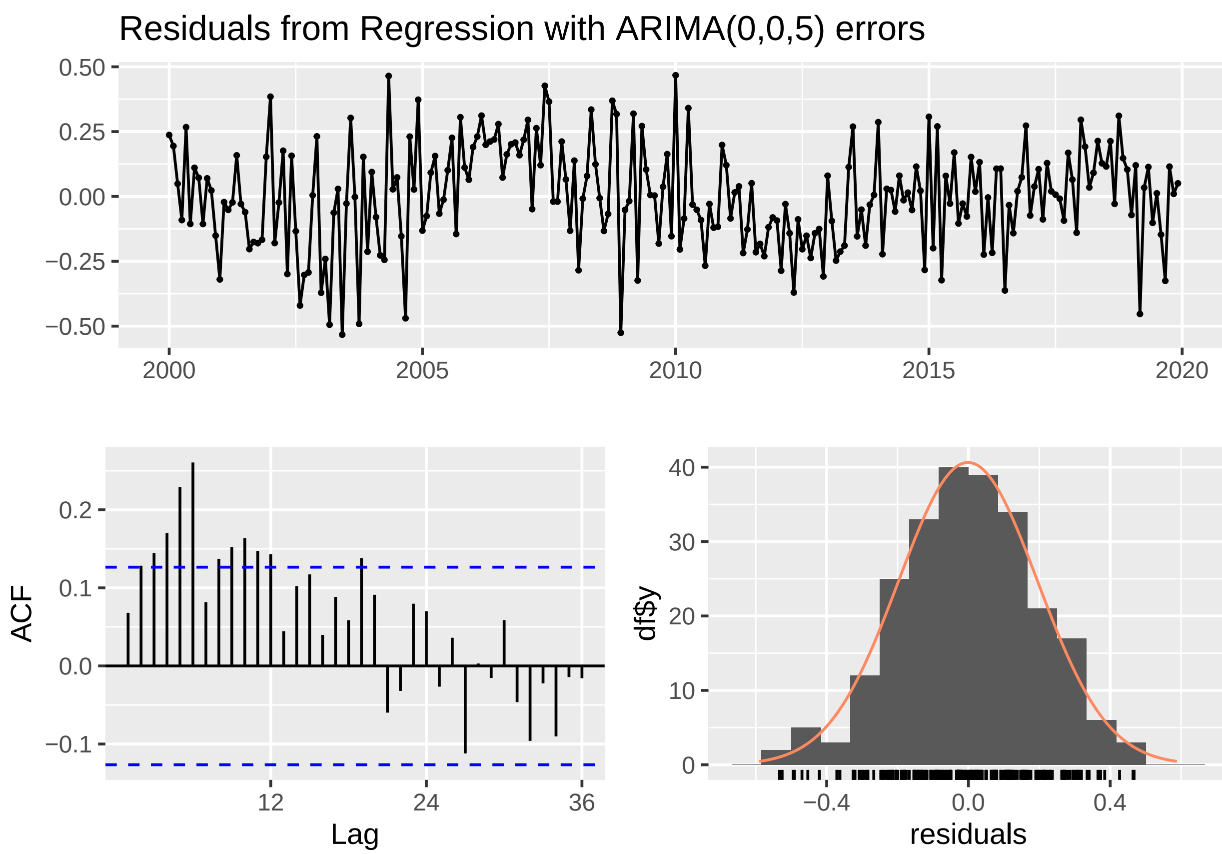

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,5) errors
## Q* = 69.173, df = 19, p-value = 1.262e-07
##
## Model df: 5.   Total lags used: 24
```

second order polynomial

```
fit2 <- auto.arima(interest_precovid, xreg = cbind(t^0, t^1, t^2), approximation = FALSE, stepwise = FA
autoplot(forecast(fit2, h=10, xreg = cbind(treg^0, treg^1, treg^2)))
```




```r
checkresiduals(fit2)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,5) errors
## Q* = 98.478, df = 19, p-value = 1.009e-12
##
## Model df: 5.   Total lags used: 24
```

linear trend (drift)

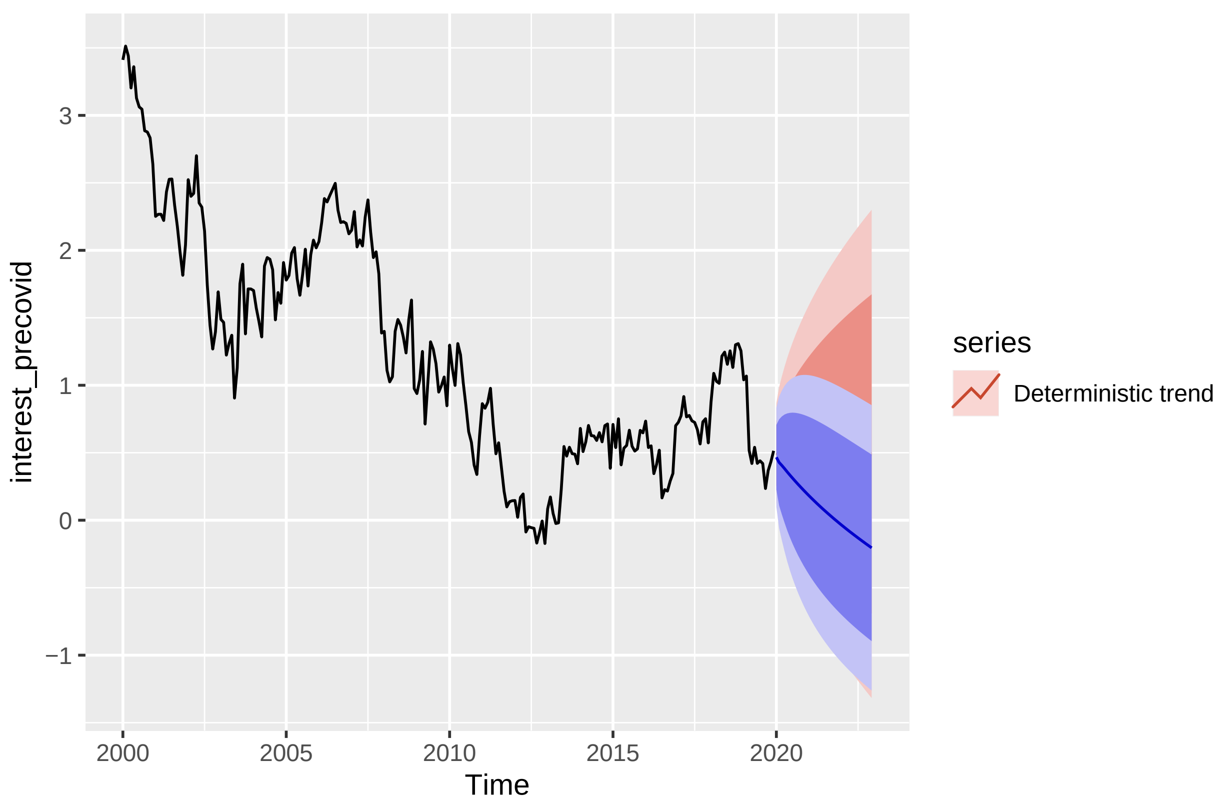
```
fit3 <- auto.arima(interest_precovid, xreg = cbind(t^0, t^1), approximation = FALSE, stepwise = FALSE)
arima_for_fit3 <- auto.arima(interest_precovid, approximation = FALSE, stepwise = FALSE)
summary(fit3)
```

```
## Series: interest_precovid
## Regression with ARIMA(3,0,0) errors
##
## Coefficients:
##          ar1      ar2     ar3   xreg1    xreg2
##       0.8765  -0.0487  0.1225  2.6363  -0.0106
## s.e.  0.0639   0.0854  0.0643  0.4024   0.0027
##
## sigma^2 = 0.03501:  log likelihood = 63.13
```

```
## AIC=-114.26   AICc=-113.9   BIC=-93.37
##
## Training set error measures:
##                        ME     RMSE       MAE       MPE     MAPE      MASE
## Training set -0.006700629 0.185156 0.1426663 -2.727971 30.58303 0.3398663
##                     ACF1
## Training set 0.0032656
```

```
autoplot(interest_precovid) +
  autolayer(forecast(arima_for_fit3, h=36), series="Deterministic trend") +
  autolayer(forecast(fit3, xreg = cbind(treg^0, treg^1), series="Stochastic trend"))
```

```
## Warning in forecast.forecast_ARIMA(fit3, xreg = cbind(treg^0, treg^1), series =
## "Stochastic trend"): The non-existent series arguments will be ignored.
```




```
checkresiduals(fit3)
```


```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,0,0) errors
## Q* = 20.861, df = 21, p-value = 0.4675
##
## Model df: 3.   Total lags used: 24
```

first order, with long periodic cycles

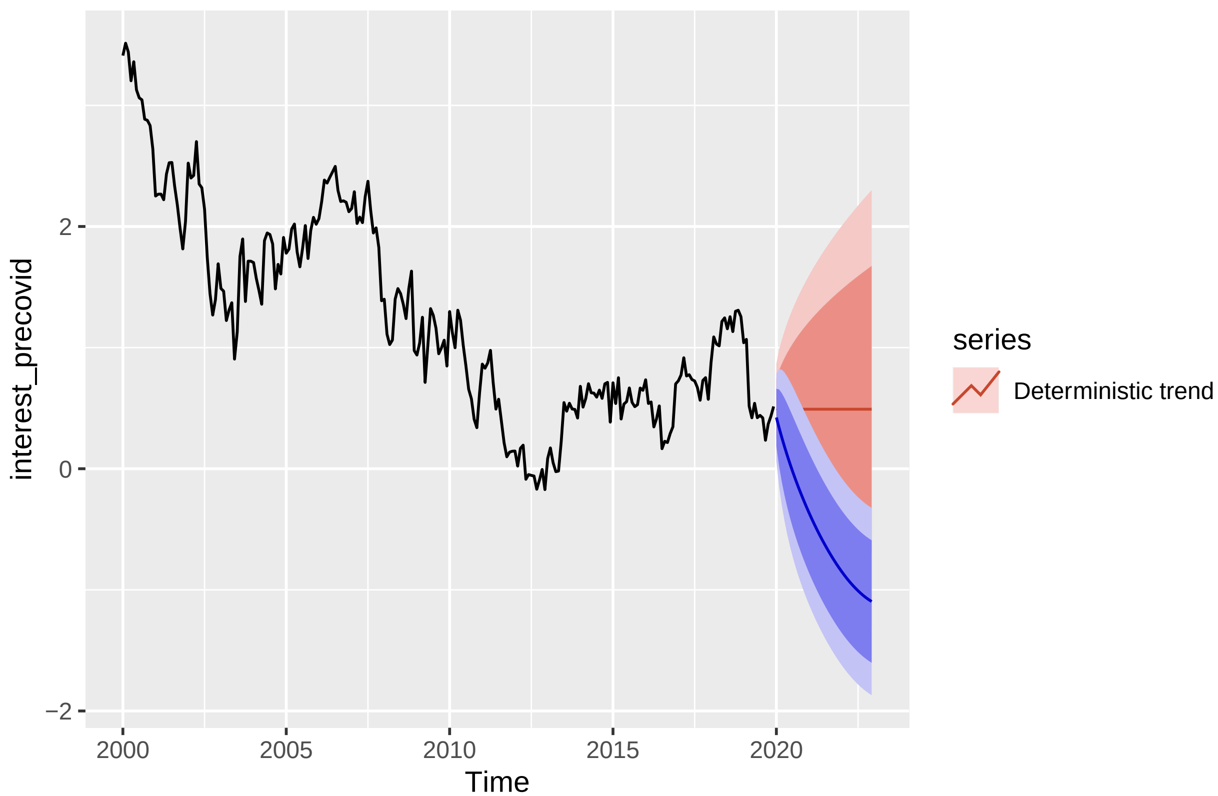
```
fit4 <- auto.arima(interest_precovid, xreg = cbind(t^0, t^1, cos(pi*t/60), sin(pi*t/60)), approximation
summary(fit4)
```

```
## Series: interest_precovid
## Regression with ARIMA(1,0,0) errors
##
## Coefficients:
##          ar1  xreg1    xreg2   xreg3    xreg4
##       0.8831  2.680  -0.0117  0.1961  -0.5213
## s.e.  0.0305  0.197   0.0014  0.1261   0.1399
##
## sigma^2 = 0.03427:  log likelihood = 66.04
## AIC=-120.08   AICc=-119.72   BIC=-99.2
```

```
##
## Training set error measures:
##                        ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.003364964 0.1831833 0.1388288 -4.547935 29.51226 0.3307242
##                     ACF1
## Training set -0.0402184
```

```
autoplot(interest_precovid) +
  autolayer(forecast(fit_arima, h=36), series="Deterministic trend") +
  autolayer(forecast(fit4, xreg = cbind(treg^0, treg^1,  cos(pi*treg/60), sin(pi*treg/60)), series="Sto
```

```
## Warning in forecast.forecast_ARIMA(fit4, xreg = cbind(treg^0, treg^1, cos(pi *
## : The non-existent series arguments will be ignored.
```



first order, with long and short periodic cycles

```
fit5 <- auto.arima(interest_precovid, xreg = cbind(t^1, cos(pi*t/60), sin(pi*t/60), cos(pi/6*t), sin(pi/
summary(fit5)
```

```
## Series: interest_precovid
## Regression with ARIMA(3,0,0) errors
##
```

```
## Coefficients:
##          ar1      ar2     ar3  intercept    xreg1   xreg2    xreg3    xreg4
##       0.8320  -0.0520  0.1255     2.7066  -0.0118  0.2092  -0.5275  -0.0297
## s.e.  0.0639   0.0835  0.0644     0.2305   0.0016  0.1384   0.1536   0.0283
##        xreg5
##       0.0274
## s.e.  0.0285
##
## sigma^2 = 0.03401:  log likelihood = 68.97
## AIC=-117.94   AICc=-116.98   BIC=-83.14
##
## Training set error measures:
##                        ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.004205549 0.1809328 0.1378388 -5.573356 30.51125 0.3283658
##                     ACF1
## Training set 0.004085056
```

```r
autoplot(interest_precovid) +
  autolayer(forecast(fit_arima, h=36), series="Deterministic trend") +
  autolayer(forecast(fit5, xreg = cbind(treg^1,  cos(pi*treg/60), sin(pi*treg/60), cos(pi/6*treg), sin(p
```

calculate adjusted r squared test

```r
vec_of_cor = c(cor(fitted(fit5), t^1),
               cor(fitted(fit5), cos(pi*t/60)),
               cor(fitted(fit5), sin(pi*t/60)),
```

```
                cor(fitted(fit5), cos(pi/6*t)),
                cor(fitted(fit5), sin(pi/6*t)))

R_squared = (t(vec_of_cor)%*%solve(cor(data.frame(t^1, cos(pi/60*t), sin(pi/60*t), cos(pi/6*t), sin(pi/6
n=length(t)
k=length(vec_of_cor)
aR_squared = 1 - ((1 - R_squared)*(n-1)/(n - k -1))
print(aR_squared)
```

```
##           [,1]
## [1,] 0.6869919
```

Parameters Stability

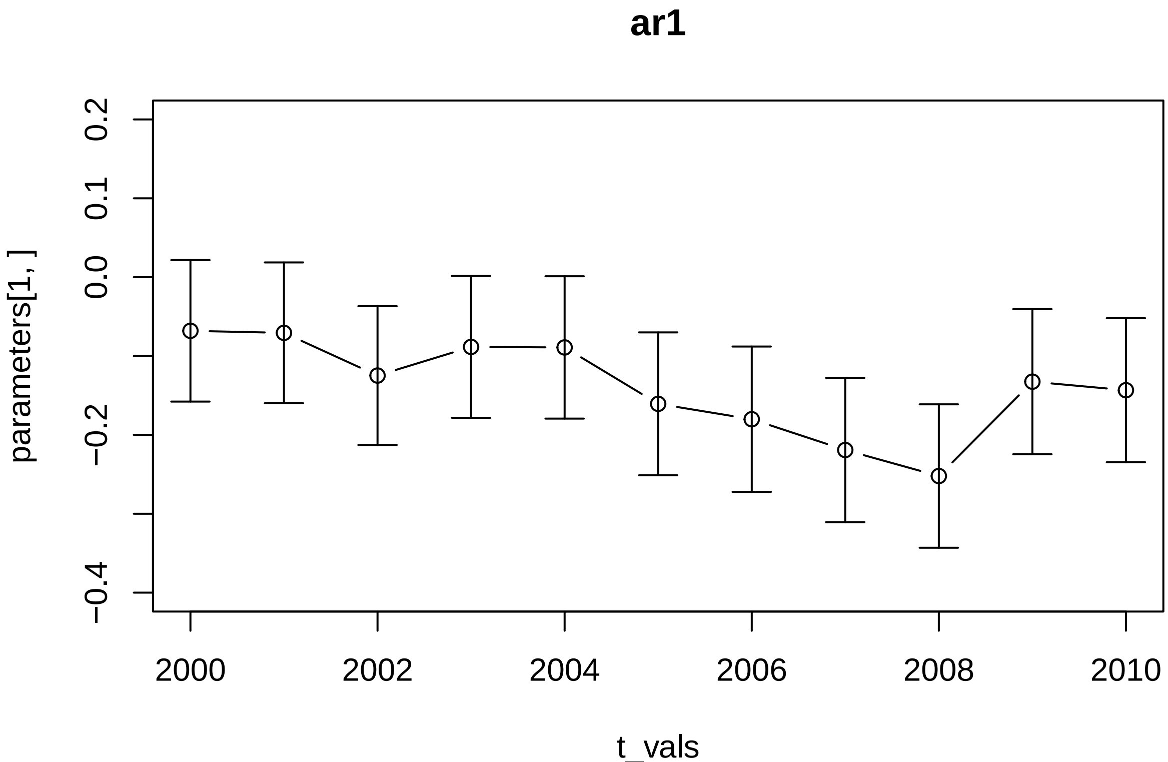
Rolling Window

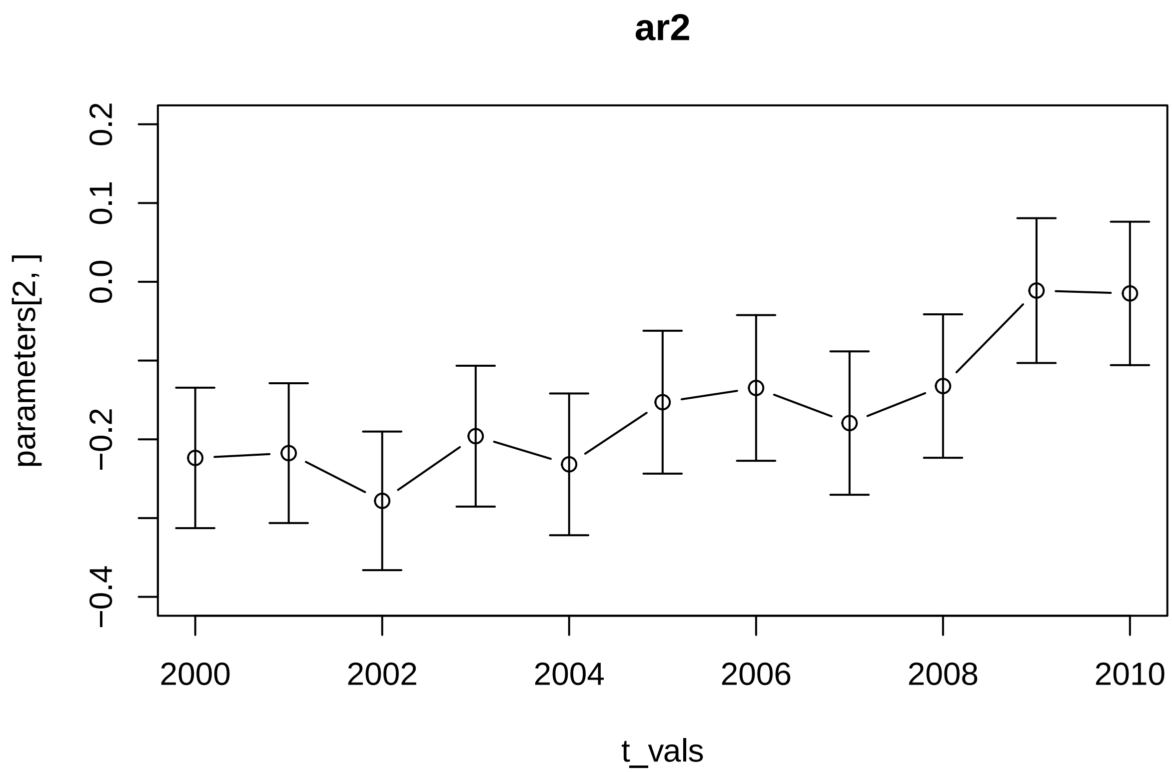
Throughout the following tests, I will be considering the

```
parameters =c()
errors = c()

s = 2000
for (i in 1:11){
    parameters <- cbind(parameters, coefficients(Arima(window(interest, frequency = 12, start=c(2000+i,
    errors <- cbind(errors, sqrt(diag(vcov(Arima(window(interest, frequency = 12, start=c(2000+i, 1), en
)
}

t_vals = c(2000:2010)
```

due to the small sample sizes I preferred to keep the default aicc optimization scheme, rather than the aic.

```
{
  plot(t_vals, parameters[1, ], type='b', main="ar1", ylim = c(-0.4, 0.2))+
  arrows(x0=t_vals, y0=parameters[1, ]-errors[1, ], x1 = t_vals, y1=parameters[1, ]+errors[1, ], code=3
  plot(t_vals, parameters[2, ], type='b', main="ar2", ylim = c(-0.4, 0.2))+
  arrows(x0=t_vals, y0=parameters[2, ]-errors[2, ], x1 = t_vals, y1=parameters[2, ]+errors[2, ], code=3
}
```


ar1
parameters[1,]
0.2
0.1
0.0
−0.2
−0.4
2000
2002
2004
2006
2008
2010
t_vals




```
## integer(0)
```

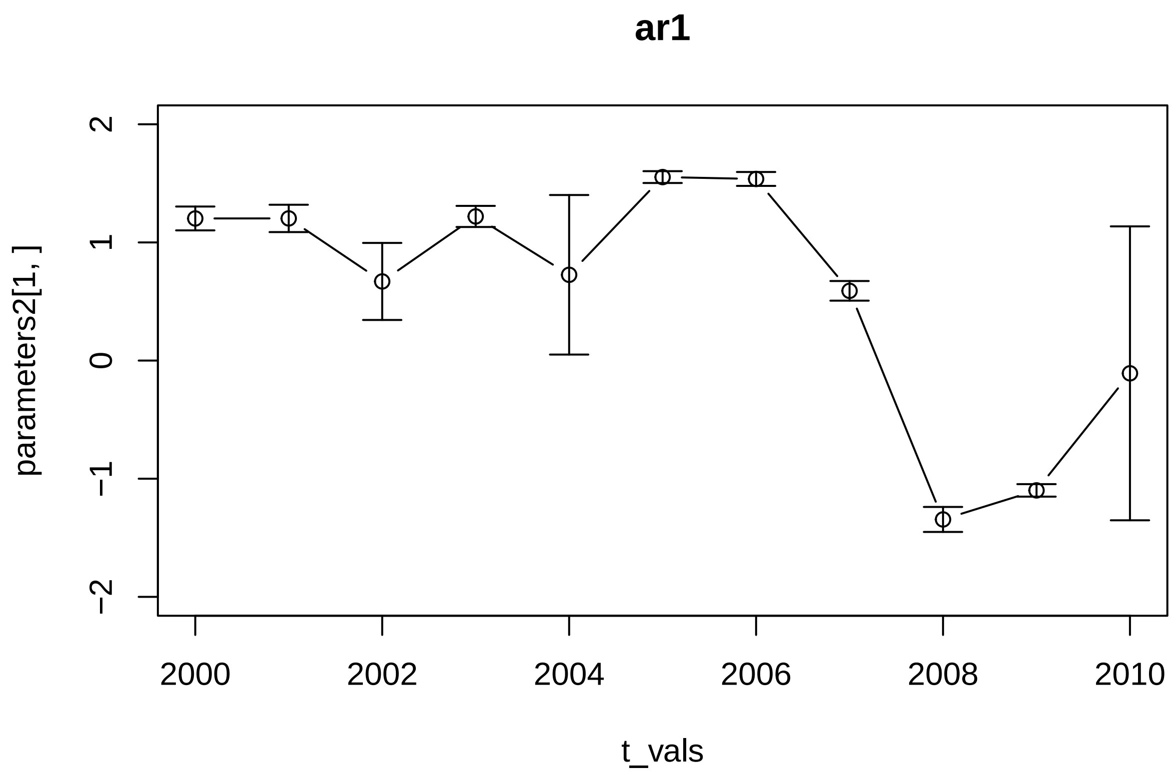
```
cbind('ar1' = parameters[1,],
      's.e(ar1)' = errors[1, ],
      'ar2' = parameters[2,],
      's.e(ar2)' = errors[2, ]
      )
```

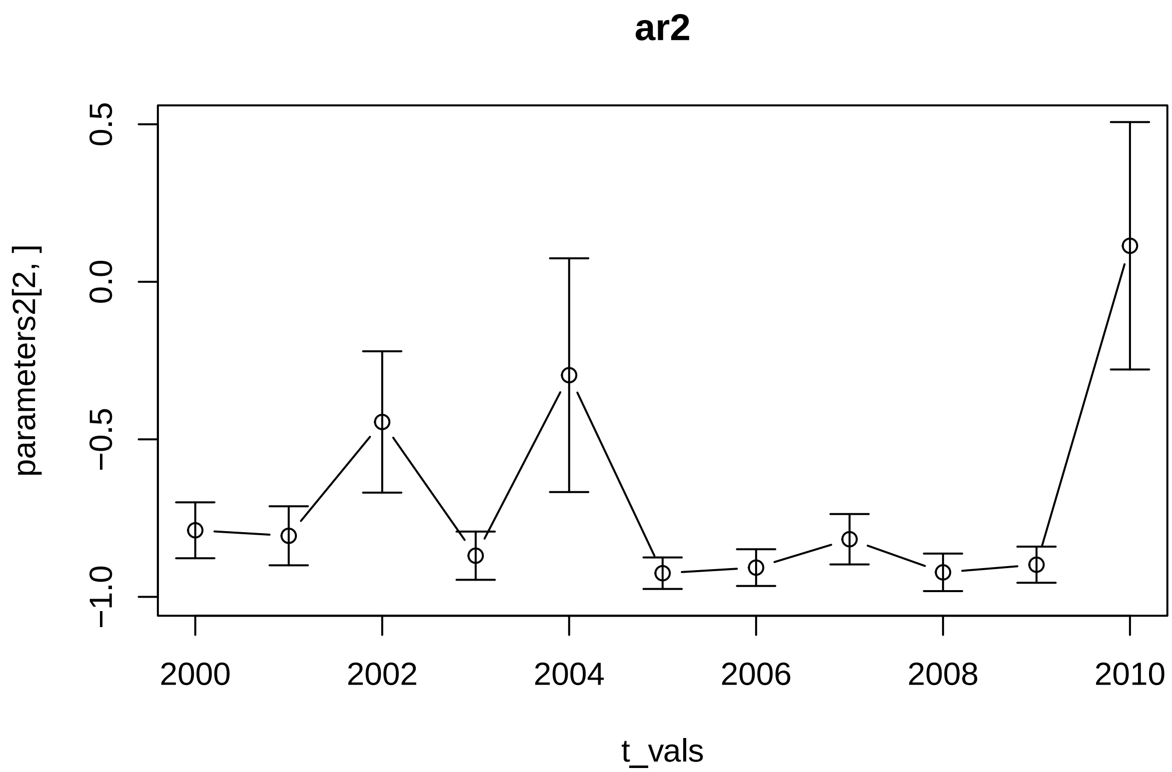
```
##              ar1   s.e(ar1)         ar2   s.e(ar2)
##  [1,] -0.06803913 0.08970171 -0.22360140 0.08916453
##  [2,] -0.07060939 0.08929109 -0.21752596 0.08874775
##  [3,] -0.12476973 0.08803037 -0.27812216 0.08798077
##  [4,] -0.08841939 0.08985708 -0.19597730 0.08942455
##  [5,] -0.08909497 0.09024782 -0.23176670 0.08998001
##  [6,] -0.16063070 0.09062497 -0.15284866 0.09071218
##  [7,] -0.18012695 0.09217377 -0.13475924 0.09248636
##  [8,] -0.21915537 0.09146068 -0.17938878 0.09101450
##  [9,] -0.25217031 0.09091678 -0.13236263 0.09105294
## [10,] -0.13252914 0.09200725 -0.01115426 0.09192574
## [11,] -0.14334047 0.09130707 -0.01476446 0.09104311
```

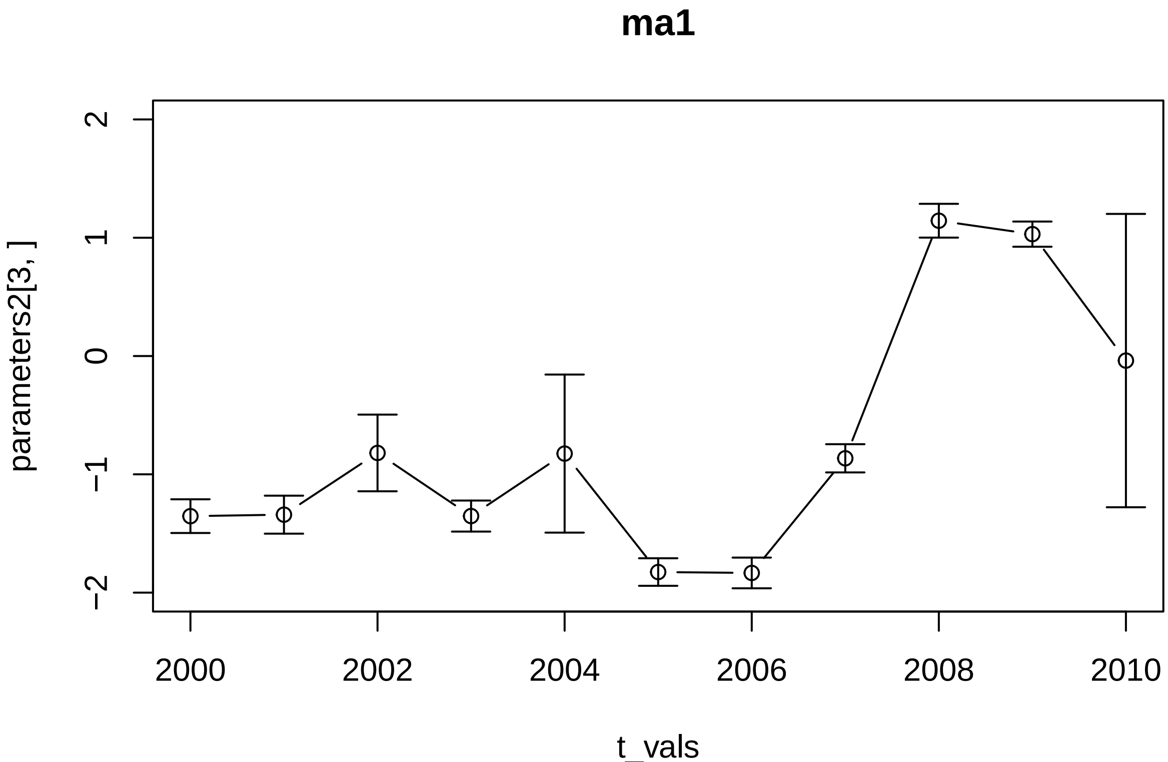
Try Arima (2, 1, 5)

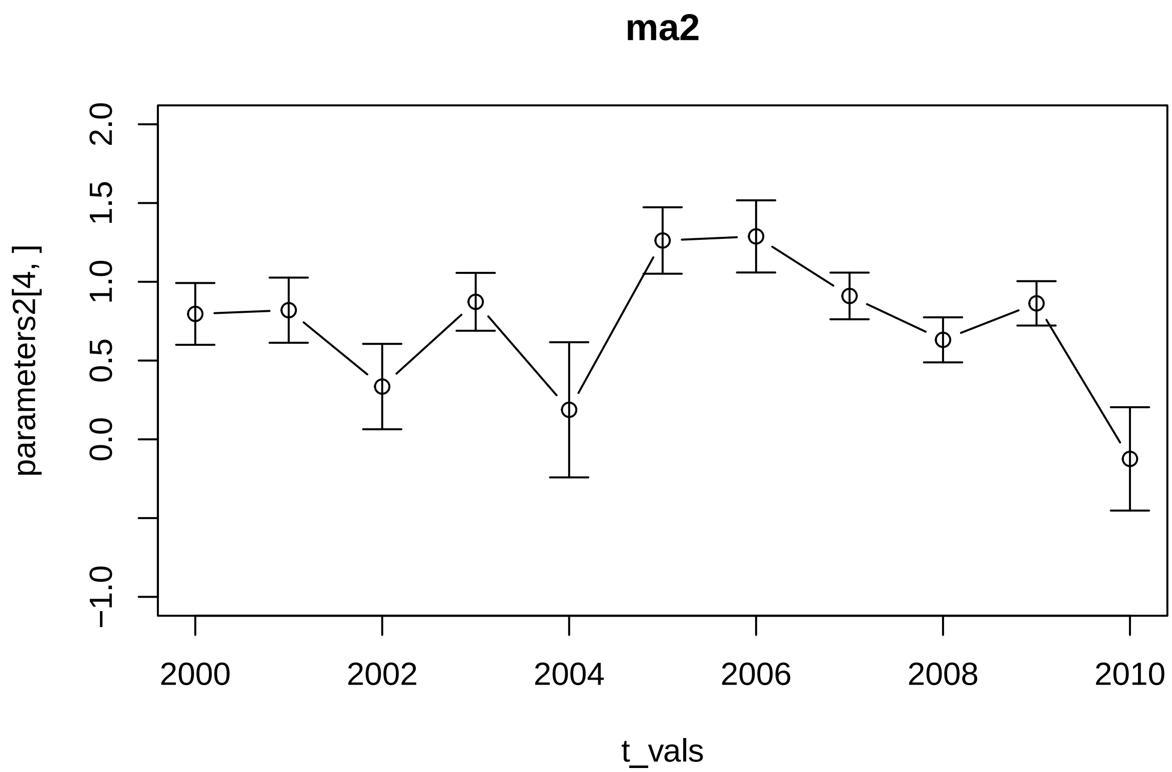
```r
parameters2 =c()
errors2 = c()

s = 2000
for (i in 1:11){
    parameters2 <- cbind(parameters2, coefficients(Arima(window(interest, frequency = 12, start=c(2000+i
    errors2 <- cbind(errors2, sqrt(diag(vcov(Arima(window(interest, frequency = 12, start=c(2000+i, 1),
)
}

t_vals = c(2000:2010)
```

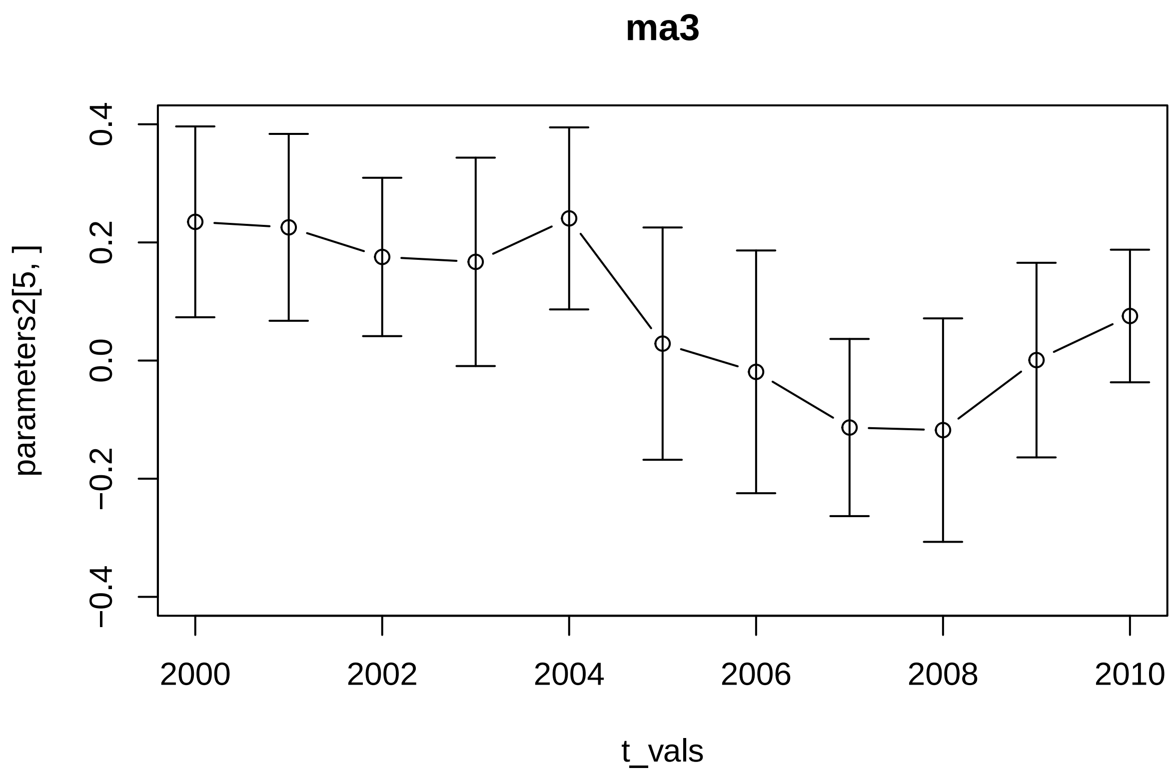
```r
{
  plot(t_vals, parameters2[1, ], type='b', main="ar1", ylim = c(-2, 2))+
  arrows(x0=t_vals, y0=parameters2[1, ]-errors2[1, ], x1 = t_vals, y1=parameters2[1, ]+errors2[1, ], co
  plot(t_vals, parameters2[2, ], type='b', main="ar2", ylim = c(-1, 0.5))+
  arrows(x0=t_vals, y0=parameters2[2, ]-errors2[2, ], x1 = t_vals, y1=parameters2[2, ]+errors2[2, ], co
  plot(t_vals, parameters2[3, ], type='b', main="ma1", ylim = c(-2, 2))+
  arrows(x0=t_vals, y0=parameters2[3, ]-errors2[3, ], x1 = t_vals, y1=parameters2[3, ]+errors2[3, ], co
  plot(t_vals, parameters2[4, ], type='b', main="ma2", ylim = c(-1, 2))+
  arrows(x0=t_vals, y0=parameters2[4, ]-errors2[4, ], x1 = t_vals, y1=parameters2[4, ]+errors2[4, ], co
  plot(t_vals, parameters2[5, ], type='b', main="ma3", ylim = c(-0.4, 0.4))+
  arrows(x0=t_vals, y0=parameters2[5, ]-errors2[5, ], x1 = t_vals, y1=parameters2[5, ]+errors2[5, ], co
  plot(t_vals, parameters2[6, ], type='b', main="ma4", ylim = c(-0.4, 0.4))+
  arrows(x0=t_vals, y0=parameters2[6, ]-errors2[6, ], x1 = t_vals, y1=parameters2[6, ]+errors2[6, ], co
  plot(t_vals, parameters2[7, ], type='b', main="ma5", ylim = c(-0.4, 0.3))+
  arrows(x0=t_vals, y0=parameters2[7, ]-errors2[7, ], x1 = t_vals, y1=parameters2[7, ]+errors2[7, ], co
}
```

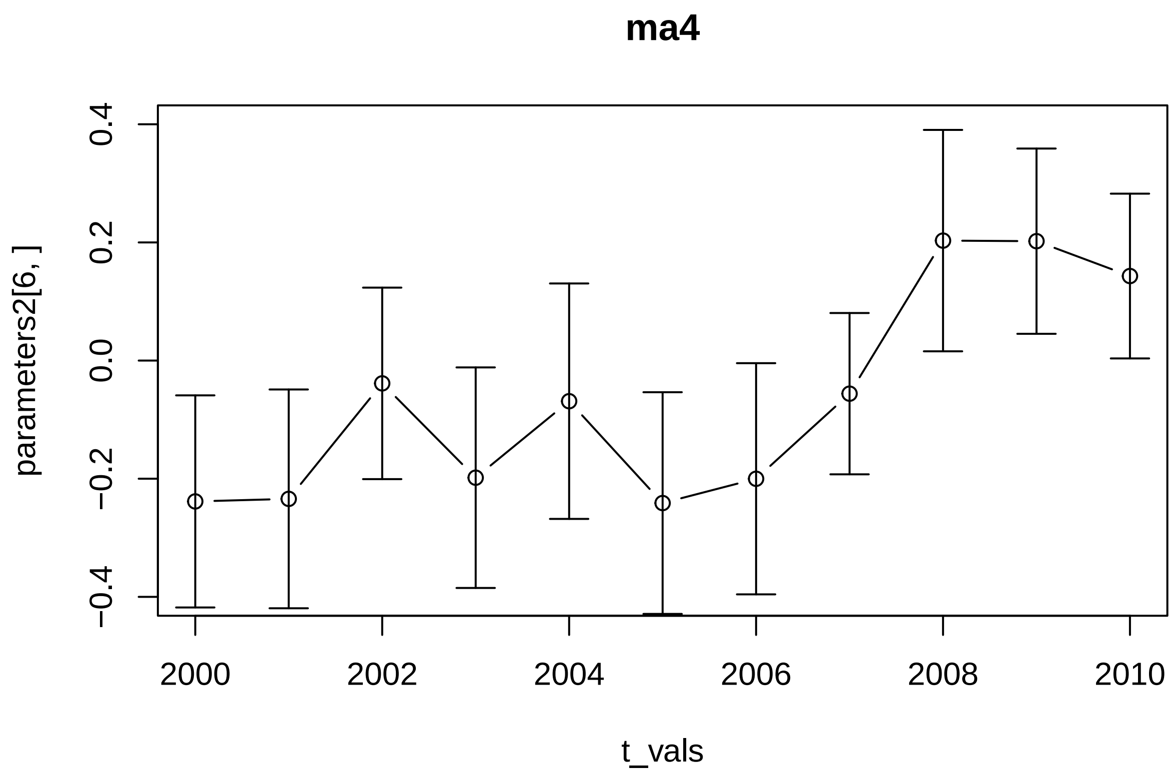

ar1
parameters2[1,]
t_vals
2
1
0
−1
−2
2000
2002
2004
2006
2008
2010

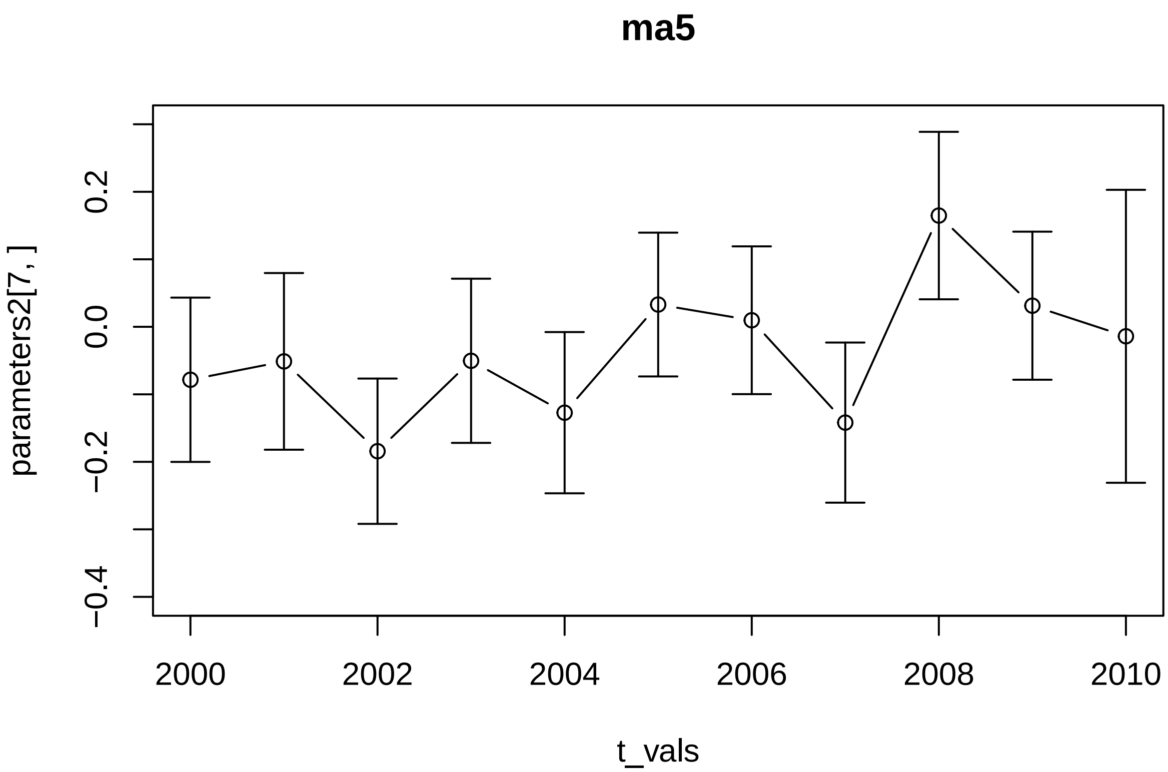

ar2
0.5
0.0
−0.5
−1.0
parameters2[2,]
2000
2002
2004
2006
2008
2010
t_vals


ma1
parameters2[3,]
2
1
0
−1
−2
2000
2002
2004
2006
2008
2010
t_vals


ma2
parameters2[4,]
2.0
1.5
1.0
0.5
0.0
−1.0
2000
2002
2004
2006
2008
2010
t_vals


ma3
parameters2[5,]
0.4
0.2
0.0
−0.2
−0.4
2000
2002
2004
2006
2008
2010
t_vals


ma4
0.4
0.2
0.0
−0.2
−0.4
parameters2[6,]
2000
2002
2004
2006
2008
2010
t_vals




```
## integer(0)
```

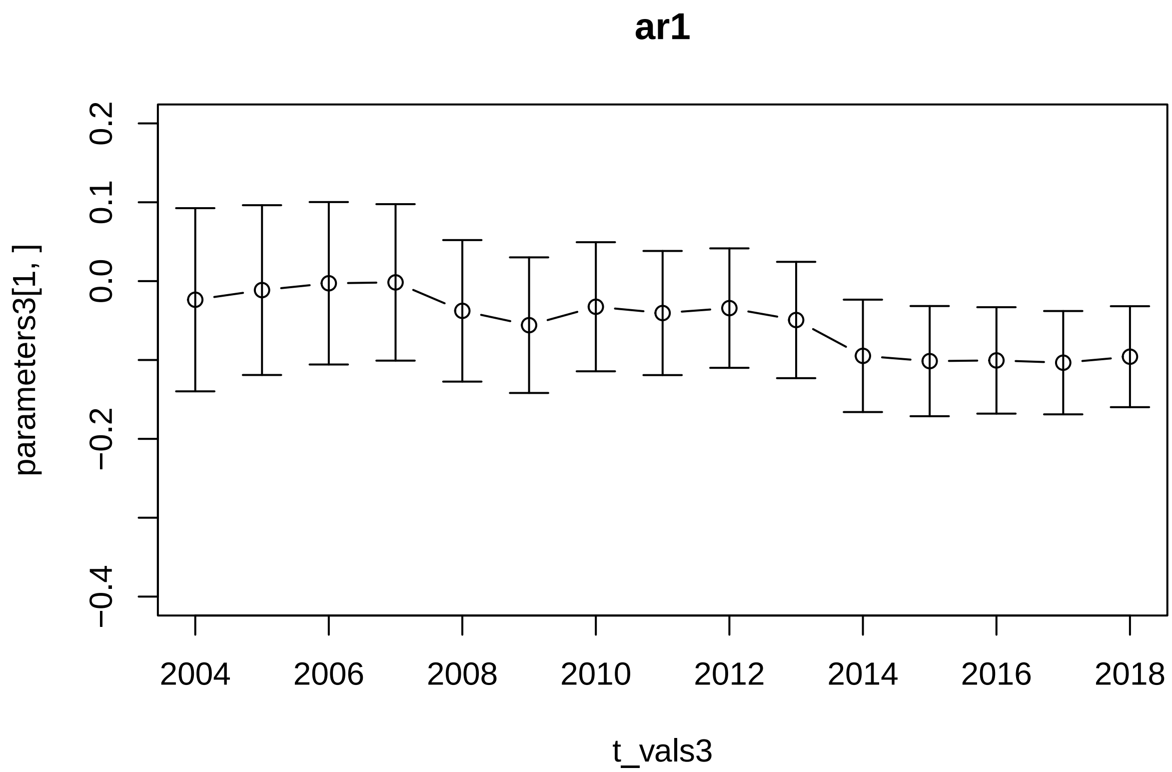
The parameters became highly unstable, therefore, in the following analysis I will be using Arima (2, 1, 0) order.

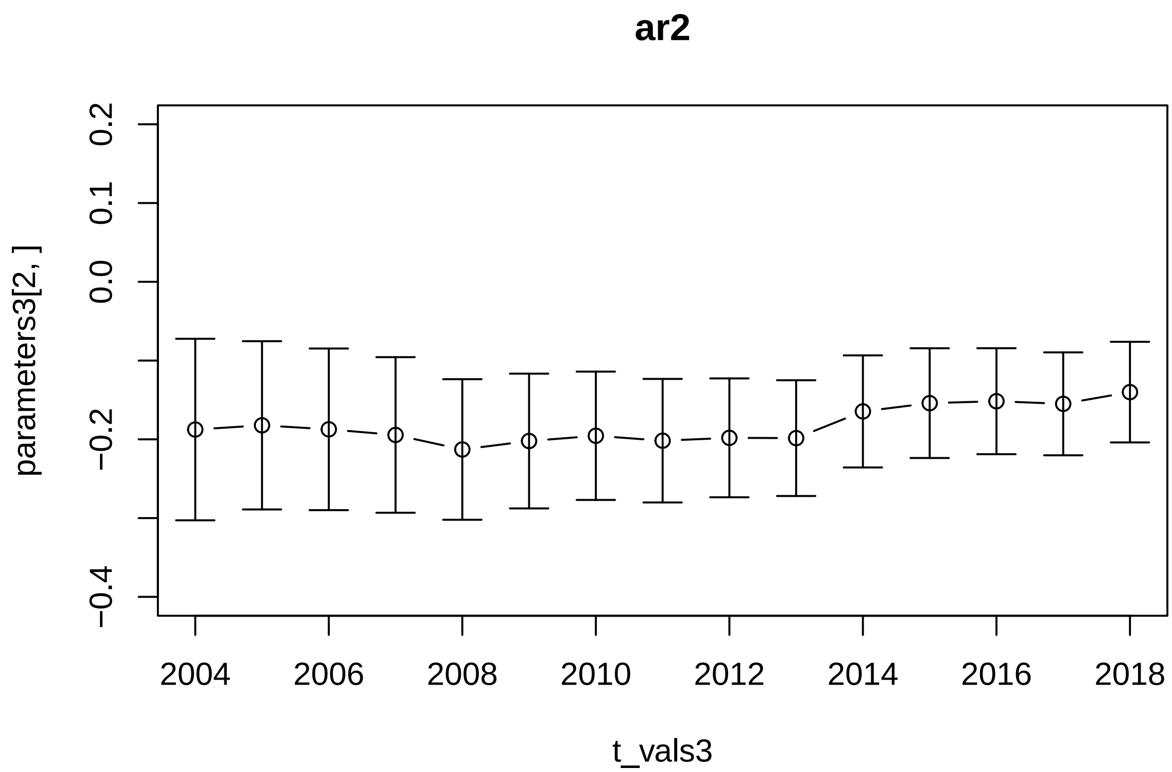
Change finishing point

```
parameters3 =c()
errors3 = c()

s = 2000
for (i in 1:15){
    parameters3 <- cbind(parameters3, coefficients(Arima(window(interest, frequency = 12, start=c(2000,
    errors3 <- cbind(errors3, sqrt(diag(vcov(Arima(window(interest, frequency = 12, start=c(2000, 1), e
)
}

t_vals3 = c(2004:2018)
```

```
{
  plot(t_vals3, parameters3[1, ], type='b', main="ar1", ylim = c(-0.4, 0.2))+
  arrows(x0=t_vals3, y0=parameters3[1, ]-errors3[1, ], x1 = t_vals3, y1=parameters3[1, ]+errors3[1, ],
  plot(t_vals3, parameters3[2, ], type='b', main="ar2", ylim = c(-0.4, 0.2))+
```

```
  arrows(x0=t_vals3, y0=parameters3[2, ]-errors3[2, ], x1 = t_vals3, y1=parameters3[2, ]+errors3[2, ], 
}
```






```
## integer(0)
```

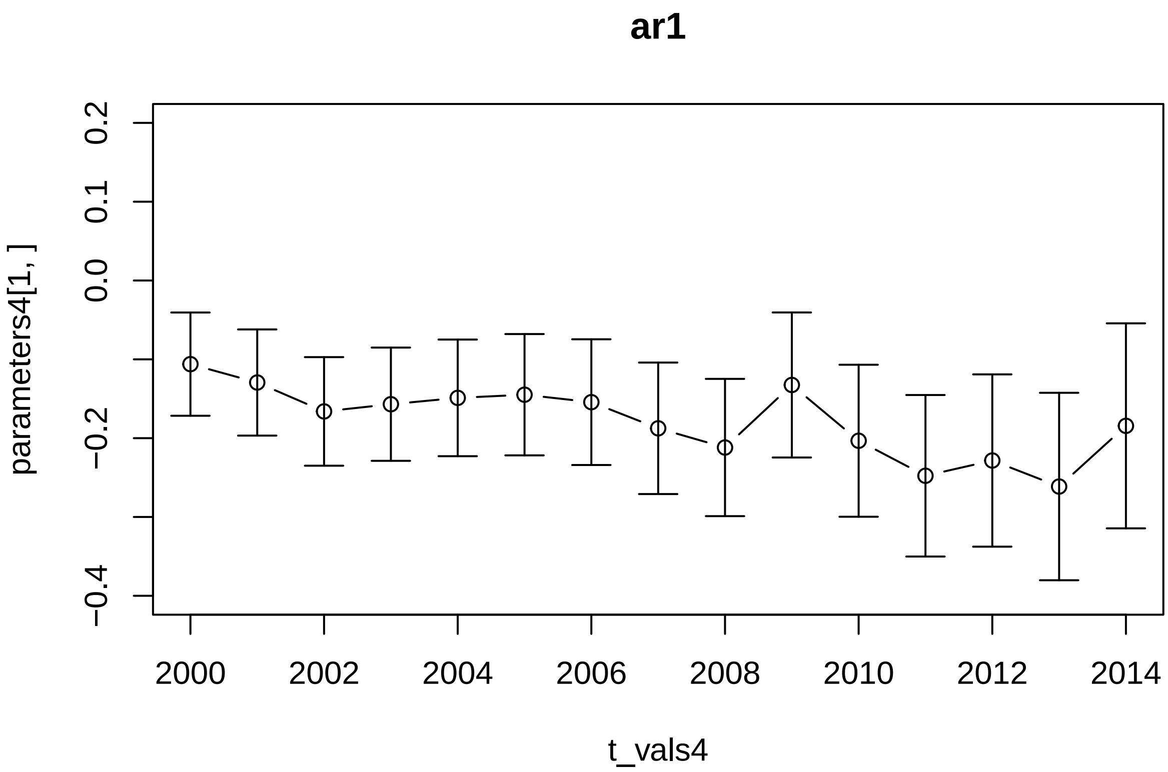
```
cbind('ar1' = parameters3[1,],
      's.e(ar1)' = errors3[1, ],
      'ar2' = parameters3[2,],
      's.e(ar2)' = errors3[2, ]
      )
```

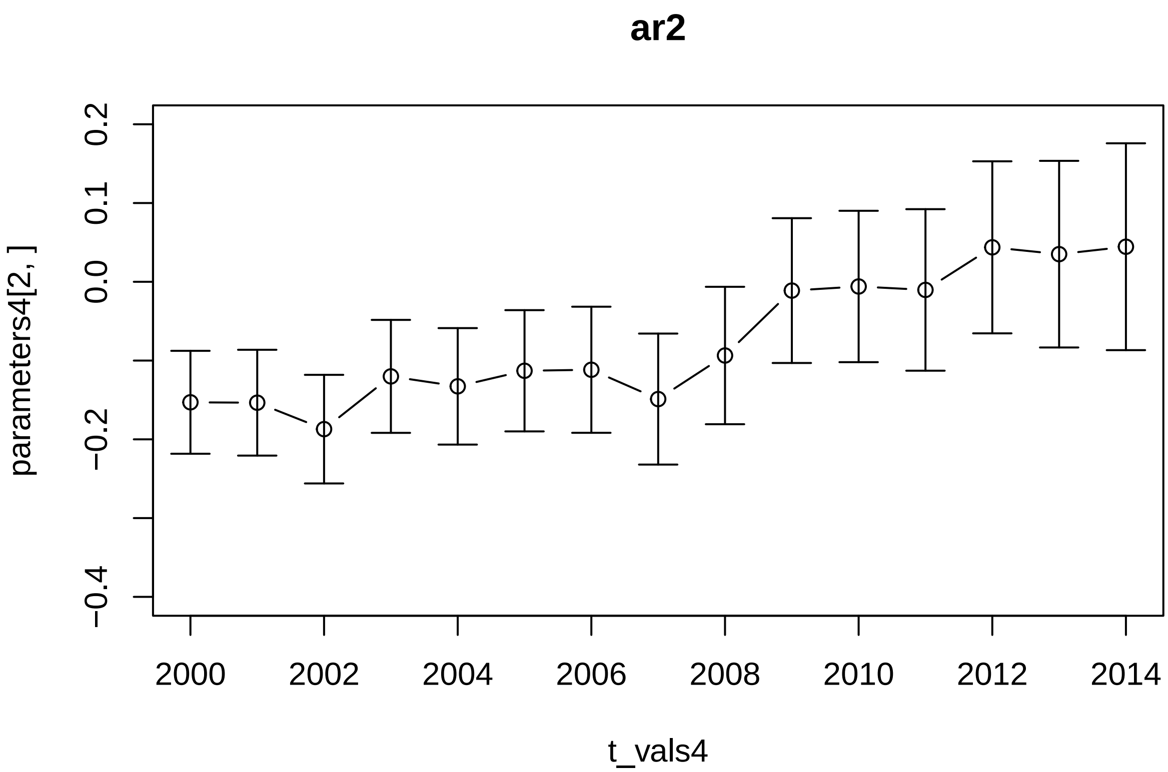
```
##                ar1   s.e(ar1)        ar2   s.e(ar2)
##  [1,] -0.023610537 0.11613062 -0.1875780 0.11525555
##  [2,] -0.011374035 0.10764232 -0.1822147 0.10682038
##  [3,] -0.002729087 0.10300361 -0.1873074 0.10261299
##  [4,] -0.001603560 0.09920253 -0.1944446 0.09877088
##  [5,] -0.037679438 0.08972393 -0.2129046 0.08924006
##  [6,] -0.055854043 0.08596541 -0.2021549 0.08555710
##  [7,] -0.032505438 0.08184439 -0.1954864 0.08148157
##  [8,] -0.040453179 0.07869515 -0.2017301 0.07840645
##  [9,] -0.034212546 0.07575054 -0.1981156 0.07545740
## [10,] -0.049299019 0.07372738 -0.1984735 0.07345834
## [11,] -0.094754388 0.07126987 -0.1645688 0.07113394
## [12,] -0.101428866 0.06984848 -0.1540586 0.06965641
## [13,] -0.100536212 0.06746326 -0.1516280 0.06728149
## [14,] -0.103412586 0.06550531 -0.1549482 0.06533649
## [15,] -0.095821175 0.06400363 -0.1400617 0.06386705
```

Change starting point

```
parameters4 =c()
errors4 = c()

for (i in 1:15){
    parameters4 <- cbind(parameters4, coefficients(Arima(window(interest, frequency = 12, start=c(2000 +
                                )
    errors4 <- cbind(errors4, sqrt(diag(vcov(Arima(window(interest, frequency = 12, start=c(2000 + i, 1)
                            )
}

t_vals4 = c(2000:2014)
```

```
{
  plot(t_vals4, parameters4[1, ], type='b', main="ar1", ylim = c(-0.4, 0.2))+
  arrows(x0=t_vals4, y0=parameters4[1, ]-errors4[1, ], x1 = t_vals4, y1=parameters4[1, ]+errors4[1, ], c
  plot(t_vals4, parameters4[2, ], type='b', main="ar2", ylim = c(-0.4, 0.2))+
  arrows(x0=t_vals4, y0=parameters4[2, ]-errors4[2, ], x1 = t_vals4, y1=parameters4[2, ]+errors4[2, ], c
}
```




ar2
parameters4[2,]
0.2
0.1
0.0
−0.2
−0.4
2000
2002
2004
2006
2008
2010
2012
2014
t_vals4

```
## integer(0)
```