Multivariate Modeling Example

Jackson Cates

9/5/2020

Libraries

```r
library(readr)
library(marima)
library(tsibble)
library(dplyr)
library(tidyr)
library(ggplot2)
library(tidyverse)
library(fable)
library(feasts)
library(urca)
library(gridExtra)
library(forecast)
library(tseries)
library(MTS)
library(vars)
```

Data Set Summary

```r
head(EuStockMarkets)
```

```
## Time Series:
## Start = c(1991, 130)
## End = c(1991, 135)
## Frequency = 260
##               DAX    SMI    CAC   FTSE
## 1991.496 1628.75 1678.1 1772.8 2443.6
## 1991.500 1613.63 1688.5 1750.5 2460.2
## 1991.504 1606.51 1678.6 1718.0 2448.2
## 1991.508 1621.04 1684.1 1708.1 2470.4
## 1991.512 1618.16 1686.6 1723.1 2484.7
## 1991.515 1610.61 1671.6 1714.3 2466.8
```

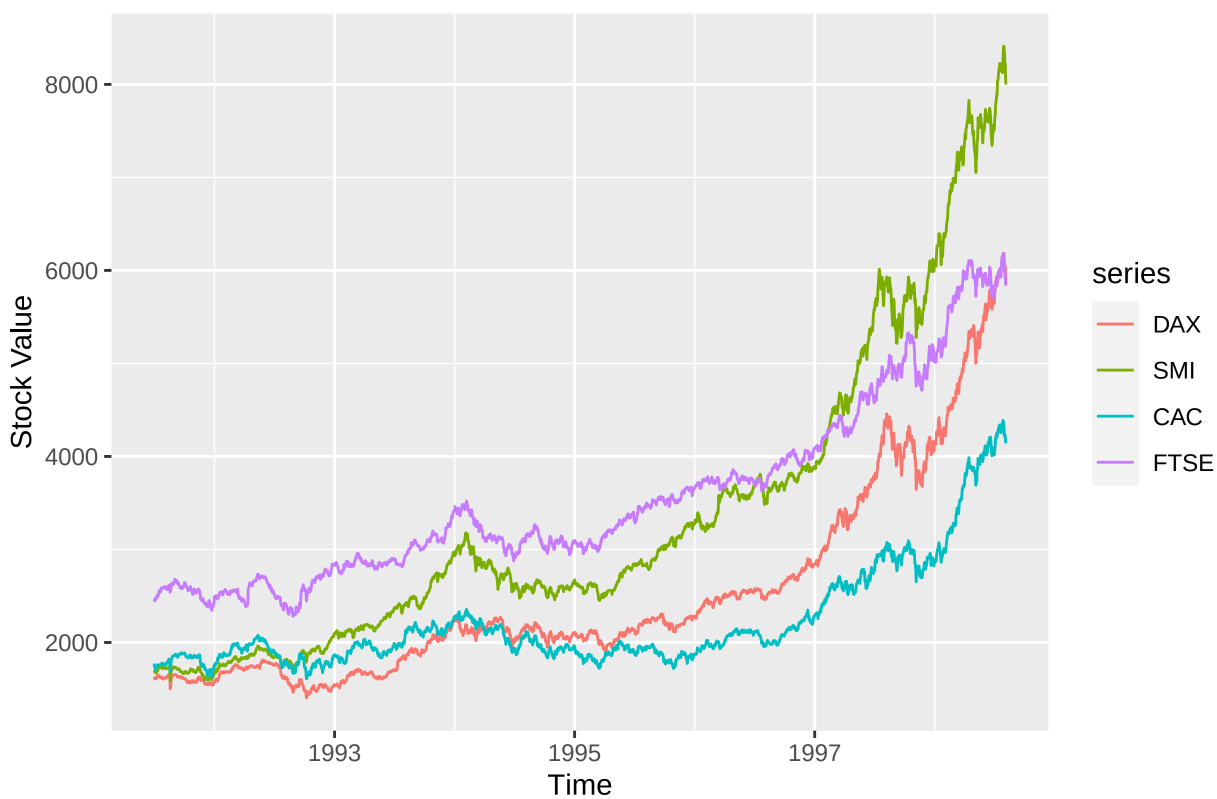
```r
tail(EuStockMarkets)
```

```
## Time Series:
## Start = c(1998, 164)
## End = c(1998, 169)
## Frequency = 260
##               DAX    SMI    CAC   FTSE
## 1998.627 5598.32 7952.9 4041.9 5680.4
## 1998.631 5460.43 7721.3 3939.5 5587.6
```

```
## 1998.635 5285.78 7447.9 3846.0 5432.8
## 1998.638 5386.94 7607.5 3945.7 5462.2
## 1998.642 5355.03 7552.6 3951.7 5399.5
## 1998.646 5473.72 7676.3 3995.0 5455.0
```

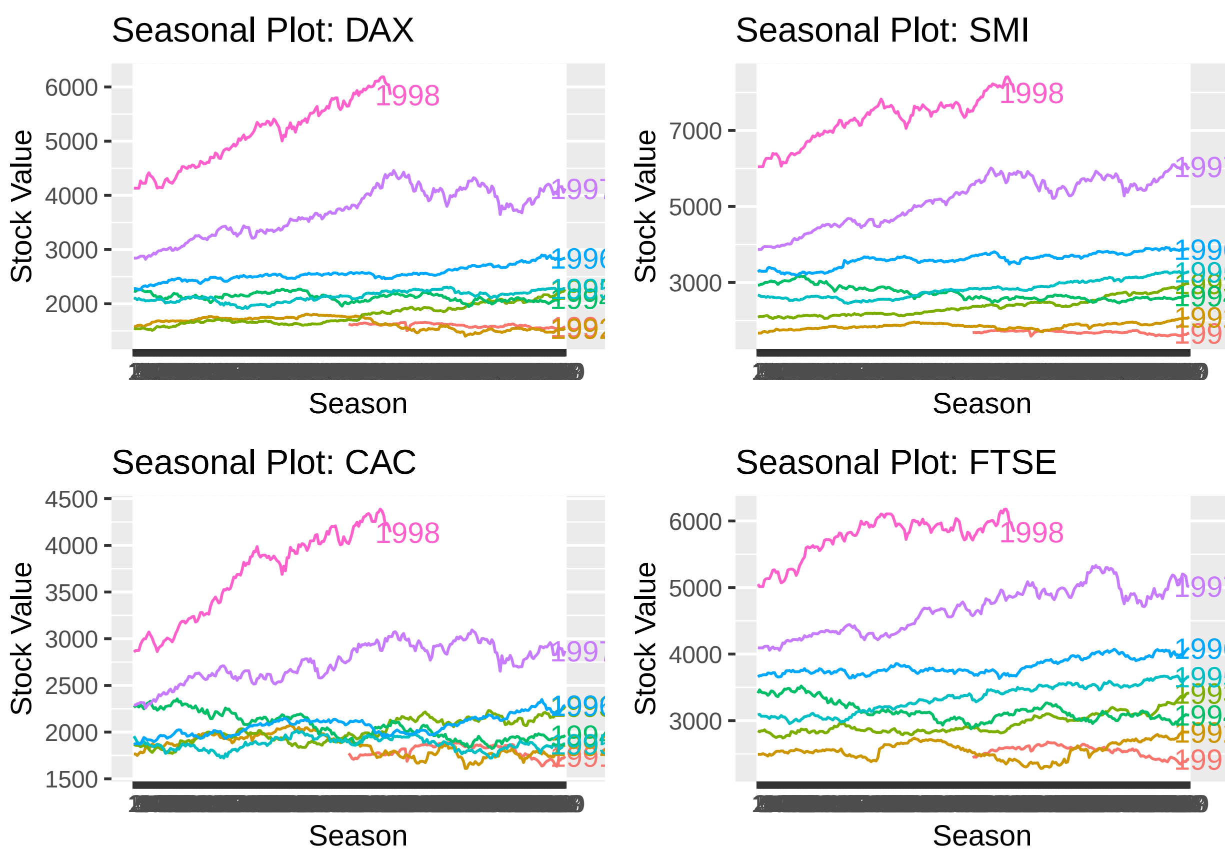
```
# Seperates the data into training and testing data
train = subset(EuStockMarkets, start = 1, end = nrow(EuStockMarkets) - 14)
test = subset(EuStockMarkets, start = nrow(EuStockMarkets) - 14, end = nrow(EuStockMarkets))

# Makes the time plot
train %>% autoplot() + ylab("Stock Value")
```



Testing Stationary

```
plot1 = ggseasonplot(train[,"DAX"], year.labels = T, main = "Seasonal Plot: DAX", ylab = "Stock Value")
plot2 = ggseasonplot(train[,"SMI"], year.labels = T, main = "Seasonal Plot: SMI", ylab = "Stock Value")
plot3 = ggseasonplot(train[,"CAC"], year.labels = T, main = "Seasonal Plot: CAC", ylab = "Stock Value")
plot4 = ggseasonplot(train[,"FTSE"], year.labels = T, main = "Seasonal Plot: FTSE", ylab = "Stock Value
grid.arrange(plot1, plot2, plot3, plot4, ncol = 2, nrow = 2)
```



Unitroot Test

H_O : The data is stationary

H_A : The data is not stationary

```
kpss.test(train[,"DAX"])[3]
```

```
## Warning in kpss.test(train[, "DAX"]): p-value smaller than printed p-value

## $p.value
## [1] 0.01
```

```
kpss.test(train[,"SMI"])[3]
```

```
## Warning in kpss.test(train[, "SMI"]): p-value smaller than printed p-value

## $p.value
## [1] 0.01
```

```
kpss.test(train[,"CAC"])[3]
```

```
## Warning in kpss.test(train[, "CAC"]): p-value smaller than printed p-value

## $p.value
## [1] 0.01
```

```r
kpss.test(train[,"FTSE"])[3]
```

```
## Warning in kpss.test(train[, "FTSE"]): p-value smaller than printed p-value

## $p.value
## [1] 0.01
```

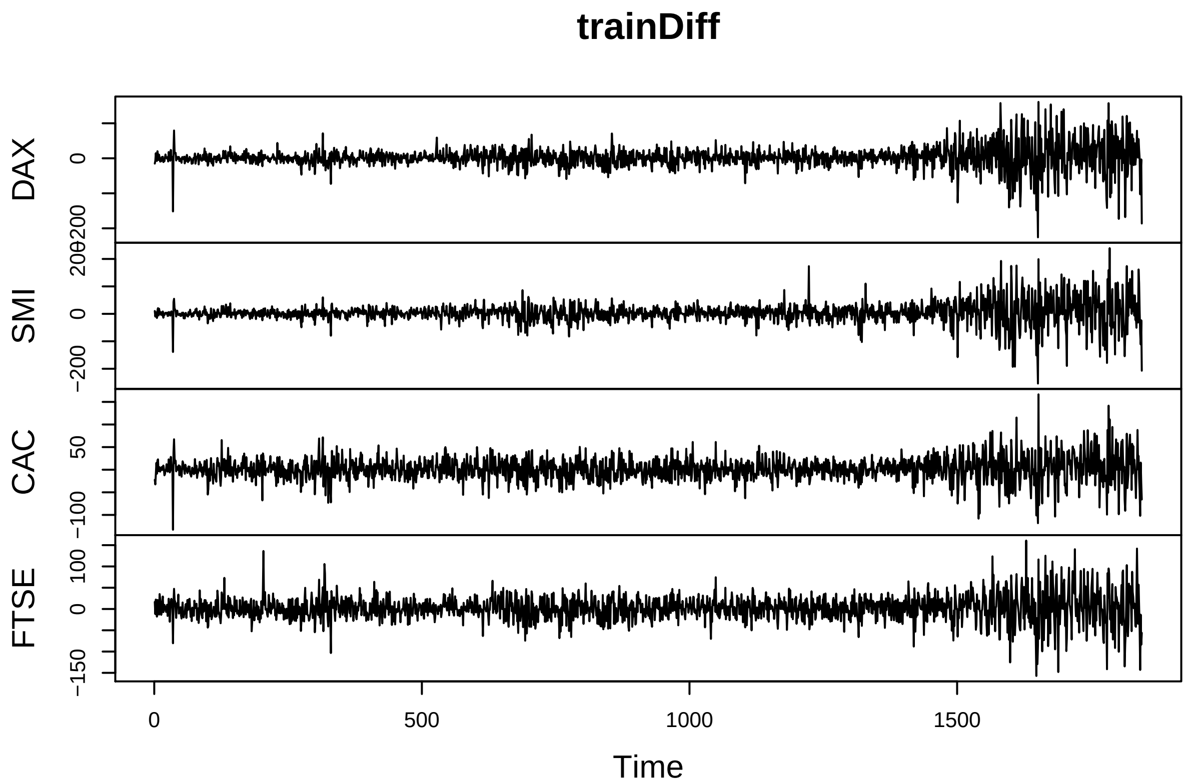
Differencing

We are going to take the change between consecutive observations

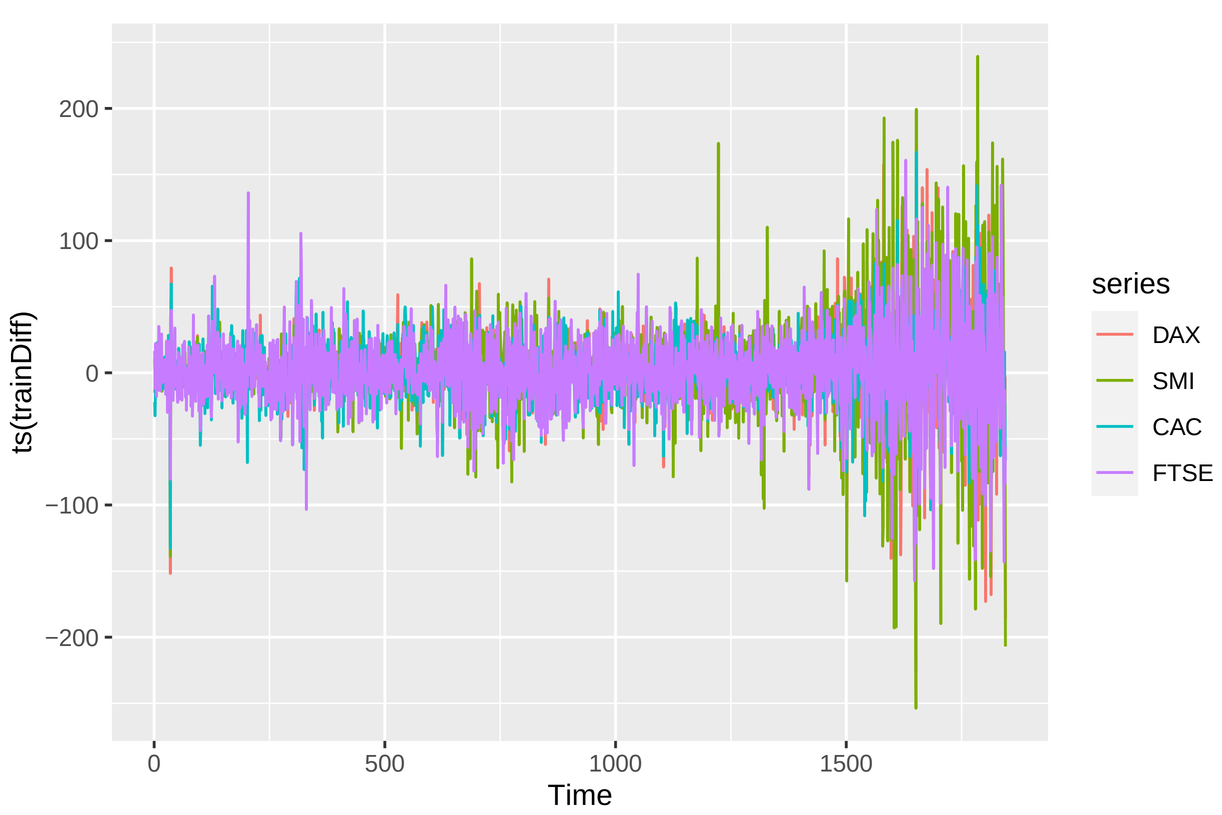
$y'_t = y_t - y_{t-1}$

```
trainDiff = train %>% diffM()

plot.ts(trainDiff)
```

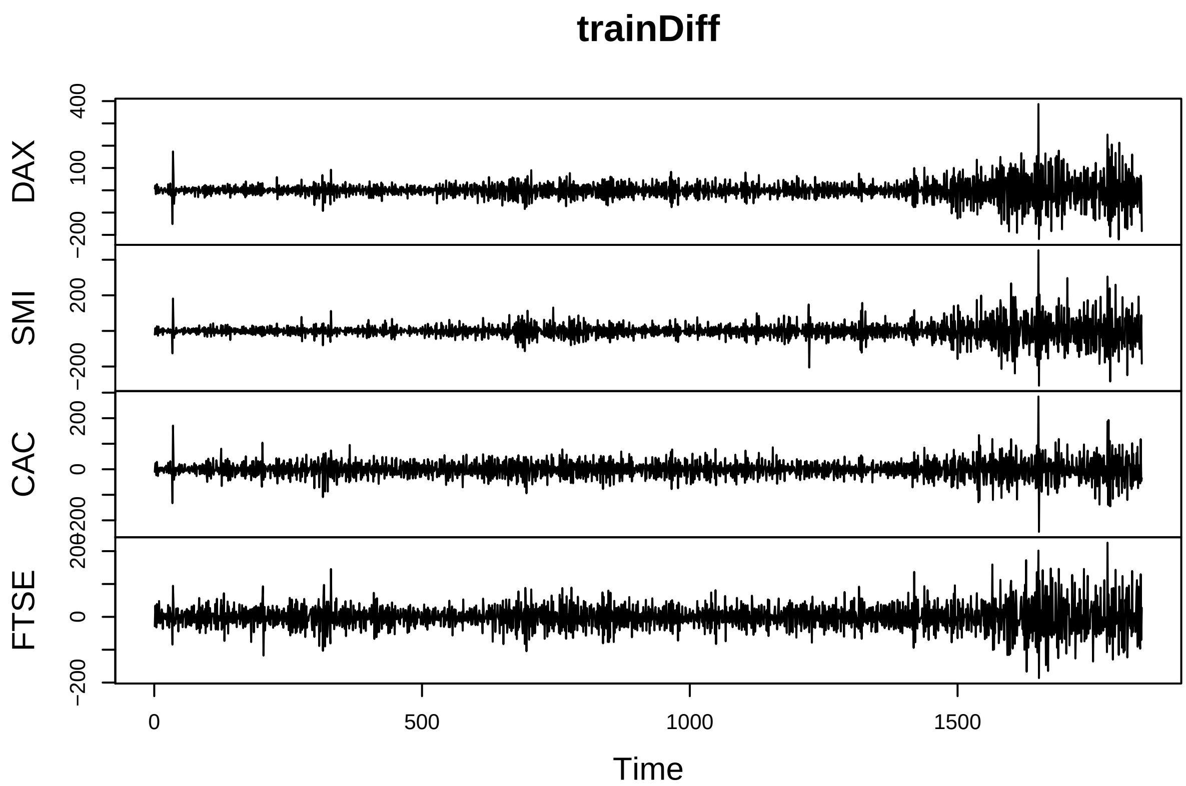



```
autoplot(ts(trainDiff))
```

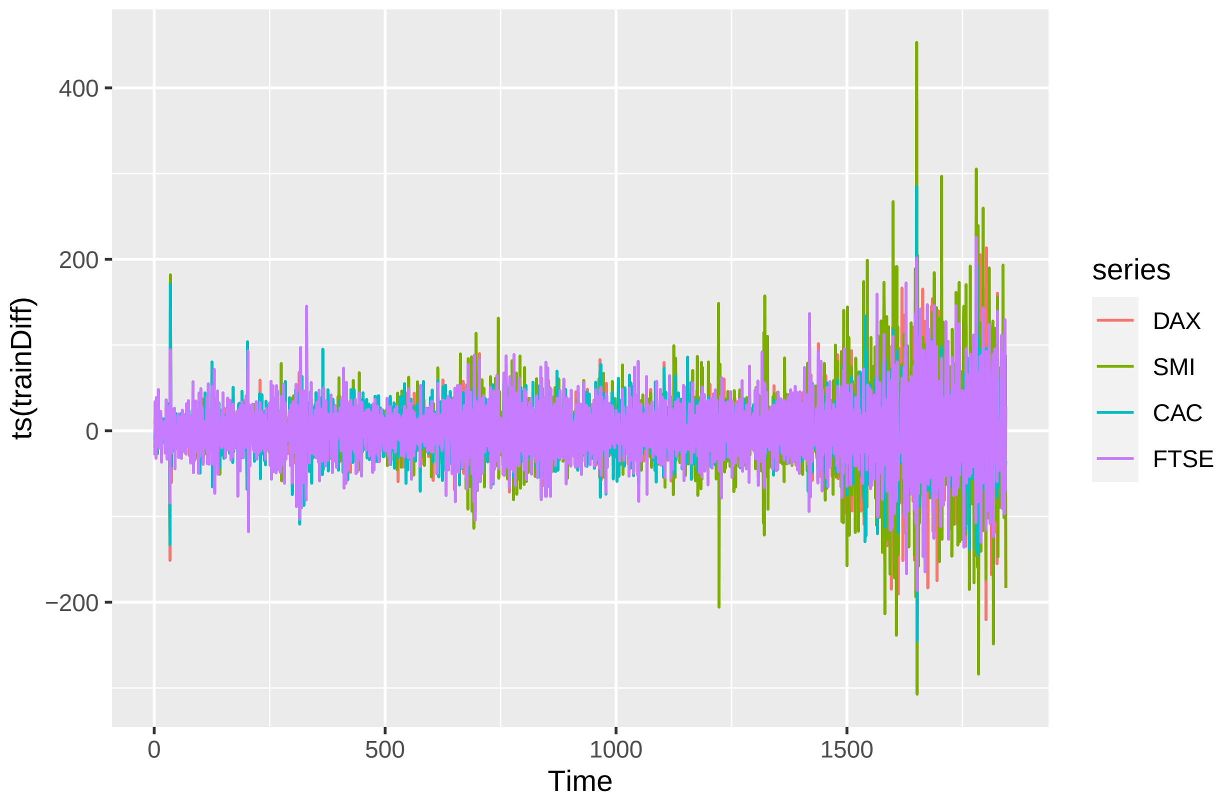



```
trainDiff = trainDiff %>% diffM()

plot.ts(trainDiff)
```




```
autoplot(ts(trainDiff))
```



Unitroot Test

H_O : The data is stationary

H_A : The data is not stationary

```
kpss.test(trainDiff[,"DAX"])[3]
```

```
## Warning in kpss.test(trainDiff[, "DAX"]): p-value greater than printed p-value

## $p.value
## [1] 0.1
```

```
kpss.test(trainDiff[,"SMI"])[3]
```

```
## Warning in kpss.test(trainDiff[, "SMI"]): p-value greater than printed p-value

## $p.value
## [1] 0.1
```

```
kpss.test(trainDiff[,"CAC"])[3]
```

```
## Warning in kpss.test(trainDiff[, "CAC"]): p-value greater than printed p-value

## $p.value
## [1] 0.1
```

```
kpss.test(trainDiff[,"FTSE"])[3]
```

```
## Warning in kpss.test(trainDiff[, "FTSE"]): p-value greater than printed p-value
```

```
## $p.value
## [1] 0.1
```

Selecting p and q

Eccm function in R gives a matrix of multivariate Ljung-Box statistics of a vector time series.

H_O : The model does not show a lack of fit (that the autocorrelations (for the chosen lags) in the population from which the sample is taken are all zero).

H_A : The model does show a lack of fit

```r
Eccm(trainDiff)
```

```
## p-values table of Extended Cross-correlation Matrices:
## Column: MA order
## Row   : AR order
##        0      1      2      3      4      5      6
## 0 0.0000 0.0000 0.0000 0.0000 0.0000 0.0001 0.0075
## 1 0.0000 0.0000 0.0092 0.0162 0.5624 0.7776 0.0726
## 2 0.0000 0.0071 0.1537 0.0096 0.5264 0.4244 0.1895
## 3 0.0000 0.3605 1.0000 0.9736 0.1257 0.1246 0.8178
## 4 0.0000 0.7653 1.0000 0.9995 0.7286 0.7455 0.8409
## 5 0.0000 1.0000 1.0000 1.0000 1.0000 0.9995 0.9980
```

Fitting the model

p	q	aic	bic	refAic	refBic	ref2Aic	ref2Bic
2	2	25.23482	25.43837	25.54427	25.71489	NA	NA
3	1	25.31184	25.51539	25.32326	25.41007	25.61058	25.79916
3	2	58.92585	59.17729	41.12731	41.36678	25.65966	25.80334

```r
fit = VARMA(trainDiff, p = 2, q = 2)
```

```
## Number of parameters:  68
## initial estimates:  -0.0114 -0.0104 -0.004 -0.0301 -0.5612 -0.2874 -0.0471 0.4812 0.0511 -0.0453 -0.0
## Par. lower-bounds:  -1.5312 -1.8423 -1.2397 -1.4672 -0.8903 -0.4931 -0.3766 0.1971 -0.0383 -0.1138 -0
## Par. upper-bounds:  1.5084 1.8215 1.2317 1.407 -0.2321 -0.0817 0.2825 0.7653 0.1405 0.0232 0.0943 0.0
## Final   Estimates:  -0.006881378 -0.01689296 -0.002222269 -0.02591792 -0.5638624 -0.2881675 -0.04650
```

```
## Warning in sqrt(diag(solve(Hessian))): NaNs produced
```

```
##
## Coefficient(s):
##        Estimate  Std. Error    t value Pr(>|t|)
## DAX  -6.881e-03   8.887e-04 -7.743e+00 9.77e-15 ***
## SMI  -1.689e-02   4.370e-03 -3.865e+00 0.000111 ***
## CAC  -2.222e-03   3.464e-03 -6.420e-01 0.521195
## FTSE -2.592e-02   2.761e-03 -9.386e+00  < 2e-16 ***
## DAX  -5.639e-01   2.053e-02 -2.747e+01  < 2e-16 ***
## SMI  -2.882e-01   2.133e-02 -1.351e+01  < 2e-16 ***
## CAC  -4.651e-02   1.814e-02 -2.563e+00 0.010372 *
## FTSE  4.834e-01   6.267e-03  7.714e+01  < 2e-16 ***
## DAX   5.301e-02   3.526e-02  1.504e+00 0.132702
## SMI  -4.366e-02   2.707e-02 -1.613e+00 0.106724
## CAC  -5.247e-03   1.395e-02 -3.760e-01 0.706765
## FTSE -4.616e-02   1.279e-02 -3.610e+00 0.000307 ***
## DAX   3.238e-03          NA         NA       NA
## SMI  -6.255e-01   2.836e-02 -2.205e+01  < 2e-16 ***
## CAC   8.754e-02   3.890e-02  2.250e+00 0.024438 *
## FTSE  2.376e-01   2.968e-02  8.004e+00 1.11e-15 ***
## DAX   1.573e-01   4.635e-02  3.394e+00 0.000688 ***
## SMI   1.825e-02   3.237e-02  5.640e-01 0.572892
## CAC  -4.797e-02   3.329e-02 -1.441e+00 0.149642
## FTSE -8.935e-02   2.659e-02 -3.361e+00 0.000777 ***
## DAX  -1.549e-01   2.048e-02 -7.563e+00 3.95e-14 ***
## SMI  -1.199e-01   1.945e-02 -6.165e+00 7.03e-10 ***
## CAC  -3.769e-01   1.367e-02 -2.757e+01  < 2e-16 ***
## FTSE  1.763e-01   1.343e-02  1.313e+01  < 2e-16 ***
## DAX   2.748e-02   3.129e-02  8.780e-01 0.379761
## SMI  -6.095e-02   2.230e-02 -2.734e+00 0.006263 **
## CAC   8.709e-02   2.147e-02  4.056e+00 4.99e-05 ***
## FTSE -4.762e-03   4.355e-03 -1.094e+00 0.274165
## DAX  -2.567e-02   3.080e-02 -8.330e-01 0.404597
## SMI  -2.274e-01   2.480e-02 -9.171e+00  < 2e-16 ***
## CAC   3.554e-02   2.604e-02  1.365e+00 0.172361
## FTSE -2.181e-01   1.972e-02 -1.106e+01  < 2e-16 ***
## DAX   5.810e-02   2.564e-02  2.266e+00 0.023477 *
```

```
## SMI  -6.543e-02   2.508e-02 -2.609e+00 0.009082 **
## CAC  -6.670e-02   2.218e-02 -3.008e+00 0.002634 **
## FTSE  8.188e-02   1.606e-02  5.099e+00 3.42e-07 ***
##      -4.011e-01   2.674e-03 -1.500e+02  < 2e-16 ***
##       1.820e-01   5.620e-05  3.238e+03  < 2e-16 ***
##       5.437e-02          NA         NA       NA
##      -3.808e-01   1.175e-05 -3.241e+04  < 2e-16 ***
##      -6.349e-01   4.610e-04 -1.377e+03  < 2e-16 ***
##      -1.866e-01   5.381e-05 -3.468e+03  < 2e-16 ***
##       3.504e-02          NA         NA       NA
##       3.759e-01   2.490e-05  1.510e+04  < 2e-16 ***
##       2.300e-02   1.672e-02  1.375e+00 0.168978
##      -3.566e-01          NA         NA       NA
##      -1.023e-01   2.626e-04 -3.895e+02  < 2e-16 ***
##      -9.294e-02          NA         NA       NA
##      -2.302e-01   2.426e-02 -9.489e+00  < 2e-16 ***
##      -5.774e-01   9.259e-06 -6.236e+04  < 2e-16 ***
##       2.404e-01          NA         NA       NA
##       1.819e-01   2.030e-04  8.960e+02  < 2e-16 ***
##       1.349e-01   1.341e-02  1.006e+01  < 2e-16 ***
##       4.871e-02          NA         NA       NA
##      -5.487e-01   2.125e-05 -2.582e+04  < 2e-16 ***
##      -7.772e-02   3.319e-05 -2.342e+03  < 2e-16 ***
##      -1.843e-01   1.612e-02 -1.143e+01  < 2e-16 ***
##      -4.981e-02          NA         NA       NA
##      -3.532e-01   4.226e-06 -8.357e+04  < 2e-16 ***
##       7.291e-02   8.074e-05  9.030e+02  < 2e-16 ***
##       2.706e-02   2.290e-02  1.182e+00 0.237341
##       1.575e-01          NA         NA       NA
##      -8.951e-02          NA         NA       NA
##      -5.424e-01   2.028e-06 -2.675e+05  < 2e-16 ***
##      -1.278e-01   2.929e-02 -4.364e+00 1.28e-05 ***
##      -1.111e-01          NA         NA       NA
##       1.517e-01          NA         NA       NA
##      -4.097e-01   4.738e-06 -8.647e+04  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ---
## Estimates in matrix form:
## Constant term:
## Estimates:  -0.006881378 -0.01689296 -0.002222269 -0.02591792
## AR coefficient matrix
## AR( 1 )-matrix
##          [,1]   [,2]    [,3]   [,4]
## [1,] -0.56386 -0.288 -0.0465  0.483
## [2,]  0.00324 -0.626  0.0875  0.238
## [3,] -0.15485 -0.120 -0.3769  0.176
## [4,] -0.02567 -0.227  0.0355 -0.218
## AR( 2 )-matrix
##        [,1]    [,2]     [,3]     [,4]
## [1,] 0.0530 -0.0437 -0.00525 -0.04616
## [2,] 0.1573  0.0183 -0.04797 -0.08935
## [3,] 0.0275 -0.0610  0.08709 -0.00476
## [4,] 0.0581 -0.0654 -0.06670  0.08188
```

```
## MA coefficient matrix
## MA( 1 )-matrix
##         [,1]    [,2]    [,3]   [,4]
## [1,]  0.4011 -0.1820 -0.0544 0.3808
## [2,] -0.0230  0.3566  0.1023 0.0929
## [3,] -0.1349 -0.0487  0.5487 0.0777
## [4,] -0.0271 -0.1575  0.0895 0.5424
## MA( 2 )-matrix
##       [,1]   [,2]   [,3]    [,4]
## [1,] 0.635 0.1866 -0.035 -0.3759
## [2,] 0.230 0.5774 -0.240 -0.1819
## [3,] 0.184 0.0498  0.353 -0.0729
## [4,] 0.128 0.1111 -0.152  0.4097
##
## Residuals cov-matrix:
##          [,1]      [,2]     [,3]     [,4]
## [1,] 983.1929  870.1295 591.6951 621.0185
## [2,] 870.1295 1443.8933 613.8826 693.7454
## [3,] 591.6951  613.8826 665.9477 505.9309
## [4,] 621.0185  693.7454 505.9309 894.6722
## ----
## aic=  25.23482
## bic=  25.43837
```