DTU Library

Multivariate Time Series Estimation using marima

Spliid, Henrik

Published in:
Symposium i anvendt statistik 2016

Publication date:
2016

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Spliid, H. (2016). Multivariate Time Series Estimation using marima. In P. Linde (Ed.), *Symposium i anvendt statistik 2016* (pp. 108-123). Danmarks Statistik. http://www.dst.dk/da/Statistik/Publikationer/VisPub?cid=025874

Multivariate Time Series Estimation using marima

Henrik Spliid, DTU Compute

A computer program, called `marima`, written in the `open source` language, *R*, has been developed. Some of `marima`'s facilities and ideas are presented in the following.

1 The multivariate ARMA(p,q) model

Let t denote (discrete) time. Consider a k-variate random vector y_t of *observations* and, correspondingly, a k-variate random vector, u_t, of *unknown innovations* with zero mean:

$$y_t = \begin{pmatrix} y_{1,t} \\ y_{2,t} \\ \cdots \\ y_{k,t} \end{pmatrix} \text{ , and } u_t = \begin{pmatrix} u_{1,t} \\ u_{2,t} \\ \cdots \\ u_{k,t} \end{pmatrix} \text{ , } \quad t = \{1, 2, \ldots, n\} \tag{1}$$

Further suppose that the random vector y_t is generated through the model

$$y_t + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} = u_t + \theta_1 u_{t-1} + \cdots + \theta_q u_{t-q} \tag{2}$$

where the coefficient matrices $\phi_1, \ldots, \phi_p$ and $\theta_1, \ldots, \theta_q$ are all of dimension $k \times k$.

The series u_t is without autocorrelation, but the individual elements (coordinates) need not be, for example, uncorrelated. The covariance matrix of u_t is $\text{Var}(u_t) = \Sigma_u$, and it is assumed to be independent of t. Most often u_t is assumed to be normally distributed.

The lefthand side of equation (2) is called the *autoregressive* or AR part of the model, while the righthand side is called the *moving average* or MA part of the model. p is the order of the AR part, and q is the order of the MA part. The model is called the *multivariate* $ARMA(p, q)$ model.

The model is often extended to include external non-random regression variables.

A further and somewhat more detailed description of **`marima`** is available from the repository where **`marima`** is located (contact: hspl[at]dtu.dk) .

2 Operator form of the ARMA(p,q) model

2.1 Matrix polynomials used in marima

Define the *k-variate backwards shift operator* B such that if B is multiplied on a time indexed k-variate random variable, the result is to be interpreted as the variable lagged a single time step.

And, in general, lagging r time steps is accomplished using:

$$B^r y_t = y_{t-r}$$

Now, introduce the operator B into model (2) which then can be written as

$$(I + \phi_1 B + \cdots + \phi_p B^p) y_t = (I + \theta_1 B + \cdots + \theta_q B^q) u_t \ ,$$

where I is the $k \times k$ unity matrix. Also, define the *matrix polynomials* $\phi(B) = I + \phi_1 B + \cdots + \phi_p B^p$ and $\theta(B) = I + \theta_1 B + \cdots + \theta_q B^q$. This leads to the general multivariate ARMA(p,q) model in operator form

$$\phi(B) y_t = \theta(B) u_t \tag{3}$$

Most often the averages of the k variables in y_t are subtracted before the model estimation. When reconstructing or forecasting the measured series analysed by `marima`, the averages of the original data can be reintroduced.

2.2 Averages and their representation in the arma model

Suppose that the vector y_t has been (for example) means-adjusted, such that $y_t = v_t - \eta$, where v_t are the original measurements, and η is the (estimated) vector of averages: $\eta \simeq E\{v_t\}$. Suppose now, that $\phi(B)y_t = \theta(B)u_t$ or, equivalently:

$$\phi(B)(v_t - \eta) = \theta(B)u_t$$

$$\phi(B)v_t = \mu + \theta(B)u_t \quad ; \quad \mu = \phi(B)\eta$$

$$\mu = \left[\sum_{i=0}^{p} \phi_i\right] \eta \tag{4}$$

Note that equation (4) applies for any transformation of the form $y_t = z_t - \eta$.

2.3 Inverse matrix polynomials and alternative forms

It is convenient to be able to write model (3) in the following form:

$$y_t = u_t + \psi_1 u_{t-1} + \cdots + \psi_\ell u_{t-\ell} + \cdots = \psi(B) u_t \tag{5}$$

Given the model (3), the model (5) can be determined if we are able to calculate the *left inverse polynomial* $\phi^{-1}(B)$ of the polynomial $\phi(B)$, such that

$$\phi^{-1}(B)\phi(B) = I \tag{6}$$

In general, if $\phi(B)$ is a finite order polynomial, the inverse, $\phi^{-1}(B)$ is of infinite order.

Pre-multiplying with $\phi^{-1}(B)$ on both sides of the equals sign in model (3) gives

$$y_t = \phi^{-1}(B)\theta(B)u_t = \psi(B)u_t = \sum_{i=0}^{\infty} \psi_i u_{t-i} \tag{7}$$

This form is called the *random shock* form and, generally (if the model includes a nonzero AR term), the polynomial $\psi(B)$ is of infinite length with decreasing coefficients, such that $\psi(\ell) \to 0$ for $\ell \to \infty$. If, more precisely, $\sum_{i=0}^{\infty} \psi(z)$ converges for all $|z| \leq 1$ the model (7) is said to be *stationary*.

Similarly if $\theta^{-1}(B)$ is the left inverse of $\theta(B)$, we may pre-multiply with $\theta^{-1}(B)$ on both sides of the equals sign in model (3). This gives the socalled *inverse form*:

$$\pi(B)y_t = u_t \tag{8}$$

where $\pi(B) = \theta^{-1}(B)\phi(B)$. If, similarly, $\sum_{i=0}^{\infty} \pi(z)$ converges for all $|z| \leq 1$ the model (8) is said to be *invertible*.

3 Simple operations for matrix polynomials

Note that, the 0'th order coefficient matrix of $\phi(B)$ (that is ϕ_0) and of $\theta(B)$ are $k \times k$ unity matrices.

The *marima* package includes routines for inverting and multiplying matrix polynomials, namely `pol.inv` and `pol.mul`.

The **left inverse** of `phi` is computed as, say, `inv.phi <- pol.inv(phi,L)` which will result in an array, `inv.phi`, of dimension $k \times k \times (1 + L)$ holding the $k \times k$ unity matrix followed by the first `L` matrix coefficients of the left inverse of `phi`.

The **product** of two matrix polynomials, $\phi(B)$ and $\theta(B)$, is computed as `pol.mul(phi,theta,L)` which will result in an array of dimension $k \times k \times (1+L)$ holding the $k \times k$ unity matrix followed by the first `L` matrix coefficients of the product $\phi(B)\theta(B)$.

Equation (7) can be carried out using, for example, `psi<-pol.mul(pol.inv(phi,L), theta,L)` giving the unity matrix of order $k \times k$ followed by the first `L` matrix terms of the $\psi(.)$ polynomium. The array `psi` will have dimension $k \times k \times (1 + L)$.

4 Differencing multivariate time series

Often it is convenient to do differencing, such as $z_t = y_t - y_{t-s}$, that is differencing over s time steps. A routine called `define.dif` included in the marima package can perform (practically) all kinds of differencing for a multivariate time series.

4.1 Single time step differencing

The polynomial $\nabla(B) = (I - B)$ can be used to difference the time series one time step, for example

$$z_t = \nabla(B)y_t = y_t - y_{t-1} \tag{9}$$

If, for example, $y_t = \{y_{1,t}, y_{2,t}\}^T$ is bivariate then the ∇ polynomial for differencing both variables once is

$$\nabla(B) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} B & 0 \\ 0 & B \end{pmatrix}$$

4.2 Seasonal differencing

The polynomial $\nabla(B^s) = (I - B^s)$ is used if a seasonal differencing with seasonality s is wanted:

$$z_t = \nabla(B^s)y_t = y_t - y_{t-s} \tag{10}$$

The polynomial for s timesteps seasonal differencing is

$$\nabla(B^s) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} B^s & 0 \\ 0 & B^s \end{pmatrix}$$

4.3 Mixed differencing

When a multivariate time series is at hand, it may be necessary to difference the individual series differently.

Suppose again, that $y_t = \{y_{1,t}, y_{2,t}\}^T$ is bivariate and that we want to difference over time periods $s = \{s_1, s_2\}$. Then we may define, a little more generally,

$$\nabla(B^s) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} B^{s_1} & 0 \\ 0 & B^{s_2} \end{pmatrix}$$

and

$$\nabla(B^s)\begin{pmatrix} y_{1,t} \\ y_{2,t} \end{pmatrix} = \begin{pmatrix} y_{1,t} \\ y_{2,t} \end{pmatrix} - \begin{pmatrix} y_{1,t-s_1} \\ y_{2,t-s_2} \end{pmatrix}$$

The routine `define.dif(...)` can perform mixed differencing of a multivariate series. The routine `define.sum(...)` included in the marima package does the reverse, that is summing a multivariate series.

4.4 Aggregated model

Suppose the differencing polynomium used (before analysing the data by `marima`), is

$$\nabla(B) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} B - \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} B^{12}$$

The differenced series is called z_t, and $z_t = \nabla(B)y_t$. Suppose now that z_t is analysed by `marima` and that the estimated model for z_t is $\widehat{\phi}(B)z_t = \widehat{\theta}(B)u_t$. The estimated aggregated (nonstationary) model for the observed time series, y_t, is then $\widehat{\phi}(B)\nabla(B)y_t = \widehat{\theta}(B)u_t$.

5 Model selection in marima

5.1 Defining the marima model

Defining models in `marima` is done by creating 0/1-indicator arrays corresponding to the ar-part and the ma-part of the model. These arrays are organised the same way as the model polynomiums wanted. Suppose the ar-indicator array is called `ar.pattern`. Then the value at position `ar.pattern[i,j,`ℓ`]` is the indicator for the $\{i, j\}$'th element in the lag=ℓ ar-parameter matrix $\phi_\ell = \{\phi_{i,j}\}_\ell$.

The value 1 (one) indicates that a parameter is to be estimated at that position. The value 0 indicates that the parameter corresponding to that position is 0. The function `define.model` can be used for setting up the indicator arrays properly.

Examples of the use of `define.model` are obtained with

```
library(marima)
example(define.model).
```

6 Estimation and identification of a marima model

As described by Spliid (1983) the estimation of the model is done with a pseudo-regression method. In the present implementation the `R` procedure `lm(...)` is used. This enables `marima` to utilize the procedure `step(...)` in order to search for a good model in a stepwise manner. The key parameter in `step` is the k-factor used in Akaike's criterion (where k=2 is Akaike's suggestion).

The k-factor can be set when calling `marima` by specifying the input parameter `penalty` to an alternative value.

After having estimated the `marima`-model as defined with the use of, for example, `define.model`, the value `penalty=0` causes `marima` to do no search for a (reduced) model. If `penalty=2` the usual AIC is used to identify a reduced model and in a stepwise manner.

This is repeated a few times and from then on, `marima` iterates on the selected model until (hopefully) convergence.

Experience has shown that a first choice `penalty=1` often results in a model which gives a good overview of which coefficients are the most important ones and which are less important. Approximate F-test values for the individual parameter estimates are given in the output object, and they serve the same purpose.

7 Australian firearms legislation

Baker & McPhedran (2007) discuss the effect of the Australian firearms legislation of (implemented) 1997 on death rates. Four different (maybe related?) death rates (firearm suicides, firearm homicides, other suicides and other homicides) are considered. Baker & McPhedran analysed the data by conventional univariate ARIMA models with separate models for each of the four death rates. All models estimated were univariate arma(1,1) models, i.e. arma models of order (ar=1,ma=1) and without differerencing.

Here, we shall illustrate the use af `marima` along the same lines, although it is by no means claimed that the results obtained are optimal or represent the best analysis of these data.

7.1 Data

The data for the study can be accessed using, for example,
`library(marima); data(australian.killings); all.data <- austr` .

The data.frame `austr` has the following appearance, in that the last 10 lines correspond to not observed future values:

```
  Year  suic.fire homi.fire  suic.other homi.other leg  acc.leg
1 1915 4.031636  0.5215052    9.166456   1.303763   0        0
```

2	1916	3.702076	0.4248284	7.970589	1.416094	0	0
3	1917	3.056176	0.4250311	7.104091	1.052458	0	0
4	1918	3.280707	0.4771938	6.621064	1.312283	0	0
5	1919	2.984728	0.8280212	7.529215	1.309429	0	0
.	.	.	.	.	.	.	.
81	1995	2.2023310	0.3209428	10.397440	1.4829770	0	0
82	1996	2.1025940	0.5406671	10.900720	1.1632530	1	1
83	1997	1.7982930	0.4050209	12.901270	1.3284680	1	2
84	1998	1.2024840	0.2885961	13.099060	1.2345500	1	3
85	1999	1.4002010	0.3275942	11.698280	1.4847410	1	4
86	2000	1.2008320	0.3132606	11.099870	1.3365790	1	5
87	2001	1.2980830	0.2575562	11.301570	1.3392920	1	6
88	2002	1.0997420	0.2138386	10.702110	1.4052250	1	7
89	2003	1.0013770	0.1861856	10.099310	1.3334910	1	8
90	2004	0.8361743	0.1592713	9.591119	1.1497400	1	9
91	2005	NA	NA	NA	NA	1	10
92	2006	NA	NA	NA	NA	1	11
.	.	.	.	.	.	.	.
100	2014	NA	NA	NA	NA	1	19

It is noted that the data.frame is organised columnwise, while the time series in principal should be organised rowwise. This is (if needed) taken care of in `marima` such that the datamatrix is transposed if the number of rows is larger than the number of columns.

The column `leg` indicates whether legislation has been imposed or not (1 or 0). The column `acc.leg` accumulates the legislation (as one possible parametrisation of the effect of the legislation).

Note, that `leg` is set to 1 already in 1996. This is because the first effect of `leg` will be for the year *after* 1996 (namely 1997). This is a general feature in time series models where present values depend on previous values. The first year where `leg` can (is believed to) have an effect is therefore 1997.

7.2 Analysis of the four-variate time series

We will estimate the four univariate the models for the four death rates for the period from 1915 to 1996 (both included) as discussed by Baker & McPhedran. In order to define the model the procedure **define.model** is used, and subsequently **marima** is called using the data from the period 1915 to 1996:

```
rm(list=ls())
library(marima)
data(australian.killings)
old.data <- t(austr)[,1:83]
ar<-c(1)
ma<-c(1)
# Define the proper model:
Model1   <- define.model(kvar=7, ar=ar, ma=ma, rem.var=c(1,6,7), indep=c(2:5))
# Now call marima:
Marima1  <- marima(old.data,means=1,
 ar.pattern=Model1$ar.pattern, ma.pattern=Model1$ma.pattern,
             Check=FALSE, Plot=FALSE, penalty=0.0)
short.form(Marima1$ar.estimates, leading=FALSE) # print estimates
short.form(Marima1$ma.estimates, leading=FALSE)
```

Using `define.model` the variables in the data which are irrelevant for the analyses are taken out, `rem.var=c(1,6,7)`, and `indep=c(2:5)` results in the variables 2, 3, 4 and 5 being analysed independently. The estimated model is as follows:

```
 > short.form(Marima1$ar.estimates, leading=FALSE)
, , Lag=0 (unity matrix, not printed here (leading=FALSE))

, , Lag=1

   x1=y1  x2=y2  x3=y3  x4=y4  x5=y5 x6=y6 x7=y7
y1     0  0.0    0.0    0.0    0.0      0     0
y2     0 -0.7932 0.0    0.0    0.0      0     0
y3     0  0.0   -0.7848 0.0    0.0      0     0
y4     0  0.0    0.0   -0.8870 0.0      0     0
y5     0  0.0    0.0    0.0   -0.9809   0     0
y6     0  0.0    0.0    0.0    0.0      0     0
y7     0  0.0    0.0    0.0    0.0      0     0
```

```
> short.form(Marima1$ma.estimates, leading=FALSE)
, , Lag=0 (unity matrix, not printed here (leading=FALSE))

, , Lag=1

   x1=y1   x2=y2   x3=y3  x4=y4   x5=y5 x6=y6 x7=y7
y1     0  0.0     0.0    0.0     0.0       0     0
y2     0 -0.1317  0.0    0.0     0.0       0     0
y3     0  0.0    -0.4162 0.0     0.0       0     0
y4     0  0.0     0.0    0.0163  0.0       0     0
y5     0  0.0     0.0    0.0    -0.7104    0     0
y6     0  0.0     0.0    0.0     0.0       0     0
y7     0  0.0     0.0    0.0     0.0       0     0
```

Further statistics are saved in the object `Marima1`. For example the covariance matrix of the residuals (`Marima1$resid.cov`):

```
> round(Marima1$resid.cov[2:5,2:5], 4)
       u2     u3      u4      u5
u2 0.1083 0.0161  0.0354  0.0112
u3 0.0161 0.0146  0.0041  0.0020
u4 0.0354 0.0041  0.6582 -0.0041
u5 0.0112 0.0020 -0.0041  0.0224
```

and the covariance matrix of the original variables (`Marima1$data.cov`):

```
> round(Marima1$data.cov[2:5,2:5], 4)
       y2     y3     y4     y5
y2 0.2348 0.0425 0.1110 0.0296
y3 0.0425 0.0199 0.0552 0.0097
y4 0.1110 0.0552 2.3522 0.0790
y5 0.0296 0.0097 0.0790 0.0573
```

The multiple correlations for the 4 variables are:

```
> round(1-(diag(Marima1$resid.cov[2:5,2:5])/diag(Marima1$data.cov[2:5,2:5])),2)
  u2   u3   u4   u5
0.54 0.27 0.72 0.61
```

We may now estimate the general four-variate arma(1,1) model. The only modification in comparison with the above analysis is the model (Model2) definition statement where `indep=c(2:5)` is taken out (`indep=NULL`).

```
ar<-c(1)
ma<-c(1)
Model2   <- define.model(kvar=7, ar=ar, ma=ma, rem.var=c(1,6,7), indep=NULL)
Marima2  <- marima(old.data, means=1, ar.pattern=Model2$ar.pattern,
     ma.pattern=Model2$ma.pattern, Check=FALSE, Plot=FALSE, penalty=0)
```

```
> short.form(Marima2$ar.estimates,leading=FALSE)
, , Lag=1

   x1=y1   x2=y2   x3=y3   x4=y4   x5=y5 x6=y6 x7=y7
y1     0  0.0     0.0     0.0     0.0        0     0
y2     0 -0.5916 -0.8260  0.0545 -0.0583     0     0
y3     0 -0.0933 -0.0868 -0.0034 -0.0861     0     0
y4     0  1.2875 -4.1046 -0.8282 -0.6410     0     0
y5     0  0.1222 -0.6610  0.0164 -0.9458     0     0
y6     0  0.0     0.0     0.0     0.0        0     0
y7     0  0.0     0.0     0.0     0.0        0     0

> short.form(Marima2$ma.estimates,leading=FALSE)
, , Lag=1

   x1=y1   x2=y2   x3=y3  x4=y4   x5=y5 x6=y6 x7=y7
y1     0  0.0     0.0    0.0     0.0        0     0
y2     0  0.0378 -0.1077 0.1750 -0.2258     0     0
y3     0 -0.0146  0.2216 0.0415 -0.0097     0     0
y4     0  1.0700 -2.3897 0.0523 -0.2860     0     0
y5     0  0.0885 -0.5220 0.0308 -0.6557     0     0
```

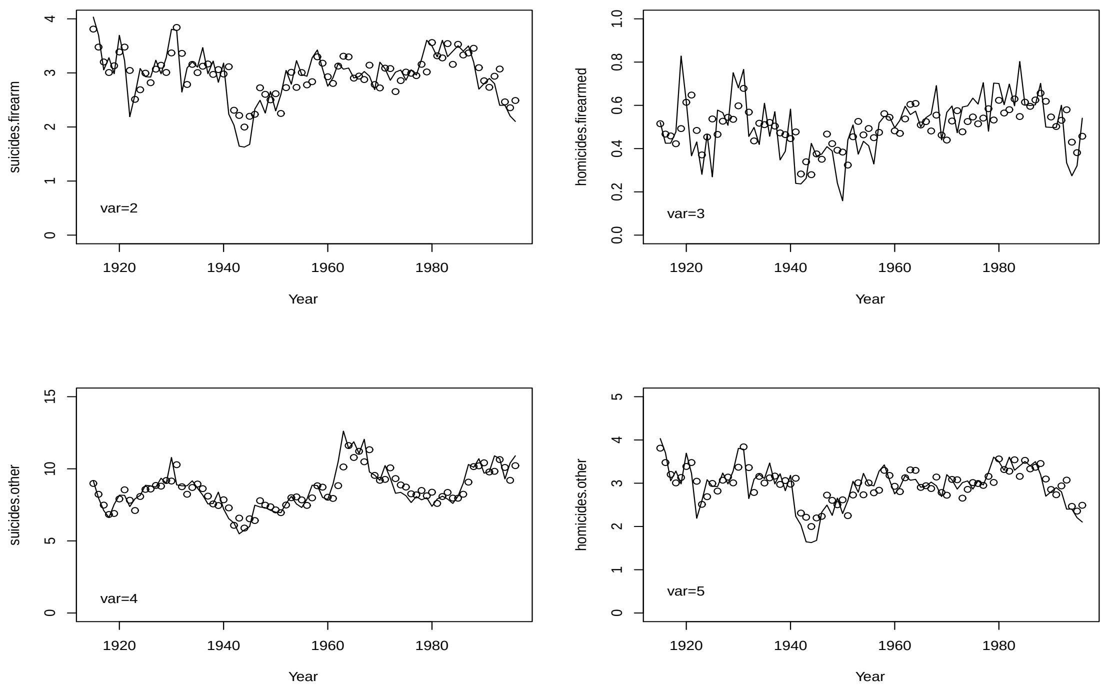
```
y6      0  0.0      0.0     0.0      0.0           0      0
y7      0  0.0      0.0     0.0      0.0           0      0
```

In order to evaluate the improvement in taking into account the correlations between the four variables we may compute:

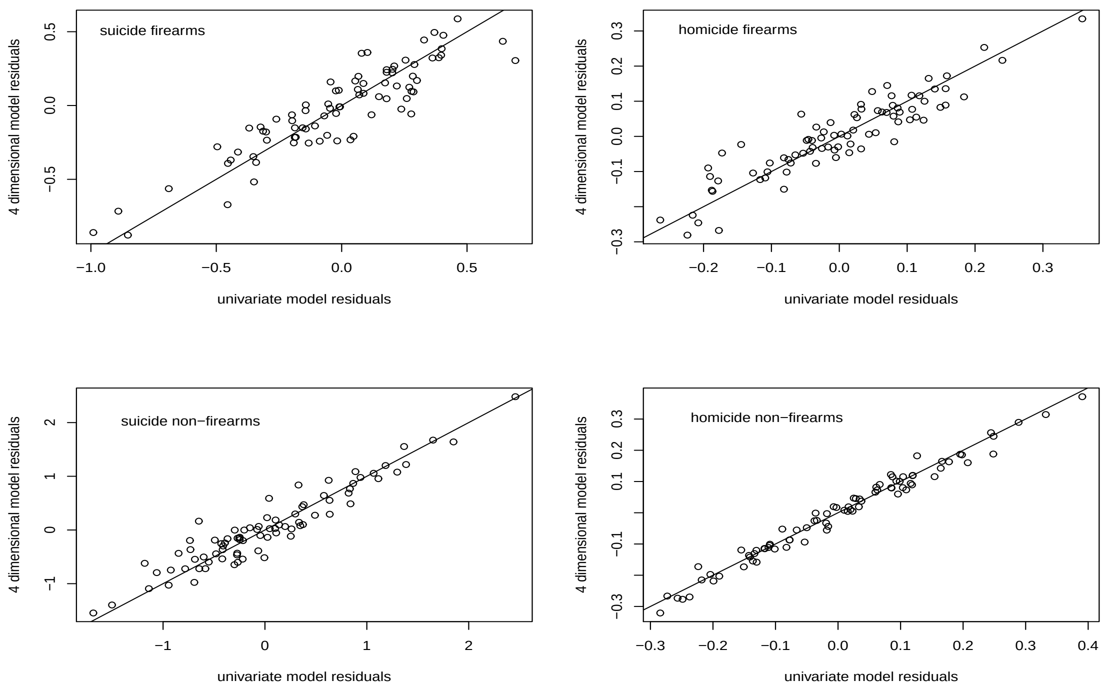
```
>  round(diag(Marima2$resid.cov/Marima1$resid.cov)[2:5], 2)
  u2   u3   u4   u5
0.84 0.89 0.85 1.00
```

so that, for example, the residual variance of the predictions for the first variable (suicides by firearms) estimated by the 4-dimensional model is (only) 84% of the corresponding residual variance for the 4-independent variables model. For the fourth variable (homicides without firearms) there is (practically) no improvement using the 4-dimensional model.

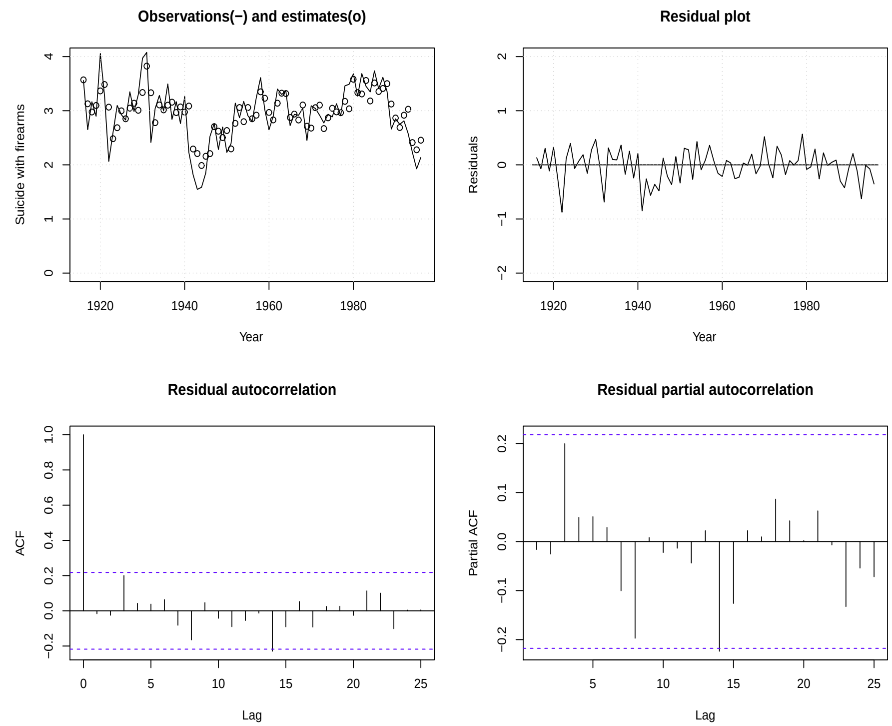
The observations and the predictions for all four variables are shown below (lines=predictions, points=data):



A comparison of the residuals from the univariate models and the 4-dimensional was performed. It turned out that the major differences were for the variable no. 2 (suicides using firearms):



Below are shown the usual model control plots for the 'suicides using firearms' data from the four-dimensional model:



The residual autocorrelations and the partial autocorrelations indicate that the model fits the data adequately.

7.3 Estimation of legislation effect

We shall now estimate a regression model in which variables 6 and 7 are acting as regression variables (use `reg.var=c(6,7)` when calling the model definition procedure `define.model`).

7.3.1 Multivariate model with legislation regression

```
library(marima)
data(australian.killings)
all.data<-t(austr)[,1:90]
ar<-c(1)
ma<-c(1)
Model3   <- define.model(kvar=7, ar=ar, ma=ma, rem.var=c(1), reg.var=c(6,7))
Marima3  <- marima(all.data,means=1, ar.pattern=Model3$ar.pattern,
    ma.pattern=Model3$ma.pattern, Check=FALSE, Plot=FALSE, penalty=0)
```

```
, , Lag=1

> short.form(Marima3$ar.estimates,leading=FALSE)
   x1=y1   x2=y2   x3=y3   x4=y4   x5=y5   x6=y6   x7=y7
y1     0  0.0     0.0     0.0     0.0     0.0     0.0
y2     0 -0.3922 -1.6062  0.0532 -0.0742  0.7155 -0.0163
y3     0 -0.0569 -0.2544 -0.0024 -0.0812  0.0773  0.0107
y4     0  0.8260 -2.7354 -0.7853 -0.5335 -0.9404  0.3087
y5     0  0.1167 -0.6289  0.0140 -0.9501 -0.0257  0.0084
y6     0  0.0     0.0     0.0     0.0     0.0     0.0
y7     0  0.0     0.0     0.0     0.0     0.0     0.0

> short.form(Marima3$ma.estimates,leading=FALSE)
, , Lag=1

   x1=y1  x2=y2   x3=y3  x4=y4   x5=y5 x6=y6 x7=y7
y1     0 0.0     0.0    0.0     0.0        0     0
y2     0 0.2052 -0.8038 0.1557 -0.3363     0     0
y3     0 0.0176  0.0605 0.0374 -0.0181     0     0
y4     0 0.7415 -1.0624 0.0903 -0.0894     0     0
y5     0 0.0879 -0.4930 0.0221 -0.6900     0     0
y6     0 0.0     0.0    0.0     0.0        0     0
```

```
y7      0 0.0     0.0    0.0     0.0         0     0
```

One can assess the significance of the estimated coefficients by means of the `Marima3$ar.fvalues` and the `Marima3$ma.fvalues`.

7.3.2 Multivariate model with legislation regression, 'penalty' reduced

The model reduction/simplification is performed using the option 'penalty', for example `penalty=1`. With the previous example but now penalty=1 we get:

```
Marima4  <- marima(all.data, means=1, ar.pattern=Model4$ar.pattern,
     ma.pattern=Model4$ma.pattern, Check=FALSE, Plot=FALSE, penalty=1)
```

```
round(short.form(Marima4$ar.estimates,leading=FALSE),4)
, , Lag=1

   x1=y1   x2=y2   x3=y3   x4=y4   x5=y5   x6=y6  x7=y7
y1     0  0.0000  0.0000  0.0000  0.0000  0.0000 0.0000
y2     0 -0.5619 -0.8153  0.0342  0.0000  0.5733 0.0000 (suicide with f.a.)
y3     0 -0.0608 -0.3171  0.0000 -0.0669  0.0966 0.0000 (homicide with f.a.)
y4     0  0.6533 -1.6631 -0.8268 -0.4720 -0.9667 0.3253 (suicide without f.a.)
y5     0  0.0000  0.0000  0.0000 -0.9801  0.0000 0.0000 (homicide without f.a.)
y6     0  0.0000  0.0000  0.0000  0.0000  0.0000 0.0000
y7     0  0.0000  0.0000  0.0000  0.0000  0.0000 0.0000

> round(short.form(Marima4$ma.estimates,leading=FALSE),4)
, , Lag=1

   x1=y1  x2=y2 x3=y3  x4=y4   x5=y5 x6=y6 x7=y7
y1     0 0.0000     0 0.0000  0.0000     0     0
y2     0 0.0000     0 0.1303 -0.2351     0     0
y3     0 0.0000     0 0.0391  0.0000     0     0
y4     0 0.5857     0 0.0000  0.0000     0     0
y5     0 0.0000     0 0.0000 -0.7163     0     0
```

```
y6       0 0.0000      0 0.0000  0.0000      0      0
y7       0 0.0000      0 0.0000  0.0000      0      0
```

It is seen that many of the regression coefficients for the intervention (x6) and the regression (x7) in the *penalty=1* reduced model are 0 (zero). For variable y2 (suicides with firearms) a constant *decrease* of 0.5733 and no annual *decrease or increase* from 1997 and onwards is found. For variable 3 (homicide with firearms) a small constant *increase* of 0.0966 and practically no annual *change* is found. For variable 4 (suicide without use of firearms) a constant *increase* of 0.9667 and an annual *decrease* of 0.3253 per year is found, but no change of level. For variable 5 (homicide without use of firearms) no effect from the legislation is found.

One might conclude that the level of the rate of suicides using firearms is decreased by about 0.5733 with no annual effect. But suicides without using firearms decreases by about 0.3253 per year after an initial increase of about 0.9667. The rate of homicides (with or without the use of firearms) is generally not affected by the legislation.

In order to assess the significance of the model found one may use the F-values of the ar-part of the estimated model:

```
> round(short.form(Marima4$ar.fvalues, leading=FALSE), 2)
, , Lag=1

   x1=y1 x2=y2 x3=y3  x4=y4  x5=y5 x6=y6 x7=y7
y1
y2       36.96  6.79   1.43         8.53
y3        3.04  7.54          1.44  2.14
y4        5.11  4.55 181.42   1.59  1.88  6.48
y5                          138.38
y6
y7                                       (zeroes removed)
```

An F-value=2.85, having 1 and around 90 degrees of freedom (length of time series), corresponds to a p-value$\simeq$10% . Therefore, the dependence of the legislation is only highly significant for variables 2 (suicides with firearms) and 4 (homicide with firearms) with p-values below 1% (1-pf(6.48,1,90)$\simeq$1.3%).

Further, it is seen that variable 5 (killings without using firearms) does not seem to depend on any of the other variables (2, 3, 4), neither in the autoregressive nor in the moving average part of the model:

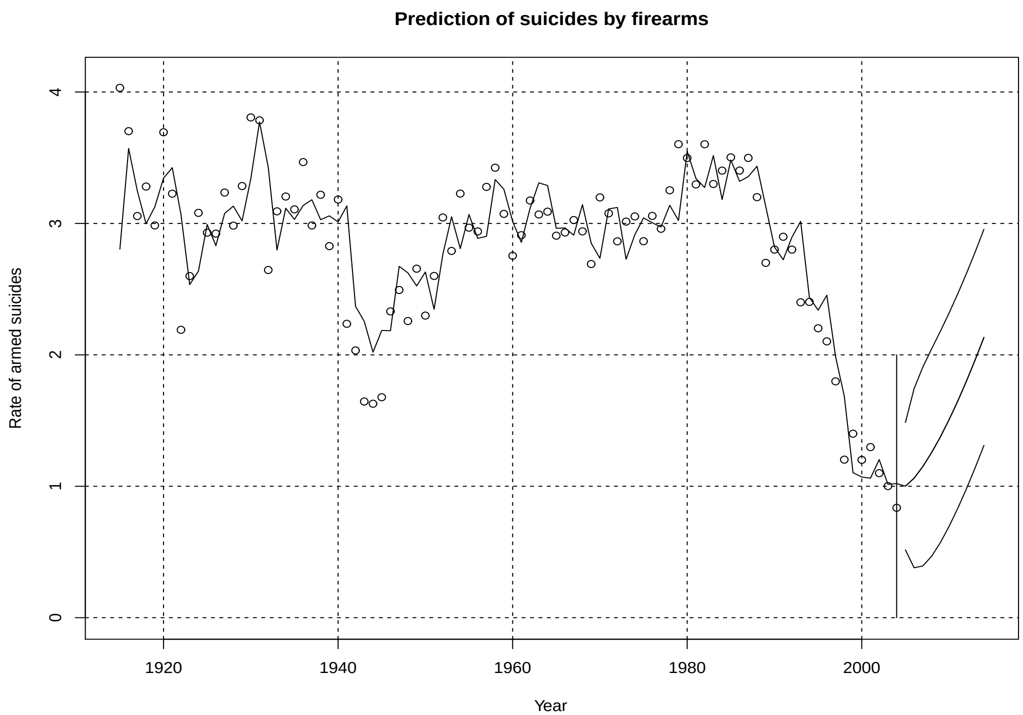
$$(y_{5,t} - 1.104) - 0.5619 \cdot (y_{5,t-1} - 1.104) = u_{5,t} - 0.7163 \cdot u_{5,t-1}$$

in that the mean of the observed y_5, 1.104, was subtracted from the observations before the marima estimation.

7.4 Prediction of timeseries

The routine called `arma.forecast` is used. We start by estimating our model (as before), and then we use the routine `arma.forecast`.

The data (o), the 1-step-ahead forecasts (–) and the nstep=10 forecast (–) and a 90% prediction interval for the forecast are shown in the plot below. Note, that the prediction interval is computed from the marima estimates and without taking the estimation uncertainty into account.



7.5 Forecasting variance

If a forecast $\widehat{y_{t+\ell}}$ over ℓ time units is calculated ($\ell \geq 1$), the variance of that forecast will be $\mathrm{Var}\{\widehat{\mathrm{y_{t+\ell}}}\} = \sum_{\mathrm{i}=0}^{\ell-1} \psi_{\mathrm{i}} \Sigma_{\mathrm{u}} \psi_{\mathrm{i}}^{\mathrm{T}}$, which can be derived from equation 7. The prediction interval shown in the above plot is calculated using this equation.

---

8 References

[1] Baker, J. & McPhedran, S. (2007) Gun Laws and Sudden Death, British Journal of Criminology, 47: 455-469.

[2] Jenkins, G.M. & Alavi, A.S. (1981) Some Aspects of Modelling and Forecasting Multivariate Time Series, Journal of Time Series Analysis, Vol. 2, no 1.

[3] Madsen, H. (2008) Time Series Analysis, Chapmann & Hall (in particular chapter 9: Multivariate time series).

[4] Reinsel G.C. (2003) Elements of Multivariate Time Series Analysis, Springer Verlag, 2^{nd} ed. pp 106-114.

[5] Spliid, H. (1983) A Fast Estimation Method for the Vector Autoregressive Moving Average Model with Exogeneous Variables, Journal of the American Statistical Association, Vol.78, no.384.

[end-of-document]