

University of Groningen

Controlling omitted variables and measurement errors by means of constrained autoregression and structural equation modeling

Suparman, Yusep

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

2015

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Suparman, Y. (2015). *Controlling omitted variables and measurement errors by means of constrained autoregression and structural equation modeling: Theory, simulations and application to measuring household preference for in-house piped water in Indonesia*. [Thesis fully internal (DIV), University of Groningen]. University of Groningen.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Chapter 4 Controlling for Omitted Variables by means of Constrained Autoregression: Testing the Constraints

Abstract

In this paper we perform Monte-Carlo simulation to evaluate Type I error probability and the power of a likelihood ratio test of constrained autoregression (CAR) as vehicle to control for time-varying omitted variables. The main finding is that the probability of Type I error is less than the pre-specified level of 0.01 virtually everywhere while the power is 1 everywhere. We have furthermore found that the test of CAR is highly sensitive to the autoregression of the omitted variable. We conclude that CAR is an acceptable vehicle to control for time-varying omitted variables.

Keywords: Constrained autoregression, Time-varying omitted variables, Monte-Carlo simulation, Type I error probability, Power of the CAR test.

1. Introduction

The problem of omitted variable bias has been well addressed in the literature, including text book like Green (2003) and Wooldridge (2002). To control the bias, panel data models, particularly fixed effect and latent fixed effect models, demeaned regression and first order difference regression models have been frequently used in applied research. However, these approaches are based on the assumption that the omitted variables are time invariant. This assumption is hardly met in practice since most natural and socioeconomic phenomena evolve over time. Thus, application of these models may

result in another kind of bias, i.e. miss-specification bias. Suparman et al. (2014) proposed constrained autoregression (CAR) as an alternative. Conceptually, CAR is more adequate than the time-invariant approaches, since it explicitly accounts for dynamics by modeling the aggregate of the time-varying omitted variables as a first order autoregression.

The number of studies examining the performance of CAR is very limited. Suparman et al.'s (2015, in press) empirical study showed that CAR estimates are more in line with theoretical expectations than the outcomes of a latent fixed effect model. In addition, Suparman and Folmer's (2015) Monte-Carlo study showed that CAR outperforms the time-invariant panel models in reducing omitted variables bias.

In this paper we further explore the performance of CAR to control for time-varying omitted variables. Specifically, we evaluate tests of the CAR specifications by means of Monte Carlo simulation.

In section 2 we summarize CAR. Next, in section 3, we present the Monte-Carlo simulation design. The results are discussed in section 4. Section 5 concludes.

2. Constrained Autoregression (CAR)

Consider the following complete regression model (Suparman and Folmer, 2015):

$$y_{it} = \beta_0 + \sum_{j=1}^a \beta_j x_{jit} + \sum_{k=1}^b \gamma_k z_{kit} + o_{it} \quad (1)$$

for cross-sectional unit i , $i=1,2,\dots,N$ at time points t , $t=1,2,\dots,T$. o_{it} is an independent-identically-distributed (IID) error term which satisfies the zero conditional mean assumption for all i and t . For any value of $a=1,2,\dots$, and $b=1,2,\dots$, let the z_k s

be omitted from (1) with at least one of them correlated with one of the x_j s. Hence, the model with omitted variables (MOV) reads:

$$y_{it} = \beta_0 + \sum_{j=1}^a \beta_j x_{jit} + o_{it} \quad (2)$$

Greene (2003) and Wooldridge (2002), amongst others, show that ordinary least squares (OLS) of $\hat{\beta}_j$ is biased.

To control for the omitted z_k s, CAR assumes that the aggregate of the omitted variables and the error term in (1) develop according to the autoregression model

$$\left(\sum_{k=1}^b \gamma_k z_{kit} + o_{it} \right) = \alpha_0 + \alpha_1 \left(\sum_{k=1}^b \gamma_k z_{kit-1} + o_{it-1} \right) + \varepsilon_{it} \quad (3)$$

(3) can be rewritten in terms of the included variables in (1) as follows:

$$\sum_{k=1}^b \gamma_k z_{kit} + o_{it} = y_{it} - \beta_0 - \sum_{j=1}^a \beta_j x_{jit} \quad (4)$$

Combining (3) and (4) and re-arranging gives

$$y_{it} = \beta_0 + \alpha_1 y_{it-1} - \alpha_1 \sum_{j=1}^a \beta_j x_{jit-1} + \sum_{j=1}^a \beta_j x_{jit} + \varepsilon_{it} \quad (5)$$

(5) is an autoregression model with lagged independent variables (x_{jit-1}) whose regression coefficients are constrained ($-\alpha_1 \beta_j$).

3. Simulation design

Starting point of the simulations is the following complete regression model with three explanatory variables

$$y_{it} = \beta_v v_{it} + \beta_x x_{it} + \beta_z z_{it} + o_{it}^y \quad (6)$$

with $i=1,2,\dots,1000$ and $t=1,2,3$ (for details, see Suparman and Folmer (2015)). We generate the explanatory variables (v , x , and z) at the first wave ($t=1$) from a multivariate normal distribution with zero mean vector and covariance matrix

$$\Sigma = \begin{bmatrix} 1 & & \\ \text{cov}(v,x) & 1 & \\ \text{cov}(v,z) & \text{cov}(x,z) & 1 \end{bmatrix}.$$

The explanatory variable at wave 2 and 3 are generated according to the autoregression model $u_{it} = \alpha_u u_{it-1} + \varepsilon_{it}^u$ with $\varepsilon_{it}^u \stackrel{IID}{\sim} N(0, 1 - \alpha_u^2)$, for $u = v, x, z$ and $t = 2, 3$. We take $\text{cov}(u_{it-1}, \varepsilon_{it}^u) = 0$ and keep the variances of the explanatory variables at unity by setting $\text{var}(\varepsilon_{it}^u) = 1 - \alpha_u^2$. Furthermore, $\varepsilon_{it}^y \stackrel{IID}{\sim} N(0, \sigma_{\varepsilon^y}^2)$. We generate y from (6) with the following values for the above parameters. First, since they have the largest impacts on omitted variable bias (Suparman and Folmer, 2015), we vary $\text{cov}(v, z)$, $\text{cov}(x, z)$, and α_z (the autoregression coefficient of the omitted variable) over the following values: 0, 0.3, 0.6, and 0.9. For the other parameters that have smaller impacts on omitted variable bias, we consider only one value. Specifically, the autoregression coefficients of the included explanatory variables (α_v and α_x) and their covariance at $t = 1$ ($\text{cov}(v, x)$) are set at 0.6, β_v and β_x at 0.3, β_z at 1.0, and $\sigma_{\varepsilon^y}^2$ at 0.1. Finally, the number of repetitions is 2500.

We test the CAR specification by means of a likelihood ratio test (Jöreskog and Sörbom, 1996). Specifically, we test

$H_0 : \alpha = 0, \alpha\beta_v = 0, \alpha\beta_x = 0$, i.e. there is no omitted variable (NOV) implying that

$$y_{it} = \beta_0 + \beta_v v_{it} + \beta_x x_{it} + \varepsilon_{it}$$

holds

versus

H_1 : at least one of the parameter constraints in H_0 is not true.

In the latter case we accept

$$y_{it} = \beta_0 + \beta_v v_{it} + \beta_x x_{it} + \alpha y_{it-1} - \alpha \beta_v v_{it-1} - \alpha \beta_x x_{it-1} + o_{it}$$

In the absence of omitted variables, acceptance of CAR implies acceptance of an over-specified model while in the presence of omitted variables rejection of CAR implies acceptance of an under-specified model. Conversely, acceptance of NOV in the absence of omitted variables implies acceptance of a correctly specified model and acceptance of NOV in the presence of omitted variables acceptance of an under-specified model. Observe that NOV is a more restricted model than CAR, since in CAR there are the lagged dependent and lagged explanatory variables. Hence, there is one additional regression coefficient (the autoregression coefficient α) and there are combinations of α and of the regression coefficients of the current explanatory variables. Accordingly, NOV is the model under the null hypothesis. The test statistic is

$$\chi^2 = -2(\ln(L_{\text{NOV}}) - \ln(L_{\text{CAR}}))$$

with L_{CAR} and L_{NOV} the likelihood scores under CAR and NOV, respectively. The statistic is Chi-square distributed with degrees of freedom equal to the degrees of freedom under NOV minus the degrees of freedom under CAR or, put differently, the number of free parameters in NOV minus the number of free parameter in CAR (Casella and Berger, 2002).

We summarize the test results in terms of the probability of rejecting NOV. In the case NOV is false, this probability is the power; when it is true it is the probability of a Type I error (significance level). The benchmark significance level is 0.01. That is, a significance level equal to or smaller than 0.01 indicates that the test performs well when NOV is true. In a similar vein, a power of 1 indicates that the test perform well when NOV is false.

3. Results

We ran the simulations in R. CAR was estimated by means of the maximum likelihood procedure in OpenMx (Boker et al., 2011). The syntax is provided in Appendix 4.1 Some combinations of $\text{cov}(v, z)$ and $\text{cov}(x, z)$ values produced non-definite-positive covariance matrices of the explanatory variables. Accordingly, the data for those combinations could not be generated. These cases are denoted NA in Table 4.1.

Table 4.1 Probability of rejecting NOV

$\text{cov}(v, z)$	$\text{cov}(x, z)$	α_z			
		0.0	0.3	0.6	0.9
0.0	0.0	0.009	1	1	1
	0.3	0.009	1	1	1
	0.6	0.012	1	1	1
	0.9	NA	NA	NA	NA
0.3	0.0	0.005	1	1	1
	0.3	0.006	1	1	1
	0.6	0.008	1	1	1
	0.9	0.009	1	1	1
0.6	0.0	0.004	1	1	1
	0.3	0.004	1	1	1
	0.6	0.004	1	1	1
	0.9	0.004	1	1	1
0.9	0.0	NA	NA	NA	NA
	0.3	0.002	1	1	1
	0.6	0.004	1	1	1
	0.9	NA	NA	NA	NA

Note: NA data generation not feasible because of singular covariance matrix

Estimations converged for all data sets generated. We present the probabilities of rejecting NOV for various combinations of parameter values in Table 4.1. The main overall result is that the significance level is smaller than the benchmark of 0.01 virtually everywhere while the power is 1 everywhere. Hence, overall, the test performs well.

Below we present some more detailed results. First, for $\alpha_z = \text{cov}(x, z) = \text{cov}(v, z) = 0$, the probability of rejecting NOV is below the significance level of 0.01. Put differently, the test fails to reject NOV at the pre-specified significance level, when the omitted variable is not auto-correlated nor correlated with the included variables.

When at least one of the correlations between the omitted and the included variables is non-zero, and the autoregression of the omitted variable is zero, the rejection probability remains well below the significance level of 0.01, except when $\text{cov}(x, z) = 0.6$ and $\alpha_z = \text{cov}(v, z) = 0$. In that case the rejection probability is slightly higher than the significance level. Although this situation does not imply NOV, it is not much of a problem because the correlation between the omitted variable and the included variables is a minor determinant of bias in a model with omitted variables (Suparman and Folmer, 2015).

When the autocorrelation of the omitted variable increases, the probability of rejecting NOV jumps to 1, whatever the values of the covariances between the omitted variable and the included variables. Hence, the test is highly responsive to the autocorrelation of the omitted variable. This is fortunate, because as shown, in Suparman and Folmer (2015), autocorrelation of the omitted variable is the most important determinant of bias reduction by CAR.

Note that when there is an omitted variable that is uncorrelated with the included variables i.e. $\text{cov}(x, z) = 0$ and $\text{cov}(v, z) = 0$, autocorrelation of the omitted variable implies autocorrelation in the error term. In this case the test also fails to reject CAR.

4. Conclusion

Constrained autoregression (CAR) has been introduced to control for omitted time-varying omitted variables and thus to reduce omitted variable bias. In this paper we analyze the performance of a likelihood ratio test of no omitted variables versus CAR by means of Monte-Carlo analysis. The main finding is that the probability of a Type I error is less than the pre-specified level of 0.01 virtually everywhere while the power is 1 everywhere. We have furthermore found that CAR is highly sensitive to the autoregression of the omitted variable.

Suparman and Folmer (2015) showed that CAR is an effective vehicle to control for omitted variables. Particularly, it is superior to a variety of alternative procedures commonly applied to reduce omitted variable bias. The main outcome of this paper is that a likelihood ratio test can be effectively applied to decide whether or not to apply CAR.

References

- Boker, S., Neale, M., Maes, H., Wilde, M., Spiegel, M., Brick, T., Spies, J., Estarbrook, R., Kenny, S., Bates, T., Mehta, P., & Fox, J. (2011). OpenMx: An open source extended structural equation modeling framework. *Psychometrika*, 76(2), 306-317. DOI: 10.1007/S11336-010-9200-6.
- Casella, G., & Berger, R. L. (2002). *Statistical inference*. Duxbury.
- Greene, W. (2003). *Econometrics analysis* (5th ed.). New Jersey: Prentice Hall.
- Suparman, Y., Folmer, H., & Oud, J. H. L. (2014). Hedonic price models with omitted variables and measurement errors: a constrained autoregression-structural equation modeling approach with application to urban Indonesia. *Journal of Geographical Systems*, 16(1), 49-70. DOI: 10.1007/s10109-013-0186-3

Suparman, Y. & Folmer, H. (2015) Controlling for Time-Varying Omitted Variables in Panel Data Models: Evidence from a Monte-Carlo Simulation.

Suparman, Y., Folmer, H., & Oud, J. H. L. (in press). The willingness to pay for in-house piped water in urban and rural Indonesia. Papers in Regional Science. DOI: 10.1111/pirs.12124

Wooldridge, J. M. (2002). *Introductory econometrics: A modern approach* (2nd ed.). South-Western.

Appendix 4.1 Simulation Syntax Testing the Constraint

```
# Simulation
# Constrained Autoregression
# Spatial Lag-Autoregression
setwd("E:/0paper4sim/")
require(mvtnorm)
require(OpenMx)
# A. Simulation setting and Population Parameters
for (av in 2:2)
{
for (ax in 2:2)
{
for (az in 0:3)
{
for (r in 2:2)
{
for (s in 0:3)
{
for (t in 0:3)
{
set.seed(-3) # Initial random number seed
K <- 5 # Number of simulation replication
N <- 100 # sample size
## constant parameters
vx1 <- 1 #var(x1)
vz1 <- 1 #var(z1)
vv1 <- 1 #var(v1)
cvx <- (0+3*r)/10 #cor(x1,z1)
cvz <- (0+3*s)/10 #cor(v1,x1)
cxz <- (0+3*t)/10 #cor(v1,z1)
alv <- (0+3*av)/10 #autoregression v
alx <- (0+3*ax)/10 #autoregression x
alz <- (0+3*az)/10 #autoregression z
vev <- 1-alv^2 #autoregression v error variance
vex <- 1-alx^2 #autoregression x error variance
vez <- 1-alz^2 #autoregression z error variance
bev <- 0.3 #be(v->y)
bex <- 0.3 #be(x->y)
bez <- 1.0 #be(z->y)
By <- matrix(c(bev,bex,bez),nrow=1,ncol=3)
```

```

vey          <- .1                                #var(ey)
# B. Data Generation
## Generating V1, X1 and Z1
Sxz          <- matrix(c(vv1,cvz,cvz,cvz,vx1,cxz,cvz,cxz,vz1),
                      nrow=3,ncol=3, byrow=TRUE)
Mxz          <- matrix(c(0,0,0),nrow=3, ncol=1)
XZ1          <- rmvnorm(N,Mxz,Sxz)
## Generating V2, X2 and Z2
Sexz        <- matrix(c(vev,0,0,0,vex,0,0,0,vez),nrow=3,ncol=3, byrow=TRUE)
Mexz        <- matrix(c(0,0,0),nrow=3, ncol=1)
T2          <- matrix(0,nrow=N,ncol=3)
Al          <- matrix(c(alv,0,0,0,alx,0,0,0,alz),nrow=3,ncol=3,
byrow=TRUE)
Exz2        <- rmvnorm(N,Mexz,Sexz)
XZ2          <- T2+(XZ1%*%Al)+Exz2
## Generating X3 and Z3
T3          <-matrix(0,nrow=N,ncol=3)
Exz3        <- rmvnorm(N,Mexz,Sexz)
XZ3          <- T3+(XZ2%*%Al)+Exz3
for (k in 1:K) # Looping for simulation replication
{
print(" ")
print(c(k,alv,alx,alz,cxz,cvz,bez))
## Generating Y and Yw
Vey          <- matrix(c(vey),nrow=1,ncol=1)
Mey          <- matrix(c(0),nrow=1,ncol=1)
Ey1          <- rmvnorm(N,Mey,Vey)
Y1           <- (XZ1%*%t(By)+Ey1)
Ey2          <- rmvnorm(N,Mey,Vey)
Y2           <- (XZ2%*%t(By)+Ey2)
Ey3          <- rmvnorm(N,Mey,Vey)
Y3           <- (XZ3%*%t(By)+Ey3)
## Data processing
Dt           <- cbind(Y1,Y2,Y3,XZ1,XZ2,XZ3)
Dt           <- as.real(Dt)
Dt           <- matrix(Dt, nrow=N, ncol=12)
YVX2        <- cbind(Y1,XZ1,XZ2)
YVX2        <- YVX2[,c(1,2,3,5,6)]
YVX2        <- matrix(YVX2, nrow=N, ncol=5)
YVX3        <- cbind(Y2,XZ2,XZ3)
YVX3        <- YVX3[,c(1,2,3,5,6)]
YVX3        <- matrix(YVX3, nrow=N, ncol=5)
Y1          <- matrix(Y1, nrow=N, ncol=1)
Y2          <- matrix(Y2, nrow=N, ncol=1)
Y3          <- matrix(Y3, nrow=N, ncol=1)
VX1         <- matrix(XZ1[,c(1,2)], nrow=N, ncol=2, byrow=T)
VX2         <- matrix(XZ2[,c(1,2)], nrow=N, ncol=2, byrow=T)
VX3         <- matrix(XZ3[,c(1,2)], nrow=N, ncol=2, byrow=T)
ONE         <- matrix(1, nrow=N, ncol=1)
varNames    <-
c("y1","y2","y3","v1","x1","z1","v2","x2","z2","v3","x3","z3")
varNames1   <- c("y1","y2","y3","v1","x1","v2","x2","v3","x3")
varNames1a  <- c("y2","y3","v2","x2","v3","x3")
latNames    <- c("y1","y2","y3","v1","x1","v2","x2","v3","x3","ksi")
varNames2   <- c("dy2","dy3","dv2","dx2","dv3","dx3")
write.table(Dt,"Dt0.dat", sep="\t", row.names=FALSE,col.names=varNames)
input       <- "Dt0.dat"
data        <- read.table(file=input, header=TRUE)
             is.na(data)=data==999
data        <- data[,c(1,2,3,4,5,6,7,8,9,10,11,12)]
data1       <- data[,c(1,2,3,4,5,7,8,10,11)]

```

```

data1a<- data[,c(2,3,7,8,10,11)]
#data0      <- matrix(data1, nrow=N, ncol=6)
diff  <- matrix(c(-1, 1,0, 0, 0,0, 0, 0,0, 0,0,0,0,
                  0,-1,1, 0, 0,0, 0, 0,0, 0,0,0,0,
                  0, 0,0,-1, 0,0, 1, 0,0, 0,0,0,0,
                  0, 0,0, 0,-1,0, 0, 1,0, 0,0,0,0,
                  0, 0,0, 0, 0,0,-1, 0,0, 1,0,0,0,
                  0, 0,0, 0, 0,0, 0,-1,0, 0,1,0),
nrow=6,ncol=12,byrow=TRUE)
Dt2  <- Dt%*%t(diff)
write.table(Dt2,"Dt2.dat", sep="\t",
row.names=FALSE,col.names=varNames2)
input2<- "Dt2.dat"
data2 <- read.table(file=input2, header=TRUE)
      is.na(data)=data==999
data2 <- data2[,c(1,2,3,4,5,6)]
datay <- data[,c(1,2,3)]
datax <- data[,c(4,5,7,8,10,11)]
My    <- rowMeans(datay)
Mx    <- rowMeans(datax)
Myx   <- cbind(My,My,My,Mx,Mx,Mx)
Dt3   <- data1-Myx
write.table(Dt3,"Dt3.dat", sep="\t",
row.names=FALSE,col.names=varNames1)
input3 <- "Dt3.dat"
data3 <- read.table(file=input3, header=TRUE)
      is.na(data)=data==999
V     <- cov(data)
M     <- colMeans(data)
Vd    <- cov(data2)
Md    <- colMeans(data2)
# C. Estimation Process
## C0. Full Model
### C0a. Specifying
Model0 <- mxModel("Model0",
  mxData(observed=data, type="raw"),
  mxMatrix(type="Full",nrow=12, ncol=12,
values=c(
0,0,0, bev,bex,bez, 0 ,0 ,0 , 0 ,0 ,0 ,
0,0,0, 0 ,0 ,0 , bev,bex,bez, 0 ,0 ,0 ,
0,0,0, 0 ,0 ,0 , 0 ,0 ,0 , bev,bex,bez,
0,0,0, 0 ,0 ,0 , 0 ,0 ,0 , 0 ,0 ,0 ,
0,0,0, 0 ,0 ,0 , 0 ,0 ,0 , 0 ,0 ,0 ,
0,0,0, 0 ,0 ,0 , 0 ,0 ,0 , 0 ,0 ,0 ,
0,0,0, 0 ,0 ,0 , 0 ,0 ,0 , 0 ,0 ,0 ,
0,0,0, 0 ,0 ,0 , 0 ,0 ,0 , 0 ,0 ,0 ,
0,0,0, 0 ,0 ,0 , 0 ,0 ,0 , 0 ,0 ,0 ,
0,0,0, 0 ,0 ,0 , 0 ,0 ,0 , 0 ,0 ,0 ,
0,0,0, 0 ,0 ,0 , 0 ,0 ,0 , 0 ,0 ,0 ,
0,0,0, 0 ,0 ,0 , 0 ,0 ,0 , 0 ,0 ,0 ,
free=c(
F,F,F, T,T,T, F,F,F, F,F,F,
F,F,F, F,F,F, T,T,T, F,F,F,
F,F,F, F,F,F, F,F,F, T,T,T,
F,F,F, F,F,F, F,F,F, F,F,F,
F,F,F, F,F,F, F,F,F, F,F,F,
F,F,F, F,F,F, F,F,F, F,F,F,
F,F,F, F,F,F, F,F,F, F,F,F,
F,F,F, F,F,F, F,F,F, F,F,F,
F,F,F, F,F,F, F,F,F, F,F,F,
F,F,F, F,F,F, F,F,F, F,F,F,
F,F,F, F,F,F, F,F,F, F,F,F,
F,F,F, F,F,F, F,F,F, F,F,F),

```

```

labels=c(  NA,NA,NA, "bv", "bx", "bz", NA ,NA ,NA , NA ,NA ,NA ,
           NA,NA,NA, NA ,NA ,NA , "bv", "bx", "bz", NA ,NA ,NA ,
           NA,NA,NA, NA ,NA ,NA , NA ,NA ,NA , "bv", "bx", "bz",
           NA,NA,NA, NA ,NA ,NA , NA ,NA ,NA , NA ,NA ,NA ,
           NA,NA,NA, NA ,NA ,NA , NA ,NA ,NA , NA ,NA ,NA ,
           NA,NA,NA, NA ,NA ,NA , NA ,NA ,NA , NA ,NA ,NA ,
           NA,NA,NA, NA ,NA ,NA , NA ,NA ,NA , NA ,NA ,NA ,
           NA,NA,NA, NA ,NA ,NA , NA ,NA ,NA , NA ,NA ,NA ,
           NA,NA,NA, NA ,NA ,NA , NA ,NA ,NA , NA ,NA ,NA ,
           NA,NA,NA, NA ,NA ,NA , NA ,NA ,NA , NA ,NA ,NA ,
           NA,NA,NA, NA ,NA ,NA , NA ,NA ,NA , NA ,NA ,NA ,
           NA,NA,NA, NA ,NA ,NA , NA ,NA ,NA , NA ,NA ,NA ),
          byrow=TRUE,name="A"),
mxMatrix(type="Symm",nrow=12,ncol=12,
values=c(  vey,0 ,0 , 0,0,0, 0,0,0, 0,0,0,
           0 ,vey,0 , 0,0,0, 0,0,0, 0,0,0,
           0 ,0 ,vey, 0,0,0, 0,0,0, 0,0,0,
           0 ,0 ,0 , V[4,4],V[5,4],V[6,4], V[7,4],V[8,4],V[9,4],
V[10,4],V[11,4],V[12,4],
           0 ,0 ,0 , V[5,4],V[5,5],V[6,5], V[7,5],V[8,5],V[9,5],
V[10,5],V[11,5],V[12,5],
           0 ,0 ,0 , V[6,4],V[6,5],V[6,6], V[7,6],V[8,6],V[9,6],
V[10,6],V[11,6],V[12,6],
           0 ,0 ,0 , V[7,4],V[7,5],V[7,6], V[7,7],V[8,7],V[9,7],
V[10,7],V[11,7],V[12,7],
           0 ,0 ,0 , V[8,4],V[8,5],V[8,6], V[8,7],V[8,8],V[9,8],
V[10,8],V[11,8],V[12,8],
           0 ,0 ,0 , V[9,4],V[9,5],V[9,6], V[9,7],V[9,8],V[9,9],
V[10,9],V[11,9],V[12,9],
           0 ,0 ,0 , V[10,4],V[10,5],V[10,6],
V[10,7],V[10,8],V[10,9], V[10,10],V[11,10],V[12,10],
           0 ,0 ,0 , V[11,4],V[11,5],V[11,6],
V[11,7],V[11,8],V[11,9], V[11,10],V[11,11],V[12,11],
           0 ,0 ,0 , V[12,4],V[12,5],V[12,6],
V[12,7],V[12,8],V[12,9], V[12,10],V[12,11],V[12,12]),
free=c(  T,F,F, F,F,F, F,F,F, F,F,F,
         F,T,F, F,F,F, F,F,F, F,F,F,
         F,F,T, F,F,F, F,F,F, F,F,F,
         F,F,F, T,T,T, T,T,T, T,T,T,
         F,F,F, T,T,T, T,T,T, T,T,T,
         F,F,F, T,T,T, T,T,T, T,T,T,
         F,F,F, T,T,T, T,T,T, T,T,T,
         F,F,F, T,T,T, T,T,T, T,T,T,
         F,F,F, T,T,T, T,T,T, T,T,T,
         F,F,F, T,T,T, T,T,T, T,T,T,
         F,F,F, T,T,T, T,T,T, T,T,T,
         F,F,F, T,T,T, T,T,T, T,T,T),
labels=c(  "p1",NA ,NA , NA,NA,NA, NA,NA,NA, NA,NA,NA,
           NA , "p1",NA , NA,NA,NA, NA,NA,NA, NA,NA,NA,
           NA ,NA , "p1", NA,NA,NA, NA,NA,NA, NA,NA,NA,
           NA ,NA ,NA , NA,NA,NA, NA,NA,NA, NA,NA,NA,
           NA ,NA ,NA , NA,NA,NA, NA,NA,NA, NA,NA,NA,
           NA ,NA ,NA , NA,NA,NA, NA,NA,NA, NA,NA,NA,
           NA ,NA ,NA , NA,NA,NA, NA,NA,NA, NA,NA,NA,
           NA ,NA ,NA , NA,NA,NA, NA,NA,NA, NA,NA,NA,
           NA ,NA ,NA , NA,NA,NA, NA,NA,NA, NA,NA,NA,
           NA ,NA ,NA , NA,NA,NA, NA,NA,NA, NA,NA,NA,
           NA ,NA ,NA , NA,NA,NA, NA,NA,NA, NA,NA,NA,
           NA ,NA ,NA , NA,NA,NA, NA,NA,NA, NA,NA,NA ),
          byrow=TRUE,name="S"),
mxMatrix(type="Iden",nrow=12,ncol=12,name="F"),

```

```

        mxMatrix(type="Full",nrow=1,ncol=12,
values=c( 0,0,0, M[4],M[5],M[6], M[7],M[8],M[9]),
free=c(    T,T,T, T,T,T, T,T,T, T,T,T),
labels=c("i1","i2","i3","m4","m5","m6","m7","m8","m9","m10","m11",
"m12"),
        name="M"),
        mxRAMObjective("A","S","F","M",dimnames=varNames)
)
### C0b. Processing
Model0Fit <- mxRun(Model0)
summary_Fit <- summary(Model0Fit)
### C0c. Writing
bv0 <- summary_Fit[[1]]$Estimate[[1]]
bx0 <- summary_Fit[[1]]$Estimate[[2]]
bz0 <- summary_Fit[[1]]$Estimate[[3]]
p10 <- summary_Fit[[1]]$Estimate[[4]]
p20 <- summary_Fit[[1]]$Estimate[[4]]
p30 <- summary_Fit[[1]]$Estimate[[4]]
i10 <- summary_Fit[[1]]$Estimate[[50]]
i20 <- summary_Fit[[1]]$Estimate[[51]]
i30 <- summary_Fit[[1]]$Estimate[[52]]
pk0 <- 0
al0 <- 0
output0<- c( bv0, bx0, bz0, p10, p20, p30, i10, i20, i30, pk0, al0)
print(output0)write.table(t(output0), file="outfile0.dat", sep=" ",
eol="\n", na="NA", dec=".", row.names=FALSE, col.names=FALSE,
append=TRUE)
## C1. Under-specified Regression Model
### Cla. Specifying
Model1 <- mxModel("Model1",
        mxData(observed=data1, type="raw"),
        mxMatrix(type="Full",nrow=9,ncol=9,
values=c( 0,0,0, 0,0, 0 ,0 , 0 ,0 ,
0,0,0, 0 ,0 , bev,bex, 0 ,0 ,
0,0,0, 0 ,0 , 0 ,0 , bev,bex,
0,0,0, 0,0, 0,0, 0,0,
0,0,0, 0,0, 0,0, 0,0,
0,0,0, 0,0, 0,0, 0,0,
0,0,0, 0,0, 0,0, 0,0,
0,0,0, 0,0, 0,0, 0,0,
0,0,0, 0,0, 0,0, 0,0),
        free=c(
F,F,F, F,F, F,F, F,F,
F,F,F, F,F, T,T, F,F,
F,F,F, F,F, F,F, T,T,
F,F,F, F,F, F,F, F,F,
F,F,F, F,F, F,F, F,F,
F,F,F, F,F, F,F, F,F,
F,F,F, F,F, F,F, F,F,
F,F,F, F,F, F,F, F,F,
F,F,F, F,F, F,F, F,F),
labels=c( NA,NA,NA, NA,NA, NA ,NA , NA ,NA ,
NA,NA,NA, NA ,NA , "bv","bx", NA ,NA ,
NA,NA,NA, NA ,NA , NA ,NA , "bv","bx",
NA,NA,NA, NA,NA, NA,NA, NA,NA,
NA,NA,NA, NA,NA, NA,NA, NA,NA,
NA,NA,NA, NA,NA, NA,NA, NA,NA,
NA,NA,NA, NA,NA, NA,NA, NA,NA,
NA,NA,NA, NA,NA, NA,NA, NA,NA ),
        byrow=TRUE,name="A" ),

```

```

        mxMatrix(type="Symm",nrow=9,ncol=9,
values=c(  V[1,1],0 ,0 , V[4,1],V[5,1], V[7,1],V[8,1],
V[10,1],V[11,1],
        0 ,vey,0, 0,0, 0,0, 0,0,
        0 ,0,vey, 0,0, 0,0, 0,0,
        V[4,1],0,0, V[4,4],V[5,4], V[7,4],V[8,4], V[10,4],V[11,4],
        V[5,1],0,0, V[5,4],V[5,5], V[7,5],V[8,5], V[10,5],V[11,5],
        V[7,1],0,0, V[7,4],V[7,5], V[7,7],V[8,7], V[10,7],V[11,7],
        V[8,1],0,0, V[8,4],V[8,5], V[8,7],V[8,8], V[10,8],V[11,8],

V[10,1],0,0,V[10,4],V[10,5],V[10,7],V[10,8],V[10,10],V[11,10],

V[11,1],0,0,V[11,4],V[11,5],V[11,7],V[11,8],V[11,10],V[11,11]),
        free=c(          T,F,F, T,T, T,T, T,T,
                        F,T,F, F,F, F,F, F,F,
                        F,F,T, F,F, F,F, F,F,
                        T,F,F, T,T, T,T, T,T,
                        T,F,F, T,T, T,T, T,T,
                        T,F,F, T,T, T,T, T,T,
                        T,F,F, T,T, T,T, T,T,
                        T,F,F, T,T, T,T, T,T,
                        T,F,F, T,T, T,T, T,T),
labels=c(  NA ,NA ,NA , NA,NA, NA,NA, NA,NA,
NA , "p1",NA, NA,NA, NA,NA, NA,NA,
NA ,NA, "p1", NA,NA, NA,NA, NA,NA,
NA ,NA ,NA , NA,NA, NA,NA, NA,NA,
NA ,NA ,NA , NA,NA, NA,NA, NA,NA,
NA ,NA ,NA , NA,NA, NA,NA, NA,NA,
NA ,NA ,NA , NA,NA, NA,NA, NA,NA,
NA ,NA ,NA , NA,NA, NA,NA, NA,NA,
NA ,NA ,NA , NA,NA, NA,NA, NA,NA),
        byrow=TRUE,name="S"),
        mxMatrix(type="Iden",nrow=9,ncol=9,name="F"),
        mxMatrix(type="Full",nrow=1,ncol=9,
values=c(  M[1],0,0, M[4],M[5], M[7],M[8], M[10],M[11]),
free=c(  T,T,T, T,T, T,T, T,T),
labels=c(  "m1","i2","i3", "m4","m5", "m7","m8",
"m10","m11"),name="M"),
        mxRAMObjective("A","S","F","M",dimnames=varNames1)
)
### Clb. Processing
ModellFit <- mxRun(Modell)
summary_Fit <- summary(ModellFit)
### Clc. Writing
bv1 <- summary_Fit[[1]]$Estimate[[1]]
bx1 <- summary_Fit[[1]]$Estimate[[2]]
bz1 <- bez
p11 <- summary_Fit[[1]]$Estimate[[3]]
p21 <- summary_Fit[[1]]$Estimate[[3]]
p31 <- summary_Fit[[1]]$Estimate[[3]]
m11 <- 0
i21 <- summary_Fit[[1]]$Estimate[[14]]
i31 <- summary_Fit[[1]]$Estimate[[15]]
pk1 <- 0
all <- 0
np1 <- summary_Fit$estimatedParameters
ll1 <- summary_Fit$Minus2LogLikelihood
b_ur <- matrix(c(bv1,bx1),nrow=2, ncol=1, byrow=T)
e_ur2 <- Y2-(ONE%*%i21)-(VX2%*%b_ur)
e_ur3 <- Y3-(ONE%*%i31)-(VX3%*%b_ur)
ssr_ur <- t(e_ur2)%*%e_ur2 + t(e_ur3)%*%e_ur3

```

```

output1<- c(bv1, bx1, bz1, p11, p21, p31, m11, i21, i31, pk1, all)
print(output1)
write.table(t(output1), file="outfile1.dat", sep=" ", eol="\n", na="NA",
            dec=".", row.names=FALSE, col.names=FALSE, append=TRUE)
## C3. Constrained Autoregression Model
### C3a. Specifying
Model3 <- mxModel("Model3",
  mxData(observed=data1, type="raw"),
  mxMatrix(type="Full",nrow=1,ncol=1,free=TRUE,values=bev,
            labels="bv",name="Mbv"),
  mxMatrix(type="Full",nrow=1,ncol=1,free=TRUE,values=bex,
            labels="bx",name="Mbx"),
  mxMatrix(type="Full",nrow=1,ncol=1,free=TRUE,values=1,
            labels="a",name="Ma"),
  mxAlgebra(expression=(-1)*Ma*Mbv,name="d"),
  mxAlgebra(expression=(-1)*Ma*Mbx,name="e"),
  mxMatrix(type="Full",nrow=9,ncol=9,
values=c( 0,0,0, 0,0,0, 0,0,0, 0,0,0,
          1,0,0, NA,NA, bev,bex, 0,0,0,
          0,1,0, 0,0, NA,NA, bev,bex,
          0,0,0, 0,0, 0,0, 0,0,
          0,0,0, 0,0, 0,0, 0,0,
          0,0,0, 0,0, 0,0, 0,0,
          0,0,0, 0,0, 0,0, 0,0,
          0,0,0, 0,0, 0,0, 0,0,
          0,0,0, 0,0, 0,0, 0,0),
            free=c(
                                F,F,F, F,F, F,F, F,F,
                                T,F,F, F,F, T,T, F,F,
                                F,T,F, F,F, F,F, T,T,
                                F,F,F, F,F, F,F, F,F,
                                F,F,F, F,F, F,F, F,F,
                                F,F,F, F,F, F,F, F,F,
                                F,F,F, F,F, F,F, F,F,
                                F,F,F, F,F, F,F, F,F,
                                F,F,F, F,F, F,F, F,F),
labels=c( NA,NA,NA,NA,NA, NA,NA,NA,NA,
          "a",NA,NA, "d[1,1]", "e[1,1]", "bv" , "bx" , NA,NA,
          NA, "a",NA, NA, NA, "d[1,1]", "e[1,1]", "bv", "bx",
          NA,NA,NA, NA,NA, NA,NA, NA,NA,
          NA,NA,NA, NA,NA, NA,NA, NA,NA,
          NA,NA,NA, NA,NA, NA,NA, NA,NA,
          NA,NA,NA, NA,NA, NA,NA, NA,NA,
          NA,NA,NA, NA,NA, NA,NA, NA,NA,
          NA,NA,NA, NA,NA, NA,NA, NA,NA ),
            byrow=TRUE,name="A"),
  mxMatrix(type="Symm",nrow=9,ncol=9,
values=c( V[1,1],0,0, V[4,1],V[5,1], V[7,1],V[8,1],
V[10,1],V[11,1],
          0,vey,0,0,0,0,0,0,0,
          0,0,vey,0,0,0,0,0,0,
          V[4,1],0,0, V[4,4],V[5,4], V[7,4],V[8,4], V[10,4],V[11,4],
          V[5,1],0,0, V[5,4],V[5,5], V[7,5],V[8,5], V[10,5],V[11,5],
          V[7,1],0,0, V[7,4],V[7,5], V[7,7],V[8,7], V[10,7],V[11,7],
          V[8,1],0,0, V[8,4],V[8,5], V[8,7],V[8,8], V[10,8],V[11,8],
          V[10,1],0,0, V[10,4],V[10,5], V[10,7],V[10,8], V[10,10],V[11,10],
          V[11,1],0,0, V[11,4],V[11,5], V[11,7],V[11,8], V[11,10],V[11,11]),
            free=c(
                                T,F,F, T,T, T,T, T,T,
                                F,T,F, F,F, F,F, F,F,
                                F,F,T, F,F, F,F, F,F,

```



```

T,F,F, T,T, T,T, T,T,
T,F,F, T,T, T,T, T,T,
T,F,F, T,T, T,T, T,T,
T,F,F, T,T, T,T, T,T,
T,F,F, T,T, T,T, T,T,
T,F,F, T,T, T,T, T,T,
labels=c( "p1",NA ,NA , NA,NA, NA,NA, NA,NA,
NA , "p2",NA , NA,NA, NA,NA, NA,NA,
NA ,NA , "p3", NA,NA, NA,NA, NA,NA,
NA ,NA ,NA , NA,NA, NA,NA, NA,NA,
NA ,NA ,NA , NA,NA, NA,NA, NA,NA,
NA ,NA ,NA , NA,NA, NA,NA, NA,NA,
NA ,NA ,NA , NA,NA, NA,NA, NA,NA,
NA ,NA ,NA , NA,NA, NA,NA, NA,NA ),
byrow=TRUE,name="S"),
mxMatrix(type="Iden",nrow=9,ncol=9,name="F"),
mxMatrix(type="Full",nrow=1,ncol=9,
values=c( M[1],0,0, M[4],M[5], M[7],M[8], M[10],M[11]),
free=c( T,T,T, T,T, T,T, T,T),
labels=c("m1","i2","i3", "m4","m5", "m7","m8", "m10","m11"),name="M"),
mxRAMObjective("A","S","F","M",dimnames=varNames1)
)
### C3b. Processing
Model3Fit <- mxRun(Model3)
summary_Fit <- summary(Model3Fit)
### C3c. Writing
bv3 <- summary_Fit[[1]]$Estimate[[1]]
bx3 <- summary_Fit[[1]]$Estimate[[2]]
bz3 <- bez
p13 <- 0.0
p23 <- summary_Fit[[1]]$Estimate[[5]]
p33 <- summary_Fit[[1]]$Estimate[[6]]
i13 <- 0.0
i23 <- summary_Fit[[1]]$Estimate[[35]]
i33 <- summary_Fit[[1]]$Estimate[[36]]
pk3 <- 0
al3 <- summary_Fit[[1]]$Estimate[[3]]
sea <- summary_Fit[[1]]$Std.Error[[3]]
np3 <- summary_Fit$estimatedParameters
l13 <- summary_Fit$Minus2LogLikelihood
b_car <- matrix(c(al3,-1*al3*bv3,-1*al3*bx3,bv3,bx3),nrow=5,
ncol=1, byrow=T)
e_car2 <- Y2-(ONE%*%i23)-(YVX2%*%b_car)
e_car3 <- Y3-(ONE%*%i33)-(YVX3%*%b_car)
ssr_car <- t(e_car2)%*%e_car2 + t(e_car3)%*%e_car3
Ft <- ((ssr_ur-ssr_car)/ssr_car)*(N-3-1)
Ta <- al3/sea
Ch <- l11-l13
df <- np3-np1
p_Ft <- pf(Ft, 1, N-3-1, lower.tail=F)
p_Ta <- pt(Ta, N-3-1, lower.tail=F)
p_Ch <- pchisq(Ch,df, lower.tail=F)
print(Ch)
print(df)
print(p_Ch)
if(p_Ft<0.01)
df01 <- 1
else
df01 <- 0
if(p_Ft<0.05)

```

```

        dF05 <- 1
    else
        dF05 <- 0
    if(p_Ft<0.1)
        dF10 <- 1
    else
        dF10 <- 0
    if(p-Ta<0.01)
        dT01 <- 1
    else
        dT01 <- 0
    if(p-Ta<0.05)
        dT05 <- 1
    else
        dT05 <- 0
    if(p-Ta<0.1)
        dT10 <- 1
    else
        dT10 <- 0
    if(p_Ch<0.01)
        dC01 <- 1
    else
        dC01 <- 0
    if(p_Ch<0.05)
        dC05 <- 1
    else
        dC05 <- 0
    if(p_Ch<0.1)
        dC10 <- 1
    else
        dC10 <- 0

output3<- c( bv3, bx3, bz3, p13, p23, p33, i13, i23, i33, pk3, al3)
print(output3)
write.table(t(output3), file="outfile3.dat", sep=" ", eol="\n", na="NA",
            dec=".", row.names=FALSE, col.names=FALSE, append=TRUE)
output4 <- c(Ft, p_Ft, dF01, dF05, dF10, Ta, p_Ta, dT01, dT05,
dT10, Ch, p_Ch, dC01, dC05, dC10, cvx, cvz, cxz, alv, alx, alz)
print(output4)
write.table(t(output4), file="outfile4.dat", sep=" ", eol="\n", na="NA",
            dec=".", row.names=FALSE, col.names=FALSE, append=TRUE)
output5 <- c(ssr_ur, ssr_car, al3, sea, l11, l13, Ft, p_Ft, dF01,
dF05, dF10, Ta, p_Ta, dT01, dT05, dT10, Ch, p_Ch, dC01, dC05, dC10, cvx,
cvz, cxz, alv, alx, alz)
write.table(t(output5), file="outfile5.dat", sep=" ", eol="\n", na="NA",
            dec=".", row.names=FALSE, col.names=FALSE, append=TRUE)
file.remove("Dt0.dat")
file.remove("Dt2.dat")
file.remove("Dt3.dat")
}
# D. Results Analysing
## D0. Full Model
out0 <- read.table(file="outfile0.dat",header=FALSE,sep=" ",dec=".")
is.na(out0)=out0==999
ME0 <-colMeans(out0)
ME0 <-matrix(ME0,nrow=1,ncol=11)
PV <-matrix(c(bev,bex,bez,vey,vey,vey,0,0,0,0,0),nrow=1,ncol=11)
BE0 <-ME0-PV
SE0 <-sqrt(diag(cov(out0)))
SE0 <-matrix(SE0,nrow=1,ncol=11)
ID0 <-matrix(c(0,N,alv,alx,alz,cvx,cvz,cxz,bez),nrow=1,ncol=9)

```

