

Computer Program for Estimation Multivariate Volatility Processes Using DVEC Model

Article Info:

Management Information Systems,
Vol. 3 (2008), No. 2,
pp. 011-017

Received 12 Jun 2008
Accepted 24 October 2008

UDC 004.42:336.76

Summary

This article presents computer program for estimation of multivariate (bivariate and trivariate) volatility processes, written in EViews Version 4.1. In order to estimate multivariate volatility processes for analysis of the Serbian financial market, I had to write new subprograms within Eviews software package. The programs are written for the diagonal vector ARCH model (DVEC) in bivariate and trivariate versions.

Key words

Multivariate GARCH model, Diagonal VEC model, volatility, EViews.

Introduction

It is often the case in finance that the covariance between some two series is of interest, just as the variances of the individual series themselves. While univariate descriptions of GARCH models are useful and important, the problems of risk assessment, asset allocation, hedging in futures markets and options pricing, portfolio Value at Risk and CAPM betas estimates require a multivariate framework. This is because all aforementioned problems require covariances as inputs. MGARCH models specify equations for how the variances and covariances move over time.

In this article I presented computer programs for estimation the multivariate GARCH models (MGARCH) with EViews, Version 4.1 using daily data from Belgrade stock exchange for two pairs of daily log returns for stocks and index. I considered bivariate and trivariate version of MGARCH models, diagonal vector ARCH (DVEC, by Bollerslev, Engle and Wooldridge, 1988) model. I used data of daily log returns for BELEX15 index, Hemofarm stock, and Energoprojekt stock, respectively. The data set covers the period from October 3, 2005 to October 6, 2006 (for detail see reference Minović, 2007b). A first simple method to estimate the parameters of a multivariate (bivariate and trivariate) GARCH models is the Berndt-Hall-Hall-Hausman (BHHH) algorithm. This algorithm uses the first derivatives of the quasi-maximum likelihood (QML) with respect to the number of parameters that are contained in multivariate GARCH models. This is an iterative

procedure, the BHHH algorithm needs suitable initial parameters (Franke, Härdle, & Hafner, 2005). Additionally, EViews does not support estimation of MGARCH models, so I extended EViews with new subprograms. This will require utilization of informatical methods (programming). For all calculations in DVEC programs for bivariate and trivariate versions, number of iteration is 100 and convergence criterion is $1 \cdot 10^{-5}$ which ought to be considered a procedure with high precision.

The rest of the paper is organized as follows. At the beginning of the second section I define vector ARCH model (VEC), diagonal VEC model (DVEC). The rest of Section 2 presents the program source code of bivariate DVEC model. Section 3 presents the program source code of trivariate DVEC model. The Section 4 concludes.

1. The Program Source Code of Bivariate DVEC Model

1.1. Definition of DVEC model

First, I introduce the VEC operator:

- VEC is the operator that stacks a matrix as a column vector:

$$VEC(\Sigma_t) = (\sigma_{11t}, \sigma_{21t}, \dots, \sigma_{N1t}, \sigma_{12t}, \sigma_{22t}, \dots, \sigma_{NNt})' \tag{2.1}$$

(Bauwens, 2005).

The general multivariate GARCH(p, q) model is given as:

$$VEC(\Sigma_t) = C + \sum_{i=1}^q A_i \cdot VEC(\varepsilon_{t-i} \varepsilon'_{t-i}) + \sum_{j=1}^p B_j \cdot VEC(\Sigma_{t-j}), \quad (2.2)$$

where A_i and B_j are parameter matrices containing $(N^*)^2$ parameters (with $N^* = N(N+1)/2$), whereas the vector C contains N^* coefficients. We will assume that all eigenvalues of the matrix $\sum_{i=1}^q A_i + \sum_{j=1}^p B_j$ have modulus smaller than one, in which case the vector process \square is covariance stationary with unconditional covariance matrix given by \square (Hafner & Herwartz, 2006).

A potentially serious issue with the unrestricted VEC model described by equation (2.2) is that it requires the estimation of a large number of parameters. This over-parameterization led to the development of the simplified diagonal VEC model, by Bollerslev, Engle and Wooldridge (1988), where the A and B matrices are forced to be diagonal. The result is a reduction of the number of parameters in the variance and covariance equations to 18 for the trivariate case (Brooks, Burke, & Persaud, 2003).

Because of the simplification that it provides, the diagonal VEC model is frequently used. Each of its variance-covariance terms is postulated to follow a GARCH-type equation. The model can be written as follows (Tse & Tsui Albert, 1999):

$$\sigma_{y,t} = c_{ij} + \sum_{h=1}^p a_{hij} \varepsilon_{t-h,i} \varepsilon_{t-h,j} + \sum_{h=1}^q b_{hij} \sigma_{t-h,ij} \quad 1 \leq i \leq j \leq k, \quad (2.3)$$

where c_{ij} , a_{hij} and b_{hij} are parameters. The diagonal VEC multivariate GARCH model could also be expressed as an infinite order multivariate ARCH model, where the covariance is expressed as a geometrically declining weighted average of past cross products of unexpected returns, with recent observations carrying higher weights. An alternative solution to the dimensionality problem would be to use orthogonal GARCH or factor GARCH models (Brooks, Burke, & Persaud, 2003).

Now, we will present the diagonal VEC model in the form:

$$\Sigma_t = C_0^* + \sum_{i=1}^m A_i^* \square (\varepsilon_{t-i} \varepsilon'_{t-i}) + \sum_{j=1}^s B_j^* \square \Sigma_{t-j}, \quad (2.4)$$

where m and s are non-negative integers, and \square denotes Hadamard product¹ (element by element matrix multiplication) (Tsay, 2005). Let us define

¹ If $X = (x_{ij})$ and $Y = (y_{ij})$ are both $m \times n$ matrices, then $X \square Y$ is the $m \times n$ matrix containing elementwise products $(x_{ij}y_{ij})$ (Bauwens, et al, 2006).

the symmetric $N \times N$ matrices A_i^* and B_j^* as the matrices implied by the relations $A = \text{diag}[\text{vec}(A^*)]^2$, and $B = \text{diag}[\text{vec}(B^*)]$ for and C_0^* as given by $C = \text{vec}(C_0^*)$ (Bauwens et al. 2006). The model which is represented by Eq. (2.4) is DVEC(m,s) model (Tsay, 2005). \square_t must be the parameter matrices, and only the lower portions of these matrices need to be parameterized and estimated. For example, Silberberg and Pafka (2001) prove that a sufficient condition to ensure the positive definiteness of the covariance matrix \square_t in Eq. (2.4) is that the constant term C_0^* is positive definite and all the other coefficient matrices, A_i^* and B_j^* , are positive semidefinite (De Goeij & Marquering, 2004).

Each element of \square_t depends only on its own past value and the corresponding product term in $\varepsilon_{t-1} \varepsilon'_{t-1}$. That is, each element of a DVEC model follows a GARCH(1,1) type model. The model is simple, but it may not produce a positive-definite covariance matrix. Furthermore, the model does not allow for dynamic dependence between volatility series (Tsay, 2005).

1.2. Bivariate DVEC model

My programs (in bivariate and trivariate version) are based on sample programs for BEKK (named after Baba, Engle, Kraft, and Kroner) model, released with EViews 4.1. However, these sample programs required significant modifications in order to be applied. For details about different versions MGARCH models and program codes you can see reference (Minović, 2007a).

The first, I define mean equation as:

$$\mathbf{r} = \mathbf{mu} + \mathbf{res},$$

where \mathbf{res} is error term (disturbance); $\mathbf{res} \sim N(0,H)$, \mathbf{res} is normally distributed with mean zero and variances H .

The second, DVEC model is:

$$\mathbf{VEC}(\mathbf{H}) = \mathbf{omega} + \mathbf{beta} * \mathbf{VEC}(\mathbf{H}(-1)) + \mathbf{alpha} * \mathbf{VEC}(\mathbf{res}(-1) \mathbf{res}(-1)'), \text{ where}$$

$$\mathbf{r} = 2 \times 1$$

$$\mathbf{mu} = 2 \times 1$$

$$\mathbf{H} = 2 \times 2 \text{ (symmetric)}$$

$$\mathbf{H}(1,1) = \text{variance of } \mathbf{r}1 \text{ (saved as var_r1)}$$

$$\mathbf{H}(1,2) = \text{covariance of } \mathbf{r}1 \text{ and } \mathbf{r}2 \text{ (saved as}$$

² If \mathbf{v} is a vector of dimensions m then $\text{diag}(\mathbf{v}) = I_m \times \mathbf{v}$ (Bauwens, et al, 2006).

```
cov_r1r2)
H(2,2) = variance of r2 (saved as var_r2)
omega = 2 x 2 low triangular
beta = 2 x 2 diagonal
alpha = 2 x 2 diagonal

%path = @runpath
cd "%path%"
Then, I load workfile:

load ar_residuals1$.wfl
```

Note: dependent variables of both series must be continues.

```
smpl @all
series r1 = resbelex15
series r2 = reshemofarm
```

Here, **r1** is residual of BELEX15 index from ARMA process and **r2** is residual of Hemofarm stock from ARMA process; the keyword "**@all**" refers to the entire workfile range; Note for seting sample: first observation of s1 need to be one or two periods before the first observation of s0.

```
sample s0 10/3/2005 09/20/2006
sample s1 10/10/2005 09/20/2006
```

Follows, initialization of parameters and starting values; change below only to change the specification of model

```
smpl s0
```

Hence, we estimate an GARCH(1,1) models with a mean equations consisting of returns (residuals from ARMA processes) regressed on a constant. The procedure will perform up to 100 iterations (termed as **m**) and, upon convergence (termed as **c**), will report Bollerslev-Wooldridge robust QML standard errors (termed as **h**). I use BHHH algorithm for maximization (termed as **b**). Thus, I get starting values from univariate GARCH

```
equation eq1.arch(m=100,c=1e-5,h,b) r1 c
equation eq2.arch(m=100,c=1e-5,h,b) r2 c
```

Then, I declare coefficient vectors to use in bivariate DVEC model.

```
coef(2) mu
mu(1) = eq1.c(1)
mu(2) = eq2.c(1)

coef(3) omega
omega(1) = eq1.c(2)
omega(2) = 0
omega(3) = eq2.c(2)

coef(3) alpha
alpha(1) = eq1.c(3)
```

```
alpha(2) = eq2.c(3)
alpha(3) = (alpha(1)*alpha(2))^.5

coef(3) beta
beta(1) = eq1.c(4)
beta(2) = eq2.c(4)
beta(3) = (beta(1)*beta(2))^.5
```

```
Constant adjustment for log likelihood is:
!mlog2pi = 2*log(2*@acos(-1))
```

I use var-cov of sample in "s1" as starting value of variance-covariance matrix.

```
series cov_r1r2 = @cov(r1-mu(1), r2-mu(2))
series var_r1 = @var(r1)
series var_r2 = @var(r2)
```

Squared error terms are named as sqres1 and sqres2; Cross product of errors is termed as res1res2.

```
series sqres1 = (r1-mu(1))^2
series sqres2 = (r2-mu(2))^2
series res1res2 = (r1-mu(1))*(r2-mu(2))
```

.....
LOG LIKELIHOOD

- set up the likelihood
1. open a new blank likelihood object (L.O.) name bvgarch
 2. specify the log likelihood model by append
-

```
logl bvgarch
bvgarch.append @logl logl
bvgarch.append sqres1 = (r1-mu(1))^2
bvgarch.append sqres2 = (r2-mu(2))^2
bvgarch.append res1res2 = (r1-mu(1))*(r2-mu(2))
```

Now, I calculate the variance and covariance series:

```
bvgarch.append var_r1 = omega(1)+
beta(1)*var_r1(-1) + alpha(1)*sqres1(-1)
bvgarch.append var_r2 = omega(3)+
beta(2)*var_r2(-1) + alpha(2)*sqres2(-1)
bvgarch.append cov_r1r2 = omega(2) +
beta(3)*cov_r1r2(-1) + alpha(3)*res1res2(-1)
```

Determinant of the variance-covariance matrix is:

```
bvgarch.append deth = var_r1*var_r2 -
cov_r1r2^2
```

Inverse elements of the variance-covariance matrix are:

```
bvgarch.append invh1 = var_r2/deth
bvgarch.append invh3 = var_r1/deth
```

```
bvgarch.append invh2 = -cov_r1r2/deth
```

Log-likelihood series is:

```
bvgarch.append logl = -0.5*(!mlog2pi +
(invh1*sqres1+2*invh2*res1res2+invh3*sqres2)
+ log(deth))
```

Now, I can estimate the bivariate DVEC model:

```
smpl s1
```

```
bvgarch.ml(showopts, m=100, c=1e-5, b)
```

where, **ml** = maximum likelihood estimation; **showopts** = Do display the starting coefficient values and estimation options in the estimation output; **m**-set maximum number of iterations; **c**-set convergence criterion; **b**-use BHHH algorithm for maximization.

Error terms are named as res1 and res2:

```
series res1 = sqres1^.5
series res2 = sqres2^.5
```

Standardized residuals of bivariate DVEC model are:

```
series stres1 = res1 / (var_r1 ^ .5)
series stres2 = res2 / (var_r2 ^ .5)
```

Squared standardized residuals of bivariate DVEC model are:

```
series sqres1st = stres1^2
series sqres2st = stres2^2
```

Cross product of standardized residuals of bivariate DVEC model is:

```
series stres1res2 = res1*res2 /
(var_r1*var_r2)^.5
```

I plot standardized residuals of bivariate DVEC model versus returns:

```
graph resids1.line stres1 r1
show resids1
graph resids2.line stres2 r2
show resids2
```

I present my results: variances of both returns and covariances between them.

```
show bvgarch.output
graph varcov.line var_r1 var_r2 cov_r1r2
show varcov
```

2. The Program Source Code of Trivariate DVEC Model

$\text{VEC}(H) = \text{omega} + \text{beta} * \text{VEC}(H(-1)) + \text{alpha} * \text{VEC}(\text{res}(-1) \text{res}(-1)')$, where

$r = 3 \times 1$

$\mu = 3 \times 1$

$H = 3 \times 3$ (symmetric)

$H(1,1)$ = variance of r_1 (saved as var_r1)

$H(1,2)$ = cov of r_1 and r_2 (saved as cov_r1r2)

$H(1,3)$ = cov of r_1 and r_3 (saved as cov_r1r3)

$H(2,2)$ = variance of r_2 (saved as var_r2)

$H(2,3)$ = cov of r_2 and r_3 (saved as cov_r2r3)

$H(3,3)$ = variance of r_3 (saved as var_r3)

omega = 3×3 low triangular

beta = 3×3 diagonal

alpha = 3×3 diagonal

```
%path = @runpath
```

```
cd "%path"
```

I have to load workfile

```
load ar_residuals3.wf1
```

Note: dependent variables of all series must be continues

```
series r1 = resbelex15
```

```
series r2 = reshemofarm
```

```
series r3 = resenergoprojekt
```

Here, **r1**, **r2**, **r3** are residuals of BELEX15 index, Hemofarm and Energoprojekt stocks, respectively from ARMA process.

Note for setting sample: first observation of s1 need to be one or two periods before the first observation of s0.

```
sample s0 10/3/2005 09/20/2006
```

```
sample s1 10/15/2005 09/05/2006
```

Initialization of parameters and starting values change below only to change the specification of model.

```
smpl s0
```

I estimate an GARCH(1,1) models with a mean equations consisting of returns (residuals from ARMA processes) regressed on a constant. The procedure will perform up to 100 iterations (termed as m) and, upon convergence (termed as c), will report Bollerslev-Wooldridge robust QML standard errors (termed as h). We use BHHH algorithm for maximization (termed as b). So, I get starting values from univariate GARCH

```
equation eq1.arch(m=100,c=1e-5,h,b) r1 c
```

```
equation eq2.arch(m=100,c=1e-5,h,b) r2 c
```

```
equation eq3.arch(m=100,c=1e-5,h,b) r3 c
```

I declare coefficient vectors to use in trivariate DVEC model:

```
coef(3) mu
```

```
mu(1) = eq1.c(1)
mu(2) = eq2.c(1)
mu(3) = eq3.c(1)
```

```
coef(6) omega
omega(1) = eq1.c(2)
omega(2) = eq2.c(2)
omega(3) = eq3.c(2)
omega(4) = 0
omega(5) = 0
omega(6) = 0
```

```
coef(6) alpha
alpha(1) = eq1.c(3)
alpha(2) = eq2.c(3)
alpha(3) = eq3.c(3)
alpha(4) = (alpha(1)*alpha(2))^.5
alpha(5) = (alpha(1)*alpha(3))^.5
alpha(6) = (alpha(2)*alpha(3))^.5
```

```
coef(6) beta
beta(1) = eq1.c(4)
beta(2) = eq2.c(4)
beta(3) = eq3.c(4)
beta(4) = (beta(1)*beta(2))^.5
beta(5) = (beta(1)*beta(3))^.5
beta(6) = (beta(2)*beta(3))^.5
```

I use sample var-cov as starting value of variance-covariance matrix:

```
series cov_r1r2 = @cov(r1-mu(1), r2-mu(2))
series cov_r1r3 = @cov(r1-mu(1), r3-mu(3))
series cov_r2r3 = @cov(r2-mu(2), r3-mu(3))
series var_r1 = @var(r1)
series var_r2 = @var(r2)
series var_r3 = @var(r3)
```

I named squared error terms of BELEX15 index, Hemofarm and Energoprojekt stocks as sqres1, sqres2 and sqres3, respectively.

```
series sqres1 = (r1-mu(1))^2
series sqres2 = (r2-mu(2))^2
series sqres3 = (r3-mu(3))^2
```

Cross product of error termes we calculate as:

```
series res1res2 = (r1-mu(1))*(r2-mu(2))
series res1res3 = (r1-mu(1))*(r3-mu(3))
series res2res3 = (r3-mu(3))*(r2-mu(2))
```

constant adjustment for log likelihood is:

```
lmlog2pi = 3*log(2*@acos(-1))
.....
```

LOG LIKELIHOOD

set up the likelihood

1) open a new blank likelihood object name tvgarch

2) specify the log likelihood model by append

```
logl tvgarch
```

Squared errors and cross errors are:

```
tvgarch.append @logl logl
tvgarch.append sqres1 = (r1-mu(1))^2
tvgarch.append sqres2 = (r2-mu(2))^2
tvgarch.append sqres3 = (r3-mu(3))^2
```

```
tvgarch.append res1res2 = (r1-mu(1))*(r2-mu(2))
tvgarch.append res1res3 = (r1-mu(1))*(r3-mu(3))
tvgarch.append res2res3 = (r3-mu(3))*(r2-mu(2))
```

Variance and covariance series are:

```
tvgarch.append var_r1 = omega(1) +
beta(1)*var_r1(-1) + alpha(1)*sqres1(-1)
tvgarch.append var_r2 = omega(2) +
beta(2)*var_r2(-1) + alpha(2)*sqres2(-1)
tvgarch.append var_r3 = omega(3) +
beta(3)*var_r3(-1) + alpha(3)*sqres3(-1)
tvgarch.append cov_r1r2 = omega(4) +
beta(4)*cov_r1r2(-1) + alpha(4)*res1res2(-1)
tvgarch.append cov_r1r3 = omega(5) +
beta(5)*cov_r1r3(-1) + alpha(5)*res1res3(-1)
tvgarch.append cov_r2r3 = omega(6) +
beta(6)*cov_r2r3(-1) + alpha(6)*res2res3(-1)
```

Determinant of the variance-covariance matrix is:

```
tvgarch.append deth = var_r1*var_r2*var_r3 -
var_r1*cov_r2r3^2-
cov_r1r2^2*var_r3+2*cov_r1r2*cov_r2r3*cov_r1r3-
cov_r1r3^2*var_r2
```

I calculate the elements of the inverse of var_cov (H) matrix and numbered as vech(inv(H)).

```
tvgarch.append invh1 = (var_r2*var_r3-
cov_r2r3^2)/deth
tvgarch.append invh2 = -(cov_r1r2*var_r3-
cov_r1r3*cov_r2r3)/deth
tvgarch.append invh3 = (cov_r1r2*cov_r2r3-
cov_r1r3*var_r2)/deth
tvgarch.append invh4 = (var_r1*var_r3-
cov_r1r3^2)/deth
tvgarch.append invh5 = -(var_r1*cov_r2r3-
cov_r1r2*cov_r1r3)/deth
tvgarch.append invh6 = (var_r1*var_r2-
cov_r1r2^2)/deth
```

Log-likelihood series is:

```
tvgarch.append logl = -0.5*(lmlog2pi +
(invh1*sqres1+invh4*sqres2+invh6*sqres3
+2*invh2*res1res2
+2*invh3*res1res3+2*invh5*res2res3) +
log(deth))
```

I estimate the trivariate DVEC model as:

```
smpl s1
tvgarch.ml(showopts, m=100, c=1e-5, b)
```

where ml = maximum likelihood estimation; showopts = Do display the starting coefficient values and estimation options in the estimation output; m-set maximum number of iterations; c-set convergence criterion; b-use BHHH algorithm for maximization.

Error terms are named as res1, res2 and res3:

```
series res1 = sqres1^.5
series res2 = sqres2^.5
series res3 = sqres3^.5
```

The standardized residuals of trivariate DVEC model are:

```
series stres1 = res1 / (var_r1 ^ .5)
series stres2 = res2 / (var_r2 ^ .5)
series stres3 = res3 / (var_r3 ^ .5)
```

The squared standardized residuals of trivariate DVEC model are:

```
series sqres1st = stres1^2
series sqres2st = stres2^2
series sqres3st = stres3^2
```

Cross product of standardized residuals of trivariate DVEC model:

```
series stres1res2 = res1*res2 /
(var_r1*var_r2)^.5
series stres1res3 = res1*res3 /
(var_r1*var_r3)^.5
series stres2res3 = res2*res3 /
(var_r2*var_r3)^.5
```

I plot standardized residuals of trivariate DVEC model vs. returns:

```
graph resid1.line(m) stres1 r1
show resid1
graph resid2.line(m) stres2 r2
show resid2
graph resid3.line(m) stres3 r3
show resid3
```

I present my results: variances for each return and covariances between them.

```
show tvgarch.output
graph var.line var_r1 var_r2 var_r3
graph cov.line cov_r1r2 cov_r1r3 cov_r2r3
```

```
show var
show cov
```

Conclusion

This article presents computer programs for calculation of multivariate GARCH models (DVEC) in bivariate and trivariate versions. In order to estimate the multivariate DVEC models with Eviews, I had to write new subprograms. I illustrated my approach by applying it to daily returns of the BELEX15 index, Hemofarm and Energoprojekt stocks. Although, new version of Eviews (6.0) supports multivariate GARCH modeling through regular program menus and commands, manually writing a program gives us more freedom and flexibility. Hence, these programs are useful for estimation volatilities of particular securities, and for co-movement estimation between two securities. It is very important, because it is often the case in finance that the covariance between some two series is of interest, just as the variances of the individual series themselves. So, using these programs one can see behaviour of conditional covariances over time for the both stocks and the index.

References

- Bauwens, L. (2005). MGARCH-slides-LB-print. Universit e catholique de Louvain.
- Bauwens, L., Laurent, S., & Rombouts, K. J. (2006). Multivariate GARCH models: A survey. *Journal of Applied Econometrics*, 79-109.
- Brooks, C. (2002). *Introductory Econometrics for Finance*. Cambridge: Cambridge University Press.
- Brooks, C., Burke, S., & Persaud, G. (2003). Multivariate GARCH Models: Software Choice and Estimation Issues. *Journal of Applied Econometrics*, 18 (6), 725-734.
- De Goeij, P., & Marquering, W. (2004). Modeling the Conditional Covariance Between Stock and Bond Returns: A Multivariate GARCH Approach. *Journal of Financial Econometrics*, 2 (4), 531-564.
- EViews 5 User's Guide. (2004). Irvine: Quantitative Micro Software.
- Franke, J., Hardle, W., & Hafner, C. (2005, July). Introduction to Statistics of Financial Markets. Retrieved November 2007, from <http://www.quantlet.com/mdstat/scripts/sfe/html/sfiframe131.html>
- Hafner, C., & Herwartz, H. (2006). Volatility impulse response for multivariate GARCH models: An exchange rate illustration. *Journal of International Money and Finance*, 1-22.
- Minovic, J. (2007). Application of multivariate GARCH models in Serbian financial market analysis. Proceedings of International Conference, International Scientific Conference. Belgrade: Faculty of Economics, University of Belgrade.
- Minovic, J. Z. (2007). Multivariate GARCH models: Theoretical Survey and Model Application, M.Sc. Thesis. Belgrade: University of Belgrade, Faculty of Economics.
- Tsay, R. S. (2005). *Analysis of Financial Time Series*. New Jersey: Wiley.
- Tse, K. Y., & Tsui Albert, C. K. (1999). A note on diagnosing multivariate conditional heteroscedasticity models. *Journal of time series analysis*, 20 (6), 679-691.

Jelena Z. Minović

Union University
Faculty for Banking, Insurance and Finance
Zmaj Jovina 12
11000 Belgrade, Serbia

Email: jelena.minovic@gmail.com; jminovic@ien.bg.ac.yu
