Aug. 12 2008, useR!2008 in Dortmund, Germany.

# ccgarch: An R package for modelling multivariate GARCH models with conditional correlations


Tomoaki Nakatani

Department of Agricultural Economics
Hokkaido University, Japan
and
Department of Economic Statistics
Stockholm School of Economics, Sweden

# 1 Multivariate GARCH models

Involve covariance estimation

- Direct:
  - VEC representation
  - BEKK representation
- **Indirect**: through conditional correlations
  - GARCH part
    - Volatility spillovers, asymmetry etc.
  - Correlation part
    - Constant Conditional Correlation (CCC)
    - Dynamic Conditional Correlation (DCC)
    - Smooth Transition Conditional Correlation (STCC)

# Conditional Correlation GARCH models

## 2 GARCH part: with/without spillovers

A vector GARCH$(1,1)$ equation:

$$\mathbf{h}_t = \mathbf{a} + \mathbf{A}\boldsymbol{\varepsilon}_{t-1}^{(2)} + \mathbf{B}\mathbf{h}_{t-1}, \quad \varepsilon_{i,t} = h_{i,t}^{1/2} z_{i,t}, \quad \mathbf{z}_t \sim \mathsf{ID}(\mathbf{0}, \mathbf{P}_t)$$

The **diagonal** specification (no volatility spillovers)

$$\mathbf{h}_t = \begin{bmatrix} a_{10} \\ a_{20} \end{bmatrix} + \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix} \begin{bmatrix} \varepsilon_{1,t-1}^2 \\ \varepsilon_{2,t-1}^2 \end{bmatrix} + \begin{bmatrix} b_{11} & 0 \\ 0 & b_{22} \end{bmatrix} \begin{bmatrix} h_{1,t-1} \\ h_{2,t-1} \end{bmatrix}$$

The **extended** specification (allowing for volatility spillovers)

$$\mathbf{h}_t = \begin{bmatrix} a_{10} \\ a_{20} \end{bmatrix} + \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \varepsilon_{1,t-1}^2 \\ \varepsilon_{2,t-1}^2 \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} h_{1,t-1} \\ h_{2,t-1} \end{bmatrix}$$

# 3 Conditional Correlation Part

CCC and ECCC of Bollerslev(1990) and Jeantheau (1998)

$$\mathbf{P}_t = \mathbf{P} \quad \text{(constant over time)}$$

DCC of Engle (2002) and Engle and Sheppard (2001)

$$\mathbf{P}_t = (\mathbf{Q}_t \odot \mathbf{I}_N)^{-1/2}\mathbf{Q}_t(\mathbf{Q}_t \odot \mathbf{I}_N)^{-1/2}$$
$$\mathbf{Q}_t = (1-\alpha-\beta)\mathbf{Q} + \alpha\mathbf{z}_{t-1}\mathbf{z}'_{t-1} + \beta\mathbf{Q}_{t-1}$$
$$\alpha+\beta<1 \quad \text{and} \quad \alpha,\beta>0$$

where $\mathbf{Q}$ is a sample covariance matrix of $\mathbf{z}_t$.

STCC of Silvennoinen and Teräsvirta(2005)

$$\mathbf{P}_t = (1-G_t)\mathbf{P}_{(1)} + G_t\mathbf{P}_{(2)}$$
$$G_t = [1+\exp\{-\gamma(s_t-c)\}]^{-1}, \gamma>0$$

# The package

# 4 Description of the package

| | |
|---|---|
| Name: | `ccgarch` |
| Version: | 0.1.0 (continuously updated) |
| Author: | Tomoaki Nakatani ⟨`naktom2@gmail.com`⟩ |
| Depends: | R 2.6.1 or later |
| Description: | Functions for estimating and simulating the family of the CC-GARCH models. |
| Simulating: | the first order (E)CCC-GARCH, (E)DCC-GARCH, (E)STCC-GARCH |
| Estimating: | the first order (E)CCC-GARCH, (E)DCC-GARCH |
| Availability: | Not yet submitted to CRAN. Available upon request. |

# 5 Functions for simulation

CCC-GARCH and Extended CCC-GARCH models

```
eccc.sim(nobs, a, A, B, R, d.f=Inf,
         cut=1000, model)
```

DCC-GARCH and Extended DCC-GARCH models

```
dcc.sim(nobs, a, A, B, R, dcc.para,
        d.f=Inf, cut=1000, model)
```

STCC-GARCH and Extended STCC-GARCH models

```
stcc.sim(nobs, a, A, B, R1, R2, tr.par,
         st.par, d.f=Inf, cut=1000, model)
```

# 6 Generating data from DCC-GARCH(1,1) (1)

Arguments for `dcc.sim`

```
dcc.sim(nobs, a, A, B, R, dcc.para,
        d.f=Inf, cut=1000, model)
```

| | |
|---|---|
| `nobs`: | number of observations to be simulated $(T)$ |
| `a`: | vector of constants in the GARCH equation $(N \times 1)$ |
| `A`: | ARCH parameter in the GARCH equation $(N \times N)$ |
| `B`: | GARCH parameter in the GARCH equation $(N \times N)$ |
| `R`: | unconditional correlation matrix $(N \times N)$ |
| `dcc.para`: | vector of the DCC parameters $(2 \times 1)$ |
| `d.f`: | degrees of freedom parameter for the $t$-distribution |
| `cut`: | number of observations to be removed |
| `model`: | character string, "diagonal" or "extended" |

# 7 Generating data from DCC-GARCH(1,1) (2)

Output from `dcc.sim` — a list with components:

- `z`: random draws from $\mathsf{N}(\mathbf{0}, \mathbf{I})$. $(T \times N)$
- `std.z`: standardised residuals, $\texttt{std.z}_t \sim \mathsf{ID}(0, \mathbf{R}_t)$. $(T \times N)$
- `dcc`: dynamic conditional correlations $\mathbf{R}_t$. $(T \times N^2)$
- `h`: simulated volatilities. $(T \times N)$
- `eps`: time series with DCC-GARCH process. $(T \times N)$

The DCC matrix at time $t = 10$, say, is obtained by

```
dcc.data <- dcc.sim(nobs, a, A, B, R, dcc.para,
                    d.f=Inf, cut=1000, model="diagonal")
dcc <- dcc.data$dcc
Rt.10 <- matrix(dcc[10,], nrow=length(a))
```

# 8 Functions for estimation

## CCC-GARCH and Extended CCC-GARCH models

```
eccc.estimation(a, A, B, R, dvar, model)
```

- Calls "optim" for simultaneous estimation of all parameters
- Uses "BFGS" algorithm

## DCC-GARCH and Extended DCC-GARCH models

```
dcc.estimation(a, A, B, dcc.para, dvar, model)
```

- Calls "optim" for the first stage (volatility part)
- Calls "constrOptim" for the second stage (DCC part)
- Uses "BFGS" algorithm

For STCC-GARCH; to be available in a future version

# 9 Estimating a DCC-GARCH model (1)

Arguments for `dcc.estimation`

```
dcc.estimation(a, A, B, dcc.para, dvar, model)
```

| | |
|---|---|
| `a`: | initial values for the constants $(N \times 1)$ |
| `A`: | initial values for the ARCH parameter $(N \times N)$ |
| `B`: | initial values for the GARCH parameter $(N \times N)$ |
| `dcc.para`: | initial values for the DCC parameters $(2 \times 1)$ |
| `dvar`: | a matrix of the observed residuals $(T \times N)$ |
| `model`: | character string, "diagonal" or "extended" |

# 10 Estimating a DCC-GARCH model (2)

Output from `dcc.estimation`—A list with components:

| | |
|---|---|
| `out`: | the estimates and their standard errors |
| `h`: | a matrix of the estimated volatilities $(T \times N)$ |
| `DCC`: | a matrix of DCC estimates $(T \times N^2)$ |
| `first`: | the results of the first stage estimation |
| `second`: | the results of the second stage estimation |

# 11 Illustrative example (1)

Simulation design:

DGPs: two diagonal DCC-GARCH(1,1) processes.

- normally and $t$-distributed (df $= 10$) innovations

Number of observations $(T)$: 3000

Number of dimensions $(N)$: 2

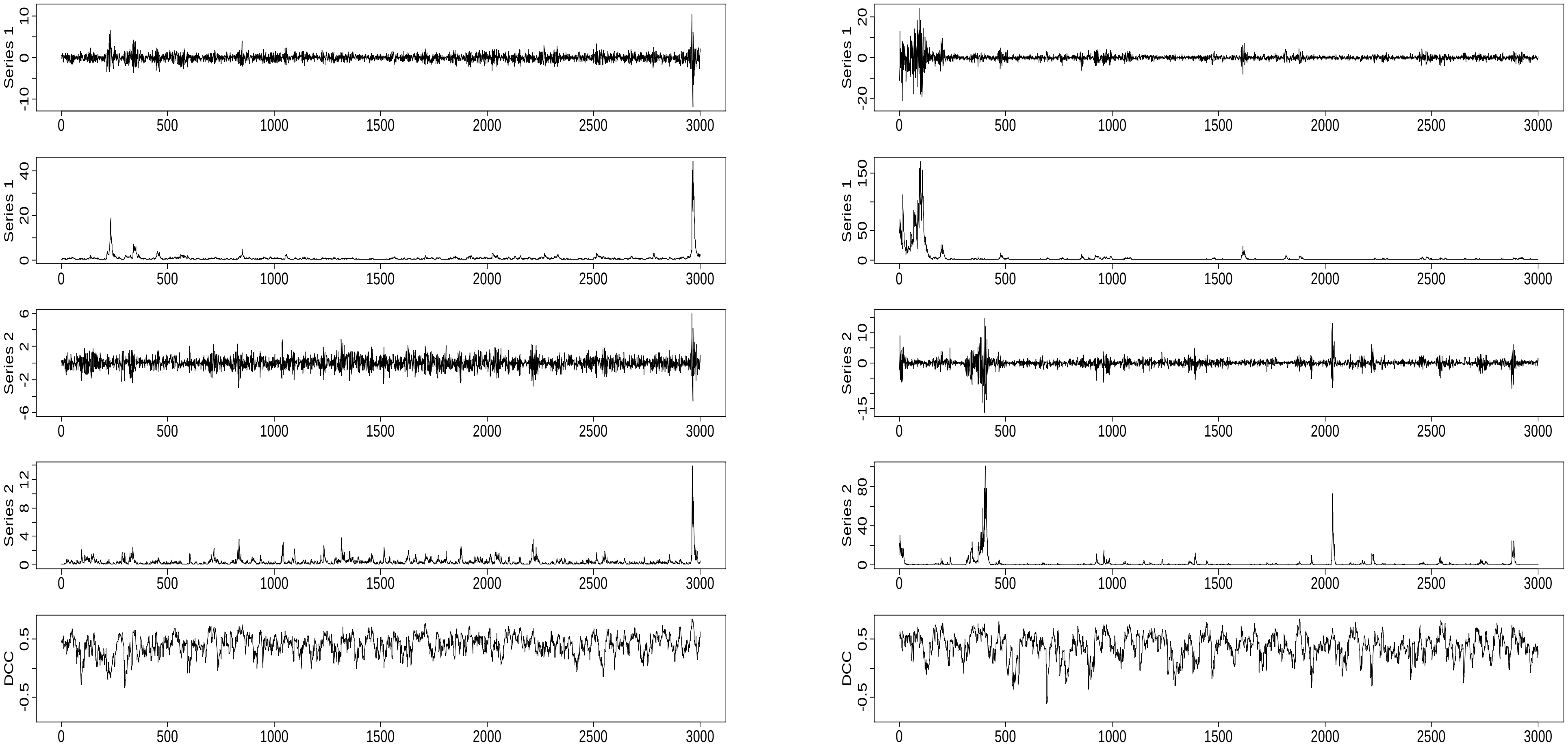
Function: `dcc.sim`

Estimation: `dcc.estimation`

Initial values: true parameter values

Note: This is just an illustrative example.

# 12 Illustrative example (2)




Normally distributed innovations    $t$-distributed innovations ($df = 10$)

Fig. 1 Two Simulated Data Series

# 13 Illustrative example (3)

Estimation results: Normally distributed innovations

| | a1 | a2 | A11 | A22 | B11 | B22 | alpha | beta |
|---|---|---|---|---|---|---|---|---|
| true.para | 0.030 | 0.050 | 0.200 | 0.300 | 0.750 | 0.600 | 0.100 | 0.800 |
| estimates | 0.044 | 0.043 | 0.236 | 0.276 | 0.709 | 0.637 | 0.106 | 0.785 |
| std.err | 0.007 | 0.022 | 0.023 | 0.006 | 0.025 | 0.028 | 0.012 | 0.029 |

Estimation results: $t$-distributed innovations

| | a1 | a2 | A11 | A22 | B11 | B22 | alpha | beta |
|---|---|---|---|---|---|---|---|---|
| true.para | 0.030 | 0.050 | 0.200 | 0.300 | 0.750 | 0.600 | 0.100 | 0.800 |
| estimates | 0.036 | 0.051 | 0.229 | 0.400 | 0.757 | 0.602 | 0.126 | 0.782 |
| std.err | 0.008 | 0.024 | 0.022 | 0.007 | 0.034 | 0.023 | 0.014 | 0.026 |

# Available utility functions

# 14 The Ljung-Box test for serial correlations

Usage:

```
ljung.box.test(x)
```

Arguments:

`x`: an univariate time series $(T \times 1)$

Example:

```
> round(ljung.box.test(eps[,1]),5)
          test stat p-value
   Lag 5  23.14052  0.00032
   Lag 10 30.54570  0.00070
             (omitted)
   Lag 45 74.74672  0.00351
   Lag 50 78.33926  0.00638
```

# 15 The Jarque-Bera test of normality

Usage:

```
jb.test(x)
```

Arguments:

`x`: a matrix of data set $(T \times N)$

Example:

```
> jb.test(eps)
             series 1 series 2
   test stat 49236.54  1908.62
   p-value       0.00     0.00
```

# 16 Robustified skewness and kurtosis

Usage: `rob.sk(x)`, `rob.kr(x)`

Description:

- Skewness and kurtosis measures
- their robustified versions by Kim and White (2004).

Arguments:

x: a matrix of data set $(T \times N)$

Example:

```
> round(rob.sk(eps),4)
           series 1 series 2
  standard   0.4596   0.2067
  robust    -0.0645  -0.0161
> round(rob.kr(eps),4)
           series 1 series 2
  standard  19.8254   3.8856
  robust     0.0520   0.1274
```

# 17 Stationarity condition of the GARCH part

Usage:

```
stationaity(A, B)
```

Arguments:

- `A`: an ARCH parameter matrix $(N \times N)$
- `B`: a GARCH parameter matrix $(N \times N)$

Value:

A module of the largest eigen value of $(\mathtt{A} + \mathtt{B})$.
Computed by $\mathtt{max}(\mathtt{Mod}(\mathtt{eigen}(\mathtt{A} + \mathtt{B})\$\mathtt{values}))$

Note:

This function is useful in the extended models.
In diagonal models, `max(A+B)` gives the answer.

# 18 Remaining tasks

- Very urgent: things mentioned in abstract
  - Procedures for diagnostic tests
  - A procedure for estimating an STCC-GARCH model
  - Allowing for negative volatility spillovers (non-trivial)
- Less urgent but important
  - The conditional mean part
  - Efficient coding for partial derivatives (use C?)
  - Allowing for higher orders in the GARCH part
  - More informative help files , eg, adding more examples
- Long term
  - Functions for graphics, eg plotting outputs, etc.
  - Improving the user interface