Package 'BEKKs'

January 20, 2022

Title Multivariate Conditional Volatility Modelling and Forecasting

Version 1.0.0

Description Methods and tools for estimating, simulating and forecasting of so-called BEKK-models (named after Baba, Engle, Kraft and Kroner) based on the fast Berndt–Hall–Hall–Hausman (BHHH) algorithm described in Hafner and Herwartz (2008) <doi:10.1007/s00184-007-0130-y>.

Depends R (>= 3.5.0)

Imports Rcpp, expm, reshape2, ggplot2, mathjaxr, gridExtra, grid, ggfortify, parallel, xts, stats, future, forecast, future.apply

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

SystemRequirements C++11

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Suggests testthat (>= 2.1.0)

RdMacros mathjaxr

RoxygenNote 7.1.2

R topics documented:

BEKKs *BEKKs: Volatility modelling*

Description

This package implements estimation, simulation and forecasting techniques for conditional volatility modelling using the BEKK model. Currently, the BEKK(1,1,1) model of Engle and Kroner (1995)

$$H_t = CC' + A' r_{t-1} r_{t-1}' A + G' H_{t-1} G$$

and the asymmetric extensions of Kroner and Ng (1998) and Grier et. al. (2004)

$$H_t = CC' + A' r_{t-1} r_{t-1}' A + B' \gamma_{t-1} \gamma_{t-1}' B + G' H_{t-1} G$$

with

$$\gamma_t = r_t I\left(r_t < 0\right)$$

are implemented.

Details

The main functions are:

bekk_spec Specifies the model type to be estimated,

- bekk_fit Estimates a BEKK(1,1,1) model of a given series and specification object bekk_spec,
- bekk_sim Simulates a BEKK(1,1,1) process using either a bekk_sim or bekk_spec object,
- bekk_forecast Forecasts conditional volatility using a bekk_fit object,
- VaR Estimates (portfolio) Value-at-Risk using a fitted BEKK(1,1,1) model.
-

Author(s)

- Markus Fülle <fuelle@uni-goettingen.de>

- Helmut Herwartz <hherwartz@uni-goettingen.de>
- Alexander Lange <alexander.lange@uni-goettingen.de>

References

Engle, R. F. and K. F. Kroner (1995). Multivariate simultaneous generalized arch. Econometric Theory 11(1),122–150.

Kroner, K. F. and V. K. Ng (1998). Modeling asymmetric comovements of asset returns. Review of Financial Studies 11(4), 817–44.

Grier, K. B., Olan T. Henry, N. Olekalns, and K. Shields (2004). The asymmetric effects of uncertainty on inflation and output growth. Journal of Applied Econometrics 19(5), 551–565.

`bekk_fit` *Estimating multivariate BEKK-type volatility models*

Description

Method for fitting a variety of N-dimensional BEKK models.

Usage

```
bekk_fit(spec, data, QML_t_ratios = FALSE, max_iter = 50, crit = 1e-09)
```

Arguments

`spec`	An object of class "bekkSpec" from function bekk_spec.
`data`	A multivariate data object. Can be a numeric matrix or ts/xts/zoo object.
`QML_t_ratios`	Logical. If QML_t_ratios = 'TRUE', the t-ratios of the BEKK parameter matrices are exactly calculated via second order derivatives.
`max_iter`	Maximum number of BHHH algorithm iterations.
`crit`	Determines the precision of the BHHH algorithm.

Details

The BEKK optimization routine is based on the Berndt–Hall–Hall–Hausman (BHHH) algorithm and is inspired by the study of Hafner and Herwartz (2008). The authors provide analytical formulas for the score and Hessian of several MGARCH models in a QML framework and show that analytical derivations significantly outperform numerical methods.

Value

Returns a S3 class "bekkFit" object containing the estimated parameters, t-values, volatility process of the model defined by the BEKK_spec object.

References

Hafner and Herwartz (2008). Analytical quasi maximum likelihood inference in multivariate volatility models. Metrika, 67, 219-239.

Examples

```
data(StocksBonds)

# Fitting a symmetric BEKK model
obj_spec <- bekk_spec()
x1 <- bekk_fit(obj_spec, StocksBonds, QML_t_ratios = FALSE, max_iter = 50, crit = 1e-9)

summary(x1)

plot(x1)

# Fitting an asymmetric BEKK model
obj_spec <- bekk_spec(model = list(type = "bekk", asymmetric = TRUE))
x1 <- bekk_fit(obj_spec, StocksBonds)

summary(x1)

plot(x1)
```

`bekk_forecast` *Forecasting conditional volatilities with BEKK models*

Description

Method for forecasting a N-dimensional BEKK covariances.

Usage

```
bekk_forecast(x, n.ahead = 1, ci = 0.95)
```

Arguments

`x`	A fitted bekk model of class bekk from the bekk_fit function
`n.ahead`	Number of periods to forecast conditional volatility. Default is a one-period ahead forecast.
`ci`	Floating point in [0,1] defining the niveau for confidence bands of the conditional volatility forecast. Default is 95 per cent niveau confidence bands.

Value

Returns a S3 class "bekkForecast" object containing the conditional volatility forecasts and respective confindence bands.

Examples

```
#'
data(StocksBonds)
obj_spec <- bekk_spec()
x1 <- bekk_fit(obj_spec, StocksBonds, QML_t_ratios = FALSE, max_iter = 50, crit = 1e-9)

x2 <- bekk_forecast(x1, n.ahead = 1)
```

---

bekk_sim *Simulating BEKK models*

Description

Method for simulating a N-dimensional BEKK model.

Usage

```
bekk_sim(spec, nobs)
```

Arguments

spec	A spec object of class "bekkSpec" from the function bekk_spec or a fitted bekk model of class "bekkFit" from the bekk_fit function
nobs	Number of observations of the simulated sample

Value

Returns a simulated time series S3 class object using the parameters of passed "bekkSpec" or "bekkFit".

Examples

```
# Simulate a BEKK with estimated parameter
obj_spec <- bekk_spec()
x1 <- bekk_fit(obj_spec, StocksBonds)

x2 <- bekk_sim(x1, 3000)
```

```
plot(x2)
```

---

bekk_spec *BEKK specification method*

Description

Method for creating a N-dimensional BEKK model specification object prior to fitting and/or simulating.

Usage

```
bekk_spec(
  model = list(type = "bekk", asymmetric = FALSE),
  init_values = NULL,
  signs = NULL,
  N = NULL,
  compare = FALSE
)
```

Arguments

model	A list containing the model type specification: Currently implemented is only "bekk" ("dbekk" and "sbekk" are forthcoming). Moreover it can be specified whether the model should be estimated allowing for asymmetric volatility structure.
init_values	initial values for bekk_fit during BHHH algorithm. It can be either a numerical vector of suitable dimension, or a character vector i.e. "random" to use a random starting value generator (set a seed in advance for reproducible results), or "simple" for relying on a simple initial values generator based on typical values for BEKK parameter found in the literature. If the object from this function is passed to bekk_sim, "init_values" are used as parameters for data generating process.
signs	An N-dimensional vector consisting of "1" or "-1" to indicate the asymmetric effects to be considered. Setting the i-th element of the vector to "1" or "-1" means that the model takes into account additional volatility if the returns of the i-th column in the data matrix are either positive or negative. If "asymmetric = TRUE", the default is set to "rep(-1, N)" i.e. it is assumed that excess volatility occurs for all series if the returns are negative.
N	Integer specifying the dimension of the BEKK model. Only relevant for bekk_sim.
compare	Boolean specifying if the outcome of an asymmetric model is compared to its symmetric estimation result.

Value

Returns a S3 class "bekkSpec" object containing the specifications of the model to be estimated.

GoldStocksBonds *Gold stock and Bond returns*

Description

Trivariate data set consisting of daily gold, S&P 500 and U.S. Treasury Bond Future returns from October 1991 to October 2021.

Usage

```
data("GoldStocksBonds")
```

Format

A data frame with 7346 observations on the following 3 variables.

Gold a numeric vector

S&P 500 a numeric vector

US Treasury Bond Future a numeric vector

Source

Yahoo Finance.

Examples

```
data(GoldStocksBonds)
## maybe str(GoldStocksBonds) ; plot(GoldStocksBonds) ...
```

StocksBonds *Daily stock and Bond returns*

Description

Bivariate data set consisting of daily S&P 500 bond and MSCI World returns from December 1995 to December 2019.

Usage

```
data("StocksBonds")
```

Format

A data frame with 6073 observations on the following 2 variables.

S&P 500 Bonds a numeric vector

MSCI World a numeric vector

Source

Yahoo Finance.

Examples

```
data(StocksBonds)
## maybe str(StocksBonds) ; plot(StocksBonds) ...
```

VaR *Calculating Value-at-Risk (VaR)*

Description

Method for calculating VaR from estimated covariance processes (bekk_fit) or predicted covariances (bekk_forecast).

Usage

```
VaR(x, p = 0.99, portfolio_weights = NULL)
```

Arguments

x	An object of class "bekkFit" from the function bekk_fit or an object of class "bekkForecast" from the function bekk_forecast.
p	A numerical value that determines the confidence level. The default value is set at 0.99 in accordance with the Basel Regulation.
portfolio_weights	A vector determining the portfolio weights to calculate the portfolio VaR. If set to "NULL", the univariate VaR for each series are calculated.

Value

Returns a S3 class "var" object containing the VaR forecast and respective confidence bands.

Examples

```
data(StocksBonds)
obj_spec <- bekk_spec()
x1 <- bekk_fit(obj_spec, StocksBonds, QML_t_ratios = FALSE, max_iter = 50, crit = 1e-9)

# single VaRs of series
x2 <- VaR(x1)
plot(x2)

# VaR of equally-weighted portfolio
portfolio_weights <- c(0.5, 0.5)
x3 <- VaR(x1, portfolio_weights = portfolio_weights)
plot(x3)

# VaR of traditional 30/70 weighted bond and stock portfolio
portfolio_weights <- c(0.3, 0.7)
x4 <- VaR(x1, portfolio_weights = portfolio_weights)
plot(x4)
```

Index