Package ‘bayesDccGarch’

February 7, 2016

Version 2.0
Date 2016-01-29
Title The Bayesian Dynamic Conditional Correlation GARCH Model
Depends R (>= 2.0), numDeriv, coda
Author Jose A Fiorucci <jafioruci@gmail.com>, Ricardo S Ehlers <ehlers@icmc.usp.br>, Francisco Louzada <louzada@icmc.usp.br>
Maintainer Jose A Fiorucci <jafioruci@gmail.com>
BugReports Send an email for <jafioruci@gmail.com> with title 'bayesDccGarch Bug'
License GPL (>= 2)
URL http://arxiv.org/abs/1412.2967
NeedsCompilation yes
Repository CRAN
Date/Publication 2016-02-07 09:28:39

R topics documented:

bayesDccGarch-package ............................................. 2
bayesDccGarch ..................................................... 3
DaxCacNik ............................................................. 7
densityFunctions ................................................... 7
logLikDccGarch ..................................................... 9
plot.bayesDccGarch ............................................... 10
plotVol ............................................................. 11
Index 13
bayesDccGarch-package  bayesDccGARCH: Methods and tools for Bayesian analysis of DCC-GARCH\(1,1\) Model.

Description

In this package we implemented functions for Bayesian analysis of DCC-GARCH\(1,1\) Model using the same modelling of Fioruci et al (2014a). Several probabilities distributions are available for the errors which can model both skewness and heavy tails. See Fioruci et al (2014b) for more details about the package.

Details

Package: bayesDccGarch
Type: Package
Version: 2.0
Date: 2016-01-29
License: GPL (>=2.14)

bayesDccGarch(mY, n_sim = 10000)

Author(s)

Jose Augusto Fiorucci, Ricardo Sandes Ehlers and Francisco Louzada. Maintainer: Jose Augusto Fiorucci <jafioruci@gmail.com> and Ricardo Sandes Ehlers <ehlers@icmc.usp.br

References


See Also

Available functions: bayesDccGarch, logLikDccGarch, dssnorm, dsst, dssged, plot, plotVol
For classical estimation see Package ccgarch.

Examples

data(DaxCacNik)
Dax = DaxCacNik[,1]
## Description

Performs a Markov Chain for all parameters of the DCC-GARCH(1,1) Model.

## Usage

```r
bayesDccGarch(mY, nSim = 10000, tail_ini = 8, omega_ini = rep(0.03, ncol(mY)),
alpha_ini = rep(0.03, ncol(mY)), beta_ini = rep(0.8, ncol(mY)),
aIni = 0.03, bIni = 0.8, gamma_ini = rep(1, ncol(mY)),
errorDist = 2, control = list())

increaseSim(x, nSim=10000)
```
## S3 method for class 'bayesDccGarch'
window(x, start = NULL, end = NULL, thin = NULL, ...)

### Arguments

- `mY`: a matrix of the data \((n \times k)\).
- `nSim`: length of Markov chain. Default: 10000.
- `tail_ini`: initial value of \(\nu\) parameter if `errorDist = 2` or initial value of \(\delta\) parameter if `errorDist = 3`. If `errorDist = 1` this argument is not used.
- `omega_ini`: a numeric vector \((k \times 1)\) with the initial values of \(\omega_i\) parameters. Default: `rep(0.03, ncol(mY))`.
- `alpha_ini`: a numeric vector \((k \times 1)\) with the initial values of \(\alpha_i\) parameters. Default: `rep(0.03, ncol(mY))`.
- `beta_ini`: a numeric vector \((k \times 1)\) with the initial values of \(\beta_i\) parameters. Default: `rep(0.8, ncol(mY))`.
- `a_ini`: a numeric value of the initial values of \(a\) parameter. Default: 0.03.
- `b_ini`: a numeric value of the initial values of \(b\) parameter. Default: 0.8.
- `gamma_ini`: a numeric vector \((k \times 1)\) with the initial values of \(\gamma_i\) parameters. Default: `rep(1.0, ncol(mY))`.
- `errorDist`: a probability distribution for errors. Use `errorDist=1` for `SSNorm`, `errorDist=2` for `SST` or `errorDist=3` for `SSGED`. Default: 2.
- `control`: list of control arguments (See *Details*).
- `x`: an object of `bayesDccGarch` class.
- `start`: the first iteration of interest from Markov chain.
- `end`: the last iteration of interest from Markov chain.
- `thin`: the required interval between successive samples.
- `...`: additional arguments for S3 generic window function

### Details

The `bayesDccGarch()` function performs a Markov Chain for all parameters of the model DCC-GARCH(1,1) (or GARCH(1,1) in the univariate case). There are three options of probability distributions for the error component. These are the standardized skew versions of normal, t-student and ged distributions. See Fioruci et al (2014a) and Fioruci et al (2014b) for any detail. The `control` argument can be used for define the prior hyper-parameters and the simulation algorithm parameters. It is a list that can supply any of the following components:

- `$mu_tail`: the value of hyper-parameter \(\mu_\nu\) if `errorDist=2` or the hyper-parameter \(\mu_\delta\) if `errorDist=3`. Default: 8
- `$mu_gamma`: a vector with the hyper-parameters \(\mu_{\gamma_i}\). Default: `rep(0, ncol(mY))`
- `$mu_omega`: a vector with the hyper-parameters \(\mu_{\omega_i}\). Default: `rep(0, ncol(mY))`
- `$mu_alpha`: a vector with the hyper-parameters \(\mu_{\alpha_i}\). Default: `rep(0, ncol(mY))`
$\mu_{\beta}$ a vector with the hyper-parameters $\mu_{\beta_i}$. Default: `rep(0, ncol(mY))`

$\mu_a$ the value of the hyper-parameter $\mu_a$. Default: 0

$\mu_b$ the value of the hyper-parameter $\mu_b$. Default: 0

$\sigma_{\text{tail}}$ the value of hyper-parameter $\sigma_{\nu}$ if `errorDist=2` or the hyper-parameter $\sigma_{\delta}$ if `errorDist=3`. Default: 10

$\sigma_{\gamma}$ a vector with the hyper-parameters $\sigma_{\gamma_i}$. Default: `rep(1.25, ncol(mY))`

$\sigma_{\omega}$ a vector with the hyper-parameters $\sigma_{\omega_i}$. Default: `rep(10, ncol(mY))`

$\sigma_{\alpha}$ a vector with the hyper-parameters $\sigma_{\alpha_i}$. Default: `rep(10, ncol(mY))`

$\sigma_{\beta}$ a vector with the hyper-parameters $\sigma_{\beta_i}$. Default: `rep(10, ncol(mY))`

$\sigma_a$ the value of the hyper-parameter $\sigma_a$. Default: 10

$\sigma_b$ the value of the hyper-parameter $\sigma_b$. Default: 10

$\text{simAlg}$ the random walk Metropolis-Hasting algorithm update. Use 1 for update all parameters as one block, use 2 for update one parameter for each time and use 3 for an automatic choice.

$\text{nPilotSim}$ number of simulation for pilot sample if `control$\text{simAlg}=3`. Default: 1000

$\text{cholCov}$ the cholesky decomposition matrix of the covariance matrix for simulation by one-block Metropolis-Hasting. It must to be passed if `control$\text{simAlg}=1`.

$\text{sdSim}$ a vector with the standard deviations for simulation by one-dimensional Metropolis-Hasting. It must to be passed if `control$\text{simAlg}=2`.

$\text{print}$ a logical variable for if the function should report the number of interactions in each 100 interactions or not. Default: TRUE

The function `increaseSim()` can be used to increase the length of Markov chain simulation.

The function `window()` can be used to filter the Markov chain simulation. In this case, all statistics are recomputed.

**Value**

An object of `bayesDccGarch` class, which contains a list with elements:

$\text{control}$ a list with the used `control` argument.

$\text{MC}$ a element of `mcmc` class with the Markov Chain simulation of all parameters. (`R` package `coda`)

$\text{H}$ a matrix with the Bayesian estimates of volatilities and co-volatilities.

$\text{IC}$ the Bayesian estimate of Akaike Information Criterion, Bayesian Information Criterion and Deviance Information Criterion.

$\text{elapsedTime}$ an object of class `proc_time` which is a numeric vector of length 5, containing the user, system, and total elapsed times of the process.

**Author(s)**

Jose Augusto Fiorucci, Ricardo Sandes Ehlers and Francisco Louzada
References


See Also

bayesDccGarch-package, logLikDccGarch, plot, plotVol

Examples

data(DaxCacNik)
Dax = DaxCacNik[,1]

### DCC-GARCH(1,1) ###
out = bayesDccGarch(DaxCacNik[,1], nSim=1000) # more data is necessary
summary(out)
plot(out)

### GARCH(1,1) ###
# out = bayesDccGarch(Dax)
# summary(out)
# plot(out)

### DCC-GARCH(1,1) with SSGED innovations ###
## out = bayesDccGarch(DaxCacNik, tail_ini=1.5, errorDist=3)

### Informative Prior for alpha and beta parameters ###
# out = bayesDccGarch(Dax, alpha_ini=0.01, beta_ini=0.90,
# control=list(mu_alpha=0.01, sigma_alpha=0.001, mu_beta=0.90, sigma_beta=0.01))
# summary(out)
# plot(out$MC) ## plot Markov Chain
# plot(out)

### more simulations ##############################################################
# out1 = bayesDccGarch(DaxCacNik)
# out2 = increaseSim(out, 10000) # more 10000 simulations
# out3 = window(out2, start=5000) # removing the first 5000 as burning
# summary(out3)
#
### Plotting volatilities #
# plot(out3)
#
### Plotting Markov Chain #
# plot(out3$MC)
Description

The matrix DaxCacNik contains daily observations of the hundredfold log-returns of daily indices of stock markets in Frankfurt (DAX), Paris (CAC40) and Tokyo (NIKKEI), from 10 October 1991 until 30 December 1997 (a total of 1627 days). The stock market data is freely available at http://robjhyndman.com/tsdldata/data/FVD1.dat.

Usage

data(DaxCacNik)

Author(s)

Jose Augusto Fioruci, Ricardo Sandes Ehlers and Francisco Louzada

References


densityFunctions

Density functions of multivariate Standard Skew Norm, t-Student and GED distributions

Description

Compute the density function of Standard Skew Normal distribution (SSNORM) or density function of Standard Skew t-Student distribution (SST) or density function of Standard Skew GED distribution (SSGED)

Usage

dssnorm(x, gamma=rep(1,length(x)), log=FALSE)

dsst(x, gamma=rep(1,length(x)), nu=10, log=FALSE)

dssged(x, gamma=rep(1,length(x)), delta=2, log=FALSE)
densityFunctions

Arguments

- **x**: a numeric vector for the point which the density will be computed.
- **gamma**: a numeric vector for skew parameters. Must be positive.
- **nu**: a numeric value of shape parameter of the multivariate Standard Skew t-Student distribution. Must be greater than 2.
- **delta**: a numeric value of shape parameter of GED distribution. Must be positive.
- **log**: logical; if TRUE, densities are returned as \( \log(p) \).

Value

Returns the computed value of the density.

Author(s)

Jose Augusto Fioruci, Ricardo Sandes Ehlers and Francisco Louzada

References


See Also

bayesDccGarch-package

Examples

### Univariate symmetric standard norm distributions ###

dssnorm(x=0)
dsst(x=0, nu=100)
dssged(x=0, delta=2)

### Univariate standard skew norm distributions ###

dssnorm(x=0, gamma=1.5)
dsst(x=0, gamma=1.5, nu=100)
dssged(x=0, gamma=1.5, delta=2)

### Multivariate standard skew norm distributions ###

dssnorm(x=c(0,0), gamma=c(1.5,0.7))
dsst(x=c(0,0), gamma=c(1.5,0.7), nu=100)
dssged(x=c(0,0), gamma=c(1.5,0.7), delta=2)
logLikDccGarch

The logarithm of likelihood function of DCC-GARCH(1,1) Model.

Description
Compute the logarithm of likelihood function of DCC-GARCH(1,1) Model if mY is a matrix or the logarithm of likelihood function of GARCH(1,1) Model if mY is numeric vector.

Usage
logLikDccGarch(mY, omega = rep(0.03, ncol(mY)), alpha = rep(0.03, ncol(mY)),
beta = rep(0.8, ncol(mY)), a = 0.03, b = 0.8, gamma = rep(1, ncol(mY)),
tail = 10, errorDist = 2)

Arguments
- mY: a matrix of the data (n x k).
- omega: a numeric vector (k x 1) with the values of \( \omega_i \) parameters. Default: rep(0.03, ncol(mY)).
- alpha: a numeric vector (k x 1) with the values of \( \alpha_i \) parameters. Default: rep(0.03, ncol(mY)).
- beta: a numeric vector (k x 1) with the values of \( \beta_i \) parameters. Default: rep(0.8, ncol(mY)).
- a: a numeric value of the \( a \) parameter. Default: 0.03.
- b: a numeric value of the \( b \) parameter. Default: 0.8.
- gamma: a numeric vector (k x 1) with the values of \( \gamma_i \) parameters. Default: rep(1.0, ncol(mY)).
- tail: a numeric value of \( \nu \) parameter if errorDist = 2 or of \( \delta \) parameter if errorDist = 3. If errorDist = 1 so this arguments is no used.
- errorDist: a probability distribution for errors. Use errorDist=1 for SSNorm, errorDist=2 for SST or errorDist=3 for SSGED. Default: 2.

Details
The log-likelihood of the model GARCH(1,1) is computed if mY has just one column. The arguments a and b are not consider in this case.

Value
Return a list with the elements:
- $H: a matrix where the lines are the \( H_t \) values for t=1,...,n.
- $value: the value of the logarithm of likelihood function.

Author(s)
Jose Augusto Fioruci, Ricardo Sandes Ehlers and Francisco Louzada
References


See Also

*bayesDccGarch-package, bayesDccGarch*

Examples

data(DaxCacNik)

Dax = DaxCacNik[,1]

#### log-likelihood function of GARCH(1,1) model with SST innovations ####
logLikDccGarch(Dax, omega=0.03, alpha=0.03, beta=0.8, gamma=0.7)$value

#### log-likelihood function of DCC-GARCH(1,1) model with SST innovations ####
logLikDccGarch(DaxCacNik, beta=c(0.82,0.91,0.85), gamma=c(0.7, 1.3, 1.7), tail=10)$value

---

**plot.bayesDccGarch**

*Plotting volatilities for Bayesian DCC-GARCH model*

Description

Produces a plot of time series and the volatilities. This is a particular case of plotVol function.

Usage

```r
## S3 method for class 'bayesDccGarch'
plot(x, ts.names=NULL, colors = c("grey","red"), ...)  
```

Arguments

- **x**
  - Object of class “bayesDccGarch”.
- **ts.names**
  - a vector of length `k` with the names of the time series.
- **colors**
  - a vector with the colors for plotting the returns and volatilities.
- **...**
  - additional arguments for `plot` function

Author(s)

Ricardo Sandes Ehlers, Jose Augusto Fiorucci and Francisco Louzada
References


See Also

bayesDccGarch-package, bayesDccGarch, plotVol

Examples

data(DaxCacNik)

mY = DaxCacNik[1:10,] # more data is necessary

out = bayesDccGarch(mY, nSim=1000)

plot(out)

HPlotting volatilities of time seriesH

Description

Plotting method for volatilities of time series.

Usage

plotVol(mY, vol, ts.names=paste("TS_", 1:ncol(mY), sep=""), colors = c("grey","red"), ...)

Arguments

mY a matrix of the data \( (n \times k) \).

vol a matrix \( (n \times k) \) with the volatility estimates.

ts.names a vector of length \( k \) with the names of the time series.

colors a vector with name of the colors for plotting the returns and volatilities.

... additional arguments for plot function

Author(s)

Ricardo Sandes Ehlers, Jose Augusto Fiorucci and Francisco Louzada
References


See Also

bayesDccGarch-package, bayesDccGarch, plot.bayesDccGarch

Examples

data(DaxCacNik)

mY = DaxCacNik[1:10,] # more data is necessary

out = bayesDccGarch(mY, nSim=1000)

## The code
plotVol(mY, out$H[,c("H_1,1","H_2,2","H_3,3")], c("DAX","CAC40","NIKKEI"))

## gives the result of ##
plot(out)
Index

* Topic Bayes
  bayesDccGarch-package, 2

* Topic DCC-GARCH
  bayesDccGarch-package, 2
  logLikDccGarch, 9

* Topic GARCH
  bayesDccGarch-package, 2
  logLikDccGarch, 9

* Topic datasets
  DaxCalcNik, 7

* Topic dcc-garch
  bayesDccGarch, 3

* Topic density-function
  densityFunctions, 7

* Topic distribution
  densityFunctions, 7

* Topic garch
  bayesDccGarch, 3

* Topic models
  bayesDccGarch, 3

* Topic multivariate-distribution
  densityFunctions, 7

* Topic multivariate
  bayesDccGarch, 3
  bayesDccGarch-package, 2

* Topic package
  bayesDccGarch-package, 2

* Topic plot
  plot.bayesDccGarch, 10
  plotVol, 11

* Topic skew-distribution
  densityFunctions, 7

* Topic ts
  bayesDccGarch, 3

* Topic volatility
  plot.bayesDccGarch, 10
  plotVol, 11

bayesDccGarch, 2, 3, 10–12
bayesDccGarch-package, 2

DaxCalcNik, 7
densityFunctions, 7
dssged, 2
dssged (densityFunctions), 7
dssnorm, 2
dssnorm (densityFunctions), 7
dsst, 2
dsst (densityFunctions), 7
increaseSim (bayesDccGarch), 3
logLikDccGarch, 2, 6, 9
plot, 2, 6
plot (plot.bayesDccGarch), 10
plot.bayesDccGarch, 10, 12
plotVol, 2, 6, 11, 11
window (bayesDccGarch), 3