

# Package ‘gogarch’

March 24, 2011

**Version** 0.7-0

**Type** Package

**Title** Generalized Orthogonal GARCH (GO-GARCH) models

**Date** 2011-03-21

**Author** Bernhard Pfaff

**Maintainer** Bernhard Pfaff <bernhard@pfaffikus.de>

**Depends** R (>= 2.10.0), methods, stats, graphics, fGarch, fastICA

**Suggests** vars

**Description** Implementation of the GO-GARCH model class

**License** GPL (>= 2)

**LazyLoad** yes

**Repository** CRAN

**Repository/R-Forge/Project** gogarch

**Repository/R-Forge/Revision** 42

**Date/Publication** 2011-03-24 16:56:23

## R topics documented:

BVDW	2
BVDWAIR	3
BVDWSTOXX	4
cora	5
goest-methods	6
Goestica-class	7
Goestml-class	8
Goestmm-class	10

Goestnls-class . . . . .	11
gogarch . . . . .	13
GoGARCH-class . . . . .	15
goinit . . . . .	16
Goinit-class . . . . .	17
gollh . . . . .	18
gonls . . . . .	19
Gopredict-class . . . . .	20
Gosum-class . . . . .	21
gotheta . . . . .	21
Orthom-class . . . . .	23
Rd2 . . . . .	24
Umatch . . . . .	24
unvech . . . . .	25
UprodR . . . . .	26
validGoinitObject . . . . .	27
validOrthomObject . . . . .	28
VDW . . . . .	29

<b>Index</b>	<b>30</b>
--------------	-----------

---

BVDW

*Dow Jones Industrial Average and Nasdaq stock indices*

---

## Description

Levels of the Dow Jones Industrial Average and NASDAQ stock indices for the period 03/23/1990 until 03/23/2000.

## Usage

`data(BVDW)`

## Format

A data frame with 2610 observations on the following 3 variables.

`Date` Date in the format YYYYMMDD.

`DJIA` Level of the DIJA.

`NASDAQ` Level of the NASDAQ.

## Details

This data set has been utilized in the source below and was kindly provided by Roy van der Weide.

## Source

Boswijk, H. Peter and van der Weide, Roy (2006), Wake me up before you GO-GARCH, *Tinbergen Institute Discussion Paper, TI 2006-079/4*, University of Amsterdam and Tinbergen Institute.

**References**

<http://www.nasdaq.com>, <http://www.djindexes.com>

**See Also**

[VDW](#)

**Examples**

```
data(BVDW)
str(BVDW)
```

---

BVDWAIR

*Stock prices transportation sector, oil and kerosene prices*

---

**Description**

This data frame contains the stock prices from American Airlines, South-West Airlines, Boeing and FedEx. In addition the spot prices for crude oil and kerosene are included. This data set was used in the article by Boswijk and van der Weide (2009). The data range is from July, 19 1993 until August, 12 2008.

**Usage**

```
data(BVDWAIR)
```

**Format**

A data frame with 3791 observations on the following 7 variables.

Date **POSIXt**: The dates of observations.

CrudeOil **Crude oil price.**

Kerosene **Kerosene price.**

AmericanAir **Stock prices of American Airlines.**

SouthWest **Stock prices of South-West Airlines.**

Boeing **Stock prices of Boeing.**

FedEx **Stock prices of Boeing.**

**Details**

The stock price data was downloaded from Yahoo Finance and the price series for crude oil and kerosene were obtained from the U.S. Energy Information Administration (EIA).

**Source**

<http://finance.yahoo.com> and <http://www.econstats.com>

**References**

Boswijk, H. Peter and van der Weide, Roy (2009), Method of Moments Estimation of GO-GARCH Models, *Working Paper*, University of Amsterdam, Tinbergen Institute and World Bank.

**Examples**

```
data(BVDWAIR)
str(BVDWAIR)
```

---

BVDWSTOXX

*Sector indices of the EURO STOXX 600*

---

**Description**

The data frame contains the following sector indices of the EURO STOXX 600 index: Automobiles \& Parts, Banks, Basic Resources, Chemicals, Construction and Materials, Financial Services, Food \& Beverages, Health Care, Industrial Goods \& Services, Insurance, Media, Oil \& Gas, Technology, Telecommunications and Utilities. The data range is from 31st December 1986 until 21st November 2008.

**Usage**

```
data(BVDWSTOXX)
```

**Format**

A data frame with 5652 observations on the following 16 variables.

Date **POSIXt**: The dates of observations.

AutoParts Sector index Automobiles \& Parts

Banks Sector index Banks

BasicRes Sector index Basic Resources

Chemicals Sector index Chemicals

ConstrMat Sector index Construction and Materials

FoodBeverage Sector index Food \& Beverages

FinService Sector index Financial Services

HealthCare Sector index Health Care

IndustrialGoods Sector index Industrial Goods \& Services

Insurance Sector index Insurance

Media Sector index Media

OilGas Sector index Oil \& Gas

Technology Sector index Technology

Telecom Sector index Telecommunications

Utilities Sector index Utilities

**Source**

<http://www.stoxx.com>

**References**

Boswijk, H. Peter and van der Weide, Roy (2009), Method of Moments Estimation of GO-GARCH Models, *Working Paper*, University of Amsterdam, Tinbergen Institute and World Bank.

**Examples**

```
data(BVDWSTOXX)
str(BVDWSTOXX)
```

---

cora

*Autocorrelations of a Matrix Process*

---

**Description**

This function computes the autocorrelation matrix for a given lag. For instance, it is used for estimating GO-GARCH models whence the method of moments is utilized.

**Usage**

```
cora(SS1, lag = 1, standardize = TRUE)
```

**Arguments**

SS1	Array with dimension $\text{dim} = c(m, m, n)$
lag	Integer, the lag for which the autocorrelation is computed.
standardize	Logical, if TRUE (the default), the autocorrelation matrix is computed, otherwise the autocovariance matrix.

**Details**

This function computes the autocorrelation matrix according to:

$$\hat{\Gamma}_k(s) = \frac{1}{n} \sum_{t=k+1}^n S_t S_{t-k}$$

$$\hat{\Phi}_k(s) = \hat{\Gamma}_0(s)^{-1/2} \hat{\Gamma}_k(s) \hat{\Gamma}_0(s)^{-1/2}$$

It is computationally assured that  $\hat{\Phi}_k(s)$  is symmetric by setting it equal to:  $\hat{\Phi}_k(s) = \frac{1}{2}(\hat{\Phi}_k(s) + \hat{\Phi}_k(s)')$ . The standardization matrix  $\hat{\Gamma}_0(s)^{-1/2}$  is derived from the singular value decomposition of the co-variance matrix at lag zero.

**Value**

cora                    Matrix with dimension  $\dim = c(m, m)$ .

**Author(s)**

Bernhard Pfaff

**References**

Boswijk, H. Peter and van der Weide, Roy (2009), Method of Moments Estimation of GO-GARCH Models, *Working Paper*, University of Amsterdam, Tinbergen Institute and World Bank.

**See Also**

[gogarch](#)

---

goest-methods                    *Methods for Function goest*

---

**Description**

These are methods for estimating GO-GARCH models. Currently only a method for estimating GO-GARCH models by Maximum-Likelihood is implemented.

**Details**

The declared estimation methods are called from function `gogarch`.

**Methods**

```
goest signature(object = "Goestica")
```

```
goest signature(object = "Goestmm")
```

```
goest signature(object = "Goestml")
```

```
goest signature(object = "Goestnls")
```

**Author(s)**

Bernhard Pfaff

**See Also**

[garchFit](#), [Goestica](#), [Goestml](#), [Goestnls](#), [Goestmm](#), [gogarch](#)

---

Goestica-class      *Class "Goestica": GO-GARCH models estimated by fast ICA*

---

### Description

This class contains the `GoGARCH` class and has the mixing matrix  $A$  as additional slot.

### Objects from the Class

Objects can be created by calls of the form `new("Goestmm", ...)`, or with the function `gogarch` whereby `method = "ica"` has been set.

### Slots

`ica`: Object of class "list": List object returned by `fastICA`.  
`Z`: Object of class "matrix": Transformation matrix.  
`U`: Object of class "matrix": Orthogonal matrix.  
`Y`: Object of class "matrix": Extracted component matrix.  
`H`: Object of class "list": List of conditional variance/covariance matrices.  
`models`: Object of class "list": List of univariate GARCH model fits.  
`estby`: Object of class "character": Estimation method.  
`X`: Object of class "matrix": The data matrix.  
`V`: Object of class "matrix": Covariance matrix of  $X$ .  
`P`: Object of class "matrix": Left singular values of Var/Cov matrix of  $X$ .  
`Dsqr`: Object of class "matrix": Square roots of eigenvalues on diagonal, else zero.  
`garchf`: Object of class "formula": Garch formula used for uncorrelated component GARCH models.  
`name`: Object of class "character": The name of the original data object.

### Extends

Class "`GoGARCH`", directly. Class "`Goinit`", by class "`GoGARCH`", distance 2.

### Methods

**`evar`** Returns the conditional variances as object with class attribute "mts" "ts".  
**`ccov`** Returns the conditional co-variances as object with class attribute "mts" "ts".  
**`ccor`** Returns the conditional correlations as object with class attribute "mts" "ts".  
**`coef`** Returns the coefficients of the component GARCH models.  
**`converged`** Returns the convergence codes of the component GARCH models.  
**`formula`** Returns the formula for the component GARCH models.  
**`goest`** Fast ICA estimation of Go-GARCH models.

**plot** Plotting of the conditional correlations.

**predict** Returns the conditional covariances and mean forecasts and the forecasts of the component GARCH models, object is of class `Gopredict`.

**residuals** Returns the residuals of the Go-GARCH model as object with class attribute `"mts"` `"ts"`.

**resid** Returns the residuals of the Go-GARCH model as object with class attribute `"mts"` `"ts"`.

**show** show-method for objects of class `Goestmm`.

**summary** summary-method for objects of class `Goestml`, object is of class `Gosum`.

**update** Updates an object of class `Goestml`.

### Author(s)

Bernhard Pfaff

### References

Broda, S.A. and Paoletta, M.S. (2008): CHICAGO: A Fast and Accurate Method for Portfolio Risk Calculation, *Swiss Finance Institute*, Research Paper Series No. 08-08, Zuerich.

### See Also

[GoGARCH](#), [Goinit](#), [Gosum](#), [Gopredict](#), [goest-methods](#) and [gogarch](#)

---

Goestml-class	<i>Class "Goestml": GO-GARCH models estimated by Maximum-Likelihood</i>
---------------	---

---

### Description

This class contains the `GoGARCH` class and has the outcome of `nlminb` as an additional slot.

### Objects from the Class

Objects can be created by calls of the form `new("Goestml", ...)`, or with the function `gogarch` whereby `method = "ml"` has been set.

### Slots

`opt`: Object of class `"list"`: List returned by `nlminb`.

`Z`: Object of class `"matrix"`: Transformation matrix.

`U`: Object of class `"matrix"`: Orthogonal matrix.

`Y`: Object of class `"matrix"`: Extracted component matrix.

`H`: Object of class `"list"`: List of conditional variance/covariance matrices.

`models`: Object of class `"list"`: List of univariate GARCH model fits.

**estby:** Object of class "character": Estimation method.

**X:** Object of class "matrix": The data matrix.

**V:** Object of class "matrix": Covariance matrix of X.

**P:** Object of class "matrix": Left singular values of Var/Cov matrix of X.

**Dsqr:** Object of class "matrix": Square roots of eigenvalues on diagonal, else zero.

**garchf:** Object of class "formula": Garch formula used for uncorrelated component GARCH models.

**name:** Object of class "character": The name of the original data object.

### Extends

Class "[GoGARCH](#)", directly. Class "[Goinit](#)", by class "GoGARCH", distance 2.

### Methods

**angles** Returns the Eulerian angles.

**cvar** Returns the conditional variances as object with class attribute "mts" "ts".

**ccov** Returns the conditional co-variances as object with class attribute "mts" "ts".

**ccor** Returns the conditional correlations as object with class attribute "mts" "ts".

**coef** Returns the coefficients of the component GARCH models.

**converged** Returns the convergence codes of the component GARCH models.

**formula** Returns the formula for the component GARCH models.

**goest** ML-Estimation of Go-GARCH models.

**logLik** Returns the value of the log-Likelihood function.

**plot** Plotting of the conditional correlations.

**predict** Returns the conditional covariances and mean forecasts and the forecasts of the component GARCH models, object is of class `Gopredict`.

**residuals** Returns the residuals of the Go-GARCH model as object with class attribute "mts" "ts".

**resid** Returns the residuals of the Go-GARCH model as object with class attribute "mts" "ts".

**show** show-method for objects of class `Goestml`.

**summary** summary-method for objects of class `Goestml`, object is of class `Gosum`.

**update** Updates an object of class `Goestml`.

### Author(s)

Bernhard Pfaff

### See Also

[GoGARCH](#), [Goinit](#), [Gosum](#), [Gopredict](#), [goest-methods](#)

---

Goestmm-class      *Class "Goestmm": Go-GARCH models estimated by Methods of Moments*

---

### Description

This class contains the `GoGARCH` class and has the weights vector and the matched orthogonal matrices  $U$  as additional slots.

### Objects from the Class

Objects can be created by calls of the form `new("Goestmm", ...)`, or with the function `gogarch` whereby `method = "mm"` has been set.

### Slots

**weights:** Object of class "numeric": Weights for aggregating the matched orthogonal matrices  $U$ .

**Umatched:** Object of class "list": List of matched orthogonal matrices  $U$ .

**Z:** Object of class "matrix": Transformation matrix.

**U:** Object of class "matrix": Orthogonal matrix.

**Y:** Object of class "matrix": Extracted component matrix.

**H:** Object of class "list": List of conditional variance/covariance matrices.

**models:** Object of class "list": List of univariate GARCH model fits.

**estby:** Object of class "character": Estimation method.

**X:** Object of class "matrix": The data matrix.

**V:** Object of class "matrix": Covariance matrix of  $X$ .

**P:** Object of class "matrix": Left singular values of Var/Cov matrix of  $X$ .

**Dsqr:** Object of class "matrix": Square roots of eigenvalues on diagonal, else zero.

**garchf:** Object of class "formula": Garch formula used for uncorrelated component GARCH models.

**name:** Object of class "character": The name of the original data object.

### Extends

Class "`GoGARCH`", directly. Class "`Goinit`", by class "`GoGARCH`", distance 2.

### Methods

**cvar** Returns the conditional variances as object with class attribute "mts" "ts".

**ccov** Returns the conditional co-variances as object with class attribute "mts" "ts".

**ccor** Returns the conditional correlations as object with class attribute "mts" "ts".

**coef** Returns the coefficients of the component GARCH models.

- converged** Returns the convergence codes of the component GARCH models.
- formula** Returns the formula for the component GARCH models.
- goest** Methods of moments estimation of Go-GARCH models.
- plot** Plotting of the conditional correlations.
- predict** Returns the conditional covariances and mean forecasts and the forecasts of the component GARCH models, object is of class `Gopredict`.
- residuals** Returns the residuals of the Go-GARCH model as object with class attribute `"mts"` `"ts"`.
- resid** Returns the residuals of the Go-GARCH model as object with class attribute `"mts"` `"ts"`.
- show** show-method for objects of class `Goestmm`.
- summary** summary-method for objects of class `Goestml`, object is of class `Gosum`.
- update** Updates an object of class `Goestml`.

**Author(s)**

Bernhard Pfaff

**References**

Boswijk, H. Peter and van der Weide, Roy (2009), Method of Moments Estimation of GO-GARCH Models, *Working Paper*, University of Amsterdam, Tinbergen Institute and World Bank.

**See Also**

[GoGARCH](#), [Goinit](#), [Gosum](#), [Gopredict](#), [goest-methods](#), [gogarch](#), [Umatch](#)

---

Goestnls-class	<i>Class "Goestnls": GO-GARCH models estimated by Non-linear Least-Squares</i>
----------------	--

---

**Description**

This class contains the `GoGARCH` class and has the outcome of `optim` as an additional slot.

**Objects from the Class**

Objects can be created by calls of the form `new("Goestnls", ...)`, or with the function `gogarch` whereby `method = "nls"` has been set.

**Slots**

**nls**: Object of class "list": List returned by `optim`.  
**Z**: Object of class "matrix": Transformation matrix.  
**U**: Object of class "matrix": Orthogonal matrix.  
**Y**: Object of class "matrix": Extracted component matrix.  
**H**: Object of class "list": List of conditional variance/covariance matrices.  
**models**: Object of class "list": List of univariate GARCH model fits.  
**estby**: Object of class "character": Estimation method.  
**X**: Object of class "matrix": The data matrix.  
**V**: Object of class "matrix": Covariance matrix of  $X$ .  
**P**: Object of class "matrix": Left singular values of Var/Cov matrix of  $X$ .  
**Dsqr**: Object of class "matrix": Square roots of eigenvalues on diagonal, else zero.  
**garchf**: Object of class "formula": Garch formula used for uncorrelated component GARCH models.  
**name**: Object of class "character": The name of the original data object.

**Extends**

Class `GoGARCH`, directly. Class `Goinit`, by class `GoGARCH`, distance 2.

**Methods**

**cvar** Returns the conditional variances as object with class attribute "mts" "ts".  
**ccov** Returns the conditional co-variances as object with class attribute "mts" "ts".  
**ccor** Returns the conditional correlations as object with class attribute "mts" "ts".  
**coef** Returns the coefficients of the component GARCH models.  
**converged** Returns the convergence codes of the component GARCH models.  
**formula** Returns the formula for the component GARCH models.  
**goest** NLS-Estimation of Go-GARCH models.  
**plot** Plotting of the conditional correlations.  
**predict** Returns the conditional covariances and mean forecasts and the forecasts of the component GARCH models, object is of class `Gopredict`.  
**residuals** Returns the residuals of the Go-GARCH model as object with class attribute "mts" "ts".  
**resid** Returns the residuals of the Go-GARCH model as object with class attribute "mts" "ts".  
**show** show-method for objects of class `Goestnls`.  
**summary** summary-method for objects of class `GoGARCH`, object is of class `Gosum`.  
**update** Updates an object of class `GoGARCH`.

**Author(s)**

Bernhard Pfaff

**See Also**

[GoGARCH](#), [Goinit](#), [Gosum](#), [Gopredict](#), [goest-methods](#), [gogarch](#)

gogarch

*Specification and estimation of GO-GARCH models***Description**

This function steers the specification and estimation of GO-GARCH models.

**Usage**

```
gogarch(data, formula, scale = FALSE, estby = c("ica", "mm", "ml", "nls"),
        lag.max = 1, initial = NULL, garchlist = list(init.rec = "mci", delta
        = 2, skew = 1, shape = 4, cond.dist = "norm", include.mean = FALSE,
        include.delta = NULL, include.skew = NULL, include.shape = NULL,
        leverage = NULL, trace = FALSE, algorithm = "nlminb", hessian =
        "ropt", control = list(), title = NULL, description = NULL), ...)
```

**Arguments**

<code>data</code>	Matrix: the original data set.
<code>formula</code>	Formula: valid formula for univariate GARCH models.
<code>scale</code>	Logical, if TRUE the data is scaled. The default is <code>scale = FALSE</code> .
<code>estby</code>	Character: by fast ICA <code>estby = "ica"</code> (the default), by Estbys of Moments <code>estby = "mm"</code> or by Maximum-Likelihood <code>estby = "ml"</code> or by non-linear Least-Squares <code>estby = "nls"</code> .
<code>initial</code>	Numeric: starting values for optimization (used if <code>estby = "ml"</code> or <code>estby = "nls"</code> has been chosen (see Details).
<code>lag.max</code>	Integer: The number of used lags for computing the matched orthogonal matrices $U$ (used if <code>estby = "mm"</code> has been chosen).
<code>garchlist</code>	List: Elements are passed to <code>garchFit</code> .
<code>...</code>	Ellipsis argument: is passed to the <code>goest</code> method (see details).

**Details**

The ellipsis argument is passed to the function `fastICA` if `estby = "ica"` has been set, or to `optim` if `estby = "nls"` is employed or to `nlminb` if the GO-GARCH model is estimated by maximum likelihood, *i.e.*, `estby = "ml"`. It is not employed if the methods of moments estimator is chosen.

If the argument `initial` is left `NULL`, the starting values are computed according `seq(3.0, 0.1, length.out = l)`, whereby `l` is the length of `initial` for `estby = "ml"` and are set to `rep(0.1, d)`, whereby for method = "nls". This length must be equal to  $m*(m-1)/2$  for estimation by Maximum-Likelihood and  $m * (m + 1)/2$  for estimation by non-linear least-Squares, whereby  $m$  is the number of columns of data.

**Value**

Dependent on the chosen estimation method either an object of class `Goestica` or `Goestmm` or `Goestml` or `Goestnls` is returned. All of these classes extend the `GoGARCH` class.

**Author(s)**

Bernhard Pfaff

**References**

Van der Weide, Roy (2002), GO-GARCH: A Multivariate Generalized Orthogonal GARCH Model, *Journal of Applied Econometrics*, **17(5)**, 549 – 564.

Boswijk, H. Peter and van der Weide, Roy (2006), Wake me up before you GO-GARCH, *Tinbergen Institute Discussion Paper*, **TI 2006-079/4**, University of Amsterdam and Tinbergen Institute.

Boswijk, H. Peter and van der Weide, Roy (2009), Method of Moments Estimation of GO-GARCH Models, *Working Paper*, University of Amsterdam, Tinbergen Institute and World Bank.

Broda, S.A. and Paolella, M.S. (2008): CHICAGO: A Fast and Accurate Method for Portfolio Risk Calculation, *Swiss Finance Institute*, Research Paper Series No. 08-08, Zuerich.

**See Also**

[GoGARCH](#), [Goestica](#), [Goestmm](#), [Goestnls](#), [Goestml](#), [goest-methods](#)

**Examples**

```
## Not run:
library(vars)
## Boswijk / van der Weide (2009)
data(BVDWSTOXX)
BVDWSTOXX <- zoo(x = BVDWSTOXX[, -1], order.by = BVDWSTOXX[, 1])
BVDWSTOXX <- window(BVDWSTOXX, end = as.POSIXct("2007-12-31"))
BVDWSTOXX <- diff(log(BVDWSTOXX))
sectors <- BVDWSTOXX[, c("AutoParts", "Banks", "OilGas")]
sectors <- apply(sectors, 2, scale, scale = FALSE)
gogmm <- gogarch(sectors, formula = ~garch(1,1), estby = "mm",
  lag.max = 100)
gogmm
## Boswijk / van der Weide (2006)
data(BVDW)
BVDW <- zoo(x = BVDW[, -1], order.by = BVDW[, 1])
BVDW <- diff(log(BVDW)) * 100
gognls <- gogarch(BVDW, formula = ~garch(1,1), scale = TRUE,
  estby = "nls")
gognls
## van der Weide (2002)
data(VDW)
var1 <- VAR(scale(VDW), p = 1, type = "const")
resid <- residuals(var1)
gogml <- gogarch(resid, ~garch(1, 1), scale = TRUE,
  estby = "ml", control = list(iter.max = 1000))
```

```

gogml
solve (gogml@Z)

## End(Not run)

```

---

GoGARCH-class

---

Class "GoGARCH": Estimated GO-GARCH Models

---

## Description

This class defines the slots for estimated GO-GARCH models. It contains the class `Goinit`.

## Objects from the Class

Objects can be created by calls of the form `new("GoGARCH", ...)`.

## Slots

**Z:** Object of class "matrix": Transformation matrix.

**U:** Object of class "Orthom": Orthonormal matrix.

**Y:** Object of class "matrix": Extracted component matrix.

**H:** Object of class "list": List of conditional variance/covariance matrices.

**models:** Object of class "list": List of univariate GARCH model fits.

**estby:** Object of class "character": Estimation method.

**CALL:** Object of class "call": Result of `match.call` in generating function.

**X:** Object of class "matrix": The data matrix.

**V:** Object of class "matrix": Covariance matrix of X.

**P:** Object of class "matrix": Left singular values of Var/Cov matrix of X.

**DsqR:** Object of class "matrix": Square roots of eigenvalues on diagonal, else zero.

**garchf:** Object of class "formula": Garch formula used for uncorrelated component GARCH models.

**name:** Object of class "character": The name of the original data object.

## Extends

Class "`Goinit`", directly.

**Methods**

- cvar** Returns the conditional variances as object with class attribute "mts" "ts".
- ccov** Returns the conditional co-variances as object with class attribute "mts" "ts".
- ccor** Returns the conditional correlations as object with class attribute "mts" "ts".
- coef** Returns the coefficients of the component GARCH models.
- converged** Returns the convergence codes of the component GARCH models.
- formula** Returns the formula for the component GARCH models.
- plot** Plotting of the conditional correlations.
- predict** Returns the conditional covariances and mean forecasts and the forecasts of the component GARCH models, object is of class Gopredict.
- residuals** Returns the residuals of the GO-GARCH model.
- show** show-method for objects of class GoGARCH.
- summary** summary-method for objects of class GoGARCH, object is of class Gosum.
- update** Updates an object of class GoGARCH.

**Author(s)**

Bernhard Pfaff

**See Also**

[Goinit](#), [Gosum](#), [Gopredict](#)

---

goinit

*Constructor function for objects of class "Goinit"*

---

**Description**

This function can be utilized to create objects of class Goinit. These objects are the starting point for estimating GO-GARCH models.

**Usage**

```
goinit(X, garchf = ~garch(1, 1), scale = FALSE)
```

**Arguments**

- |        |  |
|--------|--|
| X      | Matrix: the data matrix.   |
| garchf | Formula: A formula object that will be used in the GARCH models of the uncorrelated components.    |
| scale  | Logical, if TRUE the data X will be scaled, the default value is FALSE for no scaling of the data. |

**Details**

This function computes the variance/covariance matrix of  $X$ . Next the singular value decomposition is applied and the projection matrix as well as the diagonal matrix with the square roots of the eigen values are computed.

**Value**

An object of class `Goinit`.

**Author(s)**

Bernhard Pfaff

**See Also**

[Goinit](#)

**Examples**

```
## Not run:
library(vars)
data(VDW)
var1 <- VAR(VDW, p = 1, type = "const")
resid <- resid(var1)
goinit(resid, scale = TRUE)

## End(Not run)
```

---

`Goinit-class`

*Class "Goinit": Initialisation of GO-GARCH models*

---

**Description**

This class defines the required slots for estimating GO-GARCH models.

**Objects from the Class**

Objects can be created by calls of the form `new("Goinit", ...)`, or more conveniently by `goinit()`.

**Slots**

**X:** Object of class `"matrix"`: The data matrix.

**V:** Object of class `"matrix"`: Covariance matrix of  $X$ .

**P:** Object of class `"matrix"`: Left singular values of Var/Cov matrix of  $X$ .

**Dsqr:** Object of class `"matrix"`: Square roots of eigenvalues on diagonal, else zero.

**garchf:** Object of class `"formula"`: Garch formula used for uncorrelated component GARCH models.

**name:** Object of class `"character"`: The name of the original data object.

**Methods**

`show` Prints the slots, whereby for X only the head is displayed.

**Author(s)**

Bernhard Pfaff

**See Also**

`garchFit`, `goinit`

**Examples**

```
showClass("Goinit")
```

---

gollh

*Log-Likelihood function of GO-GARCH models*

---

**Description**

This function returns the negative of the log-Likelihood function for GO-GARCH models.

**Usage**

```
gollh(params, object, garchlist)
```

**Arguments**

`params`            Vector of initial values for `theta`.  
`object`            An object of class `Goinit` or an extension thereof.  
`garchlist`        List, elements are passed to `garchFit`.

**Details**

The log-Likelihood function of GO-GARCH models is given as:

$$L_{\theta, \alpha, \beta} = -\frac{1}{2} \sum_{t=1}^T m \log(2\pi) + \log |Z_{\theta} Z'_{\theta}| + \log |H_t| + y' H_t^{-1} y_t$$

whereby  $Z = P\Delta^{\frac{1}{2}}U_0$ ,  $y_t = Z^{-1}x_t$  and  $H_t$  is the conditional variance matrix of the independent components.

**Value**

`negll`            Scalar, the negative value of the log-Likelihood function.

**Author(s)**

Bernhard Pfaff

**References**

Van der Weide, Roy (2002), GO-GARCH: A Multivariate Generalized Orthogonal GARCH Model, *Journal of Applied Econometrics*, **17(5)**, 549 – 564.

**See Also**[garchFit](#)

gonls

*Non-linear least-squares estimation of matrix B***Description**

This is the target function for estimating the matrix  $B$  by non-linear least-squares. It is used in the estimation method `goest` if `method = "nls"` is chosen.

**Usage**`gonls(params, SSI)`**Arguments**

<code>params</code>	The initial values of the $vech(B)$ .
<code>SSI</code>	A list with two elements, each a list itself, containing $S_t = s_t s_t' - I_m$ and $S_{t-1} = s_{t-1} s_{t-1}' - I_m$ , respectively.

**Details**

Boswijk and van der Weiden (2006) proposed the following criterion function:

$$S(A) = \frac{1}{n} \sum_{t=1}^n tr([s_t s_t' - I_m - B(s_{t-1} s_{t-1}' - I_m)B]^2) = S^*(B)$$

for retrieving the matrix  $U$ . This matrix is the eigen vector matrix of  $B$ . The linear map  $Z = P\Delta^{1/2}U$  and its inverse can then be computed for calculating the component matrix  $Y = XZ^{-1}$ .

**Value**

<code>f</code>	<code>numeric</code> : The value of the target function.
----------------	--

**Author(s)**

Bernhard Pfaff

## References

Boswijk, H. Peter and van der Weide, Roy (2006), Wake me up before you GO-GARCH, *Tinbergen Institute Discussion Paper, TI 2006-079/4*, University of Amsterdam and Tinbergen Institute.

## See Also

[gogarch](#)

---

Gopredict-class      *Class "Gopredict": Prediction of GO-GARCH Models*

---

## Description

This class defines the slots for forecasts from a GO-GARCH model.

## Objects from the Class

Objects can be created by calls of the form `new("Gopredict", ...)`, or with the method `predict` of formal class objects `GoGARCH` and `Goestml`.

## Slots

**Hf:** Object of class "list": The forecasted conditional covariances.

**Xf:** Object of class "matrix": The transformed forecasts of the component GARCH mean models.

**CGARCHF:** Object of class "list": The original forecasts of the component GARCH models.

## Methods

**ccor** Returns the forecasted conditional correlations.

**ccov** Returns the forecasted conditional co-variances.

**cvar** Returns the forecasted conditional variances.

**show** show-method for objects of class `Gopredict`.

## Note

In case more than 10 forecasts steps are computed, the `show`-method displays only the `head` of the returned objects. Furthermore, the `show`-method displays the forecasted conditional variances only. The forecasted conditional co-variances and/or the forecasted conditional correlations can be retrieved with the methods `ccov` or `ccor`, respectively.

## Author(s)

Bernhard Pfaff

## See Also

[GoGARCH](#), [Goestml](#)

---

Gosum-class

---

Class "Gosum": Summary object of GO-GARCH model

---

**Description**

The formal summary class of `GoGARCH` objects or objects that extend this class.

**Objects from the Class**

Objects can be created by calls of the form `new("Gosum", ...)` or are set by the summary-method.

**Slots**

`name`: character: the name of the original data object.

`method`: character: the estimation method.

`model`: formula: The GARCH model formula for the component GARCH models.

`garchc`: list: The elements are `matcoef` matrices generated by `garchFit` for the components.

`Zinv`: matrix: The inverse of the linear map  $X = YZ$ .

**Methods**

`show` show-method for objects of class `Gosum`.

**Author(s)**

Bernhard Pfaff

**See Also**

[GoGARCH](#), [Goestml](#)

---

gotheta

---

Creates an object of class `GoGARCH` based on Euler angles

---

**Description**

This function returns an object of class `GoGARCH` based on an input vector of Euler angles.

**Usage**

```
gotheta(theta, object, garchlist = list(init.rec = "mci", delta = 2,
skew = 1, shape = 4, cond.dist = "norm", include.mean = FALSE,
include.delta = NULL, include.skew = NULL, include.shape = NULL,
leverage = NULL, trace = FALSE, algorithm = "nlminb", hessian = "ropt",
control = list(), title = NULL, description = NULL))
```

**Arguments**

<code>theta</code>	Vector of Euler angles.
<code>object</code>	An object of formal class <code>Goinit</code> or an extension thereof.
<code>garchlist</code>	List with optional elements passed to <code>garchFit</code> .

**Details**

In a first step the orthogonal matrix  $U$  is computed as the product of rotation matrices given the vector `theta` of Euler angles with the function `UprodR`. The linear map  $Z$  is computed next as  $Z = PD^{\frac{1}{2}}U'$ . The unobserved components  $Y$  are calculated as  $Y = XZ^{-1}$ . These are then utilized in the estimation of the univariate GARCH models according to `object@garchf`. The conditional variance/covariance matrices are calculated according to  $V_t = ZH_tZ'$  whereby  $H_t$  signifies a matrix with the conditional variances of the univariate GARCH models on its diagonal.

**Value**

Returns an object of class `GoGARCH`.

**Author(s)**

Bernhard Pfaff

**References**

Van der Weide, Roy (2002), GO-GARCH: A Multivariate Generalized Orthogonal GARCH Model, *Journal of Applied Econometrics*, **17(5)**, 549 – 564.

**See Also**

[Goinit](#), [GoGARCH](#), [Goestml](#), [garchFit](#)

**Examples**

```
## Not run:
library(vars)
data(VDW)
var1 <- VAR(VDW, p = 1, type = "const")
resid <- resid(var1)
gin <- goinit(resid, scale = TRUE)
gotheta(0.5, gin)

## End(Not run)
```

---

Orthom-class      *Class "Orthom": Orthogonal matrices*

---

### Description

This class defines an orthogonal matrix, which is characterized by  $\det(M) = 1$  and  $MM' = I$ .

### Objects from the Class

Objects can be created by calls of the form `new("Orthom", ...)`. In addition the function `UprodR` returns an object of formal class `Orthom`.

### Slots

**M**: Object of class "matrix".

### Methods

**M** Returns the slot `M` of class `Orthom`.

**print** print-method for objects of class `Orthom`.

**show** show-method for objects of class `Orthom`.

**t** Transpose of `object@M`.

### Note

Objects are validated by `validOrthomObject()`. This function is utilised by `validObject()`.

### Author(s)

Bernhard Pfaff

### See Also

`UprodR`, `validOrthomObject`

### Examples

```
showClass("Orthom")
```

Rd2

*Rotation matrix, 2-dimensional*

---

**Description**

Given an angle  $\theta$  whereby  $\theta \in [0, \pi/2)$  the function `Rd2` returns a 2-dimensional rotation matrix of Euler angles.

**Usage**`Rd2(theta)`**Arguments**

`theta` Numeric, angle in the interval  $[0, \pi/2)$ .

**Value**

R A 2-dimensional rotation matrix.

**Author(s)**

Bernhard Pfaff

**See Also**

[UprodR](#)

**Examples**`Rd2(pi/3)`

---

Umatch

*Matching of Orthogonal Matrices for Cayley transforms*

---

**Description**

This function matches an orthogonal matrix to the importance of the columns of the matrix to which it should be matched.

**Usage**`Umatch(from, to)`

**Arguments**

from	Matrix: orthogonal
to	Matrix: orthogonal

**Value**

mat	Matched matrix.
-----	-----------------

**Author(s)**

Bernhard Pfaff

**References**

Boswijk, H. Peter and van der Weide, Roy (2009), Method of Moments Estimation of GO-GARCH Models, *Working Paper*, University of Amsterdam, Tinbergen Institute and World Bank.

Liebeck, H. and Osborne, A. (1991), The Generation of All Rational Orthogonal Matrices, *The American Mathematical Monthly*, **98 (2)** (Feb. 1991), 131 – 133.

**See Also**

[gogarch](#)

---

unvech

*Returns a symmetric matrix from a vector*

---

**Description**

This function returns the symmetric matrix  $X$  from a vector that resulted from  $v = \text{vech}(X)$ .

**Usage**

`unvech(v)`

**Arguments**

<code>v</code>	Vector, numeric.
----------------	------------------

**Details**

The vector `v` must have length equal to  $m * (m + 1) / 2$ , whereby  $m$  is a dimension of the symmetric matrix  $X_{m \times m}$ .

**Value**

<code>x</code>	Matrix, symmetric of order $m \times m$ .
----------------	---

**Author(s)**

Bernhard Pfaff

**See Also**[vec](#)**Examples**

```
v <- c(1, 2, 3, 4, 5, 6)
unvech(v)
```

---

`UprodR`*Creation of an orthogonal matrix*

---

**Description**

This function returns an orthogonal matrix which results of the matrix products of rotation matrices.

**Usage**

```
UprodR(theta)
```

**Arguments**

`theta`            Vector, of angles of the rotation matrices.

**Details**

The length of `theta` must be equal to  $m * (m - 1) / 2$ , where  $m$  is the dimension of the orthogonal matrix. The elements of `theta` must lie in the interval  $[0, \pi/2)$ .

**Value**

`result`            Object of class `Orthom`.

**Author(s)**

Bernhard Pfaff

**References**

Vilenkin, N. Ja. (1968), Special Functions and the Theory of Group Representations, Translations of Mathematical Monographs, **22**, American Math. Soc., Providence, Rhode Island, USA.

**See Also**[Rd2](#), [Orthom](#)

**Examples**

```
theta <- c(pi/3, pi/5, pi/7)
U <- UprodR(theta)
U
```

---

validGoinitObject *Validation function for objects of class Goinit*

---

**Description**

This function validates objects of class Goinit.

**Usage**

```
validGoinitObject(object)
```

**Arguments**

object            Object of class Goinit.

**Details**

This function is utilized by `validObject()`. It is tested whether `object@V`, `object@P`, `object@Dsqr` are square matrices; `object@V` coincides with the singular value decomposition.

**Value**

TRUE            Logical, TRUE if the object passes the validation, otherwise an informative error message is returned.

**Author(s)**

Bernhard Pfaff

**See Also**

[Goinit](#), [goinit](#)

**Examples**

```
data(VDW)
go <- goinit(VDW)
validObject(go)
```

---

validOrthomObject    *Validation function for objects of class Orthom*

---

### Description

This function validates objects of class Orthom.

### Usage

```
validOrthomObject(object)
```

### Arguments

object            Object of class Orthom.

### Details

This function is utilized by `validObject()`. It is tested whether `object@M` is a square matrix, has  $\det(M) = 1$  and  $MM' = I$ .

### Value

TRUE            Logical, TRUE if the object passes the validation, otherwise an informative error message is returned.

### Author(s)

Bernhard Pfaff

### See Also

[Orthom](#)

### Examples

```
theta <- c(pi/3, pi/5, pi/7)
U <- UprodR(theta)
validObject(U)
```

---

VDW

*Dow Jones Industrial Average and Nasdaq stock indices*

---

### Description

The daily (log) returns of the Dow Jones Industrial Average and the NASDAQ composite, respectively. The daily observations start at the first of January, 1990, and end in October 2001.

### Usage

```
data (VDW)
```

### Format

A data frame with 3082 observations on the following 2 variables.

DJIA Log-return of Dow Jones Industrial Average.

NASDAQ Log-return of NASDAQ.

### Details

This data set has been utilized in the source below and can be downloaded from the web-site of the *Journal of Applied Econometrics* (see link below).

### Source

Van der Weide, Roy (2002), GO-GARCH: A Multivariate Generalized Orthogonal GARCH Model, *Journal of Applied Econometrics*, **17(5)**, 549 – 564.

### References

<http://www.nasdaq.com>

<http://www.djindexes.com>

[http://qed.econ.queensu.ca/jae/2002-v17.5/van\\_der\\_weide](http://qed.econ.queensu.ca/jae/2002-v17.5/van_der_weide)

### See Also

[BVDW](#)

### Examples

```
data (VDW)
```

```
str (VDW)
```

# Index

## \*Topic **algebra**

Orthom-class, 23  
Rd2, 24  
unvech, 25  
UprodR, 26

## \*Topic **classes**

Goestica-class, 7  
Goestml-class, 8  
Goestmm-class, 10  
Goestnls-class, 11  
GoGARCH-class, 15  
Goinit-class, 17  
Gopredict-class, 20  
Gosum-class, 21  
Orthom-class, 23

## \*Topic **datasets**

BVDW, 2  
BVDWAIR, 3  
BVDWSTOXX, 4  
VDW, 29

## \*Topic **methods**

goest-methods, 6

## \*Topic **models**

cora, 5  
gogarch, 13  
goinit, 16  
gollh, 18  
gonls, 19  
gotheta, 21  
Umatch, 24

## \*Topic **utilities**

validGoinitObject, 27  
validOrthomObject, 28

angles (*Goestml-class*), 8  
angles, *Goestml-method*  
(*Goestml-class*), 8

BVDW, 2, 29  
BVDWAIR, 3

BVDWSTOXX, 4

ccor (*Goestml-class*), 8  
ccor, *Goestica-method*  
(*Goestica-class*), 7  
ccor, *Goestml-method*  
(*Goestml-class*), 8  
ccor, *Goestmm-method*  
(*Goestmm-class*), 10  
ccor, *Goestnls-method*  
(*Goestnls-class*), 11  
ccor, *GoGARCH-method*  
(*GoGARCH-class*), 15  
ccor, *Gopredict-method*  
(*Gopredict-class*), 20  
ccov (*Goestml-class*), 8  
ccov, *Goestica-method*  
(*Goestica-class*), 7  
ccov, *Goestml-method*  
(*Goestml-class*), 8  
ccov, *Goestmm-method*  
(*Goestmm-class*), 10  
ccov, *Goestnls-method*  
(*Goestnls-class*), 11  
ccov, *GoGARCH-method*  
(*GoGARCH-class*), 15  
ccov, *Gopredict-method*  
(*Gopredict-class*), 20  
coef, *Goestica-method*  
(*Goestica-class*), 7  
coef, *Goestml-method*  
(*Goestml-class*), 8  
coef, *Goestmm-method*  
(*Goestmm-class*), 10  
coef, *Goestnls-method*  
(*Goestnls-class*), 11  
coef, *GoGARCH-method*  
(*GoGARCH-class*), 15  
converged (*Goestml-class*), 8

- converged, Goestica-method  
(*Goestica-class*), 7
- converged, Goestml-method  
(*Goestml-class*), 8
- converged, Goestmm-method  
(*Goestmm-class*), 10
- converged, Goestnls-method  
(*Goestnls-class*), 11
- converged, GoGARCH-method  
(*GoGARCH-class*), 15
- cora, 5
- cvar (*Goestml-class*), 8
- cvar, Goestica-method  
(*Goestica-class*), 7
- cvar, Goestml-method  
(*Goestml-class*), 8
- cvar, Goestmm-method  
(*Goestmm-class*), 10
- cvar, Goestnls-method  
(*Goestnls-class*), 11
- cvar, GoGARCH-method  
(*GoGARCH-class*), 15
- cvar, Gopredict-method  
(*Gopredict-class*), 20
  
- formula, Goestica-method  
(*Goestica-class*), 7
- formula, Goestml-method  
(*Goestml-class*), 8
- formula, Goestmm-method  
(*Goestmm-class*), 10
- formula, Goestnls-method  
(*Goestnls-class*), 11
- formula, GoGARCH-method  
(*GoGARCH-class*), 15
  
- garchFit, 6, 18, 19, 22
- goest (*goest-methods*), 6
- goest, Goestica-method  
(*Goestica-class*), 7
- goest, Goestml-method  
(*Goestml-class*), 8
- goest, Goestmm-method  
(*Goestmm-class*), 10
- goest, Goestnls-method  
(*Goestnls-class*), 11
- goest-methods, 8, 9, 11, 13, 14
- goest-methods, 6
- Goestica, 6, 14
- Goestica-class, 7
- Goestml, 6, 14, 20–22
- Goestml-class, 8
- Goestmm, 6, 14
- Goestmm-class, 10
- Goestnls, 6, 14
- Goestnls-class, 11
- GoGARCH, 7–14, 20–22
- gogarch, 6, 8, 11, 13, 13, 20, 25
- GoGARCH-class, 15
- Goinit, 7–13, 15–17, 22, 27
- goinit, 16, 18, 27
- Goinit-class, 17
- gollh, 18
- gonls, 19
- Gopredict, 8, 9, 11, 13, 16
- Gopredict-class, 20
- Gosum, 8, 9, 11, 13, 16
- Gosum-class, 21
- gotheta, 21
  
- logLik (*Goestml-class*), 8
- logLik, Goestml-method  
(*Goestml-class*), 8
  
- M (*Orthom-class*), 23
- M, Orthom-method (*Orthom-class*), 23
  
- Orthom, 26, 28
- Orthom-class, 23
  
- plot, Goestica, missing-method  
(*Goestica-class*), 7
- plot, Goestml, missing-method  
(*Goestml-class*), 8
- plot, Goestmm, missing-method  
(*Goestmm-class*), 10
- plot, Goestnls, missing-method  
(*Goestnls-class*), 11
- plot, GoGARCH, missing-method  
(*GoGARCH-class*), 15
- predict, Goestica-method  
(*Goestica-class*), 7
- predict, Goestml-method  
(*Goestml-class*), 8
- predict, Goestmm-method  
(*Goestmm-class*), 10
- predict, Goestnls-method  
(*Goestnls-class*), 11

- predict, GoGARCH-method  
(*GoGARCH-class*), 15
- print, Orthom-method  
(*Orthom-class*), 23
- Rd2, 24, 26
- resid, Goestica-method  
(*Goestica-class*), 7
- resid, Goestml-method  
(*Goestml-class*), 8
- resid, Goestmm-method  
(*Goestmm-class*), 10
- resid, Goestnls-method  
(*Goestnls-class*), 11
- resid, GoGARCH-method  
(*GoGARCH-class*), 15
- residuals, Goestica-method  
(*Goestica-class*), 7
- residuals, Goestml-method  
(*Goestml-class*), 8
- residuals, Goestmm-method  
(*Goestmm-class*), 10
- residuals, Goestnls-method  
(*Goestnls-class*), 11
- residuals, GoGARCH-method  
(*GoGARCH-class*), 15
- show, Goestica-method  
(*Goestica-class*), 7
- show, Goestml-method  
(*Goestml-class*), 8
- show, Goestmm-method  
(*Goestmm-class*), 10
- show, Goestnls-method  
(*Goestnls-class*), 11
- show, GoGARCH-method  
(*GoGARCH-class*), 15
- show, Goinit-method  
(*Goinit-class*), 17
- show, Gopredict-method  
(*Gopredict-class*), 20
- show, Gosum-method (*Gosum-class*),  
21
- show, Orthom-method  
(*Orthom-class*), 23
- summary, Goestica-method  
(*Goestica-class*), 7
- summary, Goestml-method  
(*Goestml-class*), 8
- summary, Goestmm-method  
(*Goestmm-class*), 10
- summary, Goestnls-method  
(*Goestnls-class*), 11
- summary, GoGARCH-method  
(*GoGARCH-class*), 15
- t, Orthom-method (*Orthom-class*), 23
- Umatch, 11, 24
- unvech, 25
- update, Goestica-method  
(*Goestica-class*), 7
- update, Goestml-method  
(*Goestml-class*), 8
- update, Goestmm-method  
(*Goestmm-class*), 10
- update, Goestnls-method  
(*Goestnls-class*), 11
- update, GoGARCH-method  
(*GoGARCH-class*), 15
- UprodR, 23, 24, 26
- validGoinitObject, 27
- validOrthomObject, 23, 28
- VDW, 3, 29
- vec, 26