

# Package ‘ufRisk’

July 22, 2025

**Type** Package

**Title** Risk Measure Calculation in Financial TS

**Version** 1.0.7

**Description** Enables the user to calculate Value at Risk (VaR) and Expected Shortfall (ES) by means of various parametric and semiparametric GARCH-type models. For the latter the estimation of the nonparametric scale function is carried out by means of a data-driven smoothing approach. Model quality, in terms of forecasting VaR and ES, can be assessed by means of various backtesting methods such as the traffic light test for VaR and a newly developed traffic light test for ES. The approaches implemented in this package are described in e.g. Feng Y., Beran J., Letmathe S. and Ghosh S. (2020) <<https://ideas.repec.org/p/pdn/ciepap/137.html>> as well as Letmathe S., Feng Y. and Uhde A. (2021) <<https://ideas.repec.org/p/pdn/ciepap/141.html>>.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Imports** esemifar, fracdiff, rugarch, smoots, stats, utils

**Depends** R (>= 2.10)

**URL** <https://wiwi.uni-paderborn.de/en/dep4/feng/>

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Author** Yuanhua Feng [aut] (Paderborn University, Germany),  
Xuehai Zhang [aut] (Former research associate at Paderborn University,  
Germany),  
Christian Peitz [aut] (Paderborn University, Germany),  
Dominik Schulz [aut] (Paderborn University, Germany),  
Shujie Li [aut] (Paderborn Universtiy, Germany),  
Sebastian Letmathe [aut, cre] (Paderborn University, Germany)

**Maintainer** Sebastian Letmathe <[sebastian.letmathe@uni-paderborn.de](mailto:sebastian.letmathe@uni-paderborn.de)>

**Repository** CRAN

**Date/Publication** 2023-10-22 08:10:02 UTC

## Contents

covtest . . . . .	2
ESTX . . . . .	5
lossfunc . . . . .	6
plot.ufRisk . . . . .	8
print.ufRisk . . . . .	9
traffttest . . . . .	10
ufRisk . . . . .	12
varcast . . . . .	14
WMT . . . . .	18

<b>Index</b>	<b>19</b>
--------------	-----------

---

covtest	<i>Unconditional and Conditional Coverage Tests, Independence Test</i>
---------	--

---

### Description

The conditional (Kupiec, 1995), the unconditional coverage test (Christoffersen, 1998) and the independence test (Christoffersen, 1998) of the Value-at-Risk (VaR) can be applied.

### Usage

```
covtest(obj = list(Loss = NULL, VaR = NULL, p = NULL), conflvl = 0.95)
```

### Arguments

obj	a list that contains the following elements: Loss a numeric vector that contains the values of a loss series ordered from past to present; is set to NULL by default. VaR a numeric vector that contains the estimated values of the VaR for the same time points of the loss series Loss; is set to NULL by default. p a numeric vector with one element; defines the probability $p$ stated in the null hypotheses of the coverage tests (see the section Details for more information); is set to NULL by default.
conflvl	a numeric vector with one element; the significance level at which the null hypotheses are evaluated; is set to 0.95 by default. Please note that a list returned by the varcast function can be directly passed to covtest.

### Details

With this function, the conditional and the unconditional coverage tests introduced by Kupiec (1995) and Christoffersen (1998) can be applied. Given a return series  $r_t$  with  $n$  observations, divide the series into  $n - K$  in-sample and  $K$  out-of-sample observations, fit a model to the in-sample data and obtain rolling one-step forecasts of the VaR for the out-of-sample time points.

Define

$$I_t = 1,$$

if  $-r_t > \widehat{VaR}_t(\alpha)$  or

$$I_t = 0,$$

otherwise,

for  $t = n+1, n+2, \dots, n+K$  as the hit sequence, where  $\alpha$  is the confidence level for the VaR (often  $\alpha = 0.95$  or  $\alpha = 0.99$ ). Furthermore, denote  $p = \alpha$  and let  $w$  be the actual covered proportion of losses in the data.

1. Unconditional coverage test:

$$H_{0,uc} : p = w$$

Let  $K_1$  be the number of ones in  $I_t$  and analogously  $K_0$  the number of zeros (all conditional on the first observation). Also calculate  $\hat{w} = K_0/(K - 1)$ . Obtain

$$L(I_t, p) = p^{K_0}(1 - p)^{K_1}$$

and

$$L(I_t, \hat{w}) = \hat{w}^{K_0}(1 - \hat{w})^{K_1}$$

and subsequently the test statistic

$$LR_{uc} = -2 * \ln\{L(I_t, p)/L(I_t, \hat{w})\}.$$

$LR_{uc}$  now asymptotically follows a chi-square-distribution with one degree of freedom.

2. Conditional coverage test:

The conditional coverage test combines the unconditional coverage test with a test on independence. Denote by  $w_{ij}$  the probability of an  $i$  on day  $t - 1$  being followed by a  $j$  on day  $t$ , where  $i$  and  $j$  correspond to the value of  $I_t$  on the respective day.

$$H_{0,cc} : w_{00} = w_{10} = p$$

with  $i = 0, 1$  and  $j = 0, 1$ .

Let  $K_{ij}$  be the number of observations, where the values on two following days follow the pattern  $ij$ . Calculate

$$L(I_t, \hat{w}_{00}, \hat{w}_{10}) = \hat{w}_{00}^{K_{00}}(1 - \hat{w}_{00})^{K_{01}} * \hat{w}_{10}^{K_{10}}(1 - \hat{w}_{10})^{K_{11}},$$

where  $\hat{w}_{00} = K_{00}/K_0$  and  $\hat{w}_{10} = K_{10}/K_1$ . The test statistic is then given by

$$LR_{cc} = -2 * \ln\{L(I_t, p)/L(I_t, \hat{w}_{00}, \hat{w}_{10})\},$$

which asymptotically follows a chi-square-distribution with two degrees of freedom.

### 3. Independence test:

$$H_{0,ind} : w_{00} = w_{10}$$

The asymptotically chi-square-distributed test statistic (one degree of freedom) is given by

$$LR_{ind} = -2 * \ln\{L(I_t, \hat{w}_{00}, \hat{w}_{10})/L(I_t, \hat{w})\}.$$

---

The function needs four inputs: the out-of-sample loss series `obj$Loss`, the corresponding estimated VaR series `obj$VaR`, the coverage level `obj$p`, for which the VaR has been calculated and the significance level `conf1vl`, at which the null hypotheses are evaluated. If an object returned by this function is entered into the R console, a detailed overview of the test results is printed.

### Value

A list of class `ufRisk` with the following four elements:

**p** probability `p` stated in the null hypotheses of the coverage tests.

**p.uc** the p-value of the unconditional coverage test.

**p.cc** the p-value of the conditional coverage test.

**p.ind** the p-value of the independence test.

**conf1vl** the significance level at which the null hypotheses are evaluated.

### Author(s)

- Sebastian Letmathe (Scientific Employee) (Department of Economics, Paderborn University)
- Dominik Schulz (Scientific Employee) (Department of Economics, Paderborn University),

### References

Christoffersen, P. F. (1998). Evaluating interval forecasts. *International economic review*, pp. 841-862.

Kupiec, P. (1995). Techniques for verifying the accuracy of risk measurement models. *The J. of Derivatives*, 3(2).

### Examples

```
# Example for Walmart Inc. (WMT)
prices <- WMT$price.close
output <- varcast(prices)
Loss <- -output$ret.out
VaR <- output$VaR.v
covtest.data <- list(Loss = Loss, VaR = VaR, p = 0.99)
```

```
covtest(covtest.data)

# directly passing an output object of 'varcast()' to 'covtest()'
output <- varcast(prices)
covtest(output)
```

---

ESTX

*EURO STOXX 50 (ESTX) Financial Time Series Data*

---

### Description

A dataset that contains the daily financial data of the ESTX from April 2007 to December 2021 (currency in EUR).

### Usage

ESTX

### Format

A data frame with 3697 rows and 10 variables:

**price.open** opening price (daily)

**price.high** highest price (daily)

**price.low** lowest price (daily)

**price.close** closing price (daily)

**volume** trading volume

**price.adjusted** adjusted closing price (daily)

**ref.date** date in format YY-MM-DD

**ticker** ticker symbol

**ret.adjusted.prices** returns obtained from the adjusted closing prices

**ret.closing.prices** returns obtained from the closing prices

### Source

The data was obtained from Yahoo Finance.

lossfunc

*Loss Functions***Description**

This functions allows for the calculation of loss functions for the selection of models.

**Usage**

```
lossfunc(obj = list(Loss = NULL, ES = NULL), beta = 1e-04)
```

**Arguments**

**obj** a list that contains the following elements:  
**Loss** a numeric vector that contains the values of a loss series ordered from past to present; is set to NULL by default  
**ES** a numeric vector that contains the estimated values of the ES for the same time points of the loss series Loss; is set to NULL by default  
Please note that a list returned by the varcast function can be directly passed to lossfunc.

**beta** a single numeric value; a measure for the opportunity cost of capital; default is 1e-04.

**Details**

Given a negative return series  $\text{obj}\$Loss$ , the corresponding Expected Shortfall (ES) estimates  $\text{obj}\$ES$  and a parameter  $\beta$  that defines the opportunity cost of capital, four different definitions of loss functions are considered.

Let  $K$  be the number of observations and  $r_t$  the observed return series. Following Sarma et al. (2003)

$$l_{t,1} = \{\widehat{ES}_t(\alpha) + r_t\}^2,$$

if  $-r_t > \widehat{ES}_t(\alpha)$

$$l_{t,1} = \beta * \widehat{ES}_t(\alpha),$$

otherwise,

is a suitable loss function (firm's loss function), where  $\beta$  is the opportunity cost of capital. The regulatory loss function is identical to the firm's loss function with the exception of  $l_{t,1} = 0$  for  $-r_t \leq \widehat{ES}_t(\alpha)$ .

Abad et al. (2015) proposed another loss function

$$l_{t,a} = \{\widehat{ES}_t(\alpha) + r_t\}^2,$$

if  $-r_t > \widehat{ES}_t(\alpha)$

$$l_{t,a} = \beta * (\widehat{ES}_t(\alpha) + r_t),$$

otherwise,

that, however, also considers opportunity costs for  $r_t > 0$ . An adjustment has been proposed by Feng. Following his idea,

$$l_{t,2} = \{\widehat{ES}_t(\alpha) + r_t\}^2,$$

if  $-r_t > \widehat{ES}_t(\alpha)$

$$l_{t,2} = \beta * \min\{\widehat{ES}_t(\alpha) + r_t, \widehat{ES}_t(\alpha)\},$$

otherwise,

should be considered as a compromise of the regulatory and the firm's loss functions. Note that instead of the ES, also a series of Value-at-Risk values can be inserted for the argument obj\$ES. However this is not possible if a list returned by the varcast function is directly passed to lossfunc.

## Value

an S3 class object, which is a list of

**loss.func1** Regulatory loss function.

**loss.func2** Firm's loss function following Sarma et al. (2003).

**loss.func3** Loss function following Abad et al. (2015).

**loss.func4** Feng's loss function. A compromise of regulatory and firm's loss function.

## Author(s)

- Sebastian Letmathe (Scientific Employee) (Department of Economics, Paderborn University)
- Dominik Schulz (Scientific Employee) (Department of Economics, Paderborn University),

## References

- Abad, P., Muela, S. B., & Martín, C. L. (2015). The role of the loss function in value-at-risk comparisons. *The Journal of Risk Model Validation*, 9(1), 1-19.
- Sarma, M., Thomas, S., & Shah, A. (2003). Selection of Value-at-Risk models. *Journal of Forecasting*, 22(4), 337-358.

## Examples

```
# Example for Walmart Inc. (WMT)
prices <- WMT$price.close
output <- varcast(prices)
Loss <- -output$ret.out
ES <- output$ES
loss.data <- list(Loss = Loss, ES = ES)
lossfunc(loss.data)

# directly passing an output object of 'varcast()' to 'lossfunc()'
x <- WMT$price.close
output <- varcast(prices)
lossfunc(output)
```

---

plot.ufRisk

*Plot Method for the Package 'ufRisk'*

---

## Description

This function regulates how objects created by the package ufRisk are plotted.

## Usage

```
## S3 method for class 'ufRisk'
plot(x, plot.option = NULL, ...)
```

## Arguments

x	an input object of class ufRisk.
plot.option	plot choice for an object of class ufRisk; viable choices are: <ol style="list-style-type: none"> <li>1 Plotting out-of-sample loss series</li> <li>2 Plotting out-of-sample losses, VaR.v &amp; breaches</li> <li>3 Plotting out-of-sample losses, VaR.e, ES &amp; breaches</li> </ol> Please note if no value is passed to plot.option a selection menu is prompted; is set to NULL by default.
...	additional arguments of the standard plot method.

## Value

None



**Author(s)**

- Sebastian Letmathe (Scientific Employee) (Department of Economics, Paderborn University)
- Dominik Schulz (Research Assistant) (Department of Economics, Paderborn University),

---

print.ufRisk	<i>Print Method for Objects of Class 'ufRisk'</i>
--------------	---

---

**Description**

This function regulates how objects of class ufRisk are printed.

**Usage**

```
## S3 method for class 'ufRisk'  
print(x, ...)
```

**Arguments**

- |     |  |
|-----|--|
| x   | an object of class ufRisk; for the current package version, only the function br_test returns such an object.              |
| ... | implemented for compatibility with the generic function; additional arguments, however, will not affect this print method. |

**Value**

None

**Author(s)**

- Sebastian Letmathe (Scientific Employee) (Department of Economics, Paderborn University)
- Dominik Schulz (Scientific Employee) (Department of Economics, Paderborn University),

---

trafftest	<i>Backtesting of Value-at-Risk and Expected Shortfall via Traffic Light Tests</i>
-----------	--

---

### Description

Backtesting methods, most importantly traffic light tests, are applied to previously calculated Value-at-Risk and Expected Shortfall series.

### Usage

trafftest(obj)

### Arguments

obj                      A list returned by the varcast function, that contains different estimated Value-at-Risk and Expected Shortfall series; any other list that follows the name conventions of the varcast function can be used as well.

### Details

The Traffic Light Test for backtesting the Value-at-Risk (VaR) was proposed by the Basel Committee on Banking Supervision (1996). A formal mathematical description was given by Constanzino and Curran (2018). Following Constanzino and Curran (2018), define the Value-at-Risk breach indicator by

$$X_{VaR}^{(i)}(\alpha) = 1_{\{L_i \geq VaR_i(\alpha)\}},$$

where  $i$  defines the corresponding trading day,  $L_i$  is the loss (denoted as a positive value) on day  $i$  and  $\alpha$  is the confidence level of the VaR (e.g. if  $\alpha = 0.95$ , the 95%-VaR is considered). The total number of breaches over all trading days  $i = 1, 2, \dots, N$  is then given by

$$X_{VaR}^N(\alpha) = \sum_{i=1}^N 1_{\{L_i \geq VaR_i(\alpha)\}}.$$

Following a Binomial Distribution, the cumulative probabilities of observing a specific number of breaches or less can be computed. Under the hypothesis that the selected volatility model is true, the cumulative probability of observing  $X_{VaR}^N(\alpha)$  breaches is therefore easily obtainable. The Basel Committee on Banking Supervision (1996) defined three zones. Depending on the zone the calculated cumulative probability can be sorted into, the suitability of the selected model can be assessed. Models with calculated cumulative probabilities  $< 95\%$  belong to the green zone and are considered appropriate. Furthermore, if the probabilities are greater or equal to 95% but smaller than 99.99%, the corresponding models are categorized into the yellow zone. The red zone is for models with cumulative probabilities greater or equal to 99.99%. If the test results in a yellow zone classification, the respective VaR values require additional monitoring. Moreover, the Basel Committee recommended to consider additional capital requirements of a bank, if its model used is in the yellow zone. Models in the red zone are considered to be heavily flawed.

Based on the same three-zone approach with the same zone boundaries, Constanzino and Curran (2018) proposed a traffic light test for the Expected Shortfall (ES). The total severity of breaches is given by

$$X_{ES}^N(\alpha) = \sum_{i=1}^N (1 - (1 - F(L_i))/(1 - \alpha)) * 1_{\{L_i \geq VaR_i(\alpha)\}},$$

with  $F(L_i)$  being the cumulative distribution of the loss at day  $i$ . As stated by Constanzino and Curran (2018),  $X_{ES}^N(\alpha)$  is approximately normally distributed  $\mathcal{N}(\mu_{ES}, N\sigma_{ES}^2)$  for large samples, where  $\mu_{ES} = 0.5(1 - \alpha)N$  and  $\sigma_{ES}^2 = (1 - \alpha)(4 - 3(1 - \alpha))/12$ , from which cumulative probabilities for the observed breaches  $X_{ES}^N$  can be easily obtained.

For semiparametric models, the backtesting of the VaR is analogous to the described approach. Backtesting the ES, however, requires minor adjustments. Given that the model's underlying innovations follow a standardized t-distribution with degrees of freedom  $\nu$ , define by  $r_t$  the demeaned returns and by  $\hat{s}_t$  the estimated total volatility.

$$\hat{\epsilon}_t^* = -r_t/\hat{s}_t\sqrt{\nu/(\nu - 2)}$$

are now suitable to calculate the total severity of breaches under the assumption that  $\epsilon_t^*$  are identically and independently distributed t-distributed random variables.

This function uses an object returned by the `varcast` function of the `uFRisk` package as an input for the function argument `obj`. A list with different elements, such as the cumulative probabilities for the VaR and ES series within `obj`, is returned. Instead of the list, only the traffic light backtesting results are printed to the R console.

NOTE:

More information on VaR and ES can be found in the documentation of the `varcast` function of the `uFRisk` package [varcast](#).

## Value

A list of class `uFRisk` is returned with the following elements.

**model** selected model for estimation

**p\_VaR.e** cumulative probability of observing the number of breaches or fewer for (1 - a.e)100%-VaR

**p\_VaR.v** cumulative probability of observing the number of breaches or fewer for (1 - a.v)100%-VaR

**p\_ES** cumulative probability of observing the number of breaches or fewer for (1 - a.e)100%-ES

**pot\_VaR.e** number of exceedances for (1 - a.e)100%-VaR

**pot\_VaR.v** number of exceedances for (1 - a.v)100%-VaR

**potES** number of exceedances for (1 - a.e)100%-ES

**br.sum** sum of breaches for (1 - a.e)100%-ES

**WAD** weighted absolute deviations - a model selection criterion

**a.v** coverage level for the (1-a.v)100% VaR

**a.e** coverage level for (1-a.e)100% VaR

**Author(s)**

- Sebastian Letmathe (Scientific Employee) (Department of Economics, Paderborn University),
- Dominik Schulz (Scientific Employee) (Department of Economics, Paderborn University),

**References**

Basel Committee on Banking Supervision (1996). Supervisory Framework For The Use of Back-Testing in Conjunction With The Internal Models Approach to Market Risk Capital Requirements. Available online: <https://www.bis.org/publ/bcbs22.htm> (accessed on 23 June 2020).

Constanzino, N., and Curran, M. (2018). A Simple Traffic Light Approach to Backtesting Expected Shortfall. In: Risks 6.1.2.

**Examples**

```
# Example for Walmart Inc. (WMT)
prices = WMT$price.close
output = varcast(prices)
traffttest(output)
```

---

ufRisk

*ufRisk: A package for user friendly and practical usage of various backtesting methods.*

---

**Description**

The goal of the `ufRisk` package (univariate financial risk) is to enable the user to compute one-step ahead forecasts of Value at Risk (VaR) and Expected Shortfall (ES) by means of various parametric and semiparametric GARCH-type models. For the latter the estimation of the nonparametric scale function is carried out by means of a data-driven smoothing approach. Currently the GARCH, the exponential GARCH (EGARCH), the Log-GARCH, the asymmetric power ARCH (APARCH), the FIGARCH and FI-Log-GARCH can be employed within the scope of `ufRisk`. Model quality, in terms of forecasting VaR and ES, can be assessed by means of various backtesting methods.

**Functions**

`varcast` is a function to calculate rolling one-step ahead forecasts of VaR and ES for a selection of parametric and semiparametric GARCH-type models (see also `varcast`).

`traffttest` is a function for backtesting VaR and ES. ES is backtested via a newly developed traffic light approach. (see also `traffttest`).

`covtest` is a function for conducting the conditional and the unconditional coverage tests introduced by Kupiec (1995) and Christoffersen (1998). (see also `covtest`).

**Author(s)**

- Yuanhua Feng (Department of Economics, Paderborn University),  
Author of the Algorithms  
Website: <https://wiwi.uni-paderborn.de/en/dep4/feng/>
- Xuehai Zhang (Former research associate at Paderborn University),  
Author
- Shujie Li (Scientific Employee) (Department of Economics, Paderborn University),  
Author
- Christian Peitz (Department of Economics, Paderborn University),  
Author
- Dominik Schulz (Scientific Employee) (Department of Economics, Paderborn University),  
Author
- Sebastian Letmathe (Scientific Employee) (Department of Economics, Paderborn University),  
Package Creator and Maintainer

**References**

- Basel Committee on Banking Supervision (1996). Supervisory Framework For The Use of Back-Testing in Conjunction With The Internal Models Approach to Market Risk Capital Requirements. Available online: <https://www.bis.org/publ/bcbs22.htm> (accessed on 23 June 2020).
- Beran, J., and Feng, Y. (2002). Local polynomial fitting with long-memory, short-memory and antipersistent errors. *Annals of the Institute of Statistical Mathematics*, 54(2), pp. 291-311.
- Constanzino, N., and Curran, M. (2018). A Simple Traffic Light Approach to Backtesting Expected Shortfall. In: *Risks* 6.1.2.
- Feng, Y. (2004). Simultaneously modeling conditional heteroskedasticity and scale change. In: *Econometric Theory*, pp. 563-596.
- Feng, Y., Beran, J., Letmathe, S., & Ghosh, S. (2020). Fractionally integrated Log-GARCH with application to value at risk and expected shortfall (No. 137). Paderborn University, CIE Center for International Economics.
- Letmathe, S., Feng, Y., & Uhde, A. (2021). Semiparametric GARCH models with long memory applied to Value at Risk and Expected Shortfall (No. 141). Paderborn University, CIE Center for International Economics.
- McNeil, A.J., Frey, R., and Embrechts, P. (2015). *Quantitative risk management: concepts, techniques and tools - revised edition*. Princeton University Press.

---

varcast	<i>Calculation of one-step ahead forecasts of Value at Risk and Expected Shortfall (parametric and semiparametric)</i>
---------	--

---

### Description

One-step ahead forecasts of Value at Risk and Expected Shortfall for a selection of short-memory and long-memory parametric as well as semiparametric GARCH-type models are computed.

### Usage

```
varcast(
  x,
  a.v = 0.99,
  a.e = 0.975,
  model = c("sGARCH", "lGARCH", "eGARCH", "apARCH", "fiGARCH", "filGARCH"),
  garchOrder = c(1, 1),
  distr = c("norm", "std"),
  n.out = 250,
  smooth = c("none", "lpr"),
  ...
)
```

### Arguments

<code>x</code>	a vector containing the price series.
<code>a.v</code>	confidence level for calculating VaR; is set to 0.99 by default.
<code>a.e</code>	confidence level for calculating ES; is set to 0.975 by default.
<code>model</code>	model to be estimated. Options are 'sGARCH', 'eGARCH', 'apARCH', 'lGARCH', 'fiGARCH' and 'filGARCH'; is set to 'sGARCH' by default.
<code>garchOrder</code>	orders to be estimated; c(1, 1), i.e. $p = q = 1$ , is the default.
<code>distr</code>	distribution to use for the innovations of the respective GARCH model; is set to 'std' by default
<code>n.out</code>	size of out-sample; is set to 250 by default.
<code>smooth</code>	a character object; defines the data-driven smoothing approach for the estimation of the nonparametric scale function; for <code>smooth = 'lpr'</code> , the scale function is obtained from the logarithm of the squared centralized returns by means of the <code>msmooth()</code> function or <code>tsmoothlm()</code> function if <code>model</code> is set to 'sGARCH', 'eGARCH', 'apARCH' and 'lGARCH' } or <code>fiGARCH</code> and 'filGARCH', respectively; is set to <code>smooth = 'none'</code> by default.
<code>...</code>	depending on the choice of <code>model</code> , further arguments can be passed to either <code>smooths::msmooth()</code> or to <code>tsmoothlm()</code> ; if no other arguments are given, the default settings are used for both functions with the exception of $p = 3$ .

## Details

Let  $Y_t$  be a (demeaned) return series. The semiparametric extension of the GARCH(p,q) model (Bollerslev, 1986) is called a Semi-GARCH model (Feng, 2004) and is defined by

$$Y_t = s(x_t)\sigma_t\eta_t,$$

with  $\eta_t \sim IID(0,1)$  and

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^p \alpha_i Y_{t-i}^2 + \sum_{j=1}^q \beta_j \sigma_{t-j}^2,$$

where  $\sigma_t > 0$  are the conditional standard deviations,  $s(x_t) > 0$  is a nonparametric scale function with  $x_t$  being the rescaled observation time points on the interval  $[0,1]$  and  $\alpha_i$  and  $\beta_j$  are non-negative real valued coefficients, except for  $\alpha_0$ , which must satisfy  $\alpha_0 > 0$ . Furthermore, it is assumed that  $\text{Var}(\sigma_t\eta_t) = 1$ . In this functions, different short-memory and long-memory GARCH-type models are selectable for the parametric part of the model. More specifically, the standard GARCH (Bollerslev, 1986), the Log-GARCH (Pantula, 1986; Geweke, 1986; Milhoj, 1988), the eGARCH (Nelson, 1991), the APARCH (Ding et al., 1993), the FIGARCH (Baillie et al., 1996) and the FI-Log-GARCH (Feng et al., 2020) model are implemented. For more information on the representations of the last three models mentioned, we refer the reader to the corresponding references listed in the references section.

While the innovations  $\eta_t$  must be i.i.d. (independent and identically distributed) with zero-mean and unit-variance and while any distribution that satisfies these conditions is suitable, the standardized t-distribution is selected for the estimation of the models and computation of the Value at Risk (VaR) as well as the Expected Shortfall (ES) within this function.

For a given level  $\alpha \in (0, 1)$ ,

$$VaR(\alpha) = \inf\{z \in R : F_L(z) \geq \alpha\}$$

defines the VaR at level alpha. In this definition,  $L$  is the loss variable (making a loss is denoted as a positive value, whereas gains are negative values) and  $F_L$  is its cumulative distribution function. Explained differently,  $VaR(\alpha)$  is the  $\alpha$ -quantile of the loss distribution.

The ES for a level  $\alpha$ , however, is given by

$$ES(\alpha) = (1/(1 - \alpha)) \int_{\alpha}^1 VaR(u)du,$$

i.e. it is the expected loss in case  $VaR(\alpha)$  is exceeded. More information on these risk measures can be found on pp. 64-72 in McNeil et al. (2015).

To apply the function, a numeric vector that contains the price series that is to be analyzed ordered from past to present must be passed to the argument  $x$ . Furthermore, the user can set different levels of alpha for the VaR and the ES via the arguments  $a.v$  and  $a.e$ , respectively. A parametric short-memory or long-memory GARCH-type model can be selected by means of `model`, which only accepts a single-element character vector as input. At the time of the release of package version 1.0.0, a standard GARCH ('sGARCH'), a Log-GARCH ('lGARCH'), an eGARCH ('eGARCH'), an APARCH ('apARCH'), a FIGARCH ('fiGARCH') and a FI-Log-GARCH ('filGARCH') model

can be selected, each with conditional t-distribution. By default, a standard GARCH model is applied. The orders of the GARCH-type models can be defined with `garchOrder`, which is a numeric vector with two elements. Its first element is the ARCH order  $p$ , whereas the GARCH order  $q$  can be adjusted via the second element. If no adjustments are made, the orders  $p = q = 1$  are selected. The number of out-sample observations is set via the argument `n.out`. If `n` is the total number of observations of the whole price series, the model is estimated for the first  $n - n.out$  observations (in-sample), while the VaR and the ES are obtained for the last `n.out` observations (out-sample) based on the estimated model for the in-sample. Moreover, the data-driven estimation method of the underlying scale function can be adjusted via the argument `smooth`. If `smooth = 'lpr'` is selected, the scale function is obtained by applying an iterative plug-in algorithm logarithm of the squared centralized returns. Depending on the setting of `model` an algorithm proposed by Feng, Gries and Fritz (2020) or by Letmathe, Feng and Uhde (2021) is employed. In the former case, the function `msmooth()` of the `smoots` package is applied and for the latter the `tsmoothlm()` function of the `esemifar` package is used. An ellipsis `...` is implemented to allow for additional arguments for `msmooth()` and `tsmoothlm()`.

**NOTE:**

This function makes use of the `arima()` function of the `stats` package, of the `fracdiff()` function of the `fracdiff` package, of the `ugarchspec()` and `ugarchfit()` functions of the `rugarch` package, of the `msmooth()` function of the `smoots` package and of the `esemifar()` function of the `esemifar` for estimation. Moreover, Log-GARCH and FI-Log-GARCH models in the parametric part of the complete models are estimated via their ARMA and FARIMA representations, respectively, and must therefore satisfy  $p \geq q$ .

**Value**

This function returns a list with the following elements.

**model** selected model for estimation

**mean** the estimated mean of the in-sample returns

**model.fit** estimated model parameters for the parametric part of the in-sample

**np.est** the estimation results for the nonparametric part of the in-sample model

**ret.in** in-sample return series

**ret.out** out-sample return series

**sig.in** estimated in-sample total volatility

**sig.fc** out-sample forecasts of the total volatility

**scale** the estimated nonparametric scale function values for the in-sample

**scale.fc** the scale function forecast for the out-sample

**VaR.e** out-sample forecasts of the  $(1-a.e)100\%$  VaR

**VaR.v** out-sample forecasts of the  $(1-a.v)100\%$  VaR

**ES** out-sample forecasts of the  $(1-a.e)100\%$  ES

**dfree** estimated degrees of freedom for the standardized returns

**a.v** coverage level for the 99 % VaR

**a.e** coverage level for 97.5 % VaR

**garchOrder** the orders  $p$  and  $q$  of the implemented GARCH-type model



**Author(s)**

- Sebastian Letmathe (Scientific Employee) (Department of Economics, Paderborn University)
- Dominik Schulz (Scientific Employee) (Department of Economics, Paderborn University),

**References**

- Baillie, R. T., Bollerslev, T., & Mikkelsen, H. O. (1996). Fractionally integrated generalized autoregressive conditional heteroskedasticity. In: *Journal of Econometrics*, 74.1, pp. 3-30.
- Bollerslev, T. (1986) Generalized autoregressive conditional heteroskedasticity. In: *Journal of Econometrics* 31.3, pp. 307-327.
- Ding, Z., Granger, C.W., and Engle, R.F. (1993). A long memory property of stock market returns and a new model. In: *Journal of Empirical Finance* 1.1, pp. 83-106.
- Feng, Y. (2004). Simultaneously modeling conditional heteroskedasticity and scale change. In: *Econometric Theory* 20.3, pp. 563-596.
- Feng, Y., Beran, J., Letmathe, S., & Ghosh, S. (2020). Fractionally integrated Log-GARCH with application to value at risk and expected shortfall (No. 137). Paderborn University, CIE Center for International Economics.
- Pantula, S.G. (1986). Modeling the persistence of conditional variances: a comment. In: *Econometric Reviews* 5, pp. 79-97.
- Geweke, J. (1986). Comment on: Modelling the persistence of conditional variances. In: *Econometric Reviews* 5, pp. 57-61.
- Letmathe, S., Feng, Y., & Uhde, A. (2021). Semiparametric GARCH models with long memory applied to Value at Risk and Expected Shortfall (No. 141). Paderborn University, CIE Center for International Economics.
- McNeil, A.J., Frey, R., and Embrechts, P. (2015). *Quantitative risk management: concepts, techniques and tools - revised edition*. Princeton University Press.
- Milhoj, A. (1988). A Multiplicative parametrization of ARCH models. Universitetets Statistiske Institut.
- Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. In: *Econometrica: Journal of the Econometric Society*, 347-370.

**Examples**

```
# Example for Walmart Inc. (WMT)
prices = WMT$price.close

# forecasting VaR and ES
results = varcast(prices, model = 'sGARCH', n.out = 250)
ret.out = results$ret.out
n.out = length(ret.out)
VaR97.5 = results$VaR.e
VaR99 = results$VaR.v
ES = results$ES
```

```
# plotting VaR at 99% coverage
matplot(1:n.out, cbind(-ret.out, VaR99),
        type = 'hl',
        xlab = 'number of out-of-sample obs.', ylab = 'losses, VaR and ES',
        main = '99% VaR (red) for the WMT return series')

# plotting VaR at 97.5% coverage and corresponding ES
matplot(1:n.out, cbind(-ret.out, ES, VaR97.5),
        type = 'hll',
        xlab = 'number of out-of-sample obs.', ylab = 'losses, VaR and ES',
        main = '97.5% VaR (green) and ES (red) for the WMT return series')
```

---

WMT

*Walmart Inc. (WMT) Financial Time Series Data*

---

### Description

A dataset that contains the daily financial data of WMT from January 2000 to December 2021 (currency in EUR).

### Usage

WMT

### Format

A data frame with 5535 rows and 10 variables:

**price.open** opening price (daily)

**price.high** highest price (daily)

**price.low** lowest price (daily)

**price.close** closing price (daily)

**volume** trading volume

**price.adjusted** adjusted closing price (daily)

**ref.date** date in format YY-MM-DD

**ticker** ticker symbol

**ret.adjusted.prices** returns obtained from the adjusted closing prices

**ret.closing.prices** returns obtained from the closing prices

### Source

The data was obtained from Yahoo Finance.

# Index

## \* datasets

ESTX, [5](#)

WMT, [18](#)

covtest, [2](#), [12](#)

ESTX, [5](#)

lossfunc, [6](#)

plot.ufRisk, [8](#)

print.ufRisk, [9](#)

trafftest, [10](#), [12](#)

ufRisk, [12](#)

ufRisk-package (ufRisk), [12](#)

varcast, [11](#), [12](#), [14](#)

WMT, [18](#)