

# Package ‘SWIM’

September 22, 2020

**Title** Scenario Weights for Importance Measurement

**Version** 0.2.2

**Author** Silvana M. Pesenti [aut, cre],  
Alberto Bettini [aut],  
Pietro Millosovich [aut],  
Andreas Tsanakas [aut]

**Maintainer** Silvana M. Pesenti <swimpackage@gmail.com>

**Description** An efficient sensitivity analysis for stochastic models based on Monte Carlo samples. Provides weights on simulated scenarios from a stochastic model, such that stressed random variables fulfil given probabilistic constraints (e.g. specified values for risk measures), under the new scenario weights. Scenario weights are selected by constrained minimisation of the relative entropy to the baseline model. The 'SWIM' package is based on Pesenti S.M, Millosovich P., Tsanakas A. (2019) "Reverse Sensitivity Testing: What does it take to break the model", <openaccess.city.ac.uk/id/eprint/18896/>.

**Depends** R (>= 3.5.0)

**Imports** Rdpack (>= 0.7), Hmisc, nleqslv, reshape2, plyr, ggplot2, stats

**RdMacros** Rdpack

**License** GPL-3

**URL** <https://github.com/spesenti/SWIM>,  
[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3515274](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3515274),  
[https://utstat.toronto.edu/pesenti/?page\\_id=138](https://utstat.toronto.edu/pesenti/?page_id=138)

**BugReports** <https://github.com/spesenti/SWIM/issues>

**Encoding** UTF-8

**LazyLoad** true

**LazyData** true

**VignetteBuilder** knitr

**RoxygenNote** 7.1.1

**Suggests** testthat, mvtnorm, spelling, Weighted.Desc.Stat, knitr, rmarkdown, bookdown, ggpubr

**Language** en-US

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-09-22 10:50:16 UTC

## R topics documented:

cdf . . . . .	2
credit_data . . . . .	3
ES_stressed . . . . .	4
get_data . . . . .	6
importance_rank . . . . .	8
merge.SWIM . . . . .	9
plot_cdf . . . . .	10
plot_hist . . . . .	12
plot_quantile . . . . .	13
plot_sensitivity . . . . .	14
plot_weights . . . . .	16
quantile_stressed . . . . .	18
sensitivity . . . . .	19
stress . . . . .	21
stress_mean . . . . .	23
stress_mean_sd . . . . .	24
stress_moment . . . . .	26
stress_prob . . . . .	28
stress_user . . . . .	30
stress_VaR . . . . .	32
stress_VaR_ES . . . . .	34
summary.SWIM . . . . .	36
SWIM . . . . .	37

**Index** **40**

---

cdf *Empirical Distribution Function of a Stressed Model*

---

### Description

Provides the empirical distribution function of a stressed model component (random variable) under the scenario weights.

### Usage

```
cdf(object, xCol = 1, wCol = 1)
```

**Arguments**

object	A SWIM object.
xCol	Numeric or character, (name of) the column of the underlying data of the object (default = 1).
wCol	Numeric, the column of the scenario weights of the object (default = 1).

**Value**

The empirical distribution function (a function) of the xCol component of the stressed model with weights wCol. The empirical distribution function can be evaluated at a vector.

**Author(s)**

Silvana M. Pesenti

**See Also**

See [plot\\_cdf](#) for plotting the empirical distribution function of the stressed model and [quantile\\_stressed](#) for sample quantiles of a stressed model.

**Examples**

```
## example with a stress on VaR
set.seed(0)
x <- as.data.frame(cbind(
  "normal" = rnorm(1000),
  "gamma" = rgamma(1000, shape = 2)))
res1 <- stress(type = "VaR", x = x,
  alpha = c(0.9, 0.95), q_ratio = 1.05)
grid <- seq(min(x$normal), max(x$normal), length.out = 5)
## stressed empirical distribution function
cdf(res1, xCol = 1, wCol = 1)(grid)
## baseline empirical distribution function
ecdf(x$normal)(grid)
```

---

credit\_data

*Credit data set*

---

**Description**

A dataset containing total aggregate losses from three sub-portfolios, generated through a binomial credit model.

**Usage**

credit\_data

**Format**

A data frame with 100,000 rows and 7 variables:

**L** total aggregate loss of a portfolio consisting of three homogeneous sub-portfolios L1, L2 and L3

**L1** aggregate loss of sub-portfolio 1

**L2** aggregate loss of sub-portfolio 2

**L3** aggregate loss of sub-portfolio 3

**H1** (conditional) default probability of sub-portfolio 1

**H2** (conditional) default probability of sub-portfolio 2

**H3** (conditional) default probability of sub-portfolio 3

**Source**

For a detailed case study of the credit data set using SWIM see Pesenti Silvana M. and Bettini A, Millosovich P, Tsanakas A (2020). "Scenario Weights for Importance Measurement (SWIM) - An R package for sensitivity analysis." Available at SSRN: <https://www.ssrn.com/abstract=3515274>.

---

ES\_stressed

*Value-at-Risk and Expected Shortfall of a Stressed Model*

---

**Description**

Provides the Value-at-Risk (VaR) and the Expected Shortfall (ES) for components (random variables) of a stochastic model.

**Usage**

```
ES_stressed(object, alpha = 0.95, xCol = "all", wCol = 1, base = FALSE)
```

```
VaR_stressed(object, alpha = 0.95, xCol = "all", wCol = 1, base = FALSE)
```

**Arguments**

object	A SWIM object.
alpha	Numeric vector, the levels of the stressed VaR and ES (default = 0.95).
xCol	Numeric or character vector, (names of) the columns of the underlying data of the object (default = "all").
wCol	Numeric, the column of the scenario weights of the object (default = 1).
base	Logical, if TRUE, statistics under the baseline are also returned (default = "FALSE").

## Details

ES\_stressed: The ES of a stressed model is the ES of a chosen stressed model component, subject to the calculated scenario weights. The ES at level alpha of a stressed model component is given by:

$$ES_{alpha} = 1/(1 - alpha) * \int_{alpha}^1 VaR_u^W du,$$

where  $VaR_u^W$  is the VaR of the stressed model component, defined below.

VaR\_stressed: The VaR of a model is the VaR (quantile) of a chosen stressed model component, subject to the calculated scenario weights. The VaR at level alpha of a stressed model component with stressed distribution function  $F^W$  is defined as its left-quantile at alpha:

$$VaR_{alpha}^W = F^{W,-1}(alpha).$$

The function VaR\_stressed provides the empirical quantile, whereas the function quantile\_stressed calculates quantiles of model components with different interpolations.

## Value

ES\_stressed: Returns a matrix with the empirical ES's at level alpha of model components specified in xCol, under the scenario weights wCol.

VaR\_stressed: Returns a matrix with the empirical VaR's at level alpha of model components specified in xCol, under the scenario weights wCol.

## Functions

- ES\_stressed: Expected Shortfall of a stressed model
- VaR\_stressed: Value-at-Risk of a stressed model.

## Author(s)

Silvana M. Pesenti

## See Also

See quantile\_stressed for quantiles other than the empirical quantiles and cdf for the empirical distribution function of a stressed model.

## Examples

```
## example with a stress on VaR
set.seed(0)
x <- as.data.frame(cbind(
  "normal" = rnorm(1000),
  "gamma" = rgamma(1000, shape = 2)))
res1 <- stress(type = "VaR", x = x,
  alpha = c(0.9, 0.95), q_ratio = 1.05)
## stressed ES
quantile_stressed(res1, probs = seq(0.9, 0.99, 0.01),
```

```

        xCol = 1, wCol = 2, type = "i/n")
quantile(x[, 1], probs = seq(0.9, 0.99, 0.01), type = 1)
VaR_stressed(res1, alpha = seq(0.9, 0.99, 0.01), xCol = 1,
             wCol = 2, base = TRUE)

## the ES of both model components under stress one
ES_stressed(res1, alpha = seq(0.9, 0.99, 0.01), xCol = "all",
            wCol = 1)
## the ES of both model components under stress two
ES_stressed(res1, alpha = seq(0.9, 0.99, 0.01), xCol = "all",
            wCol = 2)

```

---

get\_data

*Extracting from a Stressed Model*


---

## Description

Extracting the data (realisations of the stochastic model), the scenario weights, the functions generating the scenario weights, or the specifications of the stress from an object of class SWIM.

## Usage

```

get_data(object, xCol = "all")

get_weights(object)

get_weightsfun(object)

get_specs(object)

```

## Arguments

object	A SWIM object.
xCol	Numeric or character vector, (names of) the columns of the underlying data of the object (default = "all").

## Value

`get_data`: A data.frame containing the realisations of the stochastic model on which the object is based.

`get_weights`: A data.frame containing the scenario weights of the object. Columns corresponds to different stresses.

`get_weightsfun`: A list containing functions, which, when applied to a column of the data, generate the scenario weights of the object. The corresponding stressed columns can be obtained via `get_specs`.

Use `get_weights` if the SWIM object only contains scenario weights and not a list of functions.

get\_specs: A data.frame containing specifications of the stresses with each row corresponding to a different stress. Only a selection of the specifications is returned; however, all input variables are stored in the object. See also [SWIM](#).

## Functions

- get\_data: extracting data.
- get\_weights: extracting scenario weights.
- get\_weightsfun: extracting weight functions.
- get\_specs: extracting information of the stress.

## Author(s)

Silvana M. Pesenti

## See Also

[SWIM](#)

## Examples

```
## continuing example in stress_VaR
set.seed(0)
x <- as.data.frame(cbind(
  "normal" = rnorm(1000),
  "gamma" = rgamma(1000, shape = 2)))
res1 <- stress(type = "VaR", x = x,
  alpha = 0.9, q_ratio = 1.05, k = 1)

## returning the underlying data
all(get_data(res1) == x)
## the scenario weights
w <- get_weights(res1)
get_weightsfun(res1)
get_specs(res1)

## now add a stress on the means of both variables
res1 <- stress(type = "mean", x = res1, k = 1:2, new_means = c(0.5,1.5))
get_specs(res1)
## the required moments for a stress of type "mean" are not displayed
## the type of stress and the specs for the second stress can be
## extracted directly from the SWIM object.
res1$type[[2]]
res1$specs[[2]]
```

---

importance\_rank      *Importance Ranking for a Stressed Model*

---

### Description

Provides the importance ranks of the components (random variables) of a stressed model for different sensitivity measures.

### Usage

```
importance_rank(
  object,
  xCol = "all",
  wCol = "all",
  type = c("Gamma", "Wasserstein", "all"),
  f = NULL,
  k = NULL
)
```

### Arguments

object	A SWIM object.
xCol	Numeric or character vector, (names of) the columns of the underlying data of the object (default = "all"). If xCol = NULL, only the transformed data $f(x)$ is considered.
wCol	Vector, the columns of the scenario weights of the object corresponding to different stresses (default = "all").
type	Character, one of "Gamma", "Wasserstein", "all".
f	A function, or list of functions, that, applied to $x$ , constitute the transformation of the data for which the sensitivity is calculated.
k	A vector or list of vectors, same length as $f$ , indicating which columns of $x$ each function in $f$ operates on. When $f$ is a list, $k[[i]]$ corresponds to the input variables of $f[[i]]$ .

### Details

For the definition of the sensitivity measures (type), see [sensitivity](#).

### Value

A data.frame containing the importance ranks of the stressed model for different sensitivity measures. Small values correspond to large sensitivities. Different rows correspond to different random variables. The first two rows specify the stress and type of the sensitivity measure on which the ranking is calculated.



**See Also**

See [sensitivity](#) for the values of the sensitivity measures, [plot\\_sensitivity](#) for plotting sensitivity measures and [summary](#) for a summary statistic of a stressed model.

**Examples**

```
## example with a stress on VaR
set.seed(0)
x <- as.data.frame(cbind(
  "log-normal" = rlnorm(1000),
  "gamma" = rgamma(1000, shape = 2)))
res1 <- stress(type = "VaR", x = x,
  alpha = c(0.9, 0.95), q_ratio = 1.05)

importance_rank(res1, wCol = 1:2, type = "Gamma")
## sensitivity of log-transformed data
importance_rank(res1, wCol = 1, type = "all",
  f = list(function(x)log(x), function(x)log(x)), k = list(1, 2))
```

---

merge.SWIM

*Merging Two Stressed Models*

---

**Description**

This function is a method for an object of class SWIM.

**Usage**

```
## S3 method for class 'SWIM'
merge(x, y, ...)
```

**Arguments**

x, y                    Objects of class SWIM.  
...                    Additional arguments will be ignored.

**Details**

Merges two objects of class SWIM, that are based on the same data.

**Value**

An object of class SWIM containing:

- x, a data.frame containing the data;

- `new_weights`, a list, each component corresponds to a different stress and is either a vector of scenario weights or a function, that applied to a column of `x`, generates the vectors of scenario weights;
- `type`, a list, each component corresponds to a different stress and specifies the type of the stress;
- `specs`, a list, each component corresponds to a different stress and contains a list with the specifications of what has been stressed.

See [SWIM](#) for details.

### Author(s)

Silvana M. Pesenti

---

plot\_cdf

*Plotting the Empirical Distribution Functions of a Stressed Model*

---

### Description

Plots the empirical distribution function of a stressed model component (random variable) under the scenario weights.

### Usage

```
plot_cdf(
  object,
  xCol = 1,
  wCol = "all",
  base = FALSE,
  n = 500,
  x_limits,
  y_limits,
  displ = TRUE
)
```

### Arguments

<code>object</code>	A SWIM object.
<code>xCol</code>	Numeric or character, (name of) the column of the underlying data of the object (default = 1).
<code>wCol</code>	Vector, the columns of the scenario weights of the object corresponding to different stresses (default = "all").
<code>base</code>	Logical, if TRUE, statistics under the baseline are also returned (default = "FALSE").
<code>n</code>	Integer, the number of points used to plot <code>stat_ecdf</code> in <code>ggplot</code> (default = 500).

x_limits	Vector, the limits of the x-axis of the plot, the value for xlim in the coord_cartesian function in ggplot.
y_limits	Vector, the limits of the y-axis of the plot, the value for ylim in the coord_cartesian function in ggplot.
displ	Logical, if TRUE the plot is displayed, otherwise the data.frame for customised plotting with ggplot is returned (default = TRUE).

### Value

If displ = TRUE, a plot displaying the empirical distribution function of the stochastic model under the scenario weights.

If displ = FALSE, a data.frame for customised plotting with ggplot. The data.frame contains the columns: the column, xCol, of the data of the stressed model, stress (the stresses) and value (the values).

Denote by res the return of the function call, then ggplot can be called via:

```
ggplot(res, aes(x = res[,1], w = value))
+stat_ecdf(aes(color = factor(stress)), n = n).
```

Note that the ggplot2 default of stat\_ecdf does not take weight as an aesthetic. We use the workaround by Nicolas Woloszko, see Note below.

### Note

This function is based on the ggplot stat\_ecdf function. However, the stat\_ecdf does not allow for specifying weights, thus the function is based on the workaround by Nicolas Woloszko, see [https://github.com/NicolasWoloszko/stat\\_ecdf\\_weighted](https://github.com/NicolasWoloszko/stat_ecdf_weighted).

### See Also

See [cdf](#) for the empirical distribution function of a stressed model and [quantile\\_stressed](#) for sample quantiles of a stressed model.

### Examples

```
## example with a stress on VaR
set.seed(0)
x <- as.data.frame(cbind(
  "normal" = rnorm(10 ^ 5),
  "gamma" = rgamma(10 ^ 5, shape = 2)))
res1 <- stress(type = "VaR", x = x,
  alpha = c(0.75, 0.95), q_ratio = 1.15)
plot_cdf(res1, xCol = 1, wCol = 1:2, base = TRUE)
plot_cdf(res1, xCol = 1, wCol = 1:2, base = TRUE,
  x_limits = c(0, 5), y_limits = c(0.5, 1))
```

plot\_hist

*Plotting Histograms of a Stressed Model***Description**

Plots the histogram of a stressed model component (random variable) under the scenario weights.

**Usage**

```
plot_hist(
  object,
  xCol = 1,
  wCol = "all",
  base = FALSE,
  x_limits,
  displ = TRUE,
  binwidth,
  displLines = FALSE
)
```

**Arguments**

object	A SWIM object.
xCol	Numeric or character, (name of) the column of the underlying data of the object (default = 1).
wCol	Vector, the columns of the scenario weights of the object corresponding to different stresses (default = "all").
base	Logical, if TRUE, statistics under the baseline are also returned (default = "FALSE").
x_limits	Vector, the limits of the x-axis of the plot, the value for xlim in the coord_cartesian function in ggplot.
displ	Logical, if TRUE the plot is displayed, otherwise the data.frame for customised plotting with ggplot is returned (default = TRUE).
binwidth	Numeric, the width of the bins used to plot the histogram, the binwidth in the geom_freqpoly function in ggplot (default corresponds to 30 bins).
displLines	Logical, if TRUE lines are displayed instead of bins (default = FALSE).

**Value**

If displ = TRUE, a histogram of the stochastic model under the scenario weights.

If displ = FALSE, a data.frame for customised plotting with ggplot. The data.frame contains the columns: the column, xCol, of the data of the stressed model, stress (the stresses) and value (the values).

Denote by res the return of the function call, then ggplot can be called via:

```
ggplot(res, aes(x = res[, 1], y = ..density.., weight = value))
+geom_freqpoly(binwidth, aes(color = factor(stress)))
```

**See Also**

See [cdf](#) and [plot\\_cdf](#) for values and plotting of the empirical distribution function of a stressed model, respectively, and [quantile\\_stressed](#) for sample quantiles of a stressed model.

**Examples**

```
## example with a stress on VaR
set.seed(0)
x <- data.frame("gamma" = rgamma(10^5, shape = 2))
res1 <- stress(type = "VaR", x = x,
  alpha = c(0.75, 0.95), q_ratio = 1.1)
plot_hist(res1, xCol = "gamma", wCol = 1:2, base = TRUE, binwidth = 0.4)
plot_hist(res1, xCol = "gamma", wCol = 1:2, base = TRUE, binwidth = 0.4, displLines = TRUE)
```

---

plot\_quantile

*Plotting Quantile Functions of a Stressed Model*


---

**Description**

Plots the empirical quantile function of a stressed model component (random variable) under the scenario weights.

**Usage**

```
plot_quantile(
  object,
  xCol = 1,
  wCol = "all",
  base = FALSE,
  n = 500,
  x_limits,
  y_limits,
  displ = TRUE
)
```

**Arguments**

object	A SWIM object.
xCol	Numeric or character, (name of) the column of the underlying data of the object (default = 1).
wCol	Vector, the columns of the scenario weights of the object corresponding to different stresses (default = "all").
base	Logical, if TRUE, statistics under the baseline are also returned (default = "FALSE").
n	Integer, the number of points used to plot (default = 500).

x_limits	Vector, the limits of the x-axis of the plot, the value for xlim in the coord_cartesian function in ggplot.
y_limits	Vector, the limits of the y-axis of the plot, the value for ylim in the coord_cartesian function in ggplot.
displ	Logical, if TRUE the plot is displayed, otherwise the data.frame for customised plotting with ggplot is returned (default = TRUE).

### Value

If displ = TRUE, a plot displaying the empirical quantile function of the stochastic model under the scenario weights.

If displ = FALSE, a data.frame for customised plotting with ggplot. The data.frame contains the following columns: grid, the grid points to plot the quantiles, stress (the stresses) and value (the quantile values).

Denote by res the return of the function call, then ggplot can be called via:

```
ggplot(res, aes(x = res[,1], y = value))
+geom_lines(aes(color = factor(stress))).
```

### See Also

See [quantile\\_stressed](#) for sample quantiles of a stressed model and [plot\\_cdf](#) for plotting empirical distribution functions under scenario weights.

### Examples

```
## example with a stress on VaR
set.seed(0)
x <- as.data.frame(cbind(
  "normal" = rnorm(10 ^ 5),
  "gamma" = rgamma(10 ^ 5, shape = 2)))
res1 <- stress(type = "VaR", x = x,
  alpha = c(0.75, 0.95), q_ratio = 1.15)
plot_quantile(res1, xCol = 1, wCol = 1:2, base = TRUE)
plot_quantile(res1, xCol = 1, wCol = 1:2, base = TRUE, x_limits = c(0.8, 1),
  y_limits = c(0, 5))
```

### Description

Plots the sensitivity measures for components (random variables) of a stochastic model under the scenario weights.

**Usage**

```
plot_sensitivity(
  object,
  xCol = "all",
  wCol = "all",
  type = c("Gamma", "Kolmogorov", "Wasserstein"),
  f = NULL,
  k = NULL,
  displ = TRUE
)
```

**Arguments**

object	A SWIM object.
xCol	Numeric or character vector, (names of) the columns of the underlying data of the object (default = "all"). If xCol = NULL, only the transformed data $f(x)$ is considered.
wCol	Vector, the columns of the scenario weights of the object corresponding to different stresses (default = "all").
type	Character, one of "Gamma", "Kolmogorov", "Wasserstein", "all".
f	A function, or list of functions, that, applied to $x$ , constitute the transformation of the data for which the sensitivity is calculated.
k	A vector or list of vectors, same length as $f$ , indicating which columns of $x$ each function in $f$ operates on. When $f$ is a list, $k[[i]]$ corresponds to the input variables of $f[[i]]$ .
displ	Logical, if TRUE the plot is displayed, otherwise the data.frame for customised plotting with ggplot is returned (default = TRUE).

**Details**

For the definition of the sensitivity measures (type), see [sensitivity](#).

Note that the Kolmogorov distance is the same for all inputs under the same stress. Thus, it should only be used to compare different stresses, not individual components.

**Value**

If displ = TRUE, a plot displaying the sensitivity measures of the stochastic model under the scenario weights.

If displ = FALSE, a data.frame for customised plotting with ggplot. The data.frame contains the columns: stress (the stresses), type (the types of sensitivity), X\_all (the random variables), value (the values of the sensitivities).

Denote by result the return of the function call, then ggplot can be called via:

$$ggplot(result, aes(x = X_{all}, y = value))$$

$$+geom_{point}(aes(color = factor(stress), shape = type)).$$

**See Also**

See [sensitivity](#) for the values of the sensitivity measures of a stressed model and [importance\\_rank](#) for ranking of random variables according to their sensitivities.

**Examples**

```
## Consider the portfolio Y = X1 + X2 + X3 + X4 + X5,
## where (X1, X2, X3, X4, X5) are correlated normally
## distributed with equal mean and different standard deviations,
## see the README for further details.

set.seed(0)
SD <- c(70, 45, 50, 60, 75)
Corr <- matrix(rep(0.5, 5 ^ 2), nrow = 5) + diag(rep(1 - 0.5, 5))
if (!requireNamespace("mvtnorm", quietly = TRUE))
  stop("Package \"mvtnorm\" needed for this function
  to work. Please install it.")
x <- mvtnorm::rmvnorm(10 ^ 5,
  mean = rep(100, 5),
  sigma = (SD %*% t(SD)) * Corr)
data <- data.frame(rowSums(x), x)
names(data) <- c("Y", "X1", "X2", "X3", "X4", "X5")
rev.stress <- stress(type = "VaR", x = data,
  alpha = c(0.75, 0.9), q_ratio = 1.1, k = 1)

sensitivity(rev.stress, type = "all")
plot_sensitivity(rev.stress, xCol = 2:6, type = "Gamma")
plot_sensitivity(rev.stress, xCol = 6, wCol = 1, type = "all")
```

---

plot\_weights

*Plotting the scenario weights of a Stressed Model*

---

**Description**

Plots the scenario weights of a stressed model against a model component.

**Usage**

```
plot_weights(
  object,
  xCol = 1,
  wCol = "all",
  n,
  x_limits,
  y_limits,
  displ = TRUE
)
```



**Arguments**

object	A SWIM object.
xCol	Numeric or character, (name of) the column of the underlying data of the object (default = 1).
wCol	Vector, the columns of the scenario weights of the object corresponding to different stresses (default = "all").
n	Integer, the number of points used to plot (default = 5000 or the minimum of the data). If n = "all", all data points are plotted. If n is a subset of points, the plotted scenario weights are chosen in an equidistant way.
x_limits	Vector, the limits of the x-axis of the plot, the value for xlim in the coord_cartesian function in ggplot.
y_limits	Vector, the limits of the y-axis of the plot, the value for ylim in the coord_cartesian function in ggplot.
displ	Logical, if TRUE the plot is displayed, otherwise the data.frame for customised plotting with ggplot is returned (default = TRUE).

**Value**

If displ = TRUE, a plot displaying the scenario weights of a stochastic model against a model component.

If displ = FALSE, a data.frame for customised plotting with ggplot. The data.frame contains the following columns: grid, the grid points to plot the quantiles, stress (the stresses) and value (the quantile values).

Denote by res the return of the function call, then ggplot can be called via:

```
ggplot(res, aes(x = res[,1], y = value))
+geom_lines(aes(color = factor(stress))).
```

**See Also**

See [plot\\_quantile](#) for plotting sample quantiles of a stressed model and [plot\\_cdf](#) for plotting empirical distribution functions.

**Examples**

```
## example with a stress with \code{credit_data} data set:
data("credit_data")
## two stresses in VaR
model_stress <- stress_VaR(credit_data, alpha = c(0.9, 0.95), q_ratio = 1.1, k = 1)
plot_weights(model_stress, xCol = "L", wCol = 1:2)

## additional stress on VaR and ES

model_stress <- stress_VaR_ES(model_stress, alpha = 0.9, q_ratio = 1.1, s_ratio = 1.2, k = 1)
plot_weights(model_stress, xCol = "L", wCol = "all", n = 1000, x_limits = c(0, 3500),
             y_limits = c(0, 10))
```

---

quantile\_stressed      *Sample Quantiles of a Stressed Model*

---

## Description

Provides sample quantiles for components (random variables) of a stochastic model, corresponding to distribution functions under the scenario weights.

## Usage

```
quantile_stressed(
  object,
  probs = seq(0, 1, 0.25),
  xCol = "all",
  wCol = 1,
  type = c("quantile", "(i-1)/(n-1)", "i/(n+1)", "i/n")
)
```

## Arguments

object	A SWIM object.
probs	Vector of probabilities with values in $[0, 1]$ (default = $(0, 0.25, 0.5, 0.75, 1)$ ).
xCol	Numeric or character vector, (names of) the columns of the underlying data of the object (default = "all").
wCol	Numeric, the column of the scenario weights of the object (default = 1).
type	Character, one of "quantile", "(i-1)/(n-1)", "i/(n+1)", "i/n", with default = "quantile". See details below.

## Details

type defines the choice of algorithm used for calculating the estimate of the sample quantiles.

"quantile" corresponds to the default interpolation used in [quantile](#). Further options are "(i-1)/(n-1)", "i/(n+1)", "i/" the inverse of the empirical distribution function, using, respectively,  $(wt - 1)/T$ ,  $wt/(T+1)$ ,  $wt/T$ , where wt is the cumulative weight and T the total weight (usually total sample size). See [wtd.quantile](#) for further details on type, on which quantile\_stressed is based.

## Value

Returns a matrix with estimates of the distribution quantiles at the probabilities, probs, under the scenario weights wCol.

## See Also

See [wtd.quantile](#) on which the function quantile\_stressed is based.  
See [cdf](#) for the empirical distribution function of a stressed model.

**Examples**

```
## example with a stress on VaR
set.seed(0)
x <- as.data.frame(cbind(
  "normal" = rnorm(1000),
  "gamma" = rgamma(1000, shape = 2)))
res1 <- stress(type = "VaR", x = x,
  alpha = c(0.9, 0.95), q_ratio = 1.05)
## stressed sample quantiles
quantile_stressed(res1, probs = seq(0.9, 0.99, 0.01), wCol = 2)
```

---

sensitivity

*Sensitivities of a Stressed Model*


---

**Description**

Provides different sensitivity measures that compare the stressed and the baseline model.

**Usage**

```
sensitivity(
  object,
  xCol = "all",
  wCol = "all",
  type = c("Gamma", "Kolmogorov", "Wasserstein", "all"),
  f = NULL,
  k = NULL
)
```

**Arguments**

object	A SWIM object.
xCol	Numeric or character vector, (names of) the columns of the underlying data of the object (default = "all"). If xCol = NULL, only the transformed data $f(x)$ is considered.
wCol	Vector, the columns of the scenario weights of the object corresponding to different stresses (default = "all").
type	Character, one of "Gamma", "Kolmogorov", "Wasserstein", "all".
f	A function, or list of functions, that, applied to x, constitute the transformation of the data for which the sensitivity is calculated.
k	A vector or list of vectors, same length as f, indicating which columns of x each function in f operates on. When f is a list, $k[[i]]$ corresponds to the input variables of $f[[i]]$ .

## Details

Provides sensitivity measures that compare the stressed and the baseline model. Implemented sensitivity measures:

1. Gamma, the *Reverse Sensitivity Measure*, defined for a random variable  $Y$  and scenario weights  $w$  by

$$Gamma = (E(Y * w) - E(Y))/c,$$

where  $c$  is a normalisation constant such that  $|Gamma| \leq 1$ , see (Pesenti et al. 2019). Loosely speaking, the Reverse Sensitivity Measure is the normalised difference between the first moment of the stressed and the baseline distributions of  $Y$ .

2. Kolmogorov, the Kolmogorov distance, defined for distribution functions  $F, G$  by

$$Kolmogorov = \sup|F(x) - G(x)|.$$

3. Wasserstein, the Wasserstein distance of order 1, defined for two distribution functions  $F, G$  by

$$Wasserstein = \int |F(x) - G(x)|dx.$$

If  $f$  and  $k$  are provided, the sensitivity of the transformed data is returned.

## Value

A data.frame containing the sensitivity measures of the stressed model with rows corresponding to different random variables. The first two rows specify the stress and type of the sensitivity measure.

## References

Pesenti SM, Millosovich P, Tsanakas A (2019). “Reverse sensitivity testing: What does it take to break the model?” *European Journal of Operational Research*, **274**(2), 654–670.

## See Also

See [importance\\_rank](#) for ranking of random variables according to their sensitivities, [plot\\_sensitivity](#) for plotting sensitivity measures and [summary](#) for summary statistics of a stressed model.

## Examples

```
## example with a stress on VaR
set.seed(0)
x <- as.data.frame(cbind(
  "log-normal" = rlnorm(1000),
  "gamma" = rgamma(1000, shape = 2)))
res1 <- stress(type = "VaR", x = x,
  alpha = c(0.9, 0.95), q_ratio = 1.05)

sensitivity(res1, wCol = 1, type = "all")
## sensitivity of log-transformed data
sensitivity(res1, wCol = 1, type = "all",
```

```

f = list(function(x)log(x), function(x)log(x)), k = list(1,2))

## Consider the portfolio Y = X1 + X2 + X3 + X4 + X5,
## where (X1, X2, X3, X4, X5) are correlated normally
## distributed with equal mean and different standard deviations,
## see the README for further details.

## Not run:
set.seed(0)
SD <- c(70, 45, 50, 60, 75)
Corr <- matrix(rep(0.5, 5 ^ 2), nrow = 5) + diag(rep(1 - 0.5, 5))
if (!requireNamespace("mvtnorm", quietly = TRUE))
  stop("Package \"mvtnorm\" needed for this function
  to work. Please install it.")
x <- mvtnorm::rmvnorm(10 ^ 5,
  mean = rep(100, 5),
  sigma = (SD %*% t(SD)) * Corr)
data <- data.frame(rowSums(x), x)
names(data) <- c("Y", "X1", "X2", "X3", "X4", "X5")
rev.stress <- stress(type = "VaR", x = data,
  alpha = c(0.75, 0.9), q_ratio = 1.1, k = 1)

sensitivity(rev.stress, type = "all")
## sensitivity to sub-portfolios X1 + X2 and X3 + X4
sensitivity(rev.stress, xCol = NULL, type = "Gamma",
  f = rep(list(function(x)x[1] + x[2]), 2), k = list(c(2, 3), c(4, 5)))
plot_sensitivity(rev.stress, xCol = 2:6, type = "Gamma")
importance_rank(rev.stress, xCol = 2:6, type = "Gamma")

## End(Not run)

```

---

stress

*Stressing Random Variables*


---

## Description

Provides weights on simulated scenarios from a baseline stochastic model, such that stressed random variables fulfil given probabilistic constraints (e.g. specified values for risk measures), under the new scenario weights. Scenario weights are selected by constrained minimisation of the relative entropy to the baseline model.

## Usage

```

stress(
  type = c("VaR", "VaR ES", "mean", "mean sd", "moment", "prob", "user"),
  ...
)

```

### Arguments

type           Type of stress, one of "VaR", "VaR ES", "mean", "mean sd", "moment", "prob", "user".  
...            Arguments to be passed on, depending on type.

### Value

An object of class SWIM, see [SWIM](#) for details.

### Author(s)

Silvana M. Pesenti

### References

Pesenti SM, Millosovich P, Tsanakas A (2019). "Reverse sensitivity testing: What does it take to break the model?" *European Journal of Operational Research*, **274**(2), 654–670.

Pesenti Silvana M.and Bettini A, Millosovich P, Tsanakas A (2020). "Scenario Weights for Importance Measurement (SWIM) - An R package for sensitivity analysis." *Available at SSRN*: <https://www.ssrn.com/abstract=3515274>.

Csiszar I (1975). "I-divergence geometry of probability distributions and minimization problems." *The Annals of Probability*, 146–158.

### See Also

Other stress functions: [stress\\_VaR\\_ES\(\)](#), [stress\\_VaR\(\)](#), [stress\\_mean\\_sd\(\)](#), [stress\\_mean\(\)](#), [stress\\_moment\(\)](#), [stress\\_prob\(\)](#), [stress\\_user\(\)](#)

### Examples

```
set.seed(0)
x <- as.data.frame(cbind(
  "normal" = rnorm(1000),
  "gamma" = rgamma(1000, shape = 2)))
res <- stress(type = "VaR", x = x,
  alpha = 0.9, q_ratio = 1.05)
summary(res)
```

---

stress_mean	<i>Stressing Means</i>
-------------	------------------------

---

### Description

Provides weights on simulated scenarios from a baseline stochastic model, such that stressed model components (random variables) fulfil the mean constraints. Scenario weights are selected by constrained minimisation of the relative entropy to the baseline model.

### Usage

```
stress_mean(x, k, new_means, normalise = FALSE, ...)
```

### Arguments

x	A vector, matrix or data frame containing realisations of random variables. Columns of x correspond to random variables; OR A SWIM object, where x corresponds to the underlying data of the SWIM object.
k	Numeric vector, the columns of x that are stressed.
new_means	Numeric vector, same length as k, containing the stressed means.
normalise	Logical. If true, values of $f(x)$ are linearly scaled to the unit interval.
...	Additional arguments to be passed to <a href="#">nleqslv</a> or <a href="#">stress_moment</a> .

### Details

The function `stress_mean` is a wrapper for the function `stress_moment`. See [stress\\_moment](#) for details on the additional arguments to `...` and the underlying algorithm.

### Value

A SWIM object containing:

- x, a data.frame containing the data;
- new\_weights, a list, each component corresponds to a different stress and is a vector of scenario weights;
- type = "mean";
- specs, a list, each component corresponds to a different stress and contains k and new\_means.

See [SWIM](#) for details.

## References

Pesenti SM, Millosovich P, Tsanakas A (2019). “Reverse sensitivity testing: What does it take to break the model?” *European Journal of Operational Research*, **274**(2), 654–670.

Pesenti Silvana M.and Bettini A, Millosovich P, Tsanakas A (2020). “Scenario Weights for Importance Measurement (SWIM) - An R package for sensitivity analysis.” *Available at SSRN*: <https://www.ssrn.com/abstract=3515274>.

Csiszar I (1975). “I-divergence geometry of probability distributions and minimization problems.” *The Annals of Probability*, 146–158.

## See Also

See [stress\\_mean\\_sd](#) for stressing means and standard deviations jointly, and [stress\\_moment](#) for moment constraints.

Other stress functions: [stress\\_VaR\\_ES\(\)](#), [stress\\_VaR\(\)](#), [stress\\_mean\\_sd\(\)](#), [stress\\_moment\(\)](#), [stress\\_prob\(\)](#), [stress\\_user\(\)](#), [stress\(\)](#)

## Examples

```
set.seed(0)
x <- data.frame(cbind(
  "normal" = rnorm(1000),
  "gamma" = rgamma(1000, shape = 2),
  "beta" = rbeta(1000, shape1 = 2, shape2 = 2)))
## stressing means
res1 <- stress(type = "mean", x = x, k = 1:3,
  new_means = c(1, 1, 0.75))
summary(res1)
res1$specs
## calling stress_mean directly
res2 <- stress_mean(x = x, k = 1:3,
  new_means = c(1, 1, 0.75))
summary(res2)

## See also examples in stress_moment and stress_mean_sd.
```

---

stress\_mean\_sd

*Stressing Mean and Standard Deviation*

---

## Description

Provides weights on simulated scenarios from a baseline stochastic model, such that stressed model components (random variables) fulfil the mean and standard deviation constraints. Scenario weights are selected by constrained minimisation of the relative entropy to the baseline model.



**Usage**

```
stress_mean_sd(x, k, new_means, new_sd, normalise = FALSE, ...)
```

**Arguments**

x	A vector, matrix or data frame containing realisations of random variables. Columns of x correspond to random variables; OR A SWIM object, where x corresponds to the underlying data of the SWIM object.
k	Numeric vector, the columns of x that are stressed.
new_means	Numeric vector, same length as k, containing the stressed means.
new_sd	Numeric vector, same length as k, containing the stressed standard deviations.
normalise	Logical. If true, values of $f(x)$ are linearly scaled to the unit interval.
...	Additional arguments to be passed to <a href="#">nleqslv</a> or <a href="#">stress_moment</a> .

**Details**

The function `stress_mean_sd` is a wrapper for the function `stress_moment`. See [stress\\_moment](#) for details on the additional arguments to `...` and the underlying algorithm.

For stressing means only, see [stress\\_mean](#), for stressing higher moments and functions of moments, see [stress\\_moment](#).

**Value**

A SWIM object containing:

- x, a data.frame containing the data;
- new\_weights, a list, each component corresponds to a different stress and is a vector of scenario weights;
- type = "mean";
- specs, a list, each component corresponds to a different stress and contains k, new\_means and new\_sd.

See [SWIM](#) for details.

**References**

Pesenti SM, Millosovich P, Tsanakas A (2019). "Reverse sensitivity testing: What does it take to break the model?" *European Journal of Operational Research*, **274**(2), 654–670.

Pesenti Silvana M. and Bettini A, Millosovich P, Tsanakas A (2020). "Scenario Weights for Importance Measurement (SWIM) - An R package for sensitivity analysis." *Available at SSRN*: <https://www.ssrn.com/abstract=3515274>.

Csiszar I (1975). "I-divergence geometry of probability distributions and minimization problems." *The Annals of Probability*, 146–158.

**See Also**

Other stress functions: [stress\\_VaR\\_ES\(\)](#), [stress\\_VaR\(\)](#), [stress\\_mean\(\)](#), [stress\\_moment\(\)](#), [stress\\_prob\(\)](#), [stress\\_user\(\)](#), [stress\(\)](#)

**Examples**

```
set.seed(0)
x <- data.frame(cbind(
  "normal" = rnorm(1000),
  "gamma" = rgamma(1000, shape = 2),
  "beta" = rbeta(1000, shape1 = 2, shape2 = 2)))
## stressing mean and sd of column 1
res1 <- stress(type = "mean sd", x = x, k = 1, new_means = 0.1,
  new_sd = 1.1, method = "Newton",
  control = list(maxit = 1000, ftol = 1E-15))
summary(res1)
## calling stress_mean_sd directly
res2 <- stress_mean_sd(x = x, k = 1, new_means = 0.1,
  new_sd = 1.1, method = "Newton",
  control = list(maxit = 1000, ftol = 1E-15))

## See also examples in stress_moment.
```

---

stress\_moment

*Stressing Moments*

---

**Description**

Provides weights on simulated scenarios from a baseline stochastic model, such that stressed model components (random variables) fulfill the moment constraints. Scenario weights are selected by constrained minimisation of the relative entropy to the baseline model.

**Usage**

```
stress_moment(x, f, k, m, normalise = FALSE, show = FALSE, ...)
```

**Arguments**

x	A vector, matrix or data frame containing realisations of random variables. Columns of x correspond to random variables; OR A SWIM object, where x corresponds to the underlying data of the SWIM object.
f	A function, or list of functions, that, applied to x, constitute the moment constraints.
k	A vector or list of vectors, same length as f, indicating which columns of x each function in f operates on. When f is a list, k[[i]] corresponds to the input variables of f[[i]].

m	Numeric vector, same length as f, containing the stressed moments of f(x). Must be in the range of f(x).
normalise	Logical. If true, values of f(x) are linearly scaled to the unit interval.
show	Logical. If true, print the result of the call to <code>nleqslv</code> .
...	Additional arguments to be passed to <code>nleqslv</code> .

### Details

The moment constraints are given by  $E^Q(f(x)) = m$ , where  $E^Q$  denotes the expectation under the stressed model. `stress_moment` solves the subsequent set of equations with respect to theta, using `nleqslv` from package `nleqslv`:

$$E^Q(f(x)) = E(f(x) * \exp(\theta * f(x))) = m.$$

There is no guarantee that the set of equations has a solution, or that the solution is unique. SWIM will return a warning if the termination code provided by `nleqslv` is different from 1 (convergence has been achieved). It is recommended to check the result of the call to `nleqslv` using the "show" argument. The user is referred to the `nleqslv` documentation for further details.

Normalising the data may help avoiding numerical issues when the range of values is wide.

### Value

A SWIM object containing:

- x, a data.frame containing the data;
- new\_weights, a list, each component corresponds to a different stress and is a vector of scenario weights;
- type = "moment";
- specs, a list, each component corresponds to a different stress and contains f, k and m.

See `SWIM` for details.

The function call will print a message containing the termination code returned by the call to `nleqslv` and a table with the required and achieved moment, and the absolute and relative error.

### References

Pesenti SM, Millosovich P, Tsanakas A (2019). "Reverse sensitivity testing: What does it take to break the model?" *European Journal of Operational Research*, **274**(2), 654–670.

Pesenti Silvana M. and Bettini A, Millosovich P, Tsanakas A (2020). "Scenario Weights for Importance Measurement (SWIM) - An R package for sensitivity analysis." *Available at SSRN*: <https://www.ssrn.com/abstract=3515274>.

Csiszar I (1975). "I-divergence geometry of probability distributions and minimization problems." *The Annals of Probability*, 146–158.

**See Also**

See [stress\\_mean](#) for stressing means and [stress\\_mean\\_sd](#) for stressing mean and standard deviation jointly.

Other stress functions: [stress\\_VaR\\_ES\(\)](#), [stress\\_VaR\(\)](#), [stress\\_mean\\_sd\(\)](#), [stress\\_mean\(\)](#), [stress\\_prob\(\)](#), [stress\\_user\(\)](#), [stress\(\)](#)

**Examples**

```
set.seed(0)
x <- data.frame(cbind(
  "normal" = rnorm(1000),
  "gamma" = rgamma(1000, shape = 2),
  "beta" = rbeta(1000, shape1 = 2, shape2 = 2)))

## stressing covariance of columns 1, 2 while leaving the means unchanged
res1 <- stress_moment(x = x,
  f = list(function(x)x, function(x)x, function(x)x[1] * x[2]),
  k = list(1, 2, c(1, 2)), m = c(0, 2, 0.5),
  method = "Newton", control = list(maxit = 1000, ftol = 1E-10))
## means under the stressed model
summary(res1)
apply(x, 2, stats::weighted.mean, w = get_weights(res1))
## covariance of columns 1,2 under the stressed model
stats::weighted.mean(x[, 1] * x[, 2], w = get_weights(res1))

## stressing jointly the tail probabilities of columns 1, 3
res2 <- stress_moment(x = x,
  f = list(function(x)(x > 1.5), function(x)(x > 0.9)),
  k = list(1, 3), m = c(0.9, 0.9))
summary(res2)
## probabilities under the stressed model
mean((x[, 1] > 1.5) * get_weights(res2))
mean((x[, 3] > 0.9) * get_weights(res2))
```

---

stress\_prob

*Stressing Intervals*

---

**Description**

Provides weights on simulated scenarios from a baseline stochastic model, such that a stressed model component (random variable) fulfils constraints on probability of disjoint intervals. Scenario weights are selected by constrained minimisation of the relative entropy to the baseline model.

**Usage**

```
stress_prob(x, prob, lower = NULL, upper, k = 1)
```

**Arguments**

x	A vector, matrix or data frame containing realisations of random variables. Columns of x correspond to random variables; OR A SWIM object, where x corresponds to the underlying data of the SWIM object.
prob	Numeric vector, stressed probabilities corresponding to the intervals defined through lower and upper.
lower	Numeric vector, left endpoints of the intervals.
upper	Numeric vector, right endpoints of the intervals.
k	Numeric, the column of x that is stressed (default = 1).

**Details**

The intervals are treated as half open intervals, that is the lower endpoint are not included, whereas the upper endpoint are included. If upper = NULL, the intervals are consecutive and prob cumulative.

The intervals defined through lower and upper must be disjoint.

**Value**

A SWIM object containing:

- x, a data.frame containing the data;
- new\_weights, a list of functions, that applied to the kth column of x, generate the vectors of scenario weights. Each component corresponds to a different stress;
- type = "prob";
- specs, a list, each component corresponds to a different stress and contains k, lower, upper and prob.

See [SWIM](#) for details.

**Author(s)**

Silvana M. Pesenti

**References**

Pesenti SM, Millosovich P, Tsanakas A (2019). "Reverse sensitivity testing: What does it take to break the model?" *European Journal of Operational Research*, **274**(2), 654–670.

Pesenti Silvana M.and Bettini A, Millosovich P, Tsanakas A (2020). "Scenario Weights for Importance Measurement (SWIM) - An R package for sensitivity analysis." *Available at SSRN: <https://www.ssrn.com/abstract=3515274>*.

Csiszar I (1975). "I-divergence geometry of probability distributions and minimization problems." *The Annals of Probability*, 146–158.

**See Also**

Other stress functions: [stress\\_VaR\\_ES\(\)](#), [stress\\_VaR\(\)](#), [stress\\_mean\\_sd\(\)](#), [stress\\_mean\(\)](#), [stress\\_moment\(\)](#), [stress\\_user\(\)](#), [stress\(\)](#)

**Examples**

```
set.seed(0)
x <- rnorm(1000)
## consecutive intervals
res1 <- stress(type = "prob", x = x, prob = 0.008, upper = -2.4)
# probability under the stressed model
cdf(res1, xCol = 1)(-2.4)

## calling stress_prob directly
## multiple intervals
res2 <- stress_prob(x = x, prob = c(0.008, 0.06),
  lower = c(-3, -2), upper = c(-2.4, -1.6))
# probability under the stressed model
cdf(res2, xCol = 1)(c(-2.4, -1.6)) - cdf(res2, xCol = 1)(c(-3, -2))
```

---

stress\_user

*User Defined Stress*

---

**Description**

Returns a SWIM object with scenario weights defined by the user.

**Usage**

```
stress_user(x, new_weights = NULL, new_weightsfun = NULL, k = 1)
```

**Arguments**

x	A vector, matrix or data frame containing realisations of random variables. Columns of x correspond to random variables; OR A SWIM object, where x corresponds to the underlying data of the SWIM object.
new_weights	A vector, matrix or data frame containing scenario weights. Columns of new_weights correspond to different stresses. new_weights are normalised to have a mean of 1.
new_weightsfun	A list of functions, that applied to the kth column of x generate the vectors of the scenario weights. Each function corresponds to a stress. The weights generated for each stress are normalised to have a mean of 1.
k	Numeric, the column of x that is stressed (default = 1).

**Value**

A SWIM object containing:

- `x`, a `data.frame` containing the data;
- `new_weights`, a list, each component corresponds to a different stress and is either a vector of scenario weights (if `new_weights` is provided) or (if `new_weightsfun` is provided) a function, that applied to the `k`th column of `x`, generates the vectors of scenario weights;
- `type = "user"`;
- `specs`, a list, each component corresponds to a different stress and contains `k`.

See [SWIM](#) for details.

**References**

Pesenti SM, Millosovich P, Tsanakas A (2019). “Reverse sensitivity testing: What does it take to break the model?” *European Journal of Operational Research*, **274**(2), 654–670.

Pesenti Silvana M. and Bettini A, Millosovich P, Tsanakas A (2020). “Scenario Weights for Importance Measurement (SWIM) - An R package for sensitivity analysis.” *Available at SSRN*: <https://www.ssrn.com/abstract=3515274>.

Csiszar I (1975). “I-divergence geometry of probability distributions and minimization problems.” *The Annals of Probability*, 146–158.

**See Also**

Other stress functions: [stress\\_VaR\\_ES\(\)](#), [stress\\_VaR\(\)](#), [stress\\_mean\\_sd\(\)](#), [stress\\_mean\(\)](#), [stress\\_moment\(\)](#), [stress\\_prob\(\)](#), [stress\(\)](#)

**Examples**

```
set.seed(0)
x <- as.data.frame(cbind(
  "normal" = rnorm(1000),
  "gamma" = rgamma(1000, shape = 2)))
res1 <- stress(type = "user", x = x, new_weightsfun = function(x)x ^ 2, k = 1)
## plot user defined weights against the first column of x.
plot(x$normal, get_weights(res1), pch = ".")
```

---

stress\_VaR                      *Stressing Value-at-Risk*

---

### Description

Provides weights on simulated scenarios from a baseline stochastic model, such that a stressed model component (random variable) fulfils a constraint on its quantile at a given level, also known as Value-at-Risk (VaR). Scenario weights are selected by constrained minimisation of the relative entropy to the baseline model.

### Usage

```
stress_VaR(x, alpha, q_ratio = NULL, q = NULL, k = 1)
```

### Arguments

x	A vector, matrix or data frame containing realisations of random variables. Columns of x correspond to random variables; OR A SWIM object, where x corresponds to the underlying data of the SWIM object.
alpha	Numeric vector, the levels of the stressed VaR.
q_ratio	Numeric vector, the ratio of the stressed VaR to the baseline VaR. If alpha and q_ratio are vectors, they must have the same length.
q	Numeric vector, the stressed VaR at level alpha. If alpha and q are vectors, they must have the same length.
k	Numeric, the column of x that is stressed (default = 1).

### Details

The stressed VaR is the quantile of the chosen model component, subject to the calculated scenario weights. The VaR at level alpha of a random variable with distribution function F is defined as its left-quantile at alpha:

$$VaR_{alpha} = F^{-1}(alpha).$$

If one of alpha, q (q\_ratio) is a vector, the stressed VaR's of the kth column of x, at levels alpha, are equal to q.

The stressed VaR specified, either via q or q\_ratio, might not equal the attained empirical VaR of the model component. In this case, stress\_VaR will display a message and the specs contain the achieved VaR.

### Value

A SWIM object containing:

- x, a data.frame containing the data;
- new\_weights, a list of functions, that applied to the kth column of x, generates the vectors of scenario weights. Each component corresponds to a different stress;



- type = "VaR";
- specs, a list, each component corresponds to a different stress and contains k, alpha and q.

See [SWIM](#) for details.

### Author(s)

Silvana M. Pesenti

### References

Pesenti SM, Millosovich P, Tsanakas A (2019). "Reverse sensitivity testing: What does it take to break the model?" *European Journal of Operational Research*, **274**(2), 654–670.

Pesenti Silvana M. and Bettini A, Millosovich P, Tsanakas A (2020). "Scenario Weights for Importance Measurement (SWIM) - An R package for sensitivity analysis." *Available at SSRN*: <https://www.ssrn.com/abstract=3515274>.

Csiszar I (1975). "I-divergence geometry of probability distributions and minimization problems." *The Annals of Probability*, 146–158.

### See Also

Other stress functions: [stress\\_VaR\\_ES\(\)](#), [stress\\_mean\\_sd\(\)](#), [stress\\_mean\(\)](#), [stress\\_moment\(\)](#), [stress\\_prob\(\)](#), [stress\\_user\(\)](#), [stress\(\)](#)

### Examples

```
set.seed(0)
x <- as.data.frame(cbind(
  "normal" = rnorm(1000),
  "gamma" = rgamma(1000, shape = 2)))
res1 <- stress(type = "VaR", x = x,
  alpha = 0.9, q_ratio = 1.05)

## calling stress_VaR directly
## stressing "gamma"
res2 <- stress_VaR(x = x, alpha = 0.9,
  q_ratio = c(1.03, 1.05), k = 2)
get_specs(res2)
summary(res2)
```

---

stress\_VaR\_ES

*Stressing Value-at-Risk and Expected Shortfall*


---

### Description

Provides weights on simulated scenarios from a baseline stochastic model, such that a stressed model component (random variable) fulfils a constraint on its Value-at-Risk (VaR) and Expected Shortfall (ES) risk measures, both evaluated at a given level. Scenario weights are selected by constrained minimisation of the relative entropy to the baseline model.

### Usage

```
stress_VaR_ES(
  x,
  alpha,
  q_ratio = NULL,
  s_ratio = NULL,
  q = NULL,
  s = NULL,
  k = 1,
  normalise = FALSE
)
```

### Arguments

x	A vector, matrix or data frame containing realisations of random variables. Columns of x correspond to random variables; OR A SWIM object, where x corresponds to the underlying data of the SWIM object.
alpha	Numeric vector, the levels of the stressed VaR.
q_ratio	Numeric vector, the ratio of the stressed VaR to the baseline VaR. If alpha and q_ratio are vectors, they must have the same length.
s_ratio	Numeric, vector, the ratio of the stressed ES to the baseline ES. If q (q_ratio) and s_ratio are vectors, they must have the same length.
q	Numeric vector, the stressed VaR at level alpha. If alpha and q are vectors, they must have the same length.
s	Numeric, vector, the stressed ES at level alpha. If q and s are vectors, they must have the same length.
k	Numeric, the column of x that is stressed (default = 1).
normalise	Logical. If true, values of the columns to be stressed are linearly normalised to the unit interval.

## Details

The VaR at level alpha of a random variable with distribution function F is defined as its left-quantile at alpha:

$$VaR_{alpha} = F^{-1}(alpha).$$

The ES at level alpha of a random variable with distribution function F is defined by:

$$ES_{alpha} = 1/(1 - alpha) * \int_{alpha}^1 VaR_u du.$$

The stressed VaR and ES are the risk measures of the chosen model component, subject to the calculated scenario weights. If one of alpha, q, s (q\_ratio, s\_ratio) is a vector, the stressed VaR's and ES's of the kth column of x, at levels alpha, are equal to q and s, respectively.

The stressed VaR specified, either via q or q\_ratio, might not equal the attained empirical VaR of the model component. In this case, stress\_VaR will display a message and the specs contain the achieved VaR. Further, ES is then calculated on the bases of the achieved VaR.

Normalising the data may help avoiding numerical issues when the range of values is wide.

## Value

A SWIM object containing:

- x, a data.frame containing the data;
- new\_weights, a list of functions, that applied to the kth column of x, generates the vectors of scenario weights. Each component corresponds to a different stress;
- type = "VaR ES";
- specs, a list, each component corresponds to a different stress and contains k, alpha, q and s.

See [SWIM](#) for details.

## References

Pesenti SM, Millosovich P, Tsanakas A (2019). "Reverse sensitivity testing: What does it take to break the model?" *European Journal of Operational Research*, **274**(2), 654–670.

Pesenti Silvana M. and Bettini A, Millosovich P, Tsanakas A (2020). "Scenario Weights for Importance Measurement (SWIM) - An R package for sensitivity analysis." *Available at SSRN*: <https://www.ssrn.com/abstract=3515274>.

Csiszar I (1975). "I-divergence geometry of probability distributions and minimization problems." *The Annals of Probability*, 146–158.

## See Also

Other stress functions: [stress\\_VaR\(\)](#), [stress\\_mean\\_sd\(\)](#), [stress\\_mean\(\)](#), [stress\\_moment\(\)](#), [stress\\_prob\(\)](#), [stress\\_user\(\)](#), [stress\(\)](#)

**Examples**

```

set.seed(0)
x <- as.data.frame(cbind(
  "normal" = rnorm(1000),
  "gamma" = rgamma(1000, shape = 2)))
res1 <- stress(type = "VaR ES", x = x,
  alpha = c(0.9, 0.95), q_ratio = 1.05, s_ratio = 1.08)

## calling stress_VaR_ES directly
## stressing "gamma"
res2 <- stress_VaR_ES(x = x, alpha = 0.9,
  q_ratio = 1.03, s_ratio = c(1.05, 1.08), k = 2)
get_specs(res2)
summary(res2)

```

summary.SWIM

*Summarising Stressed Models***Description**

This function is a [methods](#) for an object of class SWIM. Provides summary statistics of the stochastic model, stressed using the scenario weights.

**Usage**

```

## S3 method for class 'SWIM'
summary(object, ..., xCol = "all", wCol = "all", base = FALSE)

```

**Arguments**

object	A SWIM object.
...	Additional arguments will be ignored.
xCol	Numeric or character vector, (names of) the columns of the underlying data of the object (default = "all").
wCol	Vector, the columns of the scenario weights of the object corresponding to different stresses (default = "all").
base	Logical, if TRUE, statistics under the baseline are also returned (default = "FALSE").

**Value**

summary.SWIM returns a list with components corresponding to different stresses. Components contain a summary statistic of each column of the data of the SWIM object:

mean	The sample mean.
sd	The sample standard deviation.
skewness	The sample skewness.

ex kurtosis	The sample excess kurtosis
1st Qu.	The 25% quantile.
Median	The median, 50% quantile.
3rd Qu.	The 75% quantile.

**Author(s)**

Silvana M. Pesenti

**See Also**

[summary](#), [SWIM](#)

**Examples**

```
## continuing example in stress_VaR
set.seed(0)
x <- as.data.frame(cbind(
  "normal" = rnorm(1000),
  "gamma" = rgamma(1000, shape = 2)))
res1 <- stress(type = "VaR", x = x,
  alpha = 0.9, q_ratio = 1.05)

summary(res1, xCol = "normal", base = TRUE)
```

---

SWIM

*SWIM: A Package for Sensitivity Analysis*

---

**Description**

The SWIM package provides weights on simulated scenarios from a stochastic model, such that a stressed model components (random variables) fulfil given probabilistic constraints (e.g. specified values for risk measures), under the new scenario weights. Scenario weights are selected by constrained minimisation of the relative entropy to the baseline model.

**Details**

The SWIM (Scenario Weights for Importance Measurement) package provides weights on simulated scenarios from a stochastic model, such that stressed random variables fulfil given probabilistic constraints (e.g. specified values for risk measures), under the new scenario weights. Scenario weights are selected by constrained minimisation of the relative entropy to the baseline model.

The SWIM package is based on the *reverse sensitivity framework* developed by (Pesenti et al. 2019). Consider the random vector  $X = (X_1, \dots, X_n)$ . Let  $P$  represent the probability measure under which all simulated scenarios have the same probability. Then, for a random variable  $X_i$ , the package solves:

$$\min D(P|Q)$$

s.t. constraints on the distribution of  $X_i$  under  $Q$ , where  $D(P | Q)$  is the Kullback-Leibler divergence (relative entropy) between  $P$  and  $Q$ . The solution is formed by the scenario weights representing the Radon-Nikodym derivative  $dQ / dP$ . The weighting generates a model for which the joint distribution of  $(X_1, \dots, X_n)$  is stressed.

Different elements of  $X$  can be understood as inputs or outputs of a model. For example, consider a model  $Y = g(Z)$  with input vector  $Z = (Z_1, \dots, Z_{(n-1)})$ . One can then identify  $X_1 = Y$  and  $X_2 = Z_1, \dots, X_n = Z_{(n-1)}$ . Subsequently, the user of the SWIM package can stress the model output or any of the inputs, measuring the resulting impact on the distributions of other variables.

## Stresses

Scenario weights for the following stresses are provided:

<code>stress</code>	calls one of the functions below by using <code>type</code>
<code>stress_VaR</code>	for stressing the VaR ( <code>type = "VaR"</code> )
<code>stress_VaR_ES</code>	for stressing the VaR and ES jointly ( <code>type = "VaR ES"</code> )
<code>stress_mean</code>	for stressing means ( <code>type = "mean"</code> )
<code>stress_mean_sd</code>	for stressing means and standard deviations ( <code>type = "mean sd"</code> )
<code>stress_moment</code>	for stressing moments ( <code>type = "moment"</code> )
<code>stress_prob</code>	for stressing the probabilities of intervals ( <code>type = "prob"</code> )
<code>stress_user</code>	for user defined scenario weights ( <code>type = "user"</code> )

## A SWIM object

An object of class SWIM contains a list of:

- `x`, a data.frame containing realisations of a random vector;
- `new_weights`, a list, each component corresponds to a different stress and is either a vector of scenario weights or a function, that applied to the  $k$ th column of `x`, generates the vectors of scenario weights;
- `type`: a list, each component corresponds to a different stress and specifies the type of the stress;
- `specs`, a list, each component corresponds to a different stress and contains a list with the specifications of what has been stressed. Specifications depend on the type of stress:
  - `type = "VaR"`:  $k$ , the column of `x` on which the stress is applied to;  $\alpha$ , the level of the stressed VaR;  $q$ , the stressed VaR at level  $\alpha$ .
  - `type = "VaR ES"`:  $k$ , the column of `x` on which the stress is applied to;  $\alpha$ , the level of the stressed VaR and ES;  $q$ , the stressed VaR at level  $\alpha$ .
  - `type = "mean"`:  $k$ , the columns of `x` on which the stress is applied to; `new_means`, the stressed means.
  - `type = "mean sd"`:  $k$ , the columns of `x` on which the stress is applied to; `new_means`, the stressed means; `new_sd`, the stressed standard deviations.  $s$ , the stressed ES at level  $\alpha$ .
  - `type = "moment"`:  $f$ , the list of functions, that, applied to `x`, constitute the moment constraints;  $k$ , the columns of `x` on which each function in  $f$  operates on;  $m$ , the stressed moments of  $f(x)$ .

- type = "prob": k, the column of x on which the stress is applied to; lower, the left endpoints of the intervals; upper, the right endpoints of the intervals; prob, stressed probabilities corresponding to the intervals defined through lower and upper.
- type = "user": k, the column of x on which the stress is applied to.

## References

Pesenti SM, Millosovich P, Tsanakas A (2019). "Reverse sensitivity testing: What does it take to break the model?" *European Journal of Operational Research*, **274**(2), 654–670.

Pesenti Silvana M. and Bettini A, Millosovich P, Tsanakas A (2020). "Scenario Weights for Importance Measurement (SWIM) - An R package for sensitivity analysis." *Available at SSRN*: <https://www.ssrn.com/abstract=3515274>.

Csiszar I (1975). "I-divergence geometry of probability distributions and minimization problems." *The Annals of Probability*, 146–158.

## See Also

See [get\\_data](#) for extracting the data, x; [get\\_weights](#) for extracting the scenario weights, new\_weights; [get\\_weightsfun](#) for extracting the functions generating the scenario weights; and [get\\_specs](#) for extracting the specifications of the stress on an object of class SWIM.

# Index

- \* **datasets**
  - credit\_data, 3
- \* **stress functions**
  - stress, 21
  - stress\_mean, 23
  - stress\_mean\_sd, 24
  - stress\_moment, 26
  - stress\_prob, 28
  - stress\_user, 30
  - stress\_VaR, 32
  - stress\_VaR\_ES, 34
- cdf, 2, 11, 13
- credit\_data, 3
- ES\_stressed, 4
- get\_data, 6, 39
- get\_specs, 39
- get\_specs (get\_data), 6
- get\_weights, 6, 39
- get\_weights (get\_data), 6
- get\_weightsfun, 39
- get\_weightsfun (get\_data), 6
- importance\_rank, 8, 16, 20
- merge.SWIM, 9
- methods, 36
- nleqslv, 23, 25, 27
- plot\_cdf, 3, 10, 13, 14, 17
- plot\_hist, 12
- plot\_quantile, 13, 17
- plot\_sensitivity, 9, 14, 20
- plot\_weights, 16
- quantile, 18
- quantile\_stressed, 3, 11, 13, 14, 18
- sensitivity, 8, 9, 15, 16, 19
- stress, 21, 24, 26, 28, 30, 31, 33, 35, 38
- stress\_mean, 22, 23, 25, 26, 28, 30, 31, 33, 35, 38
- stress\_mean\_sd, 22, 24, 24, 28, 30, 31, 33, 35, 38
- stress\_moment, 22–26, 26, 30, 31, 33, 35, 38
- stress\_prob, 22, 24, 26, 28, 28, 31, 33, 35, 38
- stress\_user, 22, 24, 26, 28, 30, 30, 33, 35, 38
- stress\_VaR, 22, 24, 26, 28, 30, 31, 32, 35, 38
- stress\_VaR\_ES, 22, 24, 26, 28, 30, 31, 33, 34, 38
- summary, 9, 20, 37
- summary.SWIM, 36
- SWIM, 7, 10, 22, 23, 25, 27, 29, 31, 33, 35, 37, 37
- VaR\_stressed (ES\_stressed), 4
- wtd.quantile, 18