# Package 'scoringfunctions'

August 30, 2026

**Version** 1.2

**Date** 2026-08-30

**Title** Loss Functions for Assessing Point Forecasts

**Description** Implements consistent loss functions and the corresponding identification functions for point forecasts, including those of the mean, the median, quantiles, expectiles, moments and Huber functionals (Gneiting T (2011) <doi:10.1198/jasa.2011.r10138>; Fissler T, Ziegel JF (2016) <doi:10.1214/16-AOS1439>). Pointwise losses and realised average scores are both available. Detailed documentation of the functions' properties is included for facilitating the interpretation of results.

**Depends** R (>= 4.0.0)

**License** GPL-3

**Author** Hristos Tyralis [aut, cre] (ORCID: <https://orcid.org/0000-0002-8932-4997>), Georgia Papacharalampous [aut] (ORCID: <https://orcid.org/0000-0001-5446-954X>)

**Maintainer** Hristos Tyralis <montchrister@gmail.com>

**Repository** CRAN

**NeedsCompilation** no

**Date/Publication** 2026-08-30 15:20:02 UTC

## Contents

---

`scoringfunctions-package`

*Overview of the functions in the scoringfunctions package*

---

## Description

The scoringfunctions package implements consistent scoring (loss) functions and identification functions.

## Details

The table below lists a selection of predictive functionals alongside their pointwise scoring functions (`_sf`) / realised average scores (`_rs`), and identification functions (`_if`). The complete listing of the package functions is given in the numbered sections below the table:

| **Target Functional** | **Loss Functions** (`_sf` / `_rs`) | **Identification** (`_if`) |
|---|---|---|
| Mean | `serr_sf` / `mse`, `bregman1_sf` / `bregman1_rs` `bregman2_sf` / `bregman2_rs`, `bregman3_sf` / `qlike` `bregman4_sf` / `bregman4_rs` | `mean_if` |
| Expectile ($p$) | `expectile_sf` / `expectile_rs` | `expectile_if` |
| Median | `aerr_sf` / `mae`, `maelog_sf` / `maelog_rs` `maesd_sf` / `maesd_rs` | `quantile_if` ($p = 0.5$) |
| $\beta$-Median | `bmedian_sf` / `bmedian_rs` | — |
| Quantile ($p$) | `quantile_sf` / `quantile_rs`, `gpl1_sf` / `gpl1_rs` `gpl2_sf` / `gpl2_rs` | `quantile_if` |
| Huber Mean | `huber_sf` / `huber_rs` | `hubermean_if` |

| | | |
|---|---|---|
| Huber Quantile | ghuber_sf / ghuber_rs | huberquantile_if |
| $L_q$-Mean | lqmean_sf / lqmean_rs | — |
| $L_q$-Quantile | lqquantile_sf / lqquantile_rs | — |
| Interval ($p$) | interval_sf | — |
| Mean - Variance | mv_sf | mv_if |
| Error - Spread | errorspread_sf | — |
| Relative Error | relerr_sf / mre | — |
| Percentage Error | aperr_sf / mape | — |

The package functions are categorised into six classes, each of which has its own section below:

1. Scoring functions
2. Realised (average) score functions
3. Skill score functions
4. Identification functions
5. Functions for sample levels
6. Supporting functions

## 1. Scoring functions

### 1.1. Consistent scoring functions for one-dimensional functionals:

*1.1.1. Consistent scoring functions for the mean:*
bregman1_sf: Bregman scoring function (type 1)
bregman2_sf: Bregman scoring function (type 2, Patton scoring function)
bregman3_sf: Bregman scoring function (type 3, QLIKE scoring function)
bregman4_sf: Bregman scoring function (type 4, Patton scoring function)
serr_sf: Squared error scoring function

*1.1.2. Consistent scoring functions for expectiles:*
expectile_sf: Asymmetric piecewise quadratic scoring function (expectile scoring function, expectile loss function)

*1.1.3. Consistent scoring functions for the median:*
aerr_sf: Absolute error scoring function
maelog_sf: MAE-LOG scoring function
maesd_sf: MAE-SD scoring function

*1.1.4. Consistent scoring functions for quantiles:*
gpl1_sf: Generalized piecewise linear power scoring function (type 1)
gpl2_sf: Generalized piecewise linear power scoring function (type 2)
quantile_sf: Asymmetric piecewise linear scoring function (quantile scoring function, quantile loss function)

*1.1.5. Consistent scoring functions for Huber functionals:*
ghuber_sf: Generalized Huber scoring function
huber_sf: Huber scoring function

*1.1.6. Consistent scoring functions for other functionals:*
aperr_sf: Absolute percentage error scoring function
bmedian_sf: $\beta$-median scoring function

linex_sf: LINEX scoring function
lqmean_sf: $L_q$-mean scoring function
lqquantile_sf: $L_q$-quantile scoring function
nmoment_sf: $n$-th moment scoring function
obsweighted_sf: Observation-weighted scoring function
powerweighted_sf: Power-weighted squared error scoring function
relerr_sf: Relative error scoring function (MAE-PROP scoring function)
serrexp_sf: Squared error exp scoring function
serrlog_sf: Squared error log scoring function
serrpower_sf: Squared error of power transformations scoring function
serrsq_sf: Squared error of squares scoring function
sperr_sf: Squared percentage error scoring function
srelerr_sf: Squared relative error scoring function

#### 1.2. Consistent scoring functions for two-dimensional functionals:

interval_sf: Interval scoring function (Winkler scoring function)
mv_sf: Mean - variance scoring function

#### 1.3. Consistent scoring functions for multi-dimensional functionals:

errorspread_sf: Error - spread scoring function

### 2. Realised (average) score functions

#### 2.1. Realised (average) score functions for one-dimensional functionals:

*2.1.1. Realised (average) score functions for the mean:*
bregman1_rs: Realised Bregman score (type 1)
bregman2_rs: Realised Bregman score (type 2, Patton score)
bregman4_rs: Realised Bregman score (type 4, Patton score)
mse: Mean squared error (MSE)
qlike: QLIKE

*2.1.2. Realised (average) score functions for expectiles:*
expectile_rs: Realised expectile score

*2.1.3. Realised (average) score functions for the median:*
mae: Mean absolute error (MAE)
maelog_rs: Realised MAE-LOG score
maesd_rs: Realised MAE-SD score

*2.1.4. Realised (average) score functions for quantiles:*
gpl1_rs: Realised generalized piecewise linear power score (type 1)
gpl2_rs: Realised generalized piecewise linear power score (type 2)
quantile_rs: Realised quantile score

*2.1.5. Realised (average) score functions for Huber functionals:*
ghuber_rs: Realised generalized Huber score
huber_rs: Realised Huber score

*2.1.6. Realised (average) score functions for other functionals:*
bmedian_rs: Realised $\beta$-median score
linex_rs: Realised LINEX score

lqmean_rs: Realised $L_q$-mean score
lqquantile_rs: Realised $L_q$-quantile score
mape: Mean absolute percentage error (MAPE)
mre: Mean relative error (MRE)
mspe: Mean squared percentage error (MSPE)
msre: Mean squared relative error (MSRE)
nmoment_rs: Realised $n$-th moment score
obsweighted_rs: Realised observation-weighted score
serrexp_rs: Realised squared error exp score
serrlog_rs: Realised squared error log score
serrpower_rs: Realised squared error of power transformations score
serrsq_rs: Realised squared error of squares score

## 3. Skill score functions

### 3.1. Skill score functions for one-dimensional functionals:

*3.1.1. Skill score functions for the mean:*
nse: Nash-Sutcliffe efficiency (NSE)

## 4. Identification functions

### 4.1. Identification functions for one-dimensional functionals:

expectile_if: Expectile identification function
hubermean_if: Huber mean identification function
huberquantile_if: Huber quantile identification function
mean_if: Mean identification function
meanexp_if: Exp-transformed identification function
meanlog_if: Log-transformed identification function
meanpower_if: Power-transformed identification function
nmoment_if: $n$-th moment identification function
powerweighted_if: Power-weighted identification function
quantile_if: Quantile identification function

### 4.2. Identification functions for two-dimensional functionals:

mv_if: Mean - variance identification function

## 5. Functions for sample levels

quantile_level: Sample quantile level function

## 6. Supporting functions

capping_function: Capping function

## References


Banerjee A, Guo X, Wang H (2005) On the optimality of conditional expectation as a Bregman predictor. *IEEE Transactions on Information Theory* **51(7)**:2664–2669. doi:10.1109/TIT.2005.850145.

Bellini F, Klar B, Muller A, Rosazza Gianin E (2014) Generalized quantiles as risk measures. *Insurance: Mathematics and Economics* **54**:41–48. doi:10.1016/j.insmatheco.2013.10.015.

Brehmer JR, Gneiting T (2021) Scoring interval forecasts: Equal-tailed, shortest, and modal interval. *Bernoulli* **27(3)**:1993–2010. doi:10.3150/20BEJ1298.

Chen Z (1996) Conditional $L_p$-quantiles and their application to the testing of symmetry in non-parametric regression. *Statistics and Probability Letters* **29(2)**:107–115. doi:10.1016/0167-7152(95)001638.

Christensen HM, Moroz IM, Palmer TN (2015) Evaluation of ensemble forecast uncertainty using a new proper score: Application to medium-range and seasonal forecasts. *Quarterly Journal of the Royal Meteorological Society* **141(687)(Part B)**:538–549. doi:10.1002/qj.2375.

Dimitriadis T, Fissler T, Ziegel JF (2024) Osband's principle for identification functions. *Statistical Papers* **65**:1125–1132. doi:10.1007/s0036202301428x.

Dunsmore IR (1968) A Bayesian approach to calibration. *Journal of the Royal Statistical Society, Series B (Methodological)* **30(2)**:396–405. doi:10.1111/j.25176161.1968.tb00740.x.

Ferguson TS (1967) Mathematical Statistics: A Decision-Theoretic Approach. Academic Press, New York.

Fissler T, Pesenti SM (2023) Sensitivity measures based on scoring functions. *European Journal of Operational Research* **307(3)**:1408–1423. doi:10.1016/j.ejor.2022.10.002.

Fissler T, Ziegel JF (2016) Higher order elicitability and Osband's principle. *The Annals of Statistics* **44(4)**:1680–1707. doi:10.1214/16AOS1439.

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gerber HU (1974) On additive premium calculation principles. *ASTIN Bulletin: The Journal of the IAA* **7(3)**:215–222. doi:10.1017/S0515036100006061.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Gneiting T, Raftery AE (2007) Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* **102(477)**:359–378. doi:10.1198/016214506000001437.

Houghton-Carr HA (1999) Assessment criteria for simple conceptual daily rainfall-runoff models. *Hydrological Sciences Journal* **44(2)**:237–261. doi:10.1080/02626669909492220.

Huber PJ (1964) Robust estimation of a location parameter. *Annals of Mathematical Statistics* **35(1)**:73–101. doi:10.1214/aoms/1177703732.

Koenker R, Bassett Jr G (1978) Regression quantiles. *Econometrica* **46(1)**:33–50. doi:10.2307/1913643.

Nash JE, Sutcliffe JV (1970) River flow forecasting through conceptual models Part I - A discussion of principles. *Journal of Hydrology* **10(3)**:282–290. doi:10.1016/00221694(70)902556.

Newey WK, Powell JL (1987) Asymmetric least squares estimation and testing. *Econometrica* **55(4)**:819–847. doi:10.2307/1911031.

Park H, Stefanski LA (1998) Relative-error prediction. *Statistics and Probability Letters* **40(3)**:227–236. doi:10.1016/S01677152(98)000881.

Patton AJ (2011) Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics* **160(1)**:246–256. doi:10.1016/j.jeconom.2010.03.034.

Raiffa H, Schlaifer R (1961) Applied Statistical Decision Theory. Colonial Press, Clinton.

Saerens M (2000) Building cost functions minimizing to some summary statistics. *IEEE Transactions on Neural Networks* **11(6)**:1263–1271. doi:10.1109/72.883416.

Savage LJ (1971) Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association* **66(336)**:783–801. doi:10.1080/01621459.1971.10482346.

Taggart RJ (2022) Point forecasting and forecast evaluation with generalized Huber loss. *Electronic Journal of Statistics* **16(1)**:201–231. doi:10.1214/21EJS1957.

Thirel G, Santos L, Delaigue O, Perrin C (2024) On the use of streamflow transformations for hydrological model calibration. *Hydrology and Earth System Sciences* **28(21)**:4837–4860. doi:10.5194/hess2848372024.

Thomson W (1979) Eliciting production possibilities from a well-informed manager. *Journal of Economic Theory* **20(3)**:360–380. doi:10.1016/00220531(79)900425.

Tyralis H, Papacharalampous G (2026) Variable transformations in consistent loss functions. *Knowledge-Based Systems* **336**:115202. doi:10.1016/j.knosys.2025.115202.

Varian HR (1975) A Bayesian approach to real estate assessment. In: Fienberg SE, Zellner A (eds) *Studies in Bayesian Econometrics and Statistics in Honor of Leonard J. Savage*. Amsterdam: North-Holland, pp 195–208.

Winkler RL (1972) A decision-theoretic approach to interval estimation. *Journal of the American Statistical Association* **67(337)**:187–191. doi:10.1080/01621459.1972.10481224.

Winkler RL, Murphy AH (1979) The use of probabilities in forecasts of maximum and minimum temperatures. *Meteorological Magazine* **108(1288)**:317–329.

Yeh C-C, Yeh H-W, Chan W (2008) Some equivalent forms of the arithematic-geometric mean inequality in probability: A survey. *Journal of Inequalities and Applications* **2008**:386715. doi:10.1155/2008/386715.

Zellner A (1986) Bayesian estimation and prediction using asymmetric loss functions. *Journal of the American Statistical Association* **81(394)**:446–451. doi:10.1080/01621459.1986.10478289.

---

`aerr_sf` *Absolute error scoring function*

---

## Description

The function aerr_sf computes the absolute error scoring function when $y$ materialises and $x$ is the predictive median functional.

The absolute error scoring function is defined in Table 1 in Gneiting (2011).

## Usage

```
aerr_sf(x, y)
```

## Arguments

| | |
|---|---|
| x | Predictive median functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |

## Details

The absolute error scoring function is defined by:

$$S(x, y) := |x - y|$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

Range of function:

$$S(x, y) \geq 0, \forall x, y \in \mathbb{R}$$

## Value

Vector of absolute errors.

## Note

For details on the absolute error scoring function, see Gneiting (2011).

The median functional is the median of the probability distribution $F$ of $Y$ (Gneiting 2011).

The absolute error scoring function is negatively oriented (i.e. the smaller, the better).

The absolute error scoring function is strictly $\mathbb{F}$-consistent for the median functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y]$ exists and is finite (Raiffa and Schlaifer 1961, p.196; Ferguson 1967, p.51; Thomson 1979; Saerens 2000; Gneiting 2011).

## References

Ferguson TS (1967) Mathematical Statistics: A Decision-Theoretic Approach. Academic Press, New York.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Raiffa H, Schlaifer R (1961) Applied Statistical Decision Theory. Colonial Press, Clinton.

Saerens M (2000) Building cost functions minimizing to some summary statistics. *IEEE Transactions on Neural Networks* **11(6)**:1263–1271. doi:10.1109/72.883416.

Thomson W (1979) Eliciting production possibilities from a well-informed manager. *Journal of Economic Theory* **20(3)**:360–380. doi:10.1016/00220531(79)900425.

## See Also

mae, quantile_if

## Examples

```
# Compute the absolute error scoring function.

df <- data.frame(
    y = rep(x = 0, times = 5),
    x = -2:2
)

df$absolute_error <- aerr_sf(x = df$x, y = df$y)

print(df)
```

---

aperr_sf *Absolute percentage error scoring function*

---

## Description

The function aperr_sf computes the absolute percentage error scoring function when $y$ materialises and $x$ is the predictive $\text{med}^{(-1)}(F)$ functional.

The absolute percentage error scoring function is defined in Table 1 in Gneiting (2011).

## Usage

```
aperr_sf(x, y)
```

## Arguments

| | |
|---|---|
| x | Predictive $\text{med}^{(-1)}(F)$ functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |

## Details

The absolute percentage error scoring function is defined by:

$$S(x, y) := |(x - y)/y|$$

Domain of function:

$$x > 0$$

$$y > 0$$

Range of function:

$$S(x, y) \geq 0, \forall x, y > 0$$

## Value

Vector of absolute percentage errors.

## Note

For details on the absolute percentage error scoring function, see Gneiting (2011).

The $\beta$-median functional, $\text{med}^{(\beta)}(F)$ is the median of a probability distribution whose density is proportional to $y^{\beta} f(y)$, where $f$ is the density of the probability distribution $F$ of $Y$ (Gneiting 2011).

The functional $\text{med}^{(-1)}(F)$ elicited by the absolute percentage error scoring function is the $\beta$-median at $\beta = -1$, whose density is proportional to $f(y)/y$ (p. 752 in Gneiting 2011). The $\beta$-median functional at general $\beta$ is elicited by bmedian_sf.

The absolute percentage error scoring function is negatively oriented (i.e. the smaller, the better).

The absolute percentage error scoring function is strictly $\mathbb{F}^{(w)}$-consistent for the $\text{med}^{(-1)}(F)$ functional. $\mathbb{F}$ is the family of probability distributions for which $\text{E}_F[Y]$ exists and is finite. $\mathbb{F}^{(w)}$ is the subclass of probability distributions in $\mathbb{F}$, which are such that $w(y)f(y)$, $w(y) = 1/y$ has finite integral over $(0, \infty)$, and the probability distribution $F^{(w)}$ with density proportional to $w(y)f(y)$ belongs to $\mathbb{F}$ (see Theorems 5 and 9 in Gneiting 2011).

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

mape

## Examples

```
# Compute the absolute percentage error scoring function.

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3
)

df$absolute_percentage_error <- aperr_sf(x = df$x, y = df$y)

print(df)
```

`bmedian_rs` *Realised $\beta$-median score*

## Description

The function bmedian_rs computes the realised $\beta$-median score with parameter $b$, when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised $\beta$-median score is a realised score corresponding to the $\beta$-median scoring function bmedian_sf.

## Usage

```
bmedian_rs(x, y, b)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |
| b | It can be a scalar. |

## Details

The realised $\beta$-median score is defined by:

$$S(\boldsymbol{x}, \boldsymbol{y}, b) := (1/n) \sum_{i=1}^{n} L(x_i, y_i, b)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y, b) := |1 - (y/x)^b|$$

Domain of function:

$$\boldsymbol{x} > \mathbf{0}$$

$$\boldsymbol{y} > \mathbf{0}$$

$$b \neq 0$$

where

$$\mathbf{0} = (0, ..., 0)^{\mathsf{T}}$$

is the zero vector of length $n$ and the symbol $>$ indicates pairwise inequality.

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}, b) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} > \mathbf{0}, b \neq 0$$

## Value

Value of the realised $\beta$-median score.

## Note

For details on the $\beta$-median scoring function, see bmedian_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised $\beta$-median score is the realised (average) score corresponding to the $\beta$-median scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

bmedian_sf

## Examples

```
# Compute the realised beta-median score.

set.seed(12345)

b <- 2

x <- 0.5

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(bmedian_rs(x = x, y = y, b = b))

print(bmedian_rs(x = rep(x = x, times = 100), y = y, b = b))
```

`bmedian_sf` *$\beta$-median scoring function*

## Description

The function bmedian_sf computes the $\beta$-median scoring function when $y$ materialises and $x$ is the predictive $\text{med}^{(\beta)}(F)$ functional.

The $\beta$-median scoring function is defined in eq. (4) in Gneiting (2011).

## Usage

```
bmedian_sf(x, y, b)
```

## Arguments

| | |
|---|---|
| `x` | Predictive $\text{med}^{(\beta)}(F)$ functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| `y` | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| `b` | The parameter $\beta$ of the $\beta$-median functional. It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The $\beta$-median scoring function is defined by, with $b = \beta$:

$$S(x, y, b) := |1 - (y/x)^b|$$

Domain of function:

$$x > 0$$

$$y > 0$$

$$b \neq 0$$

Range of function:

$$S(x, y, b) \geq 0, \forall x, y > 0, b \neq 0$$

## Value

Vector of $\beta$-median losses.

## Note

For details on the $\beta$-median scoring function, see Gneiting (2011).

The $\beta$-median functional, $\text{med}^{(\beta)}(F)$ is the median of a probability distribution whose density is proportional to $y^\beta f(y)$, where $f$ is the density of the probability distribution $F$ of $Y$ (Gneiting 2011).

The $\beta$-median scoring function is negatively oriented (i.e. the smaller, the better).

The $\beta$-median scoring function is strictly $\mathbb{F}^{(w)}$-consistent for the $\text{med}^{(\beta)}(F)$ functional. $\mathbb{F}$ is the family of probability distributions for which $\text{E}_F[Y]$ exists and is finite. $\mathbb{F}^{(w)}$ is the subclass of probability distributions in $\mathbb{F}$, which are such that $w(y)f(y)$, $w(y) = y^\beta$ has finite integral over $(0, \infty)$, and the probability distribution $F^{(w)}$ with density proportional to $w(y)f(y)$ belongs to $\mathbb{F}$ (see Theorems 5 and 9 in Gneiting 2011).

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

bmedian_rs

## Examples

```
# Compute the bmedian scoring function.

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3,
    b = c(-1, 1, 2)
)

df$bmedian_error <- bmedian_sf(x = df$x, y = df$y, b = df$b)

print(df)
```

---

# bregman1_rs *Realised Bregman score (type 1)*

---

## Description

The function bregman1_rs computes the realised Bregman score (type 1) with parameter $a$, when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised Bregman score (type 1) is a realised score corresponding to the Bregman scoring function (type 1) bregman1_sf.

## Usage

```
bregman1_rs(x, y, a)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |
| a | It can be a scalar. |

## Details

The realised Bregman score (type 1) is defined by:

$$S(\boldsymbol{x}, \boldsymbol{y}, a) := (1/n) \sum_{i=1}^{n} L(x_i, y_i, a)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y, a) := |y|^a - |x|^a - a\text{sign}(x)|x|^{a-1}(y - x)$$

Domain of function:

$$\boldsymbol{x} \in \mathbb{R}^n$$

$$\boldsymbol{y} \in \mathbb{R}^n$$

$$a > 1$$

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}, a) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, a > 1$$

## Value

Value of the realised Bregman score (type 1).

## Note

For details on the Bregman scoring function (type 1), see bregman1_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised Bregman score (type 1) is the realised (average) score corresponding to the Bregman scoring function (type 1).

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

bregman1_sf, mean_if

## Examples

```
# Compute the realised Bregman score (type 1).

set.seed(12345)

a <- 3

x <- 0

y <- rnorm(n = 100, mean = 0, sd = 1)

print(bregman1_rs(x = x, y = y, a = a))

print(bregman1_rs(x = rep(x = x, times = 100), y = y, a = a))
```

---

# bregman1_sf *Bregman scoring function (type 1)*

---

## Description

The function bregman1_sf computes the Bregman scoring function when $y$ materialises and $x$ is the predictive mean functional.

The Bregman scoring function is defined by eq. (18) in Gneiting (2011) and the form implemented here for $\phi(x) = |x|^a$ is defined by eq. (19) in Gneiting (2011).

## Usage

```
bregman1_sf(x, y, a)
```

## Arguments

| | |
|---|---|
| x | Predictive mean functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| a | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The Bregman scoring function (type 1) is defined by:

$$S(x, y, a) := |y|^a - |x|^a - a\text{sign}(x)|x|^{a-1}(y - x)$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$a > 1$$

Range of function:

$$S(x, y, a) \geq 0, \forall x, y \in \mathbb{R}, a > 1$$

## Value

Vector of Bregman losses.

## Note

The implemented function is named type 1 because it corresponds to a specific type of $\phi(x)$ of the general form of the Bregman scoring function defined by eq. (18) in Gneiting (2011).

For details on the Bregman scoring function, see Savage (1971), Banerjee et al. (2005) and Gneiting (2011).

The mean functional is the mean $\mathrm{E}_F[Y]$ of the probability distribution $F$ of $Y$ (Gneiting 2011).

The Bregman scoring function is negatively oriented (i.e. the smaller, the better).

The herein implemented Bregman scoring function is strictly $\mathbb{F}$-consistent for the mean functional. $\mathbb{F}$ is the family of probability distributions for which $\mathrm{E}_F[Y]$ and $\mathrm{E}_F[|Y|^a]$ exist and are finite (Savage 1971; Gneiting 2011).

## References

Banerjee A, Guo X, Wang H (2005) On the optimality of conditional expectation as a Bregman predictor. *IEEE Transactions on Information Theory* **51(7)**:2664–2669. doi:10.1109/TIT.2005.850145.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Savage LJ (1971) Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association* **66(336)**:783–801. doi:10.1080/01621459.1971.10482346.

## See Also

bregman1_rs, mean_if

## Examples

```
# Compute the Bregman scoring function (type 1).

df <- data.frame(
    y = rep(x = 0, times = 7),
    x = c(-3, -2, -1, 0, 1, 2, 3),
    a = rep(x = 3, times = 7)
)

df$bregman1_penalty <- bregman1_sf(x = df$x, y = df$y, a = df$a[1])

print(df)

# Equivalence of Bregman scoring function (type 1) and squared error scoring
# function, when a = 2.

set.seed(12345)

n <- 100

x <- runif(n = n, min = -20, max = 20)
y <- runif(n = n, min = -20, max = 20)
a <- rep(x = 2, times = n)

u <- bregman1_sf(x = x, y = y, a = a)

v <- serr_sf(x = x, y = y)

max(abs(u - v)) # values are slightly higher than 0 due to rounding error
min(abs(u - v))
```

---

bregman2_rs *Realised Bregman score (type 2, Patton score)*

---

## Description

The function bregman2_rs computes the realised Bregman score (type 2, Patton score) with parameter $b$, when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised Bregman score (type 2) is a realised score corresponding to the Bregman scoring function (type 2, Patton scoring function) bregman2_sf.

## Usage

```
bregman2_rs(x, y, b)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |
| b | It can be a scalar. |

## Details

The realised Bregman score (type 2) is defined by:

$$S(\boldsymbol{x}, \boldsymbol{y}, b) := (1/n) \sum_{i=1}^{n} L(x_i, y_i, b)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y, b) := \frac{1}{b(b-1)}(y^b - x^b) - \frac{1}{b-1} x^{b-1}(y - x)$$

Domain of function:

$$\boldsymbol{x} > \mathbf{0}$$

$$\boldsymbol{y} > \mathbf{0}$$

$$b \in \mathbb{R} \setminus \{0, 1\}$$

where

$$\mathbf{0} = (0, ..., 0)^\mathsf{T}$$

is the zero vector of length $n$ and the symbol $>$ indicates pairwise inequality.

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}, b) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} > \mathbf{0}, b \in \mathbb{R} \setminus \{0, 1\}$$

## Value

Value of the realised Bregman score (type 2).

## Note

For details on the Bregman scoring function (type 2), see bregman2_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised Bregman score (type 2) is the realised (average) score corresponding to the Bregman scoring function (type 2).

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

bregman2_sf, mean_if

## Examples

```
# Compute the realised Bregman score (type 2).

set.seed(12345)

b <- 3

x <- 0.5

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(bregman2_rs(x = x, y = y, b = b))

print(bregman2_rs(x = rep(x = x, times = 100), y = y, b = b))
```

`bregman2_sf` *Bregman scoring function (type 2, Patton scoring function)*

## Description

The function bregman2_sf computes the Bregman scoring function when $y$ materialises and $x$ is the predictive mean functional.

The Bregman scoring function is defined by eq. (18) in Gneiting (2011) and the form implemented here for $\phi(x) = \frac{1}{b(b-1)}x^b$, $b \in \mathbb{R} \setminus \{0, 1\}$ is defined by eq. (20) in Gneiting (2011).

## Usage

```
bregman2_sf(x, y, b)
```

## Arguments

| | |
|---|---|
| `x` | Predictive mean functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| `y` | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| `b` | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The Bregman scoring function (type 2) is defined by:

$$S(x, y, b) := \frac{1}{b(b-1)}(y^b - x^b) - \frac{1}{b-1}x^{b-1}(y - x)$$

Domain of function:

$$x > 0$$

$$y > 0$$

$$b \in \mathbb{R} \setminus \{0, 1\}$$

Range of function:

$$S(x, y, b) \geq 0, \forall x, y > 0, b \in \mathbb{R} \setminus \{0, 1\}$$

## Value

Vector of Bregman losses.

## Note

The implemented function is named type 2 because it corresponds to a specific type of $\phi(x)$ of the general form of the Bregman scoring function defined by eq. (18) in Gneiting (2011).

For details on the Bregman scoring function, see Savage (1971), Banerjee et al. (2005) and Gneiting (2011). For details on the specific form implemented here, see Patton (2011).

The mean functional is the mean $\mathrm{E}_F[Y]$ of the probability distribution $F$ of $Y$ (Gneiting 2011).

The Bregman scoring function is negatively oriented (i.e. the smaller, the better).

The herein implemented Bregman scoring function is strictly $\mathbb{F}$-consistent for the mean functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y]$ and $\mathrm{E}_F[\frac{1}{b(b-1)}Y^b]$ exist and are finite (Savage 1971; Gneiting 2011).

## References

Banerjee A, Guo X, Wang H (2005) On the optimality of conditional expectation as a Bregman predictor. *IEEE Transactions on Information Theory* **51(7)**:2664–2669. doi:10.1109/TIT.2005.850145.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Patton AJ (2011) Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics* **160(1)**:246–256. doi:10.1016/j.jeconom.2010.03.034.

Savage LJ (1971) Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association* **66(336)**:783–801. doi:10.1080/01621459.1971.10482346.

## See Also

bregman2_rs, mean_if

## Examples

```
# Compute the Bregman scoring function (type 2).

df <- data.frame(
    y = rep(x = 2, times = 6),
    x = rep(x = 1:3, times = 2),
    b = rep(x = c(-3, 3), each = 3)
)

df$bregman2_penalty <- bregman2_sf(x = df$x, y = df$y, b = df$b)

print(df)

# The Bregman scoring function (type 2) is half the squared error scoring
# function, when b = 2.

df <- data.frame(
    y = rep(x = 5.5, times = 10),
    x = 1:10,
    b = rep(x = 2, times = 10)
```

```
)

df$bregman2_penalty <- bregman2_sf(x = df$x, y = df$y, b = df$b)

df$squared_error <- serr_sf(x = df$x, y = df$y)

df$ratio <- df$bregman2_penalty/df$squared_error

print(df)

# When a = b > 1 the Bregman scoring function (type 2) equals the
# Bregman scoring function (type 1) up to a multiplicative constant.

df <- data.frame(
    y = rep(x = 5.5, times = 10),
    x = 1:10,
    b = rep(x = c(3, 4), each = 5)
)

df$bregman2_penalty <- bregman2_sf(x = df$x, y = df$y, b = df$b)

df$bregman1_penalty <- bregman1_sf(x = df$x, y = df$y, a = df$b)

df$ratio <- df$bregman2_penalty/df$bregman1_penalty

print(df)
```

---

`bregman3_sf` *Bregman scoring function (type 3, QLIKE scoring function)*

---

## Description

The function bregman3_sf computes the Bregman scoring function when $y$ materialises and $x$ is the predictive mean functional.

The Bregman scoring function is defined by eq. (18) in Gneiting (2011) and the form implemented here for $\phi(x) = -\log(x)$ is defined by eq. (20) in Gneiting (2011).

## Usage

```
bregman3_sf(x, y)
```

## Arguments

| | |
|---|---|
| `x` | Predictive mean functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| `y` | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |

## Details

The Bregman scoring function (type 3) is defined by:

$$S(x, y) := (y/x) - \log(y/x) - 1$$

Domain of function:

$$x > 0$$

$$y > 0$$

Range of function:

$$S(x, y) \geq 0, \forall x, y > 0$$

## Value

Vector of Bregman losses.

## Note

The implemented function is named type 3 because it corresponds to a specific type of $\phi(x)$ of the general form of the Bregman scoring function defined by eq. (18) in Gneiting (2011).

For details on the Bregman scoring function, see Savage (1971), Banerjee et al. (2005) and Gneiting (2011). For details on the specific form implemented here, see the QLIKE scoring function in Patton (2011).

The mean functional is the mean $\mathrm{E}_F[Y]$ of the probability distribution $F$ of $Y$ (Gneiting 2011).

The Bregman scoring function is negatively oriented (i.e. the smaller, the better).

The herein implemented Bregman scoring function is strictly $\mathbb{F}$-consistent for the mean functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y]$ and $\mathrm{E}_F[\log(Y)]$ exist and are finite (Savage 1971; Gneiting 2011).

## References

Banerjee A, Guo X, Wang H (2005) On the optimality of conditional expectation as a Bregman predictor. *IEEE Transactions on Information Theory* **51(7)**:2664–2669. doi:10.1109/TIT.2005.850145.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Patton AJ (2011) Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics* **160(1)**:246–256. doi:10.1016/j.jeconom.2010.03.034.

Savage LJ (1971) Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association* **66(336)**:783–801. doi:10.1080/01621459.1971.10482346.

## See Also

`qlike`, `mean_if`

## Examples

```
# Compute the Bregman scoring function (type 3, QLIKE scoring function).

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3
)

df$bregman3_penalty <- bregman3_sf(x = df$x, y = df$y)

print(df)
```

---

`bregman4_rs` *Realised Bregman score (type 4, Patton score)*

---

## Description

The function bregman4_rs computes the realised Bregman score (type 4, Patton score) when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised Bregman score (type 4) is a realised score corresponding to the Bregman scoring function (type 4, Patton scoring function) bregman4_sf.

## Usage

```
bregman4_rs(x, y)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |

## Details

The realised Bregman score (type 4) is defined by:

$$S(\boldsymbol{x},\boldsymbol{y}) := (1/n)\sum_{i=1}^{n} L(x_i, y_i)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y) := y \log(y/x) - y + x$$

Domain of function:

$$\boldsymbol{x} > \mathbf{0}$$

$$\boldsymbol{y} > \mathbf{0}$$

where

$$\mathbf{0} = (0, ..., 0)^\mathsf{T}$$

is the zero vector of length $n$ and the symbol $>$ indicates pairwise inequality.

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} > \mathbf{0}$$

## Value

Value of the realised Bregman score (type 4).

## Note

For details on the Bregman scoring function (type 4), see bregman4_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised Bregman score (type 4) is the realised (average) score corresponding to the Bregman scoring function (type 4).

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

bregman4_sf, mean_if

## Examples

```
# Compute the realised Bregman score (type 4).

set.seed(12345)

x <- 0.5

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(bregman4_rs(x = x, y = y))

print(bregman4_rs(x = rep(x = x, times = 100), y = y))
```

---

`bregman4_sf` *Bregman scoring function (type 4, Patton scoring function)*

---

## Description

The function bregman4_sf computes the Bregman scoring function when $y$ materialises and $x$ is the predictive mean functional.

The Bregman scoring function is defined by eq. (18) in Gneiting (2011) and the form implemented here for $\phi(x) = x\log(x)$ is defined by eq. (20) in Gneiting (2011).

## Usage

```
bregman4_sf(x, y)
```

## Arguments

| | |
|---|---|
| x | Predictive mean functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |

## Details

The Bregman scoring function (type 4) is defined by:

$$S(x, y) := y\log(y/x) - y + x$$

Domain of function:

$$x > 0$$

$$y > 0$$

Range of function:

$$S(x, y) \geq 0, \forall x, y > 0$$

## Value

Vector of Bregman losses.

## Note

The implemented function is named type 4 because it corresponds to a specific type of $\phi(x)$ of the general form of the Bregman scoring function defined by eq. (18) in Gneiting (2011).

For details on the Bregman scoring function, see Savage (1971), Banerjee et al. (2005) and Gneiting (2011). For details on the specific form implemented here, see Patton (2011).

The mean functional is the mean $\mathrm{E}_F[Y]$ of the probability distribution $F$ of $Y$ (Gneiting 2011).

The Bregman scoring function is negatively oriented (i.e. the smaller, the better).

The herein implemented Bregman scoring function is strictly $\mathbb{F}$-consistent for the mean functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y]$ and $\mathrm{E}_F[Y \log(Y)]$ exist and are finite (Savage 1971; Gneiting 2011).

## References

Banerjee A, Guo X, Wang H (2005) On the optimality of conditional expectation as a Bregman predictor. *IEEE Transactions on Information Theory* **51(7)**:2664–2669. doi:10.1109/TIT.2005.850145.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Patton AJ (2011) Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics* **160(1)**:246–256. doi:10.1016/j.jeconom.2010.03.034.

Savage LJ (1971) Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association* **66(336)**:783–801. doi:10.1080/01621459.1971.10482346.

## See Also

bregman4_rs, mean_if

## Examples

```
# Compute the Bregman scoring function (type 4).

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3
)

df$bregman4_penalty <- bregman4_sf(x = df$x, y = df$y)

print(df)
```

`capping_function` *Capping function*

## Description

The function capping_function computes the value of the capping function, defined in Taggart (2022), p.205.

It is used by the generalized Huber loss function among others (see Taggart 2022).

## Usage

```
capping_function(t, a, b)
```

## Arguments

| | |
|---|---|
| `t` | It can be a vector of length $n$. |
| `a` | It can be a vector of length $n$ (must have the same length as $t$). |
| `b` | It can be a vector of length $n$ (must have the same length as $t$). |

## Details

The capping function $\kappa_{a,b}(t)$ is defined by:

$$\kappa_{a,b}(t) := \max\{\min\{t, b\}, -a\}$$

or equivalently,

$$\kappa_{a,b}(t) := \begin{cases} -a, & t \leq -a \\ t, & -a < t \leq b \\ b, & t > b \end{cases}$$

Domain of function:

$$t \in \mathbb{R}$$

$$a \in [0, \infty]$$

$$b \in [0, \infty]$$

Range of function:

$$-a \leq \kappa_{a,b}(t) \leq b, \forall t \in \mathbb{R}, a, b \in [0, \infty]$$

## Value

Vector of values of the capping function.

## Note

For the definition of the capping function, see Taggart (2022), p.205.

Taggart (2022), p.205 admits the endpoints $a = \infty$ and $b = \infty$, at which the corresponding cap is removed; the examples below include those cases.

## References

Taggart RJ (2022) Point forecasting and forecast evaluation with generalized Huber loss. *Electronic Journal of Statistics* **16(1)**:201–231. doi:10.1214/21EJS1957.

## See Also

ghuber_sf, huber_sf

## Examples

```
# Compute the capping function.

df <- data.frame(
    t = c(1, -1, 1, -1, 1, -1, 1, -1, 1, 1, 2.5, 2.5, 3.5, 3.5),
    a = c(0, 0, 0, 0, Inf, Inf, Inf, Inf, 2, 3, 2, 3, 2, 3),
    b = c(0, 0, Inf, Inf, 0, 0, Inf, Inf, 3, 2, 3, 2, 3, 2)
)

df$cf <- capping_function(t = df$t, a = df$a, b = df$b)

print(df)
```

---

# errorspread_sf *Error - spread scoring function*

---

## Description

The function errorspread_sf computes the error - spread scoring function, when $y$ materialises, $x_1$ is the predictive mean, $x_2$ is the predictive variance and $x_3$ is the predictive skewness.

The error - spread scoring function is defined by eq. (14) in Christensen et al. (2015).

## Usage

```
errorspread_sf(x1, x2, x3, y)
```

## Arguments

| | |
|---|---|
| x1 | Predictive mean (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| x2 | Predictive variance (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| x3 | Predictive skewness (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x_1$, $x_2$ and $x_3$). |

## Details

The error - spread scoring function is defined by:

$$S(x_1, x_2, x_3, y) := (x_2 - (x_1 - y)^2 - (x_1 - y)x_2^{1/2}x_3)^2$$

Domain of function:

$$x_1 \in \mathbb{R}$$

$$x_2 > 0$$

$$x_3 \in \mathbb{R}$$

$$y \in \mathbb{R}$$

Range of function:

$$S(x_1, x_2, x_3, y) \geq 0, \forall x_1, x_3, y \in \mathbb{R}, x_2 > 0$$

## Value

Vector of error - spread losses.

## Note

The mean functional is the mean $\mathrm{E}_F[Y]$ of the probability distribution $F$ of $Y$ (Christensen et al. 2015).

The variance functional is the variance $\mathrm{Var}_F[Y] := \mathrm{E}_F[Y^2] - (\mathrm{E}_F[Y])^2$ of the probability distribution $F$ of $Y$ (Christensen et al. 2015).

The skewness functional is the skewness $\mathrm{Sk}_F[Y] := \mathrm{E}_F[((Y - \mathrm{E}_F[Y])/(\mathrm{Var}_F[Y])^{1/2})^3]$ (Christensen et al. 2015).

The error - spread scoring function is negatively oriented (i.e. the smaller, the better).

The error - spread scoring function is consistent for the triple (mean, variance, skewness) functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y^4]$ exists and is finite (Christensen et al. 2015).

## References

Christensen HM, Moroz IM, Palmer TN (2015) Evaluation of ensemble forecast uncertainty using a new proper score: Application to medium-range and seasonal forecasts. *Quarterly Journal of the Royal Meteorological Society* **141(687)(Part B)**:538–549. doi:10.1002/qj.2375.

## Examples

```
# Compute the error - spread scoring function.

df <- data.frame(
    y = rep(x = 0, times = 6),
    x1 = c(2, 2, -2, -2, 0, 0),
    x2 = c(1, 2, 1, 2, 1, 2),
    x3 = c(3, 3, -3, -3, 0, 0)
)

df$errorspread_penalty <- errorspread_sf(x1 = df$x1, x2 = df$x2, x3 = df$x3,
    y = df$y)

print(df)
```

---

# expectile_if *Expectile identification function*

---

## Description

The function expectile_if computes the expectile identification function at a specific level $p$, when $y$ materialises and $x$ is the predictive expectile at level $p$.

The expectile identification function is defined in Table 9 in Gneiting (2011).

## Usage

```
expectile_if(x, y, p)
```

## Arguments

| | |
|---|---|
| x | Predictive expectile (prediction) at level $p$. It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| p | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The expectile identification function is defined by:

$$V(x, y, p) := 2|\mathbf{1}\{x \geq y\} - p|(x - y)$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$0 < p < 1$$

Range of function:

$$V(x, y, p) \in \mathbb{R}$$

## Value

Vector of values of the expectile identification function.

## Note

For the definition of expectiles, see Newey and Powell (1987).

The expectile identification function is a strict $\mathbb{F}$-identification function for the $p$-expectile functional (Gneiting 2011; Fissler and Ziegel 2016; Dimitriadis et al. 2024).

$\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y]$ exists and is finite (Gneiting 2011; Fissler and Ziegel 2016; Dimitriadis et al. 2024).

## References

Dimitriadis T, Fissler T, Ziegel JF (2024) Osband's principle for identification functions. *Statistical Papers* **65**:1125–1132. doi:10.1007/s0036202301428x.

Fissler T, Ziegel JF (2016) Higher order elicitability and Osband's principle. *The Annals of Statistics* **44(4)**:1680–1707. doi:10.1214/16AOS1439.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Newey WK, Powell JL (1987) Asymmetric least squares estimation and testing. *Econometrica* **55(4)**:819–847. doi:10.2307/1911031.

## See Also

`expectile_sf`, `expectile_rs`

## Examples

```
# Compute the expectile identification function.

df <- data.frame(
    y = rep(x = 0, times = 6),
    x = c(2, 2, -2, -2, 0, 0),
    p = rep(x = c(0.05, 0.95), times = 3)
)

df$expectile_if <- expectile_if(x = df$x, y = df$y, p = df$p)

print(df)
```

---

# expectile_rs *Realised expectile score*

---

## Description

The function expectile_rs computes the realised expectile score at a specific level $p$ when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised expectile score is a realised score corresponding to the expectile scoring function expectile_sf.

## Usage

```
expectile_rs(x, y, p)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |
| p | It can be a scalar. |

## Details

The realised expectile score is defined by:

$$S(\boldsymbol{x}, \boldsymbol{y}, p) := (1/n) \sum_{i=1}^{n} L(x_i, y_i, p)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y, p) := |\mathbf{1}\{x \geq y\} - p|(x - y)^2$$

Domain of function:

$$\boldsymbol{x} \in \mathbb{R}^n$$

$$\boldsymbol{y} \in \mathbb{R}^n$$

$$0 < p < 1$$

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}, p) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, p \in (0, 1)$$

## Value

Value of the realised expectile score.

## Note

For details on the expectile scoring function, see expectile_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised expectile score is the realised (average) score corresponding to the expectile scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

`expectile_sf`, `expectile_if`

## Examples

```
# Compute the realised expectile score.

set.seed(12345)

x <- 0.5

y <- rnorm(n = 100, mean = 0, sd = 1)
```

```
print(expectile_rs(x = x, y = y, p = 0.7))

print(expectile_rs(x = rep(x = x, times = 100), y = y, p = 0.7))
```

---

`expectile_sf` *Asymmetric piecewise quadratic scoring function (expectile scoring function, expectile loss function)*

---

## Description

The function expectile_sf computes the asymmetric piecewise quadratic scoring function (expectile scoring function) at a specific level $p$, when $y$ materialises and $x$ is the predictive expectile at level $p$.

The asymmetric piecewise quadratic scoring function is defined by eq. (27) in Gneiting (2011).

## Usage

```
expectile_sf(x, y, p)
```

## Arguments

| | |
|---|---|
| x | Predictive expectile (prediction) at level $p$. It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| p | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The asymmetric piecewise quadratic scoring function is defined by:

$$S(x, y, p) := |\mathbf{1}\{x \geq y\} - p|(x - y)^2$$

or equivalently,

$$S(x, y, p) := p(\max\{-(x - y), 0\})^2 + (1 - p)(\max\{x - y, 0\})^2$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$0 < p < 1$$

Range of function:

$$S(x, y, p) \geq 0, \forall x, y \in \mathbb{R}, p \in (0, 1)$$

## Value

Vector of expectile losses.

## Note

For the definition of expectiles, see Newey and Powell (1987).

The asymmetric piecewise quadratic scoring function is negatively oriented (i.e. the smaller, the better).

The asymmetric piecewise quadratic scoring function is strictly $\mathbb{F}$-consistent for the $p$-expectile functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y]$ and $\mathrm{E}_F[Y^2]$ exist and are finite (Gneiting 2011).

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Newey WK, Powell JL (1987) Asymmetric least squares estimation and testing. *Econometrica* **55(4)**:819–847. doi:10.2307/1911031.

## See Also

expectile_rs, expectile_if

## Examples

```
# Compute the asymmetric piecewise quadratic scoring function (expectile scoring
# function).

df <- data.frame(
    y = rep(x = 0, times = 6),
    x = c(2, 2, -2, -2, 0, 0),
    p = rep(x = c(0.05, 0.95), times = 3)
)

df$expectile_penalty <- expectile_sf(x = df$x, y = df$y, p = df$p)

print(df)

# The asymmetric piecewise quadratic scoring function (expectile scoring
# function) at level p = 0.5 is half the squared error scoring function.

df <- data.frame(
    y = rep(x = 0, times = 3),
    x = c(-2, 0, 2),
    p = rep(x = c(0.5), times = 3)
```

```
)

df$expectile_penalty <- expectile_sf(x = df$x, y = df$y, p = df$p)

df$squared_error <- serr_sf(x = df$x, y = df$y)

print(df)
```

---

ghuber_rs *Realised generalized Huber score*

---

## Description

The function ghuber_rs computes the realised generalized Huber score at a specific level $p$ and parameters $a$ and $b$, when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised generalized Huber score is a realised score corresponding to the generalized Huber scoring function ghuber_sf.

## Usage

```
ghuber_rs(x, y, p, a, b)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |
| p | It can be a scalar. |
| a | It can be a value. |
| b | It can be a value. |

## Details

The realised generalized Huber score is defined by:

$$S(\boldsymbol{x}, \boldsymbol{y}, p, a, b) := (1/n) \sum_{i=1}^{n} L(x_i, y_i, p, a, b)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y, p, a, b) := |\mathbf{1}\{x \geq y\} - p| f_{a,b}(x - y)$$

where

$$f_{a,b}(t) := \kappa_{a,b}(t)(2t - \kappa_{a,b}(t))$$

and $\kappa_{a,b}(t)$ is the capping function defined by:

$$\kappa_{a,b}(t) := \max\{\min\{t, b\}, -a\}$$

Domain of function:

$$\boldsymbol{x} \in \mathbb{R}^n$$

$$\boldsymbol{y} \in \mathbb{R}^n$$

$$0 < p < 1$$

$$a > 0$$

$$b > 0$$

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}, p, a, b) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, p \in (0, 1), a, b > 0$$

## Value

Value of the realised generalized Huber score.

## Note

For details on the generalized Huber scoring function, see ghuber_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised generalized Huber score is the realised (average) score corresponding to the generalized Huber scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

ghuber_sf, huberquantile_if

## Examples

```
# Compute the realised generalized Huber score.

set.seed(12345)

p <- 0.7

a <- 0.5

b <- 0.5

x <- 0

y <- rnorm(n = 100, mean = 0, sd = 1)

print(ghuber_rs(x = x, y = y, p = p, a = a, b = b))

print(ghuber_rs(x = rep(x = x, times = 100), y = y, p = p, a = a, b = b))
```

---

# ghuber_sf *Generalized Huber scoring function*

---

## Description

The function ghuber_sf computes the generalized Huber scoring function at a specific level $p$ and parameters $a$ and $b$, when $y$ materialises and $x$ is the predictive Huber functional at level $p$.

The generalized Huber scoring function is defined by eq. (4.7) in Taggart (2022) for $\phi(t) = t^2$.

## Usage

```
ghuber_sf(x, y, p, a, b)
```

## Arguments

| | |
|---|---|
| x | Predictive Huber functional (prediction) at level $p$. It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| p | It can be a vector of length $n$ (must have the same length as $y$). |
| a | It can be a vector of length $n$ (must have the same length as $y$). |
| b | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The generalized Huber scoring function is defined by:

$$S(x, y, p, a, b) := |\mathbf{1}\{x \geq y\} - p|(y^2 - (\kappa_{a,b}(x - y) + y)^2 + 2x\kappa_{a,b}(x - y))$$

or equivalently

$$S(x, y, p, a, b) := |\mathbf{1}\{x \geq y\} - p| f_{a,b}(x - y)$$

or

$$S(x, y, p, a, b) := p f_{a,b}(-\max\{-(x - y), 0\}) + (1 - p) f_{a,b}(\max\{x - y, 0\})$$

where

$$f_{a,b}(t) := \kappa_{a,b}(t)(2t - \kappa_{a,b}(t))$$

and $\kappa_{a,b}(t)$ is the capping function defined by:

$$\kappa_{a,b}(t) := \max\{\min\{t, b\}, -a\}$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$0 < p < 1$$

$$a \in (0, \infty]$$

$$b \in (0, \infty]$$

Range of function:

$$S(x, y, p, a, b) \geq 0, \forall x, y \in \mathbb{R}, p \in (0, 1), a, b \in (0, \infty]$$

## Value

Vector of generalized Huber losses.

## Note

For the definition of Huber functionals, see definition 3.3 in Taggart (2022). The value of eq. (4.7) is twice the value of the equation in definition 4.2 in Taggart (2022).

Definition 3.3 in Taggart (2022) is given for finite $a > 0$ and $b > 0$. The endpoints $a = \infty$ and $b = \infty$ are admitted here through the capping function, which is defined on $[0, \infty]$ at p.205 in Taggart (2022); at $a = b = \infty$ the function equals expectile_sf, the $a \to \infty$ limit of part 5 of proposition 3.4 in Taggart (2022).

The generalized Huber scoring function is negatively oriented (i.e. the smaller, the better).

The generalized Huber scoring function is strictly $\mathbb{F}$-consistent for the $p$-Huber functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y^2 - (Y - a)^2]$ and $\mathrm{E}_F[Y^2 - (Y + b)^2]$ (or equivalently $\mathrm{E}_F[Y]$) exist and are finite (Taggart 2022).

## References

Taggart RJ (2022) Point forecasting and forecast evaluation with generalized Huber loss. *Electronic Journal of Statistics* **16(1)**:201–231. doi:10.1214/21EJS1957.

## See Also

ghuber_rs, huberquantile_if

## Examples

```
# Compute the generalized Huber scoring function.

set.seed(12345)

n <- 10

df <- data.frame(
    x = runif(n = n, min = -2, max = 2),
    y = runif(n = n, min = -2, max = 2),
    p = runif(n = n, min = 0, max = 1),
    a = runif(n = n, min = 0, max = 1),
    b = runif(n = n, min = 0, max = 1)
)

df$ghuber_penalty <- ghuber_sf(x = df$x, y = df$y, p = df$p, a = df$a, b = df$b)

print(df)

# Equivalence of the generalized Huber scoring function and the asymmetric
# piecewise quadratic scoring function (expectile scoring function), when
# a = Inf and b = Inf.

set.seed(12345)

n <- 100

x <- runif(n = n, min = -20, max = 20)
```

```
y <- runif(n = n, min = -20, max = 20)
p <- runif(n = n, min = 0, max = 1)
a <- rep(x = Inf, times = n)
b <- rep(x = Inf, times = n)

u <- ghuber_sf(x = x, y = y, p = p, a = a, b = b)
v <- expectile_sf(x = x, y = y, p = p)

max(abs(u - v)) # values are slightly higher than 0 due to rounding error
min(abs(u - v))

# Equivalence of the generalized Huber scoring function and the Huber scoring
# function when p = 1/2 and a = b.

set.seed(12345)

n <- 100

x <- runif(n = n, min = -20, max = 20)
y <- runif(n = n, min = -20, max = 20)
p <- rep(x = 1/2, times = n)
a <- runif(n = n, min = 0, max = 20)

u <- ghuber_sf(x = x, y = y, p = p, a = a, b = a)
v <- huber_sf(x = x, y = y, a = a)

max(abs(u - v)) # values are slightly higher than 0 due to rounding error
min(abs(u - v))
```

---

`gpl1_rs` *Realised generalized piecewise linear power score (type 1)*

---

## Description

The function gpl1_rs computes the realised generalized piecewise linear power score (type 1) at a specific level $p$ and parameter $b$, when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised generalized piecewise linear power score (type 1) is a realised score corresponding to the generalized piecewise linear power scoring function (type 1) gpl1_sf.

## Usage

```
gpl1_rs(x, y, p, b)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |
| p | It can be a scalar. |
| b | It can be a scalar. |

## Details

The realised generalized piecewise linear power score (type 1) is defined by:

$$S(\boldsymbol{x},\boldsymbol{y},p,b) := (1/n)\sum_{i=1}^{n} L(x_i,y_i,p,b)$$

where

$$\boldsymbol{x} = (x_1,...,x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1,...,y_n)^\mathsf{T}$$

and

$$L(x,y,p,b) := (1/b)(\mathbf{1}\{x \geq y\} - p)(x^b - y^b)$$

Domain of function:

$$\boldsymbol{x} > \mathbf{0}$$

$$\boldsymbol{y} > \mathbf{0}$$

$$0 < p < 1$$

$$b > 0$$

where

$$\mathbf{0} = (0,...,0)^\mathsf{T}$$

is the zero vector of length $n$ and the symbol $>$ indicates pairwise inequality.

Range of function:

$$S(\boldsymbol{x},\boldsymbol{y},p,b) \geq 0, \forall \boldsymbol{x},\boldsymbol{y} > \mathbf{0}, p \in (0,1), b > 0$$

## Value

Value of the realised generalized piecewise linear power score (type 1).

## Note

For details on the generalized piecewise linear power scoring function (type 1), see gpl1_sf.

The parameter is restricted to $b > 0$, where $|b| = b$, because $g(x) = x^b/|b|$ is strictly increasing only for $b > 0$, as required by Theorem 9(b) in Gneiting (2011). See gpl1_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised generalized piecewise linear power score (type 1) is the realised (average) score corresponding to the generalized piecewise linear power scoring function (type 1).

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

gpl1_sf, quantile_if

## Examples

```
# Compute the realised generalized piecewise linear power score (type 1).

set.seed(12345)

p <- 0.7

b <- 2

x <- qlnorm(p = p, meanlog = 0, sdlog = 1)

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(gpl1_rs(x = x, y = y, p = p, b = b))

print(gpl1_rs(x = rep(x = x, times = 100), y = y, p = p, b = b))
```

---

# gpl1_sf *Generalized piecewise linear power scoring function (type 1)*

---

## Description

The function gpl1_sf computes the generalized piecewise linear power scoring function at a specific level $p$ for $g(x) = x^b/|b|$, $b > 0$, when $y$ materialises and $x$ is the predictive quantile at level $p$.

The generalized piecewise linear power scoring function is defined by eq. (25) in Gneiting (2011) and the form implemented here for the specific $g(x)$ is defined by eq. (26) in Gneiting (2011).

## Usage

```
gpl1_sf(x, y, p, b)
```

## Arguments

| | |
|---|---|
| x | Predictive quantile (prediction) at level $p$. It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| p | It can be a vector of length $n$ (must have the same length as $y$). |
| b | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The generalized piecewise linear power scoring function (type 1) is defined by:

$$S(x, y, p, b) := (1/b)(\mathbf{1}\{x \geq y\} - p)(x^b - y^b)$$

or equivalently

$$S(x, y, p, b) := (1/b)(p|\max\{-(x^b - y^b), 0\}| + (1 - p)|\max\{x^b - y^b, 0\}|)$$

Domain of function:

$$x > 0$$

$$y > 0$$

$$0 < p < 1$$

$$b > 0$$

Range of function:

$$S(x, y, p, b) \geq 0, \forall x, y > 0, p \in (0, 1), b > 0$$

## Value

Vector of generalized piecewise linear power losses.

## Note

The implemented function is named type 1 because it corresponds to a specific type of $g(x)$ of the general form of the generalized piecewise linear power scoring function defined by eq. (25) in Gneiting (2011).

Eq. (26) in Gneiting (2011) is given for $b \in \mathbb{R} \setminus \{0\}$. The implementation herein deliberately restricts the domain to $b > 0$, where $|b| = b$, because $g(x) = x^b/|b|$ is strictly increasing only for $b > 0$, as required by Theorem 9(b) in Gneiting (2011). The two forms given above are equivalent on $b > 0$ only; they disagree in sign for $b < 0$, which is outside the domain of the function. Because no argument validation is performed, a call with $b < 0$ returns a value rather than an error.

For the definition of quantiles, see Koenker and Bassett Jr (1978).

The generalized piecewise linear power scoring function is negatively oriented (i.e. the smaller, the better).

The herein implemented generalized piecewise linear power scoring function is strictly $\mathbb{F}$-consistent for the $p$-quantile functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y^b]$ exists and is finite (Thomson 1979; Saerens 2000; Gneiting 2011).

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Koenker R, Bassett Jr G (1978) Regression quantiles. *Econometrica* **46(1)**:33–50. doi:10.2307/1913643.

Saerens M (2000) Building cost functions minimizing to some summary statistics. *IEEE Transactions on Neural Networks* **11(6)**:1263–1271. doi:10.1109/72.883416.

Thomson W (1979) Eliciting production possibilities from a well-informed manager. *Journal of Economic Theory* **20(3)**:360–380. doi:10.1016/00220531(79)900425.

## See Also

gpl1_rs, quantile_if

## Examples

```
# Compute the generalized piecewise linear scoring function (type 1).

df <- data.frame(
    y = rep(x = 2, times = 6),
    x = c(1, 2, 3, 1, 2, 3),
    p = c(rep(x = 0.05, times = 3), rep(x = 0.95, times = 3)),
    b = rep(x = 2, times = 6)
)

df$gpl1_penalty <- gpl1_sf(x = df$x, y = df$y, p = df$p, b = df$b)

print(df)

# Equivalence of generalized piecewise linear scoring function (type 1) and
# asymmetric piecewise linear scoring function (quantile scoring function), when
```

```
# b = 1.

set.seed(12345)

n <- 100

x <- runif(n = n, min = 0, max = 20)
y <- runif(n = n, min = 0, max = 20)
p <- runif(n = n, min = 0, max = 1)
b <- rep(x = 1, times = n)

u <- gpl1_sf(x = x, y = y, p = p, b = b)
v <- quantile_sf(x = x, y = y, p = p)

max(abs(u - v))

# Equivalence of generalized piecewise linear scoring function (type 1) and
# MAE-SD scoring function, when p = 1/2 and b = 1/2.

set.seed(12345)

n <- 100

x <- runif(n = n, min = 0, max = 20)
y <- runif(n = n, min = 0, max = 20)
p <- rep(x = 0.5, times = n)
b <- rep(x = 1/2, times = n)

u <- gpl1_sf(x = x, y = y, p = p, b = b)
v <- maesd_sf(x = x, y = y)

max(abs(u - v))
```

---

gpl2_rs *Realised generalized piecewise linear power score (type 2)*

---

## Description

The function gpl2_rs computes the realised generalized piecewise linear power score (type 2) at a specific level $p$, when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised generalized piecewise linear power score (type 2) is a realised score corresponding to the generalized piecewise linear power scoring function (type 2) gpl2_sf.

## Usage

```
gpl2_rs(x, y, p)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |
| p | It can be a scalar. |

## Details

The realised generalized piecewise linear power score (type 2) is defined by:

$$S(\boldsymbol{x}, \boldsymbol{y}, p) := (1/n) \sum_{i=1}^{n} L(x_i, y_i, p)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y, p) := (\mathbf{1}\{x \geq y\} - p) \log(x/y)$$

Domain of function:

$$\boldsymbol{x} > \mathbf{0}$$

$$\boldsymbol{y} > \mathbf{0}$$

$$0 < p < 1$$

where

$$\mathbf{0} = (0, ..., 0)^\mathsf{T}$$

is the zero vector of length $n$ and the symbol $>$ indicates pairwise inequality.

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}, p) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} > \mathbf{0}, p \in (0, 1)$$

## Value

Value of the realised generalized piecewise linear power score (type 2).

## Note

For details on the generalized piecewise linear power scoring function (type 2), see gpl2_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised generalized piecewise linear power score (type 2) is the realised (average) score corresponding to the generalized piecewise linear power scoring function (type 2).

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

gpl2_sf, quantile_if

## Examples

```
# Compute the realised generalized piecewise linear power score (type 2).

set.seed(12345)

p <- 0.7

x <- qlnorm(p = p, meanlog = 0, sdlog = 1)

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(gpl2_rs(x = x, y = y, p = p))

print(gpl2_rs(x = rep(x = x, times = 100), y = y, p = p))
```

---

# gpl2_sf *Generalized piecewise linear power scoring function (type 2)*

---

## Description

The function gpl2_sf computes the generalized piecewise linear power scoring function at a specific level $p$ for $g(x) = \log(x)$, when $y$ materialises and $x$ is the predictive quantile at level $p$.

The generalized piecewise linear power scoring function is defined by eq. (25) in Gneiting (2011) and the form implemented here for the specific $g(x)$ is defined by eq. (26) in Gneiting (2011) for $b = 0$.

## Usage

```
gpl2_sf(x, y, p)
```

## Arguments

| | |
|---|---|
| x | Predictive quantile (prediction) at level $p$. It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| p | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The generalized piecewise linear power scoring function (type 2) is defined by:

$$S(x, y, p) := (\mathbf{1}\{x \geq y\} - p) \log(x/y)$$

or equivalently

$$S(x, y, p) := p|\max\{-(\log(x) - \log(y)), 0\}| + (1 - p)|\max\{\log(x) - \log(y), 0\}|$$

Domain of function:

$$x > 0$$

$$y > 0$$

$$0 < p < 1$$

Range of function:

$$S(x, y, p) \geq 0, \forall x, y > 0, p \in (0, 1)$$

## Value

Vector of generalized piecewise linear power losses.

## Note

The implemented function is named type 2 because it corresponds to a specific type of $g(x)$ of the general form of the generalized piecewise linear power scoring function defined by eq. (25) in Gneiting (2011).

For the definition of quantiles, see Koenker and Bassett Jr (1978).

The generalized piecewise linear power scoring function is negatively oriented (i.e. the smaller, the better).

The herein implemented generalized piecewise linear power scoring function is strictly $\mathbb{F}$-consistent for the $p$-quantile functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[\log(Y)]$ exists and is finite (Thomson 1979; Saerens 2000; Gneiting 2011).

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Koenker R, Bassett Jr G (1978) Regression quantiles. *Econometrica* **46(1)**:33–50. doi:10.2307/1913643.

Saerens M (2000) Building cost functions minimizing to some summary statistics. *IEEE Transactions on Neural Networks* **11(6)**:1263–1271. doi:10.1109/72.883416.

Thomson W (1979) Eliciting production possibilities from a well-informed manager. *Journal of Economic Theory* **20(3)**:360–380. doi:10.1016/00220531(79)900425.

## See Also

gpl2_rs, quantile_if

## Examples

```
# Compute the generalized piecewise linear scoring function (type 2).

df <- data.frame(
    y = rep(x = 2, times = 6),
    x = c(1, 2, 3, 1, 2, 3),
    p = c(rep(x = 0.05, times = 3), rep(x = 0.95, times = 3))
)

df$gpl2_penalty <- gpl2_sf(x = df$x, y = df$y, p = df$p)

print(df)

# The generalized piecewise linear scoring function (type 2) is half the MAE-LOG
# scoring function, when p = 1/2.

df <- data.frame(
    y = rep(x = 5.5, times = 10),
    x = 1:10,
    p = rep(x = 0.5, times = 10)
)

df$gpl2_penalty <- gpl2_sf(x = df$x, y = df$y, p = df$p)

df$mae_log_penalty <- maelog_sf(x = df$x, y = df$y)

df$ratio <- df$gpl2_penalty/df$mae_log_penalty

print(df)
```

`hubermean_if` *Huber mean identification function*

## Description

The function hubermean_if computes the Huber mean identification function with parameter $a$, when $y$ materialises and $x$ is the predictive Huber mean.

The Huber mean identification function is defined by eq. (3.5) in Taggart (2022), for $\alpha = 1/2$ and $b = a$.

## Usage

```
hubermean_if(x, y, a)
```

## Arguments

| | |
|---|---|
| `x` | Predictive Huber mean (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| `y` | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| `a` | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The Huber mean identification function is defined by:

$$V(x, y, a) := (1/2)\kappa_{a,a}(x - y)$$

where $\kappa_{a,b}(t)$ is the capping function defined by:

$$\kappa_{a,b}(t) := \max\{\min\{t, b\}, -a\}$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$a > 0$$

Range of function:

$$-a/2 \leq V(x, y, a) \leq a/2, \forall x, y \in \mathbb{R}, a > 0$$

## Value

Vector of values of the Huber mean identification function.

## Note

For the definition of Huber mean, see Taggart (2022).

Eq. (3.5) in Taggart (2022) defines the general Huber quantile identification function $V(x, y, \alpha, a, b) := |\mathbf{1}\{x \geq y\} - \alpha|\kappa_{a,b}(x - y)$, implemented in huberquantile_if. The Huber mean identification function is its specialisation to $\alpha = 1/2$ and $b = a$, so that `huberquantile_if(x, y, 1/2, a, a)` and `hubermean_if(x, y, a)` agree.

The Huber mean identification function is a strict $\mathbb{F}$-identification function for the Huber mean functional (Taggart 2022).

$\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y]$ exists and is finite (Taggart 2022).

## References

Taggart RJ (2022) Point forecasting and forecast evaluation with generalized Huber loss. *Electronic Journal of Statistics* **16(1)**:201–231. doi:10.1214/21EJS1957.

## See Also

huber_sf, huber_rs

## Examples

```
# Compute the Huber mean identification function.

df <- data.frame(
    x = c(-3, -2, -1, 0, 1, 2, 3),
    y = c(0, 0, 0, 0, 0, 0, 0),
    a = c(2.7, 2.5, 0.6, 0.7, 0.9, 1.2, 5)
)

df$hubermean_if <- hubermean_if(x = df$x, y = df$y, a = df$a)

print(df)
```

---

`huberquantile_if` *Huber quantile identification function*

---

## Description

The function huberquantile_if computes the Huber quantile identification function at a specific level $p$ and parameters $a$ and $b$, when $y$ materialises and $x$ is the predictive Huber functional at level $p$.

The Huber quantile identification function is defined by eq. (3.5) in Taggart (2022), in its general form.

## Usage

```
huberquantile_if(x, y, p, a, b)
```

## Arguments

| | |
|---|---|
| x | Predictive Huber functional (prediction) at level $p$. It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| p | It can be a vector of length $n$ (must have the same length as $y$). |
| a | It can be a vector of length $n$ (must have the same length as $y$). |
| b | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The Huber quantile identification function is defined by:

$$V(x, y, p, a, b) := |\mathbf{1}\{x \geq y\} - p|\kappa_{a,b}(x - y)$$

where $\kappa_{a,b}(t)$ is the capping function defined by:

$$\kappa_{a,b}(t) := \max\{\min\{t, b\}, -a\}$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$0 < p < 1$$

$$a > 0$$

$$b > 0$$

Range of function:

$$-pa \leq V(x, y, p, a, b) \leq (1 - p)b, \forall x, y \in \mathbb{R}, p \in (0, 1), a, b > 0$$

## Value

Vector of values of the Huber quantile identification function.

## Note

For the definition of Huber quantile, see Taggart (2022).

At $p = 1/2$ and $b = a$ the Huber quantile identification function reduces to the Huber mean identification function hubermean_if.

The Huber quantile identification function is a strict $\mathbb{F}$-identification function for the Huber quantile functional (Taggart 2022).

$\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y]$ exists and is finite (Taggart 2022).

## References

Taggart RJ (2022) Point forecasting and forecast evaluation with generalized Huber loss. *Electronic Journal of Statistics* **16(1)**:201–231. doi:10.1214/21EJS1957.

## See Also

ghuber_sf, ghuber_rs

## Examples

```
# Compute the Huber quantile identification function.

set.seed(12345)

n <- 10

df <- data.frame(
    x = runif(n = n, min = -2, max = 2),
    y = runif(n = n, min = -2, max = 2),
    p = runif(n = n, min = 0, max = 1),
    a = runif(n = n, min = 0, max = 1),
    b = runif(n = n, min = 0, max = 1)
)

df$huberquantile_if <- huberquantile_if(x = df$x, y = df$y, p = df$p, a = df$a,
    b = df$b)

print(df)
```

---

# huber_rs *Realised Huber score*

---

## Description

The function huber_rs computes the realised Huber score with parameter $a$, when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised Huber score is a realised score corresponding to the Huber scoring function huber_sf.

## Usage

```
huber_rs(x, y, a)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |
| a | It can be a scalar. |

## Details

The realised Huber score is defined by:

$$S(\boldsymbol{x}, \boldsymbol{y}, a) := (1/n) \sum_{i=1}^{n} L(x_i, y_i, a)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y, a) := \begin{cases} \frac{1}{2}(x-y)^2, & |x-y| \leq a \\ a|x-y| - \frac{1}{2}a^2, & |x-y| > a \end{cases}$$

Domain of function:

$$\boldsymbol{x} \in \mathbb{R}^n$$

$$\boldsymbol{y} \in \mathbb{R}^n$$

$$a > 0$$

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}, a) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, a > 0$$

## Value

Value of the realised Huber score.

## Note

For details on the Huber scoring function, see huber_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised Huber score is the realised (average) score corresponding to the Huber scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

huber_sf, hubermean_if

## Examples

```
# Compute the realised Huber score.

set.seed(12345)

a <- 0.5

x <- 0

y <- rnorm(n = 100, mean = 0, sd = 1)

print(huber_rs(x = x, y = y, a = a))

print(huber_rs(x = rep(x = x, times = 100), y = y, a = a))
```

---

# huber_sf *Huber scoring function*

---

## Description

The function huber_sf computes the Huber scoring function with parameter $a$, when $y$ materialises and $x$ is the predictive Huber mean.

The Huber scoring function is defined in Huber (1964).

## Usage

```
huber_sf(x, y, a)
```

## Arguments

| | |
|---|---|
| x | Predictive Huber mean (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| a | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The Huber scoring function is defined by:

$$S(x, y, a) := \begin{cases} \frac{1}{2}(x-y)^2, & |x-y| \leq a \\ a|x-y| - \frac{1}{2}a^2, & |x-y| > a \end{cases}$$

or equivalently

$$S(x, y, a) := (1/2)\kappa_{a,a}(x-y)(2(x-y) - \kappa_{a,a}(x-y))$$

where $\kappa_{a,b}(t)$ is the capping function defined by:

$$\kappa_{a,b}(t) := \max\{\min\{t, b\}, -a\}$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$a > 0$$

Range of function:

$$S(x, y, a) \geq 0, \forall x, y \in \mathbb{R}, a > 0$$

## Value

Vector of Huber losses.

## Note

For the definition of Huber mean, see Taggart (2022).

The Huber scoring function is negatively oriented (i.e. the smaller, the better).

The Huber scoring function is strictly $\mathbb{F}$-consistent for the Huber mean. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y^2 - (Y-a)^2]$ and $\mathrm{E}_F[Y^2 - (Y+a)^2]$ (or equivalently $\mathrm{E}_F[Y]$) exist and are finite (Taggart 2022).

## References

Huber PJ (1964) Robust estimation of a location parameter. *Annals of Mathematical Statistics* **35(1)**:73–101. doi:10.1214/aoms/1177703732.

Taggart RJ (2022) Point forecasting and forecast evaluation with generalized Huber loss. *Electronic Journal of Statistics* **16(1)**:201–231. doi:10.1214/21EJS1957.

## See Also

huber_rs, hubermean_if,

## Examples

```
# Compute the Huber scoring function.

df <- data.frame(
    x = c(-3, -2, -1, 0, 1, 2, 3),
    y = c(0, 0, 0, 0, 0, 0, 0),
    a = c(2.7, 2.5, 0.6, 0.7, 0.9, 1.2, 5)
)

df$huber_penalty <- huber_sf(x = df$x, y = df$y, a = df$a)

print(df)
```

---

# interval_sf *Interval scoring function (Winkler scoring function)*

---

## Description

The function interval_sf computes the interval scoring function (Winkler scoring function) when $y$ materialises and $[x_1, x_2]$ is the central $1 - p$ prediction interval.

The interval scoring function is defined by eq. (43) in Gneiting and Raftery (2007).

## Usage

```
interval_sf(x1, x2, y, p)
```

## Arguments

| | |
|---|---|
| x1 | Predictive quantile (prediction) at level $p/2$. It can be a vector of length $n$ (must have the same length as $y$). |
| x2 | Predictive quantile (prediction) at level $1 - p/2$. It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x_1$ and $x_2$). |
| p | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The interval scoring function is defined by:

$$S(x_1, x_2, y, p) := (x_2 - x_1) + (2/p)(x_1 - y)\mathbf{1}\{y < x_1\} + (2/p)(y - x_2)\mathbf{1}\{y > x_2\}$$

Domain of function:

$$x_1 \in \mathbb{R}$$

$$x_2 \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$0 < p < 1$$

Range of function:

$$S(x_1, x_2, y, p) \geq 0, \forall x_1, x_2, y \in \mathbb{R}, x_1 < x_2, p \in (0, 1)$$

## Value

Vector of interval losses.

## Note

For the definition of quantiles, see Koenker and Bassett Jr (1978).

The interval scoring function is negatively oriented (i.e. the smaller, the better).

The interval scoring function is strictly $\mathbb{F}$-consistent for the central $1-p$ prediction interval $[x_1, x_2]$. $x_1$ and $x_2$ are quantile functionals at levels $p/2$ and $1 - p/2$ respectively.

$\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y]$ exists and is finite (Dunsmore 1968; Winkler 1972; Gneiting and Raftery 2007; Winkler and Murphy 1979; Fissler and Ziegel 2016; Brehmer and Gneiting 2021).

## References

Brehmer JR, Gneiting T (2021) Scoring interval forecasts: Equal-tailed, shortest, and modal interval. *Bernoulli* **27(3)**:1993–2010. doi:10.3150/20BEJ1298.

Dunsmore IR (1968) A Bayesian approach to calibration. *Journal of the Royal Statistical Society, Series B (Methodological)* **30(2)**:396–405. doi:10.1111/j.25176161.1968.tb00740.x.

Fissler T, Ziegel JF (2016) Higher order elicitability and Osband's principle. *The Annals of Statistics* **44(4)**:1680–1707. doi:10.1214/16AOS1439.

Gneiting T, Raftery AE (2007) Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* **102(477)**:359–378. doi:10.1198/016214506000001437.

Koenker R, Bassett Jr G (1978) Regression quantiles. *Econometrica* **46(1)**:33–50. doi:10.2307/1913643.

Winkler RL (1972) A decision-theoretic approach to interval estimation. *Journal of the American Statistical Association* **67(337)**:187–191. doi:10.1080/01621459.1972.10481224.

Winkler RL, Murphy AH (1979) The use of probabilities in forecasts of maximum and minimum temperatures. *Meteorological Magazine* **108(1288)**:317–329.

## Examples

```
# Compute the interval scoring function (Winkler scoring function).

df <- data.frame(
    y = rep(x = 0, times = 6),
    x1 = c(-3, -2, -1, 0, 1, 2),
    x2 = c(1, 2, 3, 4, 5, 6),
    p = rep(x = c(0.05, 0.95), times = 3)
)

df$interval_penalty <- interval_sf(x1 = df$x1, x2 = df$x2, y = df$y, p = df$p)

print(df)
```

---

# linex_rs *Realised LINEX score*

---

## Description

The function linex_rs computes the realised LINEX score with parameter $a$, when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised LINEX score is a realised score corresponding to the LINEX scoring function linex_sf.

## Usage

```
linex_rs(x, y, a)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |
| a | It can be a scalar. |

## Details

The realised LINEX score is defined by:

$$S(\boldsymbol{x}, \boldsymbol{y}, a) := (1/n) \sum_{i=1}^{n} L(x_i, y_i, a)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y, a) := \mathrm{e}^{a(x-y)} - a(x - y) - 1$$

Domain of function:

$$\boldsymbol{x} \in \mathbb{R}^n$$

$$\boldsymbol{y} \in \mathbb{R}^n$$

$$a \neq 0$$

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}, a) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, a \neq 0$$

## Value

Value of the realised LINEX score.

## Note

For details on the LINEX scoring function, see linex_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised LINEX score is the realised (average) score corresponding to the LINEX scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

linex_sf

## Examples

```
# Compute the realised LINEX score.

set.seed(12345)

a <- 1

x <- 0

y <- rnorm(n = 100, mean = 0, sd = 1)

print(linex_rs(x = x, y = y, a = a))

print(linex_rs(x = rep(x = x, times = 100), y = y, a = a))
```

---

# linex_sf *LINEX scoring function*

---

## Description

The function linex_sf computes the LINEX scoring function with parameter $a$ when $y$ materialises and $x$ is the predictive $-(1/a)\log \mathrm{E}_F[\mathrm{e}^{-aY}]$ entropic risk measure (Gerber 1974).

The LINEX scoring function is defined by Varian (1975).

## Usage

```
linex_sf(x, y, a)
```

## Arguments

| | |
|---|---|
| x | Predictive $-(1/a)\log \mathrm{E}_F[\mathrm{e}^{-aY}]$ entropic risk measure (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| a | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The LINEX scoring function is defined by:

$$S(x, y, a) := \mathrm{e}^{a(x-y)} - a(x-y) - 1$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$a \neq 0$$

Range of function:

$$S(x, y, a) \geq 0, \forall x, y \in \mathbb{R}, a \neq 0$$

## Value

Vector of LINEX losses.

## Note

For details on the LINEX scoring function, see Varian (1975) and Zellner (1986).

The LINEX scoring function is negatively oriented (i.e. the smaller, the better).

The LINEX scoring function is strictly $\mathbb{F}$-consistent for the $-(1/a) \log \mathrm{E}_F[\mathrm{e}^{-aY}]$ entropic risk measure. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[\mathrm{e}^{-aY}]$ and $\mathrm{E}_F[Y]$ exist and are finite (Varian 1975; Zellner 1986; Gneiting 2011).

The functional elicited here is the entropic risk measure of serrexp_sf at the opposite sign of the parameter: serrexp_sf elicits $(1/a) \log(\mathrm{E}_F[\exp(aY)])$, whereas the LINEX scoring function with parameter $a$ elicits $-(1/a) \log(\mathrm{E}_F[\exp(-aY)])$, which is the same one-parameter family evaluated at $-a$.

## References

Gerber HU (1974) On additive premium calculation principles. *ASTIN Bulletin: The Journal of the IAA* **7(3)**:215–222. doi:10.1017/S0515036100006061.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Varian HR (1975) A Bayesian approach to real estate assessment. In: Fienberg SE, Zellner A (eds) *Studies in Bayesian Econometrics and Statistics in Honor of Leonard J. Savage*. Amsterdam: North-Holland, pp 195–208.

Zellner A (1986) Bayesian estimation and prediction using asymmetric loss functions. *Journal of the American Statistical Association* **81(394)**:446–451. doi:10.1080/01621459.1986.10478289.

## See Also

linex_rs

## Examples

```
# Compute the LINEX scoring function.

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3,
    a = c(-1, 1, 2)
)

df$linex_loss <- linex_sf(x = df$x, y = df$y, a = df$a)

print(df)
```

---

`lqmean_rs` *Realised L_q-mean score*

---

## Description

The function lqmean_rs computes the realised $L_q$-mean score with parameter $q$, when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised $L_q$-mean score is a realised score corresponding to the $L_q$-mean scoring function lqmean_sf.

## Usage

```
lqmean_rs(x, y, q)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |
| q | It can be a scalar. |

## Details

The realised $L_q$-mean score is defined by:

$$S(\boldsymbol{x}, \boldsymbol{y}, q) := (1/n) \sum_{i=1}^{n} L(x_i, y_i, q)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y, q) := |x - y|^q$$

Domain of function:

$$\boldsymbol{x} \in \mathbb{R}^n$$

$$\boldsymbol{y} \in \mathbb{R}^n$$

$$q > 1$$

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}, q) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, q > 1$$

## Value

Value of the realised $L_q$-mean score.

## Note

For details on the $L_q$-mean scoring function, see lqmean_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised $L_q$-mean score is the realised (average) score corresponding to the $L_q$-mean scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

lqmean_sf

## Examples

```
# Compute the realised Lq-mean score.

set.seed(12345)

q <- 2

x <- 0
```

```
y <- rnorm(n = 100, mean = 0, sd = 1)

print(lqmean_rs(x = x, y = y, q = q))

print(lqmean_rs(x = rep(x = x, times = 100), y = y, q = q))
```

---

`lqmean_sf` *$L\_q$-mean scoring function*

---

## Description

The function lqmean_sf computes the $L_q$-mean scoring function, when $y$ materialises and $x$ is the predictive $L_q$-mean.

The $L_q$-mean scoring function is defined by Chen (1996). It is equivalent to the $L_q$-quantile scoring function at level $p = 1/2$, up to a multiplicative constant.

## Usage

```
lqmean_sf(x, y, q)
```

## Arguments

| | |
|---|---|
| `x` | Predictive $L_q$-mean. It can be a vector of length $n$ (must have the same length as $y$). |
| `y` | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| `q` | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The $L_q$-mean scoring function is defined by:

$$S(x, y, q) := |x - y|^q$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$q > 1$$

Range of function:

$$S(x, y, q) \geq 0, \forall x, y \in \mathbb{R}, q > 1$$

## Value

Vector of $L_q$-mean losses.

## Note

For the definition of $L_q$-means, see Chen (1996). In particular, $L_q$-means are the solution of the equation $\mathrm{E}_F[V(x, Y, q)] = 0$, where

$$V(x, y, q) := q\text{sign}(x - y)|x - y|^{q-1}$$

$L_q$-means are $L_q$-quantiles at level $p = 1/2$.

The parameter is restricted to $q > 1$. At $q = 1$ the scoring function is the absolute error scoring function aerr_sf, which elicits the median. The restriction to $q > 1$ makes the expected score strictly convex, so its minimiser is unique; at $q = 1$ the minimiser is in general set-valued (Section 2, p. 108 in Chen 1996).

The $L_q$-mean scoring function is negatively oriented (i.e. the smaller, the better).

The $L_q$-mean scoring function is strictly $\mathbb{F}$-consistent for the $L_q$-mean functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[|Y|^q]$ exists and is finite (Chen 1996; Bellini et al. 2014).

## References

Bellini F, Klar B, Muller A, Rosazza Gianin E (2014) Generalized quantiles as risk measures. *Insurance: Mathematics and Economics* **54**:41–48. doi:10.1016/j.insmatheco.2013.10.015.

Chen Z (1996) Conditional $L_p$-quantiles and their application to the testing of symmetry in non-parametric regression. *Statistics and Probability Letters* **29(2)**:107–115. doi:10.1016/0167-7152(95)001638.

## See Also

lqmean_rs

## Examples

```
# Compute the Lq-mean scoring function.

df <- data.frame(
    y = rep(x = 0, times = 6),
    x = c(2, 2, -2, -2, 0, 0),
    q = c(2, 3, 2, 3, 2, 3)
)

df$lqmean_penalty <- lqmean_sf(x = df$x, y = df$y, q = df$q)

print(df)
```

`lqquantile_rs` *Realised L_q-quantile score*

## Description

The function lqquantile_rs computes the realised $L_q$-quantile score at a specific level $p$ and parameter $q$, when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised $L_q$-quantile score is a realised score corresponding to the $L_q$-quantile scoring function lqquantile_sf.

## Usage

```
lqquantile_rs(x, y, p, q)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |
| p | It can be a scalar. |
| q | It can be a scalar. |

## Details

The realised $L_q$-quantile score is defined by:

$$S(\boldsymbol{x},\boldsymbol{y},p,q) := (1/n)\sum_{i=1}^{n} L(x_i,y_i,p,q)$$

where

$$\boldsymbol{x} = (x_1,...,x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1,...,y_n)^\mathsf{T}$$

and

$$L(x,y,p,q) := |\mathbf{1}\{x \geq y\} - p||x-y|^q$$

Domain of function:

$$\boldsymbol{x} \in \mathbb{R}^n$$

$$\boldsymbol{y} \in \mathbb{R}^n$$

$$0 < p < 1$$

$$q > 1$$

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}, p, q) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, p \in (0, 1), q > 1$$

## Value

Value of the realised $L_q$-quantile score.

## Note

For details on the $L_q$-quantile scoring function, see lqquantile_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised $L_q$-quantile score is the realised (average) score corresponding to the $L_q$-quantile scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

lqquantile_sf

## Examples

```
# Compute the realised Lq-quantile score.

set.seed(12345)

p <- 0.7

q <- 2

x <- qnorm(p = p, mean = 0, sd = 1)

y <- rnorm(n = 100, mean = 0, sd = 1)

print(lqquantile_rs(x = x, y = y, p = p, q = q))

print(lqquantile_rs(x = rep(x = x, times = 100), y = y, p = p, q = q))
```

`lqquantile_sf` *L_q-quantile scoring function*

## Description

The function lqquantile_sf computes the $L_q$-quantile scoring function at a specific level $p$, when $y$ materialises and $x$ is the predictive $L_q$-quantile at level $p$.

The $L_q$-quantile scoring function is defined by Chen (1996).

## Usage

```
lqquantile_sf(x, y, p, q)
```

## Arguments

| | |
|---|---|
| `x` | Predictive $L_q$-quantile at level $p$. It can be a vector of length $n$ (must have the same length as $y$). |
| `y` | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| `p` | It can be a vector of length $n$ (must have the same length as $y$). |
| `q` | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The $L_q$-quantile scoring function is defined by:

$$S(x, y, p, q) := |\mathbf{1}\{x \geq y\} - p||x - y|^q$$

or equivalently,

$$S(x, y, p, q) := p|\max\{-(x - y), 0\}|^q + (1 - p)|\max\{x - y, 0\}|^q$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$0 < p < 1$$

$$q > 1$$

Range of function:

$$S(x, y, p, q) \geq 0, \forall x, y \in \mathbb{R}, p \in (0, 1), q > 1$$

## Value

Vector of $L_q$-quantile losses.

## Note

For the definition of $L_q$-quantiles, see Chen (1996). In particular, $L_q$-quantiles at level $p$ are the solution of the equation $\mathrm{E}_F[V(x, Y, p, q)] = 0$, where

$$V(x, y, p, q) := q(\mathbf{1}\{x \geq y\} - p)|x - y|^{q-1}$$

The parameter is restricted to $q > 1$. At $q = 1$ the scoring function is the quantile scoring function quantile_sf, which elicits the $p$-quantile. The restriction to $q > 1$ makes the expected score strictly convex, so its minimiser is unique; at $q = 1$ the minimiser is in general set-valued (Section 2, p. 108 in Chen 1996).

The $L_q$-quantile scoring function is negatively oriented (i.e. the smaller, the better).

The $L_q$-quantile scoring function is strictly $\mathbb{F}$-consistent for the $L_q$-quantile functional at level $p$. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[|Y|^q]$ exists and is finite (Chen 1996; Bellini et al. 2014).

## References

Bellini F, Klar B, Muller A, Rosazza Gianin E (2014) Generalized quantiles as risk measures. *Insurance: Mathematics and Economics* **54**:41–48. doi:10.1016/j.insmatheco.2013.10.015.

Chen Z (1996) Conditional $L_p$-quantiles and their application to the testing of symmetry in non-parametric regression. *Statistics and Probability Letters* **29(2)**:107–115. doi:10.1016/0167-7152(95)001638.

## See Also

lqquantile_rs

## Examples

```
# Compute the Lq-quantile scoring function at level p.

df <- data.frame(
    y = rep(x = 0, times = 6),
    x = c(2, 2, -2, -2, 0, 0),
    p = rep(x = c(0.05, 0.95), times = 3),
    q = c(2, 3, 2, 3, 2, 3)
)

df$lqquantile_penalty <- lqquantile_sf(x = df$x, y = df$y, p = df$p, q = df$q)

print(df)
```

`mae` *Mean absolute error (MAE)*

## Description

The function mae computes the mean absolute error when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Mean absolute error is a realised score corresponding to the absolute error scoring function aerr_sf.

## Usage

```
mae(x, y)
```

## Arguments

| | |
|---|---|
| `x` | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| `y` | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |

## Details

The mean absolute error is defined by:

$$S(\boldsymbol{x},\boldsymbol{y}) := (1/n)\sum_{i=1}^{n} L(x_i, y_i)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y) := |x - y|$$

Domain of function:

$$\boldsymbol{x} \in \mathbb{R}^n$$

$$\boldsymbol{y} \in \mathbb{R}^n$$

Range of function:

$$S(\boldsymbol{x},\boldsymbol{y}) \geq 0, \forall \boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^n$$

## Value

Value of the mean absolute error.

## Note

For details on the absolute error scoring function, see aerr_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The mean absolute error is the realised (average) score corresponding to the absolute error scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

aerr_sf, quantile_if

## Examples

```
# Compute the mean absolute error.

set.seed(12345)

x <- 0

y <- rnorm(n = 100, mean = 0, sd = 1)

print(mae(x = x, y = y))

print(mae(x = rep(x = x, times = 100), y = y))
```

---

# maelog_rs *Realised MAE-LOG score*

---

## Description

The function maelog_rs computes the realised MAE-LOG score when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised MAE-LOG score is a realised score corresponding to the MAE-LOG scoring function maelog_sf.

## Usage

```
maelog_rs(x, y)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |

## Details

The realised MAE-LOG score is defined by:

$$S(\boldsymbol{x},\boldsymbol{y}) := (1/n)\sum_{i=1}^{n} L(x_i, y_i)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y) := |\log(x/y)|$$

Domain of function:

$$\boldsymbol{x} > \mathbf{0}$$

$$\boldsymbol{y} > \mathbf{0}$$

where

$$\mathbf{0} = (0, ..., 0)^\mathsf{T}$$

is the zero vector of length $n$ and the symbol $>$ indicates pairwise inequality.

Range of function:

$$S(\boldsymbol{x},\boldsymbol{y}) \geq 0, \forall \boldsymbol{x},\boldsymbol{y} > \mathbf{0}$$

## Value

Value of the realised MAE-LOG score.

## Note

For details on the MAE-LOG scoring function, see maelog_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised MAE-LOG score is the realised (average) score corresponding to the MAE-LOG scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

maelog_sf, quantile_if

## Examples

```
# Compute the realised MAE-LOG score.

set.seed(12345)

x <- 0.5

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(maelog_rs(x = x, y = y))

print(maelog_rs(x = rep(x = x, times = 100), y = y))
```

---

# maelog_sf *MAE-LOG scoring function*

---

## Description

The function maelog_sf computes the MAE-LOG scoring function when $y$ materialises and $x$ is the predictive median functional.

The MAE-LOG scoring function is defined by eq. (11) in Patton (2011).

## Usage

```
maelog_sf(x, y)
```

## Arguments

| | |
|---|---|
| x | Predictive median functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |

## Details

The MAE-LOG scoring function is defined by:

$$S(x, y) := |\log(x/y)|$$

Domain of function:

$$x > 0$$

$$y > 0$$

Range of function:

$$S(x, y) \geq 0, \forall x, y > 0$$

## Value

Vector of MAE-LOG losses.

## Note

For details on the MAE-LOG scoring function, see Gneiting (2011) and Patton (2011).

The median functional is the median of the probability distribution $F$ of $Y$ (Gneiting 2011).

The MAE-LOG scoring function is negatively oriented (i.e. the smaller, the better).

The MAE-LOG scoring function is strictly $\mathbb{F}$-consistent for the median functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[\log(Y)]$ exists and is finite (Thomson 1979; Saerens 2000; Gneiting 2011).

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Patton AJ (2011) Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics* **160(1)**:246–256. doi:10.1016/j.jeconom.2010.03.034.

Saerens M (2000) Building cost functions minimizing to some summary statistics. *IEEE Transactions on Neural Networks* **11(6)**:1263–1271. doi:10.1109/72.883416.

Thomson W (1979) Eliciting production possibilities from a well-informed manager. *Journal of Economic Theory* **20(3)**:360–380. doi:10.1016/00220531(79)900425.

## See Also

maelog_rs, quantile_if,

## Examples

```
# Compute the MAE-LOG scoring function.

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3
)

df$mae_log_penalty <- maelog_sf(x = df$x, y = df$y)

print(df)
```

---

# maesd_rs *Realised MAE-SD score*

---

## Description

The function maesd_rs computes the realised MAE-SD score when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised MAE-SD score is a realised score corresponding to the MAE-SD scoring function maesd_sf.

## Usage

```
maesd_rs(x, y)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |

## Details

The realised MAE-SD score is defined by:

$$S(\boldsymbol{x},\boldsymbol{y}) := (1/n)\sum_{i=1}^{n} L(x_i, y_i)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y) := |x^{1/2} - y^{1/2}|$$

Domain of function:

$$\boldsymbol{x} > \mathbf{0}$$

$$\boldsymbol{y} > \mathbf{0}$$

where

$$\mathbf{0} = (0, ..., 0)^\mathsf{T}$$

is the zero vector of length $n$ and the symbol $>$ indicates pairwise inequality.

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} > \mathbf{0}$$

## Value

Value of the realised MAE-SD score.

## Note

For details on the MAE-SD scoring function, see maesd_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised MAE-SD score is the realised (average) score corresponding to the MAE-SD scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

maesd_sf, quantile_if

## Examples

```
# Compute the realised MAE-SD score.

set.seed(12345)

x <- 0.5

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(maesd_rs(x = x, y = y))

print(maesd_rs(x = rep(x = x, times = 100), y = y))
```

---

`maesd_sf` *MAE-SD scoring function*

---

## Description

The function maesd_sf computes the MAE-SD scoring function when $y$ materialises and $x$ is the predictive median functional.

The MAE-SD scoring function is defined by eq. (12) in Patton (2011).

## Usage

```
maesd_sf(x, y)
```

## Arguments

| | |
|---|---|
| x | Predictive median functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |

## Details

The MAE-SD scoring function is defined by:

$$S(x, y) := |x^{1/2} - y^{1/2}|$$

Domain of function:

$$x > 0$$

$$y > 0$$

Range of function:

$$S(x, y) \geq 0, \forall x, y > 0$$

## Value

Vector of MAE-SD losses.

## Note

For details on the MAE-SD scoring function, see Gneiting (2011) and Patton (2011).

The median functional is the median of the probability distribution $F$ of $Y$ (Gneiting 2011).

The MAE-SD scoring function is negatively oriented (i.e. the smaller, the better).

The MAE-SD scoring function is strictly $\mathbb{F}$-consistent for the median functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y^{1/2}]$ exists and is finite (Thomson 1979; Saerens 2000; Gneiting 2011).

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Patton AJ (2011) Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics* **160(1)**:246–256. doi:10.1016/j.jeconom.2010.03.034.

Saerens M (2000) Building cost functions minimizing to some summary statistics. *IEEE Transactions on Neural Networks* **11(6)**:1263–1271. doi:10.1109/72.883416.

Thomson W (1979) Eliciting production possibilities from a well-informed manager. *Journal of Economic Theory* **20(3)**:360–380. doi:10.1016/00220531(79)900425.

## See Also

maesd_rs, quantile_if

## Examples

```
# Compute the MAE-SD scoring function.

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3
)

df$mae_sd_penalty <- maesd_sf(x = df$x, y = df$y)

print(df)
```

mape *Mean absolute percentage error (MAPE)*

## Description

The function mape computes the mean absolute percentage error when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Mean absolute percentage error is a realised score corresponding to the absolute percentage error scoring function aperr_sf.

## Usage

```
mape(x, y)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |

## Details

The mean absolute percentage error is defined by:

$$S(\boldsymbol{x},\boldsymbol{y}) := (1/n)\sum_{i=1}^{n} L(x_i, y_i)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y) := |(x - y)/y|$$

Domain of function:

$$\boldsymbol{x} > \mathbf{0}$$

$$\boldsymbol{y} > \mathbf{0}$$

where

$$\mathbf{0} = (0, ..., 0)^{\mathsf{T}}$$

is the zero vector of length $n$ and the symbol $>$ indicates pairwise inequality.

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} > \mathbf{0}$$

## Value

Value of the mean absolute percentage error.

## Note

For details on the absolute percentage error scoring function, see aperr_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The mean absolute percentage error is the realised (average) score corresponding to the absolute percentage error scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

aperr_sf

## Examples

```
# Compute the mean absolute percentage error.

set.seed(12345)

x <- 0.5

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(mape(x = x, y = y))

print(mape(x = rep(x = x, times = 100), y = y))
```

`meanexp_if` *Exp-transformed identification function*

## Description

The function meanexp_if computes the exp-transformed identification function with parameter $a$, when $y$ materialises and $(1/a)\log(\mathrm{E}_F[\exp(aY)])$ is the predictive functional.

The exp-transformed identification function is defined by Remark 1 in Tyralis and Papacharalampous (2026), applied to $g(t) = \exp(at)$.

## Usage

```
meanexp_if(x, y, a)
```

## Arguments

| | |
|---|---|
| `x` | Predictive $(1/a)\log(\mathrm{E}_F[\exp(aY)])$ functional. It can be a vector of length $n$ (must have the same length as $y$). |
| `y` | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| `a` | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The exp-transformed identification function is defined by:

$$V(x, y, a) := \exp(ax) - \exp(ay)$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$a \neq 0$$

Range of function:

$$V(x, y, a) \in \mathbb{R}$$

## Value

Vector of values of the exp-transformed identification function.

## Note

For details on the entropic risk measure functional $(1/a)\log(\mathrm{E}_F[\exp(aY)])$, see Gerber (1974) and serrexp_sf.

The exp-transformed identification function is a strict $\mathbb{F}$-identification function for the entropic risk measure functional $(1/a)\log(\mathrm{E}_F[\exp(aY)])$ (Tyralis and Papacharalampous 2026).

$\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[\exp(aY)]$ exists and is finite (Tyralis and Papacharalampous 2026).

linex_sf parameterises the same one-parameter family of entropic risk measures at the opposite sign: it elicits its functional as $-(1/a)\log(\mathrm{E}_F[\exp(-aY)])$, which equals the functional documented here evaluated at $-a$.

## References

Gerber HU (1974) On additive premium calculation principles. *ASTIN Bulletin: The Journal of the IAA* **7(3)**:215–222. doi:10.1017/S0515036100006061.

Tyralis H, Papacharalampous G (2026) Variable transformations in consistent loss functions. *Knowledge-Based Systems* **336**:115202. doi:10.1016/j.knosys.2025.115202.

## See Also

serrexp_sf, serrexp_rs

## Examples

```
# Compute the exp-transformed identification function.

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3,
    a = c(-1, 1, 2)
)

df$meanexp_if <- meanexp_if(x = df$x, y = df$y, a = df$a)

print(df)
```

---

# meanlog_if *Log-transformed identification function*

---

## Description

The function meanlog_if computes the log-transformed identification function, when $y$ materialises and $\exp(\mathrm{E}_F[\log(Y)])$, the geometric mean of $F$ (Yeh et al. 2008), is the predictive functional.

The log-transformed identification function is defined in Tyralis and Papacharalampous (2026).

## Usage

```
meanlog_if(x, y)
```

## Arguments

| | |
|---|---|
| x | Predictive $\exp(\mathrm{E}_F[\log(Y)])$ functional. It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |

## Details

The log-transformed identification function is defined by:

$$V(x, y) := \log(x) - \log(y)$$

Domain of function:

$$x > 0$$

$$y > 0$$

Range of function:

$$V(x, y) \in \mathbb{R}, \forall x, y > 0$$

## Value

Vector of values of the log-transformed identification function.

## Note

The $\exp(\mathrm{E}_F[\log(Y)])$ functional is the geometric mean of the probability distribution $F$ of $Y$, i.e. the $r = 0$ case of the generalized (power) mean of order $r$ defined by eq. (2.1) in Yeh et al. (2008).

The log-transformed identification function is a strict $\mathbb{F}$-identification function for the log-transformed expectation $\exp(\mathrm{E}_F[\log(Y)])$ (Tyralis and Papacharalampous 2026).

$\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[\log(Y)]$ exists and is finite (Tyralis and Papacharalampous 2026).

## References

Tyralis H, Papacharalampous G (2026) Variable transformations in consistent loss functions. *Knowledge-Based Systems* **336**:115202. doi:10.1016/j.knosys.2025.115202.

Yeh C-C, Yeh H-W, Chan W (2008) Some equivalent forms of the arithematic-geometric mean inequality in probability: A survey. *Journal of Inequalities and Applications* **2008**:386715. doi:10.1155/2008/386715.

## See Also

serrlog_sf, serrlog_rs

## Examples

```
# Compute the log-transformed identification function.

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3
)

df$meanlog_if <- meanlog_if(x = df$x, y = df$y)

print(df)
```

---

meanpower_if *Power-transformed identification function*

---

## Description

The function meanpower_if computes the power-transformed identification function with parameter $a$, when $y$ materialises and $(\mathrm{E}_F[Y^a])^{(1/a)}$ is the predictive functional.

The power-transformed identification function is defined by Remark 1 in Tyralis and Papacharalampous (2026), applied to $g(t) = t^a$.

## Usage

```
meanpower_if(x, y, a)
```

## Arguments

| | |
|---|---|
| x | Predictive $(\mathrm{E}_F[Y^a])^{(1/a)}$ functional. It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| a | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The power-transformed identification function is defined by:

$$V(x, y, a) := x^a - y^a$$

Domain of function:

Case #1

$$a > 0$$

$$x \geq 0$$

$$y \geq 0$$

Case #2

$$a \neq 0$$

$$x > 0$$

$$y > 0$$

Range of function:

$$V(x, y, a) \in \mathbb{R}$$

## Value

Vector of values of the power-transformed identification function.

## Note

For details on the $(\mathrm{E}_F[Y^a])^{(1/a)}$ functional, see serrpower_sf.

The power-transformed identification function is a strict $\mathbb{F}$-identification function for the $(\mathrm{E}_F[Y^a])^{(1/a)}$ functional (Tyralis and Papacharalampous 2026).

$\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y^a]$ exists and is finite (Tyralis and Papacharalampous 2026).

At $a = 2$, this is the identification function counterpart of serrsq_sf, which has no dedicated identification function of its own in this package.

## References

Tyralis H, Papacharalampous G (2026) Variable transformations in consistent loss functions. *Knowledge-Based Systems* **336**:115202. doi:10.1016/j.knosys.2025.115202.

## See Also

serrpower_sf, serrpower_rs

## Examples

```
# Compute the power-transformed identification function.

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3,
    a = c(1, 2, 3)
)

df$meanpower_if <- meanpower_if(x = df$x, y = df$y, a = df$a)

print(df)
```

---

`mean_if` *Mean identification function*

---

## Description

The function mean_if computes the mean identification function, when $y$ materialises and $x$ is the predictive mean.

The mean identification function is defined in Table 9 in Gneiting (2011).

## Usage

```
mean_if(x, y)
```

## Arguments

| | |
|---|---|
| x | Predictive mean (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |

## Details

The mean identification function is defined by:

$$V(x, y) := x - y$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

Range of function:

$$V(x, y) \in \mathbb{R}$$

## Value

Vector of values of the mean identification function.

## Note

The mean functional is the mean $\mathrm{E}_F[Y]$ of the probability distribution $F$ of $Y$ (Gneiting 2011).

The mean functional is the expectile at level $p = 1/2$ (Newey and Powell 1987), and the mean identification function equals the expectile identification function expectile_if at $p = 1/2$.

The mean identification function is a strict $\mathbb{F}$-identification function for the mean functional (Gneiting 2011; Fissler and Ziegel 2016; Dimitriadis et al. 2024).

$\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y]$ exists and is finite (Gneiting 2011; Fissler and Ziegel 2016; Dimitriadis et al. 2024).

## References

Dimitriadis T, Fissler T, Ziegel JF (2024) Osband's principle for identification functions. *Statistical Papers* **65**:1125–1132. doi:10.1007/s0036202301428x.

Fissler T, Ziegel JF (2016) Higher order elicitability and Osband's principle. *The Annals of Statistics* **44(4)**:1680–1707. doi:10.1214/16AOS1439.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Newey WK, Powell JL (1987) Asymmetric least squares estimation and testing. *Econometrica* **55(4)**:819–847. doi:10.2307/1911031.

## See Also

serr_sf, mse

## Examples

```
# Compute the mean identification function.

df <- data.frame(
    y = rep(x = 0, times = 3),
    x = c(-2, 0, 2)
)

df$mean_if <- mean_if(x = df$x, y = df$y)

print(df)
```

`mre` *Mean relative error (MRE)*

## Description

The function mre computes the mean relative error when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Mean relative error is a realised score corresponding to the relative error scoring function relerr_sf.

## Usage

```
mre(x, y)
```

## Arguments

| | |
|---|---|
| `x` | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| `y` | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |

## Details

The mean relative error is defined by:

$$S(\boldsymbol{x},\boldsymbol{y}) := (1/n)\sum_{i=1}^{n} L(x_i, y_i)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y) := |(x - y)/x|$$

Domain of function:

$$\boldsymbol{x} > \mathbf{0}$$

$$\boldsymbol{y} > \mathbf{0}$$

where

$$\mathbf{0} = (0, ..., 0)^\mathsf{T}$$

is the zero vector of length $n$ and the symbol $>$ indicates pairwise inequality.

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} > \mathbf{0}$$

## Value

Value of the mean relative error.

## Note

For details on the relative error scoring function, see relerr_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The mean relative error is the realised (average) score corresponding to the relative error scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

relerr_sf

## Examples

```
# Compute the mean relative error.

set.seed(12345)

x <- 0.5

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(mre(x = x, y = y))

print(mre(x = rep(x = x, times = 100), y = y))
```

`mse` *Mean squared error (MSE)*

## Description

The function mse computes the mean squared error when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Mean squared error is a realised score corresponding to the squared error scoring function serr_sf.

## Usage

```
mse(x, y)
```

## Arguments

| | |
|---|---|
| `x` | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| `y` | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |

## Details

The mean squared error is defined by:

$$S(\boldsymbol{x},\boldsymbol{y}) := (1/n)\sum_{i=1}^{n} L(x_i, y_i)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y) := (x - y)^2$$

Domain of function:

$$\boldsymbol{x} \in \mathbb{R}^n$$

$$\boldsymbol{y} \in \mathbb{R}^n$$

Range of function:

$$S(\boldsymbol{x},\boldsymbol{y}) \geq 0, \forall \boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^n$$

## Value

Value of the mean squared error.

## Note

For details on the squared error scoring function, see serr_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The mean squared error is the realised (average) score corresponding to the squared error scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

serr_sf, mean_if

## Examples

```
# Compute the mean squared error.

set.seed(12345)

x <- 0

y <- rnorm(n = 100, mean = 0, sd = 1)

print(mse(x = x, y = y))

print(mse(x = rep(x = x, times = 100), y = y))
```

---

# mspe *Mean squared percentage error (MSPE)*

---

## Description

The function mspe computes the mean squared percentage error when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Mean squared percentage error is a realised score corresponding to the squared percentage error scoring function sperr_sf.

## Usage

```
mspe(x, y)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |

## Details

The mean squared percentage error is defined by:

$$S(\boldsymbol{x},\boldsymbol{y}) := (1/n)\sum_{i=1}^{n} L(x_i, y_i)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y) := ((x - y)/y)^2$$

Domain of function:

$$\boldsymbol{x} > \mathbf{0}$$

$$\boldsymbol{y} > \mathbf{0}$$

where

$$\mathbf{0} = (0, ..., 0)^\mathsf{T}$$

is the zero vector of length $n$ and the symbol $>$ indicates pairwise inequality.

Range of function:

$$S(\boldsymbol{x},\boldsymbol{y}) \geq 0, \forall \boldsymbol{x},\boldsymbol{y} > \mathbf{0}$$

## Value

Value of the mean squared percentage error.

## Note

For details on the squared percentage error scoring function, see sperr_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The mean squared percentage error is the realised (average) score corresponding to the squared percentage error scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

sperr_sf

## Examples

```
# Compute the mean squared percentage error.

set.seed(12345)

x <- 0.5

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(mspe(x = x, y = y))

print(mspe(x = rep(x = x, times = 100), y = y))
```

---

# msre *Mean squared relative error (MSRE)*

---

## Description

The function msre computes the mean squared relative error when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Mean squared relative error is a realised score corresponding to the squared relative error scoring function srelerr_sf.

## Usage

```
msre(x, y)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |

## Details

The mean squared relative error is defined by:

$$S(\boldsymbol{x},\boldsymbol{y}) := (1/n)\sum_{i=1}^{n} L(x_i, y_i)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y) := ((x - y)/x)^2$$

Domain of function:

$$\boldsymbol{x} > \mathbf{0}$$

$$\boldsymbol{y} > \mathbf{0}$$

where

$$\mathbf{0} = (0, ..., 0)^\mathsf{T}$$

is the zero vector of length $n$ and the symbol $>$ indicates pairwise inequality.

Range of function:

$$S(\boldsymbol{x},\boldsymbol{y}) \geq 0, \forall \boldsymbol{x},\boldsymbol{y} > \mathbf{0}$$

## Value

Value of the mean squared relative error.

## Note

For details on the squared relative error scoring function, see srelerr_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The mean squared relative error is the realised (average) score corresponding to the squared relative error scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

srelerr_sf

## Examples

```
# Compute the mean squared relative error.

set.seed(12345)

x <- 0.5

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(msre(x = x, y = y))

print(msre(x = rep(x = x, times = 100), y = y))
```

---

# mv_if *Mean - variance identification function*

---

## Description

The function mv_if computes the mean - variance identification function, when $y$ materialises, $x_1$ is the predictive mean and $x_2$ is the predictive variance.

The mean - variance identification function is defined in Proposition 3.18 in Fissler and Ziegel (2019).

## Usage

```
mv_if(x1, x2, y)
```

## Arguments

| | |
|---|---|
| x1 | Predictive mean (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| x2 | Predictive variance (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x_1$ and $x_2$). |

## Details

The mean - variance identification function is defined by:

$$V(x_1, x_2, y) := (x_1 - y, x_2 + x_1^2 - y^2)^\mathsf{T}$$

Domain of function:

$$x_1 \in \mathbb{R}$$

$$x_2 > 0$$

$$y \in \mathbb{R}$$

Range of function:

$$V(x_1, x_2, y) \in \mathbb{R}^2, \forall x_1, y \in \mathbb{R}, x_2 > 0$$

## Value

Matrix of mean - variance values of the identification function.

## Note

The mean functional is the mean $\mathrm{E}_F[Y]$ of the probability distribution $F$ of $Y$ (Gneiting 2011).

The variance functional is the variance $\mathrm{Var}_F[Y] := \mathrm{E}_F[Y^2] - (\mathrm{E}_F[Y])^2$ of the probability distribution $F$ of $Y$ (Gneiting 2011).

The mean - variance identification function is a strict $\mathbb{F}$-identification function for the pair (mean, variance) functional (Gneiting 2011; Fissler and Ziegel 2019; Dimitriadis et al. 2024).

$\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y]$ and $\mathrm{E}_F[Y^2]$ exist and are finite (Gneiting 2011; Fissler and Ziegel 2019; Dimitriadis et al. 2024).

## References

Dimitriadis T, Fissler T, Ziegel JF (2024) Osband's principle for identification functions. *Statistical Papers* **65**:1125–1132. doi:10.1007/s0036202301428x.

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

mv_sf

## Examples

```
# Compute the mean - variance identification function.

df <- data.frame(
    y = rep(x = 0, times = 6),
    x1 = c(2, 2, -2, -2, 0, 0),
    x2 = c(1, 2, 1, 2, 1, 2)
)

v <- as.data.frame(mv_if(x1 = df$x1, x2 = df$x2, y = df$y))

print(cbind(df, v))
```

---

# mv_sf *Mean - variance scoring function*

---

## Description

The function mv_sf computes the mean - variance scoring function, when $y$ materialises, $x_1$ is the predictive mean and $x_2$ is the predictive variance.

The mean - variance scoring function is defined by eq. (3.11) in Fissler and Ziegel (2019).

## Usage

```
mv_sf(x1, x2, y)
```

## Arguments

| | |
|---|---|
| x1 | Predictive mean (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| x2 | Predictive variance (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x_1$ and $x_2$). |

## Details

The mean - variance scoring function is defined by:

$$S(x_1, x_2, y) := x_2^{-2}(x_1^2 - 2x_2 - 2x_1 y + y^2)$$

Domain of function:

$$x_1 \in \mathbb{R}$$

$$x_2 > 0$$

$$y \in \mathbb{R}$$

Range of function:

$$S(x_1, x_2, y) \geq -2/x_2, \forall x_1, y \in \mathbb{R}, x_2 > 0$$

## Value

Vector of mean - variance losses.

## Note

The mean functional is the mean $\mathrm{E}_F[Y]$ of the probability distribution $F$ of $Y$ (Gneiting 2011).

The variance functional is the variance $\mathrm{Var}_F[Y] := \mathrm{E}_F[Y^2] - (\mathrm{E}_F[Y])^2$ of the probability distribution $F$ of $Y$ (Gneiting 2011).

The mean - variance scoring function is negatively oriented (i.e. the smaller, the better).

The mean - variance scoring function is strictly $\mathbb{F}$-consistent for the pair (mean, variance) functional (eq. (3.11) and Example 3.19 in Fissler and Ziegel (2019); Proposition 4.4 in Fissler and Ziegel (2016)). $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y]$ and $\mathrm{E}_F[Y^2]$ exist and are finite (eq. (3.11) and Example 3.19 in Fissler and Ziegel (2019); Proposition 4.4 in Fissler and Ziegel (2016)).

## References

Fissler T, Ziegel JF (2016) Higher order elicitability and Osband's principle. *The Annals of Statistics* **44(4)**:1680–1707. doi:10.1214/16AOS1439.

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

## See Also

mv_if

## Examples

```
# Compute the mean - variance scoring function.

df <- data.frame(
    y = rep(x = 0, times = 6),
    x1 = c(2, 2, -2, -2, 0, 0),
    x2 = c(1, 2, 1, 2, 1, 2)
)

df$mv_penalty <- mv_sf(x1 = df$x1, x2 = df$x2, y = df$y)

print(df)
```

---

# nmoment_if *$n$-th moment identification function*

---

## Description

The function nmoment_if computes the $n$-th moment identification function, when $y$ materialises and $x$ is the predictive $n$-th moment.

The $n$-th moment identification function is defined in Table 9 in Gneiting (2011) by setting $r(t) = t^n$ and $s(t) = 1$.

## Usage

```
nmoment_if(x, y, n)
```

## Arguments

| | |
|---|---|
| x | Predictive $n$-th moment. It can be a vector of length $m$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $m$ (must have the same length as $x$). |
| n | $n$ is the moment order. It can be a vector of length $m$ (must have the same length as $x$). |

## Details

The $n$-th moment identification function is defined by:

$$V(x, y, n) := x - y^n$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$n \in \mathbb{N}$$

Range of function:

$$V(x, y, n) \in \mathbb{R}$$

## Value

Vector of values of the $n$-th moment identification function.

## Note

The $n$-th moment functional is the expectation $\mathrm{E}_F[Y^n]$ of the probability distribution $F$ of $Y$.

The $n$-th moment identification function is a strict $\mathbb{F}$-identification function for the $n$-th moment functional (Gneiting 2011).

$\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y^n]$ exists and is finite (Gneiting 2011).

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

nmoment_sf, nmoment_rs

## Examples

```
# Compute the n-th moment identification function.

df <- data.frame(
    y = rep(x = 2, times = 6),
    x = c(1, 2, 3, 1, 2, 3),
    n = c(2, 2, 2, 3, 3, 3)
)

df$nmoment_if <- nmoment_if(x = df$x, y = df$y, n = df$n)

print(df)
```

`nmoment_rs` *Realised $n$-th moment score*

## Description

The function nmoment_rs computes the realised $n$-th moment score, when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised $n$-th moment score is a realised score corresponding to the $n$-th moment scoring function nmoment_sf.

## Usage

```
nmoment_rs(x, y, n)
```

## Arguments

| | |
|---|---|
| `x` | Prediction. It can be a vector of length $m$ (must have the same length as $\boldsymbol{y}$). |
| `y` | Realisation (true value) of process. It can be a vector of length $m$ (must have the same length as $\boldsymbol{x}$). |
| `n` | Moment order. It can be a scalar. |

## Details

The realised $n$-th moment score is defined by:

$$S(\boldsymbol{x}, \boldsymbol{y}, n) := (1/m) \sum_{i=1}^{m} L(x_i, y_i, n)$$

where

$$\boldsymbol{x} = (x_1, ..., x_m)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_m)^\mathsf{T}$$

and

$$L(x, y, n) := -x^2 - 2x(y^n - x)$$

Domain of function:

$$\boldsymbol{x} \in \mathbb{R}^m$$

$$\boldsymbol{y} \in \mathbb{R}^m$$

$$n \in \mathbb{N}$$

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}, n) \geq -(1/m) \sum_{i=1}^{m} y_i^{2n}, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m, n \in \mathbb{N}$$

## Value

Value of the realised $n$-th moment score.

## Note

For details on the $n$-th moment scoring function, see nmoment_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised $n$-th moment score is the realised (average) score corresponding to the $n$-th moment scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

nmoment_sf, nmoment_if

## Examples

```
# Compute the realised n-th moment score.
# x = 1 is the predictive 2nd moment E[Y^2] of a standard normal distribution.

set.seed(12345)

n <- 2

x <- 1

y <- rnorm(n = 100, mean = 0, sd = 1)

print(nmoment_rs(x = x, y = y, n = n))

print(nmoment_rs(x = rep(x = x, times = 100), y = y, n = n))
```

`nmoment_sf` *$n$-th moment scoring function*

## Description

The function nmoment_sf computes the $n$-th moment scoring function, when $y$ materialises, and $\mathrm{E}_F[Y^n]$ is the predictive $n$-th moment.

The $n$-th moment scoring function is defined by eq. (22) in Gneiting (2011) by setting $r(t) = t^n$, $s(t) = 1$, $\phi(t) = t^2$ and removing all terms that are not functions of $x$.

## Usage

```
nmoment_sf(x, y, n)
```

## Arguments

| | |
|---|---|
| `x` | Predictive $n$-th moment. It can be a vector of length $m$ (must have the same length as $y$). |
| `y` | Realisation (true value) of process. It can be a vector of length $m$ (must have the same length as $x$). |
| `n` | $n$ is the moment order. It can be a vector of length $m$ (must have the same length as $x$). |

## Details

The $n$-th moment scoring function is defined by:

$$S(x, y, n) := x^2 - 2xy^n$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$n \in \mathbb{N}$$

Range of function:

$$S(x, y, n) \geq -y^{2n}, \forall x, y \in \mathbb{R}, n \in \mathbb{N}$$

## Value

Vector of $n$-th moment losses.

## Note

The $n$-th moment functional is the expectation $\mathrm{E}_F[Y^n]$ of the probability distribution $F$ of $Y$.

The $n$-th moment scoring function is negatively oriented (i.e. the smaller, the better).

The $n$-th moment scoring function is strictly $\mathbb{F}$-consistent for the $n$-th moment functional $\mathrm{E}_F[Y^n]$ (Theorem 8 in Gneiting 2011). $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y]$, $\mathrm{E}_F[Y^2]$, $\mathrm{E}_F[Y^n]$ and $\mathrm{E}_F[Y^{n+1}]$ exist and are finite (Theorem 8 in Gneiting 2011).

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

nmoment_rs, nmoment_if, serr_sf, lqmean_sf

## Examples

```
# Compute the n-th moment scoring function.

df <- data.frame(
    y = rep(x = 2, times = 6),
    x = c(1, 2, 3, 1, 2, 3),
    n = c(2, 2, 2, 3, 3, 3)
)

df$nmoment_penalty <- nmoment_sf(x = df$x, y = df$y, n = df$n)

print(df)
```

---

# nse *Nash-Sutcliffe efficiency (NSE)*

---

## Description

The function nse computes the Nash-Sutcliffe efficiency when ***y*** materialises and ***x*** is the prediction.

Nash-Sutcliffe efficiency is a skill score corresponding to the squared error scoring function serr_sf. It is defined in eq. (3) in Nash and Sutcliffe (1970).

## Usage

```
nse(x, y)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as ***y***). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as ***x***). |

## Details

The Nash-Sutcliffe efficiency is defined by:

$$S_{\text{skill}}(\boldsymbol{x},\boldsymbol{y}) := 1 - S_{\text{meth}}(\boldsymbol{x},\boldsymbol{y})/S_{\text{ref}}(\boldsymbol{y})$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

$$\boldsymbol{1} = (1, ..., 1)^\mathsf{T}$$

$$\bar{\boldsymbol{y}} := (1/n)\boldsymbol{1}^\mathsf{T}\boldsymbol{y} = (1/n)\sum_{i=1}^{n} y_i$$

$$L(x, y) := (x - y)^2$$

and the predictions of the method of interest as well as the reference method are evaluated respectively by:

$$S_{\text{meth}}(\boldsymbol{x},\boldsymbol{y}) := (1/n)\sum_{i=1}^{n} L(x_i, y_i)$$

$$S_{\text{ref}}(\boldsymbol{y}) := (1/n)\sum_{i=1}^{n} L(\bar{\boldsymbol{y}}, y_i)$$

The reference score $S_{\text{ref}}$ is the score of the constant prediction $\bar{\boldsymbol{y}}$ and therefore depends on $\boldsymbol{y}$ only.

$\boldsymbol{1}$ is the vector of ones.

Domain of function:

$$\boldsymbol{x} \in \mathbb{R}^n$$

$$\boldsymbol{y} \in \mathbb{R}^n$$

$$S_{\text{ref}}(\boldsymbol{y}) > 0$$

Range of function:

$$S_{\text{skill}}(\boldsymbol{x},\boldsymbol{y}) \leq 1, \forall \boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^n \text{ with } S_{\text{ref}}(\boldsymbol{y}) > 0$$

## Value

Value of the Nash-Sutcliffe efficiency.

## Note

For details on the squared error scoring function, see serr_sf.

The concept of skill scores is defined by Gneiting (2011).

The Nash-Sutcliffe efficiency is defined in eq. (3) in Nash and Sutcliffe (1970).

The Nash-Sutcliffe efficiency is positively oriented (i.e. the larger, the better).

The Nash-Sutcliffe efficiency is undefined when the reference score $S_{\text{ref}}(\boldsymbol{y})$ is zero, which happens if and only if $\boldsymbol{y}$ is constant. In that degenerate case the function returns `NaN` or `-Inf`.

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Nash JE, Sutcliffe JV (1970) River flow forecasting through conceptual models Part I - A discussion of principles. *Journal of Hydrology* **10(3)**:282–290. doi:10.1016/00221694(70)902556.

## See Also

mse, serr_sf, mean_if

## Examples

```
# Compute the Nash-Sutcliffe efficiency.

set.seed(12345)

x <- 0

y <- rnorm(n = 100, mean = 0, sd = 1)

print(nse(x = x, y = y))

print(nse(x = rep(x = x, times = 100), y = y))

print(nse(x = mean(y), y = y))

print(nse(x = y, y = y))
```

obsweighted_rs *Realised observation-weighted score*

## Description

The function obsweighted_rs computes the realised observation-weighted score when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised observation-weighted score is a realised score corresponding to the observation-weighted scoring function obsweighted_sf.

## Usage

```
obsweighted_rs(x, y)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |

## Details

The realised observation-weighted score is defined by:

$$S(\boldsymbol{x},\boldsymbol{y}) := (1/n)\sum_{i=1}^{n} L(x_i, y_i)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y) := y(x - y)^2$$

Domain of function:

$$\boldsymbol{x} > \boldsymbol{0}$$

$$\boldsymbol{y} > \boldsymbol{0}$$

where

$$\mathbf{0} = (0, ..., 0)^{\mathsf{T}}$$

is the zero vector of length $n$ and the symbol $>$ indicates pairwise inequality.

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} > \mathbf{0}$$

## Value

Value of the realised observation-weighted score.

## Note

For details on the observation-weighted scoring function, see obsweighted_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised observation-weighted score is the realised (average) score corresponding to the observation-weighted scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

obsweighted_sf

## Examples

```
# Compute the realised observation-weighted score.

set.seed(12345)

x <- 0.5

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(obsweighted_rs(x = x, y = y))

print(obsweighted_rs(x = rep(x = x, times = 100), y = y))
```

`obsweighted_sf` *Observation-weighted scoring function*

## Description

The function obsweighted_sf computes the observation-weighted scoring function when $y$ materialises and $x$ is the predictive $\frac{\mathrm{E}_F[Y^2]}{\mathrm{E}_F[Y]}$ functional.

The observation-weighted scoring function is defined on p. 752 in Gneiting (2011).

## Usage

```
obsweighted_sf(x, y)
```

## Arguments

| | |
|---|---|
| `x` | Predictive $\frac{\mathrm{E}_F[Y^2]}{\mathrm{E}_F[Y]}$ functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| `y` | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |

## Details

The observation-weighted scoring function is defined by:

$$S(x, y) := y(x - y)^2$$

Domain of function:

$$x > 0$$

$$y > 0$$

Range of function:

$$S(x, y) \geq 0, \forall x, y > 0$$

## Value

Vector of observation-weighted errors.

## Note

For details on the observation-weighted scoring function, see Gneiting (2011).

The observation-weighted scoring function is negatively oriented (i.e. the smaller, the better).

The observation-weighted scoring function is strictly $\mathbb{F}$-consistent for the $\frac{\mathrm{E}_F[Y^2]}{\mathrm{E}_F[Y]}$ functional (Theorem 5 and eq. (12) in Gneiting 2011). $\mathbb{F}$ is the family of probability distributions $F$ concentrated on $(0, \infty)$ for which $\mathrm{E}_F[Y]$, $\mathrm{E}_F[Y^2]$ and $\mathrm{E}_F[Y^3]$ exist and are finite (Theorem 5 in Gneiting 2011).

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

obsweighted_rs

## Examples

```
# Compute the observation-weighted scoring function.

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3
)

df$obsweighted_penalty <- obsweighted_sf(x = df$x, y = df$y)

print(df)
```

---

# powerweighted_if *Power-weighted identification function*

---

## Description

The function powerweighted_if computes the power-weighted identification function with parameter $a$, when $y$ materialises and $\frac{\mathrm{E}_F[Y^{a+1}]}{\mathrm{E}_F[Y^a]}$ is the predictive functional.

The power-weighted identification function is defined by Table 9 in Gneiting (2011), applied to $r(y) = y^{a+1}$ and $s(y) = y^a$.

## Usage

```
powerweighted_if(x, y, a)
```

## Arguments

| | |
|---|---|
| x | Predictive $\frac{\mathrm{E}_F[Y^{a+1}]}{\mathrm{E}_F[Y^a]}$ functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| a | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The power-weighted identification function is defined by:

$$V(x, y, a) := xy^a - y^{a+1}$$

or equivalently,

$$V(x, y, a) := y^a(x - y)$$

Domain of function:

$$x > 0$$

$$y > 0$$

$$a \in \mathbb{R}$$

Range of function:

$$V(x, y, a) \in \mathbb{R}$$

## Value

Vector of values of the power-weighted identification function.

## Note

The $\frac{\mathrm{E}_F[Y^{a+1}]}{\mathrm{E}_F[Y^a]}$ functional is the ratio of expectations formed by weighting the probability distribution $F$ of $Y$ with $w(y) = y^a$ (eq. (11) in Gneiting 2011). For details on this functional, see powerweighted_sf.

The power-weighted identification function is a strict $\mathbb{F}$-identification function for the $\frac{\mathrm{E}_F[Y^{a+1}]}{\mathrm{E}_F[Y^a]}$ functional (Table 9 and Theorem 8 in Gneiting 2011). $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y^a]$ and $\mathrm{E}_F[Y^{a+1}]$ exist and are finite (Theorem 8 in Gneiting 2011).

At $a = 0$ the power-weighted identification function reduces to mean_if and the functional to the mean $\mathrm{E}_F[Y]$.

This is the identification function counterpart of powerweighted_sf, and thereby also of obsweighted_sf and srelerr_sf at $a = 1$ and of sperr_sf at $a = -2$.

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

powerweighted_sf

## Examples

```
# Compute the power-weighted identification function.

df <- data.frame(
    y = rep(x = 2, times = 6),
    x = c(1, 2, 3, 1, 2, 3),
    a = rep(x = c(1, -2), each = 3)
)

df$powerweighted_if <- powerweighted_if(x = df$x, y = df$y, a = df$a)

print(df)

# The power-weighted identification function reduces to the mean identification
# function at a = 0.

set.seed(12345)

n <- 10

x <- runif(n = n, min = 0, max = 2)
y <- runif(n = n, min = 0, max = 2)

max(abs(powerweighted_if(x = x, y = y, a = 0) - mean_if(x = x, y = y)))

# values are slightly higher than 0 due to rounding error
```

---

# powerweighted_sf *Power-weighted squared error scoring function*

---

## Description

The function powerweighted_sf computes the power-weighted squared error scoring function with parameter $a$, when $y$ materialises and $x$ is the predictive $\frac{\mathrm{E}_F[Y^{a+1}]}{\mathrm{E}_F[Y^a]}$ functional.

The power-weighted squared error scoring function is defined by eqs. (10) and (11) in Gneiting (2011), applied to the weight function $w(y) = y^a$ and to the squared error scoring function.

## Usage

```
powerweighted_sf(x, y, a)
```

## Arguments

| | |
|---|---|
| x | Predictive $\frac{\mathrm{E}_F[Y^{a+1}]}{\mathrm{E}_F[Y^a]}$ functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| a | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The power-weighted squared error scoring function is defined by:

$$S(x, y, a) := y^a(x - y)^2$$

Domain of function:

$$x > 0$$

$$y > 0$$

$$a \in \mathbb{R}$$

Range of function:

$$S(x, y, a) \geq 0, \forall x, y > 0, a \in \mathbb{R}$$

## Value

Vector of power-weighted squared errors.

## Note

The $\frac{\mathrm{E}_F[Y^{a+1}]}{\mathrm{E}_F[Y^a]}$ functional is the ratio of expectations formed by weighting the probability distribution $F$ of $Y$ with $w(y) = y^a$ (eq. (11) in Gneiting 2011).

The power-weighted squared error scoring function is negatively oriented (i.e. the smaller, the better).

The power-weighted squared error scoring function is strictly $\mathbb{F}$-consistent for the $\frac{\mathrm{E}_F[Y^{a+1}]}{\mathrm{E}_F[Y^a]}$ functional (Theorem 5 in Gneiting 2011). $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y^a]$, $\mathrm{E}_F[Y^{a+1}]$ and $\mathrm{E}_F[Y^{a+2}]$ exist and are finite (Theorem 5 in Gneiting 2011).

At $a = 0$ the power-weighted squared error scoring function reduces to serr_sf and the functional to the mean $\mathrm{E}_F[Y]$, at $a = 1$ it reduces to obsweighted_sf, and at $a = -2$ it reduces to sperr_sf.

srelerr_sf targets the same functional as the case $a = 1$, but is not a member of this family, because it weights with the prediction $x$ rather than with the realisation $y$ (p. 752 in Gneiting 2011).

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

powerweighted_if

## Examples

```
# Compute the power-weighted squared error scoring function.

df <- data.frame(
    y = rep(x = 2, times = 6),
    x = c(1, 2, 3, 1, 2, 3),
    a = rep(x = c(1, -2), each = 3)
)

df$powerweighted_penalty <- powerweighted_sf(x = df$x, y = df$y, a = df$a)

print(df)

# The power-weighted squared error scoring function reduces to the squared
# error scoring function at a = 0, to the observation-weighted scoring function
# at a = 1 and to the squared percentage error scoring function at a = -2.

set.seed(12345)

n <- 10

x <- runif(n = n, min = 0, max = 2)
y <- runif(n = n, min = 0, max = 2)

max(abs(powerweighted_sf(x = x, y = y, a = 0) - serr_sf(x = x, y = y)))

max(abs(powerweighted_sf(x = x, y = y, a = 1) - obsweighted_sf(x = x, y = y)))

max(abs(powerweighted_sf(x = x, y = y, a = -2) - sperr_sf(x = x, y = y)))

# values are slightly higher than 0 due to rounding error
```

`qlike` *QLIKE*

## Description

The function qlike computes the QLIKE score when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

QLIKE is a realised score corresponding to the Bregman scoring function (type 3, QLIKE scoring function) bregman3_sf.

## Usage

```
qlike(x, y)
```

## Arguments

| | |
|---|---|
| `x` | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| `y` | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |

## Details

QLIKE is defined by:

$$S(\boldsymbol{x},\boldsymbol{y}) := (1/n)\sum_{i=1}^{n} L(x_i, y_i)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y) := (y/x) - \log(y/x) - 1$$

Domain of function:

$$\boldsymbol{x} > \boldsymbol{0}$$

$$\boldsymbol{y} > \boldsymbol{0}$$

where

$$\mathbf{0} = (0, ..., 0)^\mathsf{T}$$

is the zero vector of length $n$ and the symbol $>$ indicates pairwise inequality.

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} > \mathbf{0}$$

## Value

Value of QLIKE.

## Note

For details on the Bregman scoring function (type 3, QLIKE scoring function), see bregman3_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

QLIKE is the realised (average) score corresponding to the Bregman scoring function (type 3, QLIKE scoring function).

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

bregman3_sf, mean_if

## Examples

```
# Compute QLIKE.

set.seed(12345)

x <- 0.5

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(qlike(x = x, y = y))

print(qlike(x = rep(x = x, times = 100), y = y))
```

quantile_if *Quantile identification function*

## Description

The function quantile_if computes the quantile identification function at a specific level $p$, when $y$ materialises and $x$ is the predictive quantile at level $p$.

The quantile identification function is defined in Table 9 in Gneiting (2011).

## Usage

```
quantile_if(x, y, p)
```

## Arguments

| | |
|---|---|
| x | Predictive quantile (prediction) at level $p$. It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| p | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The quantile identification function is defined by:

$$V(x, y, p) := \mathbf{1}\{x \geq y\} - p$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$0 < p < 1$$

Range of function:

$$V(x, y, p) \in \{-p, 1 - p\}, \forall x, y \in \mathbb{R}, p \in (0, 1)$$

## Value

Vector of values of the quantile identification function.

## Note

For the definition of quantiles, see Koenker and Bassett Jr (1978).

The quantile identification function is a strict $\mathbb{F}_p$-identification function for the $p$-quantile functional (Gneiting 2011; Dimitriadis et al. 2024).

$\mathbb{F}_p$ is the family of probability distributions $F$ for which there exists an $y$ with $F(y) = p$ (Gneiting 2011; Dimitriadis et al. 2024).

## References

Dimitriadis T, Fissler T, Ziegel JF (2024) Osband's principle for identification functions. *Statistical Papers* **65**:1125–1132. doi:10.1007/s0036202301428x.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Koenker R, Bassett Jr G (1978) Regression quantiles. *Econometrica* **46(1)**:33–50. doi:10.2307/1913643.

## See Also

quantile_sf, quantile_rs

## Examples

```
# Compute the quantile identification function.

df <- data.frame(
    y = rep(x = 0, times = 6),
    x = c(2, 2, -2, -2, 0, 0),
    p = rep(x = c(0.05, 0.95), times = 3)
)

df$quantile_if <- quantile_if(x = df$x, y = df$y, p = df$p)

print(df)
```

---

# quantile_level *Sample quantile level function*

---

## Description

The function quantile_level computes the sample quantile level, when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the predictive quantile at level $p$.

## Usage

```
quantile_level(x, y)
```

## Arguments

| | |
|---|---|
| x | Predictive quantile (prediction) at level $p$. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |

## Details

The sample quantile level function is defined by:

$$P(\boldsymbol{x}, \boldsymbol{y}) := (1/n) \sum_{i=1}^{n} \mathbf{1}\{x_i \geq y_i\}$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

Domain of function:

$$\boldsymbol{x} \in \mathbb{R}^n$$

$$\boldsymbol{y} \in \mathbb{R}^n$$

Range of function:

$$0 \leq P(\boldsymbol{x}, \boldsymbol{y}) \leq 1, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$$

## Value

Value of the sample quantile level.

## Note

For the definition of quantiles, see Koenker and Bassett Jr (1978).

The sample quantile level is directly related to the quantile identification function quantile_if, which is defined in Table 9 in Gneiting (2011). The sample quantile level equals the sample mean of $V(x_i, y_i, p) + p$, where $V$ is the quantile identification function at level $p$.

If $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the predictive quantile at level $p$, then ideally, the sample quantile level should be equal to the nominal quantile level $p$.

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Koenker R, Bassett Jr G (1978) Regression quantiles. *Econometrica* **46(1)**:33–50. doi:10.2307/1913643.

## See Also

quantile_sf, quantile_rs, quantile_if

## Examples

```
# Compute the sample quantile level.

set.seed(12345)

x <- qnorm(p = 0.75, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)

y <- rnorm(n = 1000, mean = 0, sd = 1)

print(quantile_level(x = x, y = y))
```

---

# quantile_rs *Realised quantile score*

---

## Description

The function quantile_rs computes the realised quantile score at a specific level $p$ when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised quantile score is a realised score corresponding to the quantile scoring function quantile_sf.

## Usage

```
quantile_rs(x, y, p)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |
| p | It can be a scalar. |

## Details

The realised quantile score is defined by:

$$S(\boldsymbol{x}, \boldsymbol{y}, p) := (1/n) \sum_{i=1}^{n} L(x_i, y_i, p)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y, p) := (\mathbf{1}\{x \geq y\} - p)(x - y)$$

Domain of function:

$$\boldsymbol{x} \in \mathbb{R}^n$$

$$\boldsymbol{y} \in \mathbb{R}^n$$

$$0 < p < 1$$

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}, p) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, p \in (0, 1)$$

## Value

Value of the realised quantile score.

## Note

For details on the quantile scoring function, see quantile_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised quantile score is the realised (average) score corresponding to the quantile scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

quantile_sf, quantile_if

## Examples

```
# Compute the realised quantile score.

set.seed(12345)

x <- qnorm(p = 0.7, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)

y <- rnorm(n = 1000, mean = 0, sd = 1)

print(quantile_rs(x = x, y = y, p = 0.7))

print(quantile_rs(x = rep(x = x, times = 1000), y = y, p = 0.7))

print(quantile_rs(x = rep(x = x, times = 1000) - 0.1, y = y, p = 0.7))
```

---

# quantile_sf *Asymmetric piecewise linear scoring function (quantile scoring function, quantile loss function)*

---

## Description

The function quantile_sf computes the asymmetric piecewise linear scoring function (quantile scoring function) at a specific level $p$, when $y$ materialises and $x$ is the predictive quantile at level $p$.

The asymmetric piecewise linear scoring function is defined by eq. (24) in Gneiting (2011).

## Usage

```
quantile_sf(x, y, p)
```

## Arguments

| | |
|---|---|
| x | Predictive quantile (prediction) at level $p$. It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| p | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The asymmetric piecewise linear scoring function is defined by:

$$S(x, y, p) := (\mathbf{1}\{x \geq y\} - p)(x - y)$$

or equivalently,

$$S(x, y, p) := p|\max\{-(x-y), 0\}| + (1-p)|\max\{x-y, 0\}|$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$0 < p < 1$$

Range of function:

$$S(x, y, p) \geq 0, \forall x, y \in \mathbb{R}, p \in (0, 1)$$

## Value

Vector of quantile losses.

## Note

For the definition of quantiles, see Koenker and Bassett Jr (1978).

The asymmetric piecewise linear scoring function is negatively oriented (i.e. the smaller, the better).

The asymmetric piecewise linear scoring function is strictly $\mathbb{F}$-consistent for the $p$-quantile functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y]$ exists and is finite (Raiffa and Schlaifer 1961, p.196; Ferguson 1967, p.51; Thomson 1979; Saerens 2000; Gneiting 2011).

## References

Ferguson TS (1967) Mathematical Statistics: A Decision-Theoretic Approach. Academic Press, New York.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Koenker R, Bassett Jr G (1978) Regression quantiles. *Econometrica* **46(1)**:33–50. doi:10.2307/1913643.

Raiffa H, Schlaifer R (1961) Applied Statistical Decision Theory. Colonial Press, Clinton.

Saerens M (2000) Building cost functions minimizing to some summary statistics. *IEEE Transactions on Neural Networks* **11(6)**:1263–1271. doi:10.1109/72.883416.

Thomson W (1979) Eliciting production possibilities from a well-informed manager. *Journal of Economic Theory* **20(3)**:360–380. doi:10.1016/00220531(79)900425.

## See Also

quantile_rs, quantile_if

## Examples

```
# Compute the asymmetric piecewise linear scoring function (quantile scoring
# function).

df <- data.frame(
    y = rep(x = 0, times = 6),
    x = c(2, 2, -2, -2, 0, 0),
    p = rep(x = c(0.05, 0.95), times = 3)
)

df$quantile_penalty <- quantile_sf(x = df$x, y = df$y, p = df$p)

print(df)

# The absolute error scoring function is twice the asymmetric piecewise linear
# scoring function (quantile scoring function) at level p = 0.5.

df <- data.frame(
    y = rep(x = 0, times = 3),
    x = c(-2, 0, 2),
    p = rep(x = c(0.5), times = 3)
)

df$quantile_penalty <- quantile_sf(x = df$x, y = df$y, p = df$p)

df$absolute_error <- aerr_sf(x = df$x, y = df$y)

print(df)
```

---

# relerr_sf *Relative error scoring function (MAE-PROP scoring function)*

---

## Description

The function relerr_sf computes the relative error scoring function when $y$ materialises and $x$ is the predictive $\text{med}^{(1)}(F)$ functional.

The relative error scoring function is defined in Table 1 in Gneiting (2011).

The relative error scoring function is referred to as MAE-PROP scoring function in eq. (13) in Patton (2011).

## Usage

```
relerr_sf(x, y)
```

## Arguments

| | |
|---|---|
| x | Predictive $\text{med}^{(1)}(F)$ functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |

y — Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$).

## Details

The relative error scoring function is defined by:

$$S(x, y) := |(x - y)/x|$$

Domain of function:

$$x > 0$$

$$y > 0$$

Range of function:

$$S(x, y) \geq 0, \forall x, y > 0$$

## Value

Vector of relative errors.

## Note

For details on the relative error scoring function, see Gneiting (2011).

The $\beta$-median functional, $\text{med}^{(\beta)}(F)$ is the median of a probability distribution whose density is proportional to $y^\beta f(y)$, where $f$ is the density of the probability distribution $F$ of $Y$ (Gneiting 2011).

The functional $\text{med}^{(1)}(F)$ elicited by the relative error scoring function is the $\beta$-median at $\beta = 1$, whose density is proportional to $yf(y)$ (p. 748 and p. 749 in Gneiting 2011). The $\beta$-median functional at general $\beta$ is elicited by bmedian_sf.

Patton (2011) adds that the MAE-PROP scoring function is also known as the mean absolute percentage error (MAPE). That name is used differently here: the package follows Gneiting (2011), where the absolute percentage error is $|(x-y)/y|$, elicited by aperr_sf and averaged by mape, whose elicited functional is $\text{med}^{(-1)}(F)$. A reader arriving from Patton (2011) should use relerr_sf, not mape.

The relative error scoring function is negatively oriented (i.e. the smaller, the better).

The relative error scoring function is strictly $\mathbb{F}^{(w)}$-consistent for the $\text{med}^{(1)}(F)$ functional. $\mathbb{F}$ is the family of probability distributions for which $\text{E}_F[Y]$ exists and is finite. $\mathbb{F}^{(w)}$ is the subclass of probability distributions in $\mathbb{F}$, which are such that $w(y)f(y)$, $w(y) = y$ has finite integral over $(0, \infty)$, and the probability distribution $F^{(w)}$ with density proportional to $w(y)f(y)$ belongs to $\mathbb{F}$ (see Theorems 5 and 9 in Gneiting 2011).

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Patton AJ (2011) Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics* **160(1)**:246–256. doi:10.1016/j.jeconom.2010.03.034.

## See Also

mre

## Examples

```
# Compute the relative error scoring function.

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3
)

df$relative_error <- relerr_sf(x = df$x, y = df$y)

print(df)
```

---

# serrexp_rs *Realised squared error exp score*

---

## Description

The function serrexp_rs computes the realised squared error exp score with parameter $a$, when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised squared error exp score is a realised score corresponding to the squared error exp scoring function serrexp_sf.

## Usage

```
serrexp_rs(x, y, a)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |
| a | It can be a scalar. |

## Details

The realised squared error exp score is defined by:

$$S(\boldsymbol{x}, \boldsymbol{y}, a) := (1/n) \sum_{i=1}^{n} L(x_i, y_i, a)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y, a) := (\mathrm{e}^{ax} - \mathrm{e}^{ay})^2$$

Domain of function:

$$\boldsymbol{x} \in \mathbb{R}^n$$

$$\boldsymbol{y} \in \mathbb{R}^n$$

$$a \neq 0$$

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}, a) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, a \neq 0$$

## Value

Value of the realised squared error exp score.

## Note

For details on the squared error exp scoring function, see serrexp_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised squared error exp score is the realised (average) score corresponding to the squared error exp scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

serrexp_sf, meanexp_if

## Examples

```
# Compute the realised squared error exp score.

set.seed(12345)

a <- 1

x <- 0

y <- rnorm(n = 100, mean = 0, sd = 1)

print(serrexp_rs(x = x, y = y, a = a))

print(serrexp_rs(x = rep(x = x, times = 100), y = y, a = a))
```

---

# serrexp_sf *Squared error exp scoring function*

---

## Description

The function serrexp_sf computes the squared error exp scoring function when $y$ materialises and $x$ is the $(1/a)\log(\mathrm{E}_F[\exp(aY)])$ predictive entropic risk measure (Gerber 1974).

The squared error exp scoring function is defined in Fissler and Pesenti (2023).

## Usage

```
serrexp_sf(x, y, a)
```

## Arguments

| | |
|---|---|
| x | Predictive $(1/a)\log(\mathrm{E}_F[\exp(aY)])$ functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| a | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The squared error exp scoring function is defined by:

$$S(x, y, a) := (\exp(ax) - \exp(ay))^2$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$a \neq 0$$

Range of function:

$$S(x, y, a) \geq 0, \forall x, y \in \mathbb{R}, a \neq 0$$

## Value

Vector of squared errors of exp-transformed variables.

## Note

For details on the squared error exp scoring function, see Fissler and Pesenti (2023).

Appendix A1 in Fissler and Pesenti (2023) is given for $a > 0$. The domain $a \neq 0$ used here is the one given for $g(t) = \exp(at)$ in Table 1 in Tyralis and Papacharalampous (2026), where the transformation is a bijection for every $a \neq 0$.

The squared error exp scoring function is negatively oriented (i.e. the smaller, the better).

The squared error exp scoring function is strictly $\mathbb{F}$-consistent for the $(1/a)\log(\mathrm{E}_F[\exp(aY)])$ entropic risk measure functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[\exp(2aY)]$ exists and is finite (Fissler and Pesenti 2023; Tyralis and Papacharalampous 2026). The squared transform doubles the moment requirement: $\mathrm{E}_F[\exp(aY)] < \infty$ alone makes the functional well defined, but the expected score is finite only under $\mathrm{E}_F[\exp(2aY)] < \infty$.

linex_sf parameterises the same one-parameter family of entropic risk measures at the opposite sign: it elicits its functional as $-(1/a)\log(\mathrm{E}_F[\exp(-aY)])$, which equals the functional documented here evaluated at $-a$.

## References

Fissler T, Pesenti SM (2023) Sensitivity measures based on scoring functions. *European Journal of Operational Research* **307(3)**:1408–1423. doi:10.1016/j.ejor.2022.10.002.

Gerber HU (1974) On additive premium calculation principles. *ASTIN Bulletin: The Journal of the IAA* **7(3)**:215–222. doi:10.1017/S0515036100006061.

Tyralis H, Papacharalampous G (2026) Variable transformations in consistent loss functions. *Knowledge-Based Systems* **336**:115202. doi:10.1016/j.knosys.2025.115202.

## See Also

serrexp_rs, meanexp_if

## Examples

```
# Compute the squared error exp scoring function.

df <- data.frame(
    y = rep(x = 0, times = 5),
    x = -2:2,
    a = c(-2, -1, 1, 2, 3)
)

df$squaredexp_error <- serrexp_sf(x = df$x, y = df$y, a = df$a)

print(df)
```

---

`serrlog_rs` *Realised squared error log score*

---

## Description

The function serrlog_rs computes the realised squared error log score when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised squared error log score is a realised score corresponding to the squared error log scoring function serrlog_sf.

## Usage

```
serrlog_rs(x, y)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |

## Details

The realised squared error log score is defined by:

$$S(\boldsymbol{x},\boldsymbol{y}) := (1/n) \sum_{i=1}^{n} L(x_i, y_i)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y) := (\log(x) - \log(y))^2$$

Domain of function:

$$\boldsymbol{x} > \mathbf{0}$$

$$\boldsymbol{y} > \mathbf{0}$$

where

$$\mathbf{0} = (0, ..., 0)^\mathsf{T}$$

is the zero vector of length $n$ and the symbol $>$ indicates pairwise inequality.

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} > \mathbf{0}$$

## Value

Value of the realised squared error log score.

## Note

For details on the squared error log scoring function, see serrlog_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised squared error log score is the realised (average) score corresponding to the squared error log scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

serrlog_sf, meanlog_if

## Examples

```
# Compute the realised squared error log score.

set.seed(12345)

x <- 0.5

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(serrlog_rs(x = x, y = y))

print(serrlog_rs(x = rep(x = x, times = 100), y = y))
```

---

`serrlog_sf` *Squared error log scoring function*

---

## Description

The function serrlog_sf computes the squared error log scoring function when $y$ materialises and $x$ is the $\exp(\mathrm{E}_F[\log(Y)])$ predictive functional, i.e. the geometric mean of $F$ (Yeh et al. 2008).

The squared error log scoring function is described by eq. (2) in Houghton-Carr (1999).

## Usage

```
serrlog_sf(x, y)
```

## Arguments

| | |
|---|---|
| x | Predictive $\exp(\mathrm{E}_F[\log(Y)])$ functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |

## Details

The squared error log scoring function is defined by:

$$S(x, y) := (\log(x) - \log(y))^2$$

Domain of function:

$$x > 0$$

$$y > 0$$

Range of function:

$$S(x, y) \geq 0, \forall x, y > 0$$

## Value

Vector of squared errors of log-transformed variables.

## Note

For details on the squared error log scoring function, see Houghton-Carr (1999).

The $\exp(\mathrm{E}_F[\log(Y)])$ functional is the geometric mean of the probability distribution $F$ of $Y$, i.e. the $r = 0$ case of the generalized (power) mean of order $r$ defined by eq. (2.1) in Yeh et al. (2008).

The squared error log scoring function is negatively oriented (i.e. the smaller, the better).

The squared error log scoring function is strictly $\mathbb{F}$-consistent for the $\exp(\mathrm{E}_F[\log(Y)])$ functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[(\log(Y))^2]$ exists and is finite (Tyralis and Papacharalampous 2026).

## References

Houghton-Carr HA (1999) Assessment criteria for simple conceptual daily rainfall-runoff models. *Hydrological Sciences Journal* **44(2)**:237–261. doi:10.1080/02626669909492220.

Tyralis H, Papacharalampous G (2026) Variable transformations in consistent loss functions. *Knowledge-Based Systems* **336**:115202. doi:10.1016/j.knosys.2025.115202.

Yeh C-C, Yeh H-W, Chan W (2008) Some equivalent forms of the arithematic-geometric mean inequality in probability: A survey. *Journal of Inequalities and Applications* **2008**:386715. doi:10.1155/2008/386715.

## See Also

serrlog_rs, meanlog_if

## Examples

```
# Compute the squared error log scoring function.

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3
)

df$squaredlog_error <- serrlog_sf(x = df$x, y = df$y)

print(df)
```

`serrpower_rs` *Realised squared error of power transformations score*

## Description

The function serrpower_rs computes the realised squared error of power transformations score with parameter $a$, when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised squared error of power transformations score is a realised score corresponding to the squared error of power transformations scoring function serrpower_sf.

## Usage

```
serrpower_rs(x, y, a)
```

## Arguments

| | |
|---|---|
| `x` | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| `y` | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |
| `a` | It can be a scalar. |

## Details

The realised squared error of power transformations score is defined by:

$$S(\boldsymbol{x}, \boldsymbol{y}, a) := (1/n) \sum_{i=1}^{n} L(x_i, y_i, a)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y, a) := (x^a - y^a)^2$$

Domain of function:

Case #1

$$a > 0$$

$$\boldsymbol{x} \geq \boldsymbol{0}$$

$$\boldsymbol{y} \geq \mathbf{0}$$

Case #2

$$a \neq 0$$

$$\boldsymbol{x} > \mathbf{0}$$

$$\boldsymbol{y} > \mathbf{0}$$

where

$$\mathbf{0} = (0, ..., 0)^{\mathsf{T}}$$

is the zero vector of length $n$ and the symbols $\geq$ and $>$ indicate pairwise inequality.

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}, a) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y}, a$$

## Value

Value of the realised squared error of power transformations score.

## Note

For details on the squared error of power transformations scoring function, see serrpower_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised squared error of power transformations score is the realised (average) score corresponding to the squared error of power transformations scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

`serrpower_sf`, `meanpower_if`

## Examples

```
# Compute the realised squared error of power transformations score.

set.seed(12345)

a <- 2

x <- 0.5

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(serrpower_rs(x = x, y = y, a = a))

print(serrpower_rs(x = rep(x = x, times = 100), y = y, a = a))
```

---

`serrpower_sf` *Squared error of power transformations scoring function*

---

## Description

The function serrpower_sf computes the squared error of power transformations scoring function when $y$ materialises and $x$ is the $(\mathrm{E}_F[Y^a])^{(1/a)}$ predictive functional.

The squared error of power transformations scoring function is defined in Tyralis and Papacharalampous (2026).

## Usage

```
serrpower_sf(x, y, a)
```

## Arguments

| | |
|---|---|
| `x` | Predictive $(\mathrm{E}_F[Y^a])^{(1/a)}$ functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| `y` | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |
| `a` | It can be a vector of length $n$ (must have the same length as $y$). |

## Details

The squared error of power transformations scoring function is defined by:

$$S(x, y, a) := (x^a - y^a)^2$$

Domain of function:

Case #1

$$a > 0$$

$$x \geq 0$$

$$y \geq 0$$

Case #2

$$a \neq 0$$

$$x > 0$$

$$y > 0$$

Range of function:

Case #1

$$S(x, y, a) \geq 0, \forall x, y \geq 0, a > 0$$

Case #2

$$S(x, y, a) \geq 0, \forall x, y > 0, a \neq 0$$

## Value

Vector of squared errors of power-transformed variables.

## Note

For details on the squared error of power transformations scoring function, see Tyralis and Papacharalampous (2026).

The squared error of power transformations scoring function is negatively oriented (i.e. the smaller, the better).

The squared error of power transformations scoring function is strictly $\mathbb{F}$-consistent for the $(\mathrm{E}_F[Y^a])^{(1/a)}$ functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y^{2a}]$ exists and is finite (Tyralis and Papacharalampous 2026).

## References

Tyralis H, Papacharalampous G (2026) Variable transformations in consistent loss functions. *Knowledge-Based Systems* **336**:115202. doi:10.1016/j.knosys.2025.115202.

## See Also

serrpower_rs, meanpower_if

## Examples

```
# Compute the squared error of power transformations scoring function.

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3,
    a = 1:3
)

df$squaredpower_error <- serrpower_sf(x = df$x, y = df$y, a = df$a)

print(df)
```

---

`serrsq_rs` *Realised squared error of squares score*

---

## Description

The function serrsq_rs computes the realised squared error of squares score when $\boldsymbol{y}$ materialises and $\boldsymbol{x}$ is the prediction.

Realised squared error of squares score is a realised score corresponding to the squared error of squares scoring function serrsq_sf.

## Usage

```
serrsq_rs(x, y)
```

## Arguments

| | |
|---|---|
| x | Prediction. It can be a vector of length $n$ (must have the same length as $\boldsymbol{y}$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $\boldsymbol{x}$). |

## Details

The realised squared error of squares score is defined by:

$$S(\boldsymbol{x},\boldsymbol{y}) := (1/n)\sum_{i=1}^{n} L(x_i, y_i)$$

where

$$\boldsymbol{x} = (x_1, ..., x_n)^\mathsf{T}$$

$$\boldsymbol{y} = (y_1, ..., y_n)^\mathsf{T}$$

and

$$L(x, y) := (x^2 - y^2)^2$$

Domain of function:

$$\boldsymbol{x} \geq \mathbf{0}$$

$$\boldsymbol{y} \geq \mathbf{0}$$

where

$$\mathbf{0} = (0, ..., 0)^\mathsf{T}$$

is the zero vector of length $n$ and the symbol $\geq$ indicates pairwise inequality.

Range of function:

$$S(\boldsymbol{x}, \boldsymbol{y}) \geq 0, \forall \boldsymbol{x}, \boldsymbol{y} \geq \mathbf{0}$$

## Value

Value of the realised squared error of squares score.

## Note

For details on the squared error of squares scoring function, see serrsq_sf.

The concept of realised (average) scores is defined by Gneiting (2011) and Fissler and Ziegel (2019).

The realised squared error of squares score is the realised (average) score corresponding to the squared error of squares scoring function.

## References

Fissler T, Ziegel JF (2019) Order-sensitivity and equivariance of scoring functions. *Electronic Journal of Statistics* **13(1)**:1166–1211. doi:10.1214/19EJS1552.

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

serrsq_sf

## Examples

```
# Compute the realised squared error of squares score.

set.seed(12345)

x <- 0.5

y <- rlnorm(n = 100, meanlog = 0, sdlog = 1)

print(serrsq_rs(x = x, y = y))

print(serrsq_rs(x = rep(x = x, times = 100), y = y))
```

---

`serrsq_sf` *Squared error of squares scoring function*

---

## Description

The function serrsq_sf computes the squared error of squares scoring function when $y$ materialises and $x$ is the $\sqrt{\mathrm{E}_F[Y^2]}$ predictive functional.

The squared transformation used by the squared error of squares scoring function is described in Thirel et al. (2024).

## Usage

```
serrsq_sf(x, y)
```

## Arguments

| | |
|---|---|
| `x` | Predictive $\sqrt{\mathrm{E}_F[Y^2]}$ functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| `y` | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |

## Details

The squared error of squares scoring function is defined by:

$$S(x, y) := (x^2 - y^2)^2$$

Domain of function:

$$x \geq 0$$

$$y \geq 0$$

Range of function:

$$S(x, y) \geq 0, \forall x, y \geq 0$$

## Value

Vector of squared errors of squared-transformed variables.

## Note

For details on the squared transformation, see Thirel et al. (2024).

The squared error of squares scoring function is negatively oriented (i.e. the smaller, the better).

The squared error of squares scoring function is strictly $\mathbb{F}$-consistent for the $\sqrt{\mathrm{E}_F[Y^2]}$ functional. $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y^4]$ exists and is finite (Tyralis and Papacharalampous 2026).

## References

Thirel G, Santos L, Delaigue O, Perrin C (2024) On the use of streamflow transformations for hydrological model calibration. *Hydrology and Earth System Sciences* **28(21)**:4837–4860. doi:10.5194/hess2848372024.

Tyralis H, Papacharalampous G (2026) Variable transformations in consistent loss functions. *Knowledge-Based Systems* **336**:115202. doi:10.1016/j.knosys.2025.115202.

## See Also

serrsq_rs

## Examples

```
# Compute the squared error of squares scoring function.

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3
)

df$squaredsq_error <- serrsq_sf(x = df$x, y = df$y)

print(df)
```

`serr_sf` *Squared error scoring function*

## Description

The function serr_sf computes the squared error scoring function when $y$ materialises and $x$ is the predictive mean functional.

The squared error scoring function is defined in Table 1 in Gneiting (2011).

## Usage

```
serr_sf(x, y)
```

## Arguments

| | |
|---|---|
| x | Predictive mean functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |

## Details

The squared error scoring function is defined by:

$$S(x, y) := (x - y)^2$$

Domain of function:

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

Range of function:

$$S(x, y) \geq 0, \forall x, y \in \mathbb{R}$$

## Value

Vector of squared errors.

## Note

For details on the squared error scoring function, see Savage (1971), Gneiting (2011).

The mean functional is the mean $\mathrm{E}_F[Y]$ of the probability distribution $F$ of $Y$ (Gneiting 2011).

The squared error scoring function is negatively oriented (i.e. the smaller, the better).

The squared error scoring function is strictly $\mathbb{F}$-consistent for the mean functional. $\mathbb{F}$ is the family of probability distributions $F$ for which the second moment exists and is finite (Savage 1971; Gneiting 2011).

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Savage LJ (1971) Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association* **66(336)**:783–801. doi:10.1080/01621459.1971.10482346.

## See Also

mse, mean_if

## Examples

```
# Compute the squared error scoring function.

df <- data.frame(
    y = rep(x = 0, times = 5),
    x = -2:2
)

df$squared_error <- serr_sf(x = df$x, y = df$y)

print(df)
```

---

# sperr_sf *Squared percentage error scoring function*

---

## Description

The function sperr_sf computes the squared percentage error scoring function when $y$ materialises and $x$ is the predictive $\frac{\mathrm{E}_F[Y^{-1}]}{\mathrm{E}_F[Y^{-2}]}$ functional.

The squared percentage error scoring function is defined on p. 752 in Gneiting (2011).

## Usage

```
sperr_sf(x, y)
```

## Arguments

| | |
|---|---|
| x | Predictive $\frac{\mathrm{E}_F[Y^{-1}]}{\mathrm{E}_F[Y^{-2}]}$ functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |

## Details

The squared percentage error scoring function is defined by:

$$S(x, y) := ((x - y)/y)^2$$

Domain of function:

$$x > 0$$

$$y > 0$$

Range of function:

$$S(x, y) \geq 0, \forall x, y > 0$$

## Value

Vector of squared percentage errors.

## Note

For details on the squared percentage error scoring function, see Park and Stefanski (1998) and Gneiting (2011).

The squared percentage error scoring function is negatively oriented (i.e. the smaller, the better).

The squared percentage error scoring function is strictly $\mathbb{F}$-consistent for the $\frac{\mathrm{E}_F[Y^{-1}]}{\mathrm{E}_F[Y^{-2}]}$ functional (Park and Stefanski 1998; Theorem 5 and eq. (11) on p. 752 in Gneiting 2011). $\mathbb{F}$ is the family of probability distributions $F$ for which $\mathrm{E}_F[Y^{-1}]$ and $\mathrm{E}_F[Y^{-2}]$ exist and are finite (Theorem 5 in Gneiting 2011).

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

Park H, Stefanski LA (1998) Relative-error prediction. *Statistics and Probability Letters* **40(3)**:227–236. doi:10.1016/S01677152(98)000881.

## See Also

mspe

## Examples

```
# Compute the squared percentage error scoring function.

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3
)

df$squared_percentage_error <- sperr_sf(x = df$x, y = df$y)

print(df)
```

---

# srelerr_sf *Squared relative error scoring function*

---

## Description

The function srelerr_sf computes the squared relative error scoring function when $y$ materialises and $x$ is the predictive $\frac{\mathrm{E}_F[Y^2]}{\mathrm{E}_F[Y]}$ functional.

The squared relative error scoring function is defined on p. 752 in Gneiting (2011).

## Usage

```
srelerr_sf(x, y)
```

## Arguments

| | |
|---|---|
| x | Predictive $\frac{\mathrm{E}_F[Y^2]}{\mathrm{E}_F[Y]}$ functional (prediction). It can be a vector of length $n$ (must have the same length as $y$). |
| y | Realisation (true value) of process. It can be a vector of length $n$ (must have the same length as $x$). |

## Details

The squared relative error scoring function is defined by:

$$S(x, y) := ((x - y)/x)^2$$

Domain of function:

$$x > 0$$

$$y > 0$$

Range of function:

$$S(x, y) \geq 0, \forall x, y > 0$$

## Value

Vector of squared relative errors.

## Note

For details on the squared relative error scoring function, see Gneiting (2011).

The squared relative error scoring function is negatively oriented (i.e. the smaller, the better).

The squared relative error scoring function is strictly $\mathbb{F}$-consistent for the $\frac{\mathrm{E}_F[Y^2]}{\mathrm{E}_F[Y]}$ functional (eq. (12) on p. 752 in Gneiting 2011). $\mathbb{F}$ is the family of probability distributions $F$ concentrated on $(0, \infty)$ for which $\mathrm{E}_F[Y^2]$ exists and is finite (Gneiting 2011).

The squared relative error scoring function weights with the prediction $x$ rather than with the realisation $y$, so it is not a member of the weighted family of Theorem 5 in Gneiting (2011); it targets the same functional as obsweighted_sf, which is $\frac{\mathrm{E}_F[Y^2]}{\mathrm{E}_F[Y]}$ (eq. (12) on p. 752 in Gneiting 2011).

## References

Gneiting T (2011) Making and evaluating point forecasts. *Journal of the American Statistical Association* **106(494)**:746–762. doi:10.1198/jasa.2011.r10138.

## See Also

msre

## Examples

```
# Compute the squared relative error scoring function.

df <- data.frame(
    y = rep(x = 2, times = 3),
    x = 1:3
)

df$squared_relative_error <- srelerr_sf(x = df$x, y = df$y)

print(df)
```

# Index