Package ‘BivLaplaceRL’

April 30, 2026

Type Package

Title Bivariate Laplace Transforms, Stochastic Orders, and Entropy Measures in Reliability

Version 1.0.0

Description Implements methods for bivariate and univariate Laplace transforms of residual lives and reversed residual lives, associated stochastic ordering concepts, and entropy measures for reliability analysis. The package covers:
(1) Bivariate Laplace transform of residual lives and stochastic comparisons based on the bivariate Laplace transform order of residual lives (BLt-rl), including weak bivariate hazard rate, mean residual life, and relative mean residual life orders, nonparametric estimation, and NBUHR/NWUHR aging class characterisation; Jayalekshmi, Rajesh, and Nair (2022) ``Bivariate Laplace Transform of Residual Lives and Their Properties'' <doi:10.1080/03610926.2022.2085874>;
(2) Bivariate Laplace transform order of reversed residual lives (BLt-Rrl), reversed hazard gradient, reversed mean residual life, and the associated stochastic orders (weak bivariate reversed hazard rate, weak bivariate reversed mean residual life); Jayalekshmi, Rajesh, and Nair (2022) ``Bivariate Laplace Transform Order and Ordering of Reversed Residual Lives'' <doi:10.1142/S0218539322500061>;
(3) Univariate Laplace transform of residual life, hazard rate, mean residual life, and the corresponding stochastic orders (Lt-rl order, hazard rate order, MRL order), together with a nonparametric estimator.
Shannon entropy and Golomb's (1966) information generating function are also provided. Parametric families supported include the Gumbel bivariate exponential, Farlie-Gumbel-Morgenstern (FGM), bivariate power, and Schur-constant distributions. Plotting utilities and a simulation framework for evaluating estimator performance are also provided.

License GPL-3

URL `https://itsmdivakaran.github.io/BivLaplaceRL/`, `https://github.com/itsmdivakaran/BivLaplaceRL`

BugReports `https://github.com/itsmdivakaran/BivLaplaceRL/issues`

Encoding UTF-8

Config/Needs/website pkgdown

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports stats, graphics

Suggests testthat (>= 3.0.0), knitr, rmarkdown, MASS, survival, covr

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Language en-GB

Author Mahesh Divakaran [aut, cre] (ORCID:
<`https://orcid.org/0000-0002-3488-0857`>, affiliation: Amity School of Applied Sciences, Amity University Lucknow),
S. Jayalekshmi [aut, ctb] (affiliation: Department of Statistics, Cochin University of Science and Technology),
G. Rajesh [aut] (affiliation: Department of Statistics, Cochin University of Science and Technology),
N. Unnikrishnan Nair [aut] (affiliation: Department of Statistics, Cochin University of Science and Technology)

Maintainer Mahesh Divakaran <`imaheshdivakaran@gmail.com`>

Repository CRAN

Date/Publication 2026-04-30 10:40:02 UTC

Contents

`BivLaplaceRL-package` *BivLaplaceRL: Bivariate Laplace Transforms, Stochastic Orders, and Entropy Measures in Reliability*

Description

BivLaplaceRL provides a unified framework for reliability analysis covering bivariate and univariate Laplace transforms of residual lives, associated stochastic orders, and entropy measures:

1. Bivariate Laplace Transform of Residual Lives

Implements the vector Laplace transform of bivariate residual lives $(L_{X_{t_1|t_2}}(s_1), L_{X_{t_2|t_1}}(s_2))$, the associated stochastic ordering BLt-rl, and its relationships with the weak bivariate hazard rate order, weak bivariate mean residual life order, and bivariate relative mean residual life order. Nonparametric estimation and NBUHR/NWUHR aging class tests are included. Reference: Jayalekshmi, Rajesh, and Nair (2022) doi:10.1080/03610926.2022.2085874.

2. Bivariate Laplace Transform of Reversed Residual Lives

Implements the bivariate Laplace transform of reversed (past) residual lives, the BLt-Rrl stochastic order, and connections with weak bivariate reversed hazard rate and reversed mean residual life orders. Reference: Jayalekshmi, Rajesh, and Nair (2022) doi:10.1142/S0218539322500061.

3. Univariate Residual Life Analysis

Implements the univariate Laplace transform of residual life $L_X(s,t) = E[e^{-sX} \mid X > t]$, hazard rate, mean residual life, and the corresponding stochastic orders (Lt-rl order, hazard rate order, MRL order), together with a nonparametric estimator.

Parametric families

Gumbel bivariate exponential, Farlie-Gumbel-Morgenstern (FGM), bivariate power, and Schur-constant distributions are built in.

Plotting

`plot_blt_residual` and `plot_blt_reversed` provide ready-made visualisations.

Author(s)

Maintainer: Mahesh Divakaran `<imaheshdivakaran@gmail.com>` (ORCID) (Amity School of Applied Sciences, Amity University Lucknow)

Authors:

- S. Jayalekshmi (Department of Statistics, Cochin University of Science and Technology) [contributor]
- G. Rajesh `<rajeshgstat@gmail.com>` (Department of Statistics, Cochin University of Science and Technology)
- N. Unnikrishnan Nair (Department of Statistics, Cochin University of Science and Technology)

References

Jayalekshmi S., Rajesh G., Nair N.U. (2022). Bivariate Laplace transform of residual lives and their properties. *Communications in Statistics - Theory and Methods*. doi:10.1080/03610926.2022.2085874

Jayalekshmi S., Rajesh G., Nair N.U. (2022). Bivariate Laplace transform order and ordering of reversed residual lives. *International Journal of Reliability, Quality and Safety Engineering*. doi:10.1142/S0218539322500061

Belzunce F., Ortega E., Ruiz J.M. (1999). The Laplace order and ordering of residual lives. *Statistics & Probability Letters*, 42(2), 145–156. doi:10.1016/S01677152(98)002028

Golomb S.W. (1966). The information generating function of a probability distribution. *IEEE Transactions on Information Theory*, 12(1), 75–77.

See Also

Useful links:

- `https://itsmdivakaran.github.io/BivLaplaceRL/`
- `https://github.com/itsmdivakaran/BivLaplaceRL`
- Report bugs at `https://github.com/itsmdivakaran/BivLaplaceRL/issues`

`bivpower` *Bivariate Power Distribution*

Description

Distribution function, survival function, density, and random generation for the bivariate power distribution:

$$F(x_1, x_2) = x_1^{p_1+\theta \log x_2}\, x_2^{p_2}, \quad 0 \le x_1, x_2 \le p_2,\ p_1, p_2 > 0,\ 0 \le \theta \le 1.$$

Usage

```
pbivpower(x1, x2, p1 = 1, p2 = 1, theta = 0)

sbivpower(x1, x2, p1 = 1, p2 = 1, theta = 0)

dbivpower(x1, x2, p1 = 1, p2 = 1, theta = 0)

rbivpower(n, p1 = 1, p2 = 1, theta = 0)
```

Arguments

`x1, x2`	Values in the support.
`p1, p2`	Positive shape parameters.
`theta`	Association parameter, $0 \leq \theta \leq 1$.
`n`	Number of random observations.

Value

Numeric vector or two-column matrix (`rbivpower`).

References

Jayalekshmi S., Rajesh G. Bivariate Laplace transform order and ordering of reversed residual lives. *International Journal of Reliability, Quality and Safety Engineering*.

Examples

```
pbivpower(0.4, 0.5, p1 = 2, p2 = 2, theta = 0.3)
set.seed(7); head(rbivpower(30, p1 = 2, p2 = 2, theta = 0.2))
```

`biv_brlmr_order` *Bivariate Relative Mean Residual Life Order*

Description

Checks whether $X \leq_{\mathrm{brlmr}} Y$: the ratio $m_{i,Y}(t_1, t_2)/m_{i,X}(t_1, t_2)$ is increasing in t_i.

Usage

```
biv_brlmr_order(
  surv_X,
  surv_Y,
  t2_fixed = 0.5,
  t1_grid = seq(0.2, 3, by = 0.3),
  tol = 1e-06
)
```

Arguments

`surv_X, surv_Y`	Joint survival functions.
`t2_fixed`	Fixed value of t_2 for the univariate slices.
`t1_grid`	Grid of t_1 values.
`tol`	Tolerance.

Value

List with `order_holds` and `ratio_values`.

References

Kayid M., Izadkhah S., Alshami S. (2016). Residual probability function, associated orderings, and related aging classes. *Statistics and Probability Letters*, 116, 37–47.

Examples

```
sX <- function(x1, x2) sgumbel_biv(x1, x2, k1 = 2, k2 = 1)
sY <- function(x1, x2) sgumbel_biv(x1, x2, k1 = 1, k2 = 1)
biv_brlmr_order(sX, sY, t2_fixed = 0.5)
```

`biv_hazard_gradient` *Bivariate Hazard Gradient*

Description

Computes the bivariate hazard gradient

$$h_i(t_1, t_2) = -\frac{\partial}{\partial t_i} \log \bar{F}(t_1, t_2), \quad i = 1, 2.$$

Usage

```
biv_hazard_gradient(
  t1,
  t2,
  surv_fn = NULL,
  k1 = 1,
  k2 = 1,
  theta = 0,
  eps = 1e-07
)
```

Arguments

`t1, t2`	Evaluation points (non-negative).
`surv_fn`	A function `function(x1, x2)` returning the joint survival probability. Defaults to Gumbel bivariate exponential.
`k1, k2, theta`	Parameters for the default Gumbel distribution.
`eps`	Step size for numerical differentiation (default `1e-7`).

Value

A named numeric vector `(h1, h2)`.

References

Jayalekshmi S., Rajesh G., Nair N.U. (2022). doi:10.1080/03610926.2022.2085874

Examples

```
biv_hazard_gradient(t1 = 1, t2 = 1)
biv_hazard_gradient(t1 = 0.5, t2 = 0.5, k1 = 2, k2 = 1.5, theta = 0.3)
```

`biv_mean_residual` *Bivariate Mean Residual Life Function*

Description

Computes the bivariate mean residual life (MRL) function

$$m_1(t_1, t_2) = E(X_{t_1|t_2}) = \frac{\int_{t_1}^{\infty} \bar{F}(x_1, t_2)\, dx_1}{\bar{F}(t_1, t_2)}$$

and analogously $m_2(t_1, t_2)$.

Usage

```
biv_mean_residual(
  t1,
  t2,
  surv_fn = NULL,
  k1 = 1,
  k2 = 1,
  theta = 0,
  upper = 100
)
```

Arguments

`t1, t2`	Non-negative thresholds.
`surv_fn`	Joint survival function; defaults to Gumbel bivariate exponential.
`k1, k2, theta`	Gumbel parameters (used when `surv_fn = NULL`).
`upper`	Upper integration bound (default `100`).

Value

A named numeric vector `(m1, m2)`.

References

Jayalekshmi S., Rajesh G., Nair N.U. (2022). doi:10.1080/03610926.2022.2085874

Examples

```
biv_mean_residual(t1 = 0.5, t2 = 0.5)
biv_mean_residual(t1 = 1, t2 = 0.5, k1 = 1, k2 = 2, theta = 0.2)
```

`biv_rhazard_gradient` *Bivariate Reversed Hazard Gradient*

Description

Computes the bivariate reversed hazard gradient

$$h_i(x_1, x_2) = \frac{\partial}{\partial x_i} \log F(x_1, x_2), \quad i = 1, 2.$$

Usage

```
biv_rhazard_gradient(x1, x2, cdf_fn = NULL, theta = 0, eps = 1e-07)
```

Arguments

`x1, x2`	Positive evaluation points.
`cdf_fn`	Joint CDF function; defaults to FGM with parameter `theta`.
`theta`	FGM parameter (used when `cdf_fn = NULL`).
`eps`	Numerical differentiation step.

Value

Named numeric vector `(rh1, rh2)`.

References

Jayalekshmi S., Rajesh G. — IJRQSE, Section 2.

Examples

```
biv_rhazard_gradient(x1 = 0.5, x2 = 0.5, theta = 0.3)
```

`biv_rmrl` *Bivariate Reversed Mean Residual Life Function*

Description

Computes the bivariate reversed mean residual life (RMRL):

$$m_1(x_1, x_2) = \frac{\int_0^{x_1} F(t_1, x_2)\, dt_1}{F(x_1, x_2)}, \quad m_2(x_1, x_2) = \frac{\int_0^{x_2} F(x_1, t_2)\, dt_2}{F(x_1, x_2)}.$$

Usage

```
biv_rmrl(x1, x2, cdf_fn = NULL, theta = 0)
```

Arguments

`x1, x2`	Positive evaluation points.
`cdf_fn`	Joint CDF; defaults to FGM.
`theta`	FGM parameter.

Value

Named numeric vector `(m1, m2)`.

References

Jayalekshmi S., Rajesh G. — IJRQSE, Section 2.

Examples

```
biv_rmrl(x1 = 0.5, x2 = 0.5, theta = 0.3)
biv_rmrl(x1 = 0.7, x2 = 0.4, theta = -0.2)
```

`biv_whr_order` *Weak Bivariate Hazard Rate Order*

Description

Checks whether $X \leq_{\text{whr}} Y$: the ratio $\bar{F}_Y(x_1, x_2)/\bar{F}_X(x_1, x_2)$ is increasing in (x_1, x_2), i.e.\ $h_{i,X}(t_1, t_2) \geq h_{i,Y}(t_1, t_2),\ i = 1, 2$.

Usage

```
biv_whr_order(
  surv_X,
  surv_Y,
  t1_grid = seq(0.1, 3, by = 0.5),
  t2_grid = seq(0.1, 3, by = 0.5),
  tol = 1e-06
)
```

Arguments

`surv_X, surv_Y`	Joint survival functions.
`t1_grid, t2_grid`	Evaluation grids.
`tol`	Tolerance.

Value

A list: `order_holds` (logical), `violations` (data frame).

References

Shaked M., Shanthikumar J.G. (2007). *Stochastic Orders*. Springer. Jayalekshmi S., Rajesh G., Nair N.U. (2022). doi:10.1080/03610926.2022.2085874

Examples

```
sX <- function(x1, x2) sgumbel_biv(x1, x2, k1 = 1, k2 = 1, theta = 0)
sY <- function(x1, x2) sgumbel_biv(x1, x2, k1 = 2, k2 = 2, theta = 0)
biv_whr_order(sX, sY)
```

`biv_wmrl_order` *Weak Bivariate Mean Residual Life Order*

Description

Checks whether $X \leq_{\mathrm{wmrl}} Y$: $m_{i,X}(t_1, t_2) \leq m_{i,Y}(t_1, t_2),\ i = 1, 2$.

Usage

```
biv_wmrl_order(
  surv_X,
  surv_Y,
  t1_grid = seq(0.1, 2, by = 0.5),
  t2_grid = seq(0.1, 2, by = 0.5),
  tol = 1e-06
)
```

Arguments

`surv_X`, `surv_Y`	Joint survival functions.
`t1_grid`, `t2_grid`	Evaluation grids.
`tol`	Tolerance.

Value

List with `order_holds` and `violations`.

References

Jayalekshmi S., Rajesh G., Nair N.U. (2022). doi:10.1080/03610926.2022.2085874

Examples

```
sX <- function(x1, x2) sgumbel_biv(x1, x2, k1 = 1.5, k2 = 1.5)
sY <- function(x1, x2) sgumbel_biv(x1, x2, k1 = 1, k2 = 1)
biv_wmrl_order(sX, sY)
```

`biv_wrhr_order` *Weak Bivariate Reversed Hazard Rate Order*

Description

Checks $X \leq_{\mathrm{wrhr}} Y$: the ratio $F_X(x_1, x_2)/F_Y(x_1, x_2)$ is decreasing in (x_1, x_2), i.e.\ $h_{i,X}(x_1, x_2) \leq h_{i,Y}(x_1, x_2),\ i = 1, 2$.

Usage

```
biv_wrhr_order(
  cdf_X,
  cdf_Y,
  x1_grid = seq(0.1, 0.9, by = 0.2),
  x2_grid = seq(0.1, 0.9, by = 0.2),
  tol = 1e-06
)
```

Arguments

`cdf_X`, `cdf_Y`	Joint CDFs.
`x1_grid`, `x2_grid`	Evaluation grids (positive values).
`tol`	Tolerance.

Value

List with `order_holds` and `violations`.

References

Jayalekshmi S., Rajesh G. — IJRQSE, Section 2.

Examples

```
cX <- function(x1, x2) pfgm_biv(x1, x2, theta = 0.2)
cY <- function(x1, x2) pfgm_biv(x1, x2, theta = 0.5)
biv_wrhr_order(cX, cY)
```

`biv_wrmrl_order` *Weak Bivariate Reversed Mean Residual Life Order*

Description

Checks $X \leq_{\text{wrmrl}} Y$: $m_{i,X}(x_1, x_2) \geq m_{i,Y}(x_1, x_2),\ i = 1, 2$.

Usage

```
biv_wrmrl_order(
  cdf_X,
  cdf_Y,
  x1_grid = seq(0.2, 0.8, by = 0.2),
  x2_grid = seq(0.2, 0.8, by = 0.2),
  tol = 1e-06
)
```

Arguments

`cdf_X, cdf_Y`	Joint CDFs.
`x1_grid, x2_grid`	Evaluation grids.
`tol`	Tolerance.

Value

List with `order_holds` and `violations`.

References

Jayalekshmi S., Rajesh G. — IJRQSE, Section 2.

Examples

```
cX <- function(x1, x2) pfgm_biv(x1, x2, theta = 0.1)
cY <- function(x1, x2) pfgm_biv(x1, x2, theta = 0.8)
biv_wrmrl_order(cX, cY)
```

`blt_order_residual` *Bivariate Laplace Transform Order of Residual Lives (BLt-rl)*

Description

Checks whether random vector X is smaller than Y in the bivariate Laplace transform order of residual lives (BLt-rl).

$X \leq_{\text{BLt-rl}} Y$ if and only if for all (t_1, t_2) in the support:

$$\frac{\int_{t_1}^{\infty} e^{-s_1 x_1} \bar{F}_X(x_1, t_2)\, dx_1}{e^{-s_1 t_1} \bar{F}_X(t_1, t_2)} \geq \frac{\int_{t_1}^{\infty} e^{-s_1 x_1} \bar{F}_Y(x_1, t_2)\, dx_1}{e^{-s_1 t_1} \bar{F}_Y(t_1, t_2)}$$

i.e.\ $L^*_{X_{t_1|t_2}}(s_1) \geq L^*_{Y_{t_1|t_2}}(s_1)$ for all t_1, t_2, s_1. The function evaluates this inequality at a grid.

Usage

```
blt_order_residual(
  surv_X,
  surv_Y,
  s1 = 1,
  s2 = 1,
  t1_grid = seq(0.1, 3, by = 0.5),
  t2_grid = seq(0.1, 3, by = 0.5),
  tol = 1e-06
)
```

Arguments

`surv_X`, `surv_Y`	Joint survival functions for X and Y respectively, each of the form `function(x1, x2)`.
`s1`, `s2`	Laplace parameters to evaluate at.
`t1_grid`, `t2_grid`	Grids of truncation times (default 0.1 to 3).
`tol`	Tolerance for declaring inequality (default `1e-6`).

Value

A list with elements:

`order_holds` Logical; `TRUE` if $X \leq Y$ at all grid points.

`violations` Data frame of grid points where the ordering fails.

`ratio_matrix` Matrix of L^*_X / L^*_Y values.

References

Jayalekshmi S., Rajesh G., Nair N.U. (2022), Definition 4.1 & Prop. 4.1. doi:10.1080/03610926.2022.2085874

See Also

`blt_residual`, `biv_whr_order`

Examples

```
# Compare two Gumbel distributions
sX <- function(x1, x2) sgumbel_biv(x1, x2, k1 = 1, k2 = 1, theta = 0.2)
sY <- function(x1, x2) sgumbel_biv(x1, x2, k1 = 2, k2 = 2, theta = 0.2)
blt_order_residual(sX, sY, s1 = 1, s2 = 1,
                   t1_grid = c(0.1, 0.5, 1), t2_grid = c(0.1, 0.5))
```

`blt_order_reversed` *Bivariate Laplace Transform Order of Reversed Residual Lives (BLt-Rrl)*

Description

Checks whether $X \leq_{\text{BLt-Rrl}} Y$: for all (t_1, t_2), $L^X_{t_1|t_2}(s_1) \geq L^Y_{t_1|t_2}(s_2)$.

Usage

```
blt_order_reversed(
  cdf_X,
  cdf_Y,
  s1 = 1,
  s2 = 1,
  t1_grid = seq(0.2, 0.8, by = 0.2),
  t2_grid = seq(0.2, 0.8, by = 0.2),
  tol = 1e-06
)
```

Arguments

`cdf_X`, `cdf_Y`	Joint CDF functions for X and Y.
`s1`, `s2`	Laplace parameters.
`t1_grid`, `t2_grid`	Grids of truncation times.
`tol`	Tolerance.

Value

Same structure as `blt_order_residual`.

References

Jayalekshmi S., Rajesh G. — IJRQSE, Definition 2.

See Also

blt_reversed, biv_wrhr_order

Examples

```
cX <- function(x1, x2) pfgm_biv(x1, x2, theta = 0.3)
cY <- function(x1, x2) pfgm_biv(x1, x2, theta = 0.5)
blt_order_reversed(cX, cY, s1 = 1, s2 = 1,
                   t1_grid = c(0.2, 0.5), t2_grid = c(0.2, 0.5))
```

`blt_residual` *Bivariate Laplace Transform of Residual Lives*

Description

Computes the bivariate Laplace transform of residual lives $L_{X_{t_1|t_2}}(s_1)$ and $L_{X_{t_2|t_1}}(s_2)$ defined as

$$L_{X_{t_1|t_2}}(s_1) = \frac{\int_{t_1}^{\infty} e^{-s_1 x_1} f(x_1|X_2 > t_2)\, dx_1}{e^{-s_1 t_1} \bar{F}(t_1|X_2 > t_2)}$$

and analogously for the second component.

The *star* version (used for ordering) is

$$L^*_{X_{t_1|t_2}}(s_1) = \frac{1 - L_{X_{t_1|t_2}}(s_1)}{s_1} = \frac{\int_{t_1}^{\infty} e^{-s_1 x_1} \bar{F}(x_1, t_2)}{e^{-s_1 t_1} \bar{F}(t_1, t_2)}\, dx_1.$$

Usage

```
blt_residual(
  s1,
  s2,
  t1 = 0,
  t2 = 0,
  surv_fn = NULL,
  k1 = 1,
  k2 = 1,
  theta = 0,
  upper = 50,
  star = FALSE
)
```

Arguments

`s1, s2`	Positive Laplace parameters.
`t1, t2`	Non-negative time thresholds (truncation ages).
`surv_fn`	A function `function(x1, x2)` returning the joint survival probability $\bar{F}(x_1, x_2)$. Defaults to the Gumbel bivariate exponential with `k1`, `k2`, `theta`.
`k1, k2`	Rate parameters (used only when `surv_fn = NULL`).
`theta`	Association parameter (used only when `surv_fn = NULL`).
`upper`	Integration upper bound (default `50`).
`star`	Logical; if `TRUE` returns the star version L^*.

Value

A named numeric vector with elements `L1` and `L2` (or `L1_star` and `L2_star` when `star = TRUE`).

References

Jayalekshmi S., Rajesh G., Nair N.U. (2022). doi:10.1080/03610926.2022.2085874

See Also

`blt_residual_gumbel`, `blt_order_residual`, `np_blt_residual`

Examples

```
# Gumbel bivariate exponential, default parameters
blt_residual(s1 = 1, s2 = 1, t1 = 0.5, t2 = 0.5)

# Star version used in ordering
blt_residual(s1 = 0.5, s2 = 0.5, t1 = 1, t2 = 1, star = TRUE)

# User-supplied survival function (Schur-constant exponential)
surv <- function(x1, x2) exp(-(x1 + x2))
blt_residual(s1 = 1, s2 = 1, t1 = 0.3, t2 = 0.3, surv_fn = surv)
```

`blt_residual_gumbel` *Closed-Form Bivariate Laplace Transform of Residual Lives (Gumbel)*

Description

Returns the *star* bivariate Laplace transform of residual lives for the Gumbel bivariate exponential distribution in closed form:

$$L^*_{X_{t_1|t_2}}(s_1) = \frac{1}{k_1 + s_1 + \theta t_2}, \quad L^*_{X_{t_2|t_1}}(s_2) = \frac{1}{k_2 + s_2 + \theta t_1}.$$

Usage

```
blt_residual_gumbel(s1, s2, t1 = 0, t2 = 0, k1 = 1, k2 = 1, theta = 0)
```

Arguments

`s1, s2`	Positive Laplace parameters.
`t1, t2`	Non-negative truncation ages.
`k1, k2`	Positive rate parameters.
`theta`	Non-negative association parameter.

Value

A named numeric vector (`L1_star`, `L2_star`).

References

Jayalekshmi S., Rajesh G., Nair N.U. (2022), Example 3.1. doi:10.1080/03610926.2022.2085874

See Also

`blt_residual`

Examples

```
blt_residual_gumbel(s1 = 1, s2 = 1, t1 = 0.5, t2 = 0.5)
blt_residual_gumbel(s1 = 2, s2 = 0.5, t1 = 0, t2 = 1, k1 = 2, k2 = 1, theta = 0.3)
```

`blt_reversed` *Bivariate Laplace Transform of Reversed Residual Lives*

Description

Computes the bivariate Laplace transform of reversed (past) residual lives. For a random pair (X_1, X_2) with joint distribution function $F(x_1, x_2)$, the transform is

$$L_{t_1|t_2}(s_1) = e^{-s_1 t_1} + \frac{s_1 \int_0^{t_1} e^{-s_1 x_1} F(x_1, t_2)\, dx_1}{F(t_1, t_2)}$$

and the associated G function (useful for ordering) is

$$G_1(t_1, t_2) = \frac{\int_0^{t_1} e^{-s_1 x_1} F(x_1, t_2)\, dx_1}{e^{-s_1 t_1} F(t_1, t_2)}.$$

Usage

```
blt_reversed(s1, s2, t1, t2, cdf_fn = NULL, theta = 0, g_form = FALSE)
```

Arguments

`s1`, `s2`	Positive Laplace parameters.
`t1`, `t2`	Positive truncation times ($t_i > 0$).
`cdf_fn`	A function `function(x1, x2)` returning the joint CDF $F(x_1, x_2)$. Defaults to the FGM distribution.
`theta`	FGM association parameter (used when `cdf_fn = NULL`).
`g_form`	Logical; if `TRUE` returns the G form instead of L.

Value

A named numeric vector `(L1, L2)` or `(G1, G2)`.

References

Jayalekshmi S., Rajesh G. Bivariate Laplace transform order and ordering of reversed residual lives. *International Journal of Reliability, Quality and Safety Engineering*.

See Also

`blt_reversed_fgm`, `blt_order_reversed`

Examples

```
# FGM distribution (default)
blt_reversed(s1 = 1, s2 = 1, t1 = 0.6, t2 = 0.6)

# G form
blt_reversed(s1 = 1, s2 = 1, t1 = 0.6, t2 = 0.6, g_form = TRUE)

# User-supplied CDF
cdf <- function(x1, x2) pfgm_biv(x1, x2, theta = 0.5)
blt_reversed(s1 = 1, s2 = 1, t1 = 0.5, t2 = 0.5, cdf_fn = cdf)
```

`blt_reversed_fgm` *Closed-Form BLT of Reversed Residual Lives for the FGM Distribution*

Description

Computes the bivariate Laplace transform of the FGM distribution in closed form (Jayalekshmi and Rajesh, Example 1):

$$L_X(s_1, s_2) = \frac{(1 - e^{-s_1})(1 - e^{-s_2})}{s_1 s_2} + \theta\Phi(s_1, s_2)$$

where Φ captures the dependence correction.

Usage

```
blt_reversed_fgm(s1, s2, theta = 0)
```

Arguments

`s1, s2`	Positive Laplace parameters.
`theta`	FGM association parameter, $|\theta| \leq 1$.

Value

Scalar numeric; the joint bivariate Laplace transform.

References

Jayalekshmi S., Rajesh G. Bivariate Laplace transform order and ordering of reversed residual lives. *International Journal of Reliability, Quality and Safety Engineering*, Example 1.

Examples

```
blt_reversed_fgm(s1 = 1, s2 = 1, theta = 0.5)
blt_reversed_fgm(s1 = 2, s2 = 0.5, theta = -0.3)
```

`blt_reversed_power` *Bivariate Laplace Transform of Reversed Residual Lives — Power Distribution*

Description

Computes the G_1 function for the bivariate power distribution:

$$G_1(t_1, t_2) = \frac{\int_0^{t_1} e^{-s_1 x_1} x_1^{p_1+\theta \log t_2} \, dx_1}{e^{-s_1 t_1} \, t_1^{p_1+\theta \log t_2}}$$

evaluated numerically.

Usage

```
blt_reversed_power(s1, t1, t2, p1 = 1, p2 = 1, theta = 0)
```

Arguments

`s1`	Positive Laplace parameter.
`t1, t2`	Positive truncation times.
`p1, p2`	Positive shape parameters.
`theta`	Association parameter, $0 \leq \theta \leq 1$.

Value

Named numeric vector (`G1, reversed_hazard_h1`).

References

Jayalekshmi S., Rajesh G. — IJRQSE, Example 2.

Examples

```
blt_reversed_power(s1 = 1, t1 = 0.5, t2 = 0.5, p1 = 2, p2 = 2, theta = 0.3)
```

`fgm_biv` *Farlie-Gumbel-Morgenstern (FGM) Bivariate Distribution*

Description

Density, distribution function, survival function, and random generation for the FGM bivariate distribution on $[0,1]^2$:

$$F(x_1,x_2) = x_1 x_2 \big[1 + \theta(1-x_1)(1-x_2)\big], \quad 0 \le x_1, x_2 \le 1,\ |\theta| \le 1.$$

Usage

```
pfgm_biv(x1, x2, theta = 0)

dfgm_biv(x1, x2, theta = 0)

sfgm_biv(x1, x2, theta = 0)

rfgm_biv(n, theta = 0)
```

Arguments

`x1, x2`	Values in $[0,1]$.
`theta`	Association parameter, $\lvert\theta\rvert \le 1$.
`n`	Number of random observations.

Value

Numeric vector (scalar functions) or two-column matrix (`rfgm_biv`).

References

Jayalekshmi S., Rajesh G. Bivariate Laplace transform order and ordering of reversed residual lives. *International Journal of Reliability, Quality and Safety Engineering*.

Examples

```
pfgm_biv(0.4, 0.6, theta = 0.5)
dfgm_biv(0.4, 0.6, theta = 0.5)
set.seed(1); head(rfgm_biv(50, theta = 0.5))
```

`gumbel_biv` *Gumbel Bivariate Exponential Distribution*

Description

Density, distribution (survival) function, and random generation for the Gumbel bivariate exponential distribution with parameters k_1, k_2, and association parameter θ.

The joint survival function is

$$\bar{F}(x_1, x_2) = \exp(-k_1 x_1 - k_2 x_2 - \theta x_1 x_2), \quad x_1, x_2 > 0,\ k_1, k_2 > 0,\ 0 \le \theta \le k_1 k_2.$$

Usage

```
dgumbel_biv(x1, x2, k1 = 1, k2 = 1, theta = 0, log.p = FALSE)

sgumbel_biv(x1, x2, k1 = 1, k2 = 1, theta = 0, log.p = FALSE)

pgumbel_biv(x1, x2, k1 = 1, k2 = 1, theta = 0)

rgumbel_biv(n, k1 = 1, k2 = 1, theta = 0)
```

Arguments

`x1, x2`	Non-negative numeric values or vectors.
`k1, k2`	Positive rate parameters.
`theta`	Non-negative association parameter; must satisfy $0 \le \theta \le k_1 k_2$.
`log.p`	Logical; if `TRUE` probabilities are given as $\log(p)$.
`n`	Number of random observations.

Value

`dgumbel_biv`	Numeric vector of density values.
`pgumbel_biv`	Numeric vector of joint CDF values.
`sgumbel_biv`	Numeric vector of joint survival function values.
`rgumbel_biv`	A two-column matrix with columns `X1` and `X2` containing the simulated observations.

References

Gumbel E.J. (1960). Bivariate exponential distributions. *Journal of the American Statistical Association*, 55(292), 698–707.

Jayalekshmi S., Rajesh G., Nair N.U. (2022). doi:10.1080/03610926.2022.2085874

Examples

```
# Survival function
sgumbel_biv(1, 2, k1 = 1, k2 = 1, theta = 0.5)

# Density
dgumbel_biv(0.5, 0.5, k1 = 1, k2 = 1.5, theta = 0.3)

# Random sample
set.seed(42)
dat <- rgumbel_biv(100, k1 = 1, k2 = 1, theta = 0.5)
head(dat)
```

`hazard_rate` *Univariate Hazard Rate Function*

Description

Computes the hazard rate (failure rate) of a non-negative continuous random variable:

$$h(t) = \frac{f(t)}{\bar{F}(t)}, \quad t \geq 0.$$

Usage

```
hazard_rate(dens_fn, surv_fn = NULL, t, upper = 100)
```

Arguments

`dens_fn`	Density function $f(t)$.
`surv_fn`	Survival function $\bar{F}(t)$; computed by numerical integration if `NULL`.
`t`	Scalar or numeric vector of time points.
`upper`	Upper integration limit (used only when `surv_fn` = `NULL`).

Value

Numeric vector of hazard rates at `t`.

See Also

`mean_residual`, `hr_order`

Examples

```
# Exp(1): constant hazard rate = 1
f  <- function(x) dexp(x, 1)
Fb <- function(x) pexp(x, 1, lower.tail = FALSE)
hazard_rate(f, Fb, t = c(0.5, 1, 2))

# Gamma(2,1): increasing hazard rate
fG  <- function(x) dgamma(x, shape = 2, rate = 1)
FbG <- function(x) pgamma(x, shape = 2, rate = 1, lower.tail = FALSE)
hazard_rate(fG, FbG, t = c(0.5, 1, 2))
```

`hr_order` *Hazard Rate Order*

Description

Checks whether $X \leq_{\mathrm{hr}} Y$ (hazard rate order): the hazard rate of X is pointwise no greater than that of Y:

$$h_X(t) \leq h_Y(t) \quad \forall\, t \geq 0.$$

Under this order X is stochastically longer-lived than Y. For exponential distributions, $\mathrm{Exp}(\lambda_1) \leq_{\mathrm{hr}} \mathrm{Exp}(\lambda_2)$ iff $\lambda_1 \leq \lambda_2$.

Usage

```
hr_order(
  dens_fn_X,
  surv_fn_X = NULL,
  dens_fn_Y,
  surv_fn_Y = NULL,
  t_grid = seq(0.1, 3, by = 0.5),
  upper = 100
)
```

Arguments

`dens_fn_X, dens_fn_Y`	Density functions of X and Y.
`surv_fn_X, surv_fn_Y`	Survival functions; computed if `NULL`.
`t_grid`	Grid of time points.
`upper`	Integration upper bound.

Value

A list with:

- `order_holds` Logical.
- `max_violation` Maximum of $h_X(t) - h_Y(t)$ over the grid.
- `hazard_X` Hazard rate of X at `t_grid`.
- `hazard_Y` Hazard rate of Y at `t_grid`.

See Also

`hazard_rate`, `mrl_order`, `lt_rl_order`

Examples

```
# Exp(1) <=_hr Exp(2): h_X(t)=1 <= 2=h_Y(t)
fX  <- function(x) dexp(x, 1)
FbX <- function(x) pexp(x, 1, lower.tail = FALSE)
fY  <- function(x) dexp(x, 2)
FbY <- function(x) pexp(x, 2, lower.tail = FALSE)
hr_order(fX, FbX, fY, FbY, t_grid = c(0.5, 1, 2))$order_holds
```

`info_gen_function` *Golomb Information Generating Function*

Description

Computes the information generating function (IGF) introduced by Golomb (1966):

$$\mathcal{I}_\alpha(f) = \int_0^\infty f^\alpha(x)\,dx, \quad \alpha > 0.$$

When $\alpha \to 1$, $-d\mathcal{I}_\alpha/d\alpha|_{\alpha=1} = H(f)$.

Usage

```
info_gen_function(dens_fn, alpha = 1, upper = 100)
```

Arguments

`dens_fn`	A function of one argument returning the density.
`alpha`	Positive parameter (default 1).
`upper`	Upper integration limit.

Value

Scalar numeric.

References

Golomb S.W. (1966). The information generating function of a probability distribution. *IEEE Transactions on Information Theory*, 12(1), 75–77.

See Also

shannon_entropy

Examples

```
# Exponential(1) with alpha=1 gives 1
info_gen_function(function(x) dexp(x, rate = 1), alpha = 1)

# alpha = 2
info_gen_function(function(x) dexp(x, rate = 1), alpha = 2)
```

`lt_residual` *Univariate Laplace Transform of Residual Life*

Description

Computes the Laplace transform of the residual life of a non-negative continuous random variable conditioned on survival past time t:

$$L_X(s,t) = E[e^{-sX} \mid X > t] = \frac{1}{\bar{F}(t)} \int_t^{\infty} e^{-sx} f(x)\, dx, \quad s \geq 0,\ t \geq 0.$$

At $t = 0$ this reduces to the standard Laplace transform $L_X(s) = E[e^{-sX}]$.

Usage

```
lt_residual(dens_fn, surv_fn = NULL, s, t = 0, upper = 100)
```

Arguments

`dens_fn`	Density function $f(x)$.
`surv_fn`	Survival function $\bar{F}(x)$; computed by numerical integration of `dens_fn` if `NULL`.
`s`	Non-negative Laplace parameter.
`t`	Truncation time (default 0).
`upper`	Upper integration limit.

Value

Scalar numeric.

References

Belzunce F., Ortega E., Ruiz J.M. (1999). The Laplace order and ordering of residual lives. *Statistics & Probability Letters*, 42(2), 145–156.

See Also

`hazard_rate`, `mean_residual`, `np_lt_residual`, `lt_rl_order`, `blt_residual`

Examples

```
# Exp(1): L_X(s, 0) = 1/(1+s) = 0.5 at s=1
f  <- function(x) dexp(x, 1)
Fb <- function(x) pexp(x, 1, lower.tail = FALSE)
lt_residual(f, Fb, s = 1, t = 0)

# Memoryless property: L_X(s,t) should equal L_X(s,0) for Exp
lt_residual(f, Fb, s = 1, t = 0.5)
```

`lt_rl_order` *Univariate Laplace Transform Order of Residual Lives*

Description

Checks whether $X \leq_{\text{Lt-rl}} Y$: the Laplace transform of the residual life of X is dominated by that of Y pointwise over a grid of (s, t) values:

$$L_X(s,t) \leq L_Y(s,t) \quad \forall\, s \geq 0,\ t \geq 0.$$

The order is verified numerically on `s_grid` x `t_grid`.

Usage

```
lt_rl_order(
  dens_fn_X,
  surv_fn_X = NULL,
  dens_fn_Y,
  surv_fn_Y = NULL,
  s_grid = seq(0.5, 3, by = 0.5),
  t_grid = seq(0, 2, by = 0.5),
  upper = 100
)
```

Arguments

`dens_fn_X, dens_fn_Y`	Density functions of X and Y.
`surv_fn_X, surv_fn_Y`	Survival functions; computed if `NULL`.
`s_grid`	Numeric vector of Laplace parameter values to check.
`t_grid`	Numeric vector of truncation times to check.
`upper`	Integration upper bound.

Value

A list with:

`order_holds`	Logical; `TRUE` if the order holds at all grid points.
`max_violation`	Maximum violation $\max(L_X - L_Y, 0)$.
`ratio_matrix`	Matrix of $L_X(s,t)/L_Y(s,t)$ values (rows = `s_grid`, columns = `t_grid`).

See Also

`lt_residual`, `hr_order`, `mrl_order`, `blt_order_residual`

Examples

```
# Exp(1) <=_Lt-rl Exp(2): L_{Exp(lambda)}(s,t) = lambda*exp(-s*t)/(s+lambda)
# For s>0: 1/(s+1) < 2/(s+2), so Exp(1) has smaller LT of residual life
fX  <- function(x) dexp(x, 1)
FbX <- function(x) pexp(x, 1, lower.tail = FALSE)
fY  <- function(x) dexp(x, 2)
FbY <- function(x) pexp(x, 2, lower.tail = FALSE)
lt_rl_order(fX, FbX, fY, FbY,
            s_grid = c(0.5, 1, 2), t_grid = c(0, 0.5, 1))$order_holds
```

`mean_residual` *Univariate Mean Residual Life*

Description

Computes the mean residual life (mean excess function):

$$m(t) = E[X - t \mid X > t] = \frac{1}{\bar{F}(t)} \int_t^{\infty} \bar{F}(x)\, dx, \quad t \geq 0.$$

Usage

```
mean_residual(surv_fn, t = 0, upper = 100)
```

Arguments

`surv_fn`	Survival function $\bar{F}(x)$.
`t`	Scalar or numeric vector of time points.
`upper`	Upper integration limit.

Value

Numeric vector of MRL values at `t`.

See Also

`hazard_rate`, `mrl_order`

Examples

```
# Exp(1): constant MRL = 1 (memoryless)
Fb <- function(x) pexp(x, 1, lower.tail = FALSE)
mean_residual(Fb, t = c(0, 0.5, 1, 2))

# Gamma(2,1): decreasing MRL
FbG <- function(x) pgamma(x, shape = 2, rate = 1, lower.tail = FALSE)
mean_residual(FbG, t = c(0, 0.5, 1, 2))
```

`mrl_order` *Mean Residual Life Order*

Description

Checks whether $X \leq_{\mathrm{mrl}} Y$ (mean residual life order): the MRL of X is pointwise no greater than that of Y:

$$m_X(t) \leq m_Y(t) \quad \forall t \geq 0.$$

Usage

```
mrl_order(surv_fn_X, surv_fn_Y, t_grid = seq(0, 3, by = 0.5), upper = 100)
```

Arguments

`surv_fn_X`, `surv_fn_Y`	Survival functions of X and Y.
`t_grid`	Grid of time points.
`upper`	Integration upper bound.

Value

A list with:

`order_holds` Logical.

`max_violation` Maximum of $m_X(t) - m_Y(t)$ over the grid.

`mrl_X` MRL of X at `t_grid`.

`mrl_Y` MRL of Y at `t_grid`.

See Also

`mean_residual`, `hr_order`, `lt_rl_order`

Examples

```
# Exp(2) <=_mrl Exp(1): m_X(t)=0.5 <= 1=m_Y(t)
FbX <- function(x) pexp(x, 2, lower.tail = FALSE)
FbY <- function(x) pexp(x, 1, lower.tail = FALSE)
mrl_order(FbX, FbY, t_grid = c(0, 0.5, 1, 2))$order_holds
```

`nbuhr_test` *Test NBUHR / NWUHR Aging Class*

Description

Checks whether a bivariate lifetime distribution belongs to the NBUHR (New Better than Used in Hazard Rate) or NWUHR (New Worse than Used) aging class. A distribution is NBUHR if

$$h_1(0, t_2) \geq h_1(t_1, t_2) \text{ for all } t_1 > 0$$

and similarly for the second component. The function evaluates this at a grid of t_1 values.

Usage

```
nbuhr_test(
  t2 = 1,
  t1_grid = seq(0.1, 5, by = 0.1),
  surv_fn = NULL,
  k1 = 1,
  k2 = 1,
  theta = 0
)
```

Arguments

`t2`	Fixed value of the second age coordinate.
`t1_grid`	Numeric vector of t_1 values to check (default 0.1 to 5 in steps of 0.1).
`surv_fn`	Joint survival function; defaults to Gumbel bivariate exponential.
`k1`, `k2`, `theta`	Gumbel parameters.

Value

A list with components:

`class1` Character: `"NBUHR"`, `"NWUHR"`, or `"neither"` for the first component.

`class2` Same for the second component.

`h1_grid` Numeric vector of $h_1(t_1, t_2)$ values.

`h2_grid` Numeric vector of $h_2(t_1, t_2)$ values.

References

Jayalekshmi S., Rajesh G., Nair N.U. (2022), Definition 3.2. doi:10.1080/03610926.2022.2085874

Examples

```
nbuhr_test(t2 = 1, k1 = 1, k2 = 1, theta = 0.3)
nbuhr_test(t2 = 0.5,
           surv_fn = function(x1, x2) exp(-(x1 + x2)))
```

`np_blt_residual` *Nonparametric Estimator for the Bivariate Laplace Transform of Residual Lives*

Description

Given a bivariate sample $(X_{1i}, X_{2i}),\ i = 1, \ldots, n$, estimates

$$\hat{L}_1^*(s_1; t_1, t_2) = \frac{\sum_{i:X_{1i}>t_1,\, X_{2i}>t_2} \int_{t_1}^{X_{1i}} e^{-s_1 u}\, du}{e^{-s_1 t_1} \cdot \#\{X_{1i} > t_1, X_{2i} > t_2\}}$$

and analogously for the second component, using the empirical survival function as described in Jayalekshmi et al. (2022), Section 6.

Usage

```
np_blt_residual(data, s1, s2, t1 = 0, t2 = 0)
```

Arguments

`data`	A two-column numeric matrix or data frame with columns for X_1 and X_2.
`s1, s2`	Positive Laplace parameters.
`t1, t2`	Non-negative truncation ages.

Value

A named numeric vector (`L1_hat, L2_hat`).

References

Jayalekshmi S., Rajesh G., Nair N.U. (2022), Section 6. doi:10.1080/03610926.2022.2085874

See Also

`blt_residual_gumbel`, `sim_blt_residual`

Examples

```
set.seed(123)
dat <- rgumbel_biv(200, k1 = 1, k2 = 1, theta = 0.5)
np_blt_residual(dat, s1 = 1, s2 = 1, t1 = 0.3, t2 = 0.3)

# Compare with closed form
blt_residual_gumbel(s1 = 1, s2 = 1, t1 = 0.3, t2 = 0.3, theta = 0.5)
```

`np_lt_residual` *Nonparametric Estimator for the Univariate Laplace Transform of Residual Life*

Description

Given a sample $X_1, \ldots, X_n$, estimates the Laplace transform of the residual life using the empirical survival function:

$$\hat{L}_X(s,t) = \frac{\sum_{i:\, X_i > t} e^{-sX_i}}{\#\{i : X_i > t\}}.$$

Usage

```
np_lt_residual(x, s, t = 0)
```

Arguments

`x`	Numeric vector; observed sample.
`s`	Non-negative Laplace parameter.
`t`	Truncation time (default 0).

Value

Scalar numeric estimate of $L_X(s,t)$.

See Also

`lt_residual`

Examples

```
set.seed(1)
x <- rexp(300, rate = 1)

# Estimate at s=1, t=0: true value 1/(1+1) = 0.5
np_lt_residual(x, s = 1, t = 0)

# Estimate at s=1, t=0.5: true value still approx 0.5 (memoryless)
np_lt_residual(x, s = 1, t = 0.5)
```

`plot_blt_residual` *Plot Bivariate Laplace Transform of Residual Lives*

Description

Plots the star Laplace transform of residual lives $L^*_{X_{t_1|t_2}}(s_1)$ as a function of t_1 for fixed s_1, t_2. Optionally overlays two distributions for visual comparison of the BLt-rl order.

Usage

```
plot_blt_residual(
  surv_fn,
  surv_fn2 = NULL,
  s1 = 1,
  t2 = 0.5,
  t1_grid = seq(0.1, 3, by = 0.1),
  k1 = 1,
  k2 = 1,
  theta = 0,
  xlab = expression(t[1]),
  ylab = expression(L^"*"[X[t[1] * "|" * t[2]]](s[1])),
  main = "Bivariate LT of Residual Lives",
  col1 = "steelblue",
  col2 = "firebrick",
  lwd = 2,
  legend_labels = c("Distribution 1", "Distribution 2")
)
```

Arguments

`surv_fn`	Joint survival function. If a second distribution is to be overlaid, pass it as `surv_fn2`.
`surv_fn2`	Optional second survival function for comparison.
`s1`	Laplace parameter (default 1).
`t2`	Fixed second truncation age (default 0.5).

`t1_grid`	Grid of first truncation ages.
`k1`, `k2`, `theta`	Parameters for the default Gumbel distribution; used only when `surv_fn = NULL`.
`xlab`, `ylab`, `main`	Plot labels.
`col1`, `col2`	Line colours.
`lwd`	Line width.
`legend_labels`	Length-2 character vector for legend (ignored if `surv_fn2 = NULL`).

Value

Invisibly returns the data frame used for plotting.

Examples

```
sX <- function(x1, x2) sgumbel_biv(x1, x2, k1 = 1, k2 = 1, theta = 0.3)
sY <- function(x1, x2) sgumbel_biv(x1, x2, k1 = 2, k2 = 1, theta = 0.3)
plot_blt_residual(sX, sY, s1 = 1, t2 = 0.5,
                  legend_labels = c("k1=1", "k1=2"))
```

`plot_blt_reversed` *Plot Bivariate Laplace Transform of Reversed Residual Lives*

Description

Plots the reversed-life Laplace transform $L_{t_1|t_2}(s_1)$ as a function of t_1 for fixed s_1 and t_2.

Usage

```
plot_blt_reversed(
  cdf_fn,
  cdf_fn2 = NULL,
  s1 = 1,
  t2 = 0.5,
  t1_grid = seq(0.1, 0.9, by = 0.05),
  theta = 0,
  xlab = expression(t[1]),
  ylab = expression(L[t[1] * "|" * t[2]](s[1])),
  main = "Bivariate LT of Reversed Residual Lives",
  col1 = "darkgreen",
  col2 = "darkorange",
  lwd = 2,
  legend_labels = c("Distribution 1", "Distribution 2")
)
```

Arguments

`cdf_fn`	Joint CDF function.
`cdf_fn2`	Optional second CDF for comparison.
`s1`	Laplace parameter.
`t2`	Fixed second truncation time.
`t1_grid`	Grid of first truncation times.
`theta`	FGM parameter (used if `cdf_fn = NULL`).
`xlab, ylab, main`	Plot labels.
`col1, col2`	Line colours.
`lwd`	Line width.
`legend_labels`	Legend labels.

Value

Invisibly returns the data frame used for plotting.

Examples

```
cX <- function(x1, x2) pfgm_biv(x1, x2, theta = 0.2)
cY <- function(x1, x2) pfgm_biv(x1, x2, theta = 0.7)
plot_blt_reversed(cX, cY, s1 = 1, t2 = 0.5,
                  legend_labels = c("theta=0.2", "theta=0.7"))
```

`schur_biv` *Schur-Constant Bivariate Distribution*

Description

Random generation and survival function for a Schur-constant bivariate distribution with survival function

$$\bar{F}(x_1, x_2) = S(x_1 + x_2), \quad x_1, x_2 > 0,$$

where S is a given univariate survival function. The default marginal is exponential with rate `lambda`.

Usage

```
sschur_biv(x1, x2, lambda = 1)

rschur_biv(n, lambda = 1)
```

Arguments

`x1, x2`	Non-negative values.
`lambda`	Exponential rate parameter for the generating survival function.
`n`	Number of random observations.

Value

Numeric vector (`sschur_biv`) or two-column matrix (`rschur_biv`).

References

Barlow R.E., Mendel M.B. (1992). De Finetti-type representations for life distributions. *Journal of the American Statistical Association*, 87(420), 1116–1122.

Examples

```
sschur_biv(0.5, 1, lambda = 1)
set.seed(2); head(rschur_biv(40, lambda = 1))
```

`shannon_entropy` *Shannon Differential Entropy*

Description

Computes the Shannon differential entropy

$$H(f) = -\int_0^\infty f(x) \log f(x)\, dx$$

for a non-negative continuous random variable with density `dens_fn`.

Usage

```
shannon_entropy(dens_fn, upper = 100)
```

Arguments

`dens_fn`	A function of one argument returning the density $f(x)$.
`upper`	Upper integration limit (default 100).

Value

Scalar numeric.

References

Shannon C.E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3), 379–423.

See Also

`info_gen_function`

Examples

```
# Exponential(1): H = 1
shannon_entropy(function(x) dexp(x, rate = 1))

# Exponential(2): H = 1 - log(2)
shannon_entropy(function(x) dexp(x, rate = 2))
```

`sim_blt_residual` *Monte-Carlo Simulation Study for the BLT Residual Estimator*

Description

Evaluates the performance of `np_blt_residual` via repeated simulation from the Gumbel bivariate exponential distribution and compares estimates to the closed-form `blt_residual_gumbel` values. Returns bias, variance, and mean squared error (MSE).

Usage

```
sim_blt_residual(
  n_obs = 200,
  n_sim = 100,
  s1 = 1,
  s2 = 1,
  t1 = 0.3,
  t2 = 0.3,
  k1 = 1,
  k2 = 1,
  theta = 0.5,
  seed = 42L
)
```

Arguments

`n_obs`	Sample size per replicate.
`n_sim`	Number of simulation replicates.
`s1`, `s2`	Laplace parameters.
`t1`, `t2`	Truncation ages.
`k1`, `k2`, `theta`	Gumbel parameters.
`seed`	Random seed for reproducibility.

Value

A data frame with columns `component`, `true_value`, `mean_est`, `bias`, `variance`, `mse`.

References

Jayalekshmi S., Rajesh G., Nair N.U. (2022), Section 6. doi:10.1080/03610926.2022.2085874

See Also

`np_blt_residual`

Examples

```
sim_blt_residual(n_obs = 100, n_sim = 50, s1 = 1, s2 = 1,
                 t1 = 0.3, t2 = 0.3, k1 = 1, k2 = 1, theta = 0.5)
```

Index