Package: tailDepFun (via r-universe)

November 3, 2024

Type Package

Title Minimum Distance Estimation of Tail Dependence Models

Description Provides functions implementing minimal distance estimation methods for parametric tail dependence models, as proposed in Einmahl, J.H.J., Kiriliouk, A., Krajina, A., and Segers, J. (2016) <doi:10.1111/rssb.12114> and Einmahl, J.H.J., Kiriliouk, A., and Segers, J. (2018) <doi:10.1007/s10687-017-0303-7>.

Version 1.0.1

Date 2021-05-27

Author Anna Kiriliouk

Maintainer Anna Kiriliouk <anna.kiriliouk@unamur.be>

Depends R (>= 3.2.3)

Imports cubature, mvtnorm, SpatialExtremes, copula, stats, utils

Suggests R.rsp

VignetteBuilder R.rsp

License GPL-3

LazyData true

Collate 'Other.R' 'helpFunctions.R' 'EstimationBR.R' 'EstimationGumbel.R' 'EstimationML.R' 'empbetacop.R'

RoxygenNote 7.1.1

Encoding UTF-8

NeedsCompilation yes

Date/Publication 2021-06-03 11:00:02 UTC

Repository https://akiriliouk.r-universe.dev

RemoteUrl https://github.com/cran/tailDepFun

RemoteRef HEAD

RemoteSha d9625211ef0f628d7352578a1582a03db6570708

Contents

`AsymVarBR` *Asymptotic variance matrix for the Brown-Resnick process.*

Description

Computes the asymptotic variance matrix for the Brown-Resnick process, estimated using the pairwise M-estimator or the weighted least squares estimator.

Usage

```
AsymVarBR(locations, indices, par, method, Tol = 1e-05)
```

Arguments

`locations`	A d x 2 matrix containing the Cartesian coordinates of d points in the plane.
`indices`	A q x d matrix containing exactly 2 ones per row, representing a pair of points from the matrix `locations`, and zeroes elsewhere.
`par`	The parameters of the Brown-Resnick process. Either (α, ρ) for an isotropic process or (α, ρ, β, c) for an anisotropic process.
`method`	Choose between "Mestimator" and "WLS".
`Tol`	For "Mestimator" only. The tolerance in the numerical integration procedure. Defaults to 1e-05.

Details

The parameters of a The matrix `indices` can be either user-defined or returned from the function `selectGrid` with `cst = c(0,1)`. Calculation might be rather slow for `method = "Mestimator"`.

Value

A q by q matrix.

References

Einmahl, J.H.J., Kiriliouk, A., Krajina, A., and Segers, J. (2016). An Mestimator of spatial tail dependence. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 78(1), 275-298.

Einmahl, J.H.J., Kiriliouk, A., and Segers, J. (2018). A continuous updating weighted least squares estimator of tail dependence in high dimensions. Extremes 21(2), 205-233.

See Also

selectGrid

Examples

```
locations <- cbind(rep(1:2, 3), rep(1:3, each = 2))
indices <- selectGrid(cst = c(0,1), d = 6, locations = locations, maxDistance = 1)
AsymVarBR(locations, indices, par = c(1.5,3), method = "WLS")
```

AsymVarGumbel *Asymptotic variance matrix for the Gumbel model.*

Description

Computes the asymptotic variance matrix for the Gumbel model, estimated using the pairwise M-estimator or the weighted least squares estimator.

Usage

```
AsymVarGumbel(indices, par, method)
```

Arguments

indices	A q x d matrix containing at least 2 non-zero elements per row, representing the values in which we will evaluate the stable tail dependence function. For method = Mestimator, this matrix should contain exactly two ones per row.
par	The parameter of the Gumbel model.
method	Choose between "Mestimator" and "WLS".

Details

The matrix `indices` can be either user defines or returned by `selectGrid`. For `method = "Mestimator"`, only a grid with exactly two ones per row is accepted, representing the pairs to be used.

Value

A q by q matrix.

References

Einmahl, J.H.J., Kiriliouk, A., Krajina, A., and Segers, J. (2016). An Mestimator of spatial tail dependence. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 78(1), 275-298.

Einmahl, J.H.J., Kiriliouk, A., and Segers, J. (2018). A continuous updating weighted least squares estimator of tail dependence in high dimensions. Extremes 21(2), 205-233.

See Also

`selectGrid`

Examples

```
indices <- selectGrid(c(0,1), d = 3, nonzero = c(2,3))
AsymVarGumbel(indices, par = 0.7, method = "WLS")
```

AsymVarMaxLinear *Asymptotic variance matrix for the max-linear model.*

Description

Computes the asymptotic variance matrix for the max-linear model, estimated using the weighted least squares estimator.

Usage

```
AsymVarMaxLinear(indices, par, Bmatrix = NULL)
```

Arguments

`indices`	A q x d matrix containing at least 2 non-zero elements per row, representing the values in which we will evaluate the stable tail dependence function.
`par`	The parameter vector.
`Bmatrix`	A function that converts the parameter vector theta to a parameter matrix B. If `NULL`, then a simple 2-factor model is assumed.

Value

A q by q matrix.

References

Einmahl, J.H.J., Kiriliouk, A., and Segers, J. (2018). A continuous updating weighted least squares estimator of tail dependence in high dimensions. Extremes 21(2), 205-233.

See Also

`selectGrid`

Examples

```
indices <- selectGrid(c(0,0.5,1), d = 3, nonzero = 3)
AsymVarMaxLinear(indices, par = c(0.1,0.55,0.75))
```

`dataEUROSTOXX` *EUROSTOXX50 weekly negative log-returns.*

Description

The first three columns represent the weekly negative log-returns of the index prices of the EUROSTOXX50 and of its subindices correspoding to the supersectors chemicals and insurance. The fourth and fifth columns represent the weekly negative log-returns of the index prices of the DAX and the CAC40 indices. The sixth to tenth columnds represent the weekly negative log-returns of the stock prices of Bayer, BASF, Allianz, AXA, and Airliquide respectively.

Format

`dataEUROSTOXX` is a matrix with 711 rows and 10 columns.

Source

Yahoo Finance

References

Einmahl, J.H.J., Kiriliouk, A., and Segers, J. (2018). A continuous updating weighted least squares estimator of tail dependence in high dimensions. Extremes 21(2), 205-233.

Examples

```
data(dataEUROSTOXX)
## Transform data to unit Pareto margins
n <- nrow(dataEUROSTOXX)
x <- apply(dataEUROSTOXX, 2, function(i) n/(n + 0.5 - rank(i)))
## Define indices in which we evaluate the estimator
indices <- selectGrid(c(0,0.5,1), d = 10, nonzero = c(2,3))
start <- c(0.67,0.8,0.77,0.91,0.41,0.47,0.25,0.7,0.72,0.19,0.37,0.7,0.09,0.58)
## Estimate the parameters. Lasts up to ten minutes.

EstimationMaxLinear(x, indices, k = 40, method = "WLS", startingValue = start,
covMat = FALSE, EURO = TRUE)
```

`dataKNMI` *Wind speeds in the Netherlands.*

Description

Daily maximal speeds of wind gusts, measured in 0.1 m/s. The data are observed at 22 inland weather stations in the Netherlands. Only the summer months are presented here (June, July, August). Also included are the Euclidian coordinates of the 22 weather stations, where a distance of 1 corresponds to 100 kilometers.

Format

dataKNMI$data is a matrix with 672 rows and 22 columns, dataKNMI$loc is a matrix with 22 rows and 2 columns.

Source

KNMI

References

Einmahl, J.H.J., Kiriliouk, A., Krajina, A., and Segers, J. (2016). An Mestimator of spatial tail dependence. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 78(1), 275-298.

Examples

```
data(dataKNMI)
n <- nrow(dataKNMI$data)
locations <- dataKNMI$loc
x <- apply(dataKNMI$data, 2, function(i) n/(n + 0.5 - rank(i)))
indices <- selectGrid(cst = c(0,1), d = 22, locations = locations, maxDistance = 0.5)
EstimationBR(x, locations, indices, k = 60, method = "Mestimator", isotropic = TRUE,
covMat = FALSE)$theta
```

`EstimationBR` *Estimation of the parameters of the Brown-Resnick process*

Description

Estimation the parameters of the Brown-Resnick process, using either the pairwise M-estimator or weighted least squares (WLS).

Usage

```
EstimationBR(
  x,
  locations,
  indices,
  k,
  method,
  isotropic = FALSE,
  biascorr = FALSE,
  Tol = 1e-05,
  k1 = (nrow(x) - 10),
  tau = 5,
  startingValue = NULL,
  Omega = diag(nrow(indices)),
  iterate = FALSE,
  covMat = TRUE
)
```

Arguments

`x`	An n x d data matrix.
`locations`	A d x 2 matrix containing the Cartesian coordinates of d points in the plane.
`indices`	A q x d matrix containing exactly 2 ones per row, representing a pair of points from the matrix `locations`, and zeroes elsewhere.
`k`	An integer between 1 and $n-1$; the threshold parameter in the definition of the empirical stable tail dependence function.
`method`	Choose between `Mestimator` and `WLS`.
`isotropic`	A Boolean variable. If `FALSE` (the default), then an anisotropic process is estimated.
`biascorr`	For `method = "WLS"` only. If `TRUE`, then the bias-corrected estimator of the stable tail dependence function is used. Defaults to `FALSE`.
`Tol`	For `method = "Mestimator"` only. The tolerance parameter used in the numerical integration procedure. Defaults to 1e-05.
`k1`	For `biascorr = TRUE` only. The value of k_1 in the definition of the bias-corrected estimator of the stable tail dependence function.
`tau`	For `biascorr = TRUE` only. The parameter of the power kernel.
`startingValue`	Initial value of the parameters in the minimization routine. Defaults to $c(1, 1.5)$ if `isotropic = TRUE` and $c(1, 1.5, 0.75, 0.75)$ if `isotropic = FALSE`.
`Omega`	A q x q matrix specifying the metric with which the distance between the parametric and nonparametric estimates will be computed. The default is the identity matrix, i.e., the Euclidean metric.
`iterate`	A Boolean variable. If `TRUE`, then for `method = "Mestimator"` the estimator is calculated twice, first with `Omega` specified by the user, and then a second time with the optimal `Omega` calculated at the initial estimate. If `method = "WLS"`, then continuous updating is used.

covMat — A Boolean variable. If TRUE (the default), the covariance matrix is calculated. Standard errors are obtained by taking the square root of the diagonal elements.

Details

The parameters of the Brown-Resnick process are either (α, ρ) for an isotropic process or (α, ρ, β, c) for an anisotropic process. The matrix `indices` can be either user-defined or returned from the function `selectGrid` with `cst = c(0,1)`. Estimation might be rather slow when `iterate = TRUE` or even when `covMat = TRUE`.

Value

A list with the following components:

`theta`	The estimator using the optimal weight matrix.
`theta_pilot`	The estimator without the optimal weight matrix.
`covMatrix`	The estimated covariance matrix for the estimator.
`value`	The value of the minimized function at `theta`.

References

Einmahl, J.H.J., Kiriliouk, A., and Segers, J. (2018). A continuous updating weighted least squares estimator of tail dependence in high dimensions. Extremes 21(2), 205-233.

Einmahl, J.H.J., Kiriliouk, A., Krajina, A., and Segers, J. (2016). An Mestimator of spatial tail dependence. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 78(1), 275-298.

See Also

`selectGrid`

Examples

```
## define the locations of 9 stations
locations <- cbind(rep(c(1:3), each = 3), rep(1:3, 3))
## select the pairs of locations
indices <- selectGrid(cst = c(0,1), d = 9, locations = locations, maxDistance = 1.5)
## The Brown-Resnick process
set.seed(1)
x <- SpatialExtremes::rmaxstab(n = 1000, coord = locations, cov.mod = "brown",
                               range = 3, smooth = 1)
## Calculate the estimtors.
EstimationBR(x, locations, indices, 100, method = "Mestimator", isotropic = TRUE,
            covMat = FALSE)$theta
EstimationBR(x, locations, indices, 100, method = "WLS", isotropic = TRUE,
covMat = FALSE)$theta
```

EstimationGumbel *Estimation of the parameter of the Gumbel model*

Description

Estimation the parameter of the Gumbel model, using either the pairwise M-estimator or weighted least squares (WLS).

Usage

```
EstimationGumbel(
  x,
  indices,
  k,
  method,
  biascorr = FALSE,
  k1 = (nrow(x) - 10),
  tau = 5,
  covMat = TRUE
)
```

Arguments

`x`	An n x d data matrix.
`indices`	A q x d matrix containing at least 2 non-zero elements per row, representing the values in which we will evaluate the stable tail dependence function. For `method = Mestimator`, this matrix should contain exactly two ones per row.
`k`	An integer between 1 and $n - 1$; the threshold parameter in the definition of the empirical stable tail dependence function.
`method`	Choose between `Mestimator` and `WLS`.
`biascorr`	For `method = "WLS"` only. If `TRUE`, then the bias-corrected estimator of the stable tail dependence function is used. Defaults to `FALSE`.
`k1`	For `biascorr = TRUE` only. The value of k_1 in the definition of the bias-corrected estimator of the stable tail dependence function.
`tau`	For `biascorr = TRUE` only. The parameter of the power kernel.
`covMat`	A Boolean variable. If `TRUE` (the default), the covariance matrix is calculated. Standard errors are obtained by taking the square root of the diagonal elements.

Details

The matrix `indices` can be either user defined or returned by `selectGrid`. For `method = "Mestimator"`, only a grid with exactly two ones per row is accepted, representing the pairs to be used.

Value

For WLS, a list with the following components:

theta	The estimator with weight matrix identity.
covMatrix	The estimated covariance matrix for the estimator.
value	The value of the minimized function at theta.

References

Einmahl, J.H.J., Kiriliouk, A., and Segers, J. (2018). A continuous updating weighted least squares estimator of tail dependence in high dimensions. Extremes 21(2), 205-233.

Einmahl, J.H.J., Kiriliouk, A., Krajina, A., and Segers, J. (2016). An Mestimator of spatial tail dependence. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 78(1), 275-298.

See Also

selectGrid

Examples

```
## Generate data with theta = 0.5
set.seed(1)
n <- 1000
cop <- copula::gumbelCopula(param = 2, dim = 3)
data <- copula::rCopula(n = n,copula = cop)
## Transform data to unit Pareto margins
x <- apply(data, 2, function(i) n/(n + 0.5 - rank(i)))
## Define indices in which we evaluate the estimator
indices <- selectGrid(c(0,1), d = 3)
EstimationGumbel(x, indices, k = 50, method = "WLS", biascorr = TRUE)
```

EstimationMaxLinear *Estimation of the parameters of the max-linear model*

Description

Estimation the parameters of the max-linear model, using either the pairwise M-estimator or weighted least squares (WLS).

Usage

```
EstimationMaxLinear(
  x,
  indices,
  k,
```

```
    method,
    Bmatrix = NULL,
    Ldot = NULL,
    biascorr = FALSE,
    k1 = (nrow(x) - 10),
    tau = 5,
    startingValue,
    Omega = diag(nrow(indices)),
    iterate = FALSE,
    covMat = TRUE,
    GoFtest = FALSE,
    dist = 0.01,
    EURO = FALSE
)
```

Arguments

x	An n x d data matrix.
indices	A q x d matrix containing at least 2 non-zero elements per row, representing the values in which we will evaluate the stable tail dependence function.
k	An integer between 1 and $n - 1$; the threshold parameter in the definition of the empirical stable tail dependence function.
method	Choose between `Mestimator` and `WLS`.
Bmatrix	A function that converts the parameter vector theta to a parameter matrix B. If nothing is provided, then a simple 2-factor model is assumed.
Ldot	For `method = "WLS"` only. A q x p matrix, where p is the number of parameters of the model. Represents the total derivative of the function L defined in Einmahl et al. (2016). If nothing is provided, then a simple 2-factor model is assumed.
biascorr	For `method = "WLS"` only. If `TRUE`, then the bias-corrected estimator of the stable tail dependence function is used. Defaults to `FALSE`.
k1	For `biascorr = TRUE` only. The value of k_1 in the definition of the bias-corrected estimator of the stable tail dependence function.
tau	For `biascorr = TRUE` only. The parameter of the power kernel.
startingValue	Initial value of the parameters in the minimization routine.
Omega	A q x q matrix specifying the metric with which the distance between the parametric and nonparametric estimates will be computed. The default is the identity matrix, i.e., the Euclidean metric.
iterate	A Boolean variable. For `method = "WLS"` only. If `TRUE`, then continuous updating is used. Defaults to `FALSE`.
covMat	A Boolean variable. For `method = "WLS"` only. If `TRUE` (the default), the covariance matrix is calculated.
GoFtest	A Boolean variable. For `method = "WLS"` only. If `TRUE`, then the goodness-of-fit test of Corollary 2.6 from Einmahl et al. (2016) is performed. Defaults to `FALSE`.

dist	A positive scalar. If GoFtest = TRUE, only eigenvalues ν larger than dist are used; see Corollary 2.6 (Einmahl et al., 2016). Defaults to 0.01.
EURO	A Boolean variable. If TRUE, then the model from Einmahl et al. (2016, Section 4) is assumed, and the corresponding Bmatrix and Ldot are used.

Details

The matrix indices can be either user defined or returned by selectGrid. For method = "Mestimator", only a grid with exactly two ones per row is accepted, representing the pairs to be used. The functions Bmatrix and Ldot can be defined such that they represent a max-linear model on a directed acyclic graph: see the vignette for this package for an example.

Value

For Mestimator, the estimator theta is returned. For WLS, a list with the following components:

theta	The estimator with estimated optimal weight matrix.
theta_pilot	The estimator without the optimal weight matrix.
covMatrix	The estimated covariance matrix for the estimator.
value	The value of the minimized function at theta.
GoFresult	A list of length two, returning the value of the test statistic and s.

References

Einmahl, J.H.J., Kiriliouk, A., and Segers, J. (2018). A continuous updating weighted least squares estimator of tail dependence in high dimensions. Extremes 21(2), 205-233.

See Also

selectGrid

Examples

```
## Generate data
set.seed(1)
n <- 1000
fr <- matrix(-1/log(runif(2*n)), nrow = n, ncol = 2)
data <- cbind(pmax(0.3*fr[,1],0.7*fr[,2]),pmax(0.5*fr[,1],0.5*fr[,2]),pmax(0.9*fr[,1],0.1*fr[,2]))
## Transform data to unit Pareto margins
x <- apply(data, 2, function(i) n/(n + 0.5 - rank(i)))
## Define indices in which we evaluate the estimator
indices <- selectGrid(cst = c(0,0.5,1), d = 3)
EstimationMaxLinear(x, indices, k = 100, method = "WLS", startingValue = c(0.3,0.5,0.9))
indices <- selectGrid(cst = c(0,1), d = 3)
EstimationMaxLinear(x, indices, k = 100, method = "Mestimator", startingValue = c(0.3,0.5,0.9))
```

selectGrid — *Selects a grid of indices.*

Description

Returns a regular grid of indices in which to evaluate the stable tail dependence function.

Usage

```
selectGrid(cst, d, nonzero = 2, locations = NULL, maxDistance = 10^6)
```

Arguments

`cst`	A vector containing the values used to construct the grid. Must contain 0.
`d`	An integer, representing the dimension.
`nonzero`	An vector containing integers between 2 and d, representing the number of non-zero elements in every row of the grid. Defaults to 2.
`locations`	A `d` x 2 matrix containing the Cartesian coordinates of d points in the plane. Used for the Brown-Resnick process only. If not `NULL`, then `cst` must be `c(0,1)` and `nonzero` must be `2`.
`maxDistance`	If `locations` is not `NULL`, pairs of locations with distance not larger than `maxDistance` will be selected.

Value

A matrix with `q` rows and `d` columns, where every row represents a vector in which we will evaluate the stable tail dependence function (for the weighted least squares estimator) or where every row indicates which pairs of variables to use (for the M-estimator)

Examples

```
selectGrid(cst = c(0,0.5,1), d = 3, nonzero = c(2,3))
locations <- cbind(rep(1:3, each = 3), rep(1:3,3))
selectGrid(c(0,1), d = 9, locations = locations, maxDistance = 1.5)
```

stdfEmp — *Empirical stable tail dependence function*

Description

Returns the stable tail dependence function in dimension `d`, evaluated in a point `cst`.

Usage

```
stdfEmp(ranks, k, cst = rep(1, ncol(ranks)))
```

Arguments

`ranks`	A `n x d` matrix, where each column is a permutation of the integers `1:n`, representing the ranks computed from a sample of size `n`.
`k`	An integer between 1 and $n - 1$; the threshold parameter in the definition of the empirical stable tail dependence function.
`cst`	The value in which the tail dependence function is evaluated: defaults to `rep(1,d)`, i.e., the extremal coefficient.

Value

A scalar between $\max(x_1, \ldots, x_d)$ and $x_1 + \cdots + x_d$.

References

Einmahl, J.H.J., Kiriliouk, A., and Segers, J. (2018). A continuous updating weighted least squares estimator of tail dependence in high dimensions. Extremes 21(2), 205-233.

See Also

`stdfEmpCorr`

Examples

```
## Simulate data from the Gumbel copula and compute the extremal coefficient in dimension four.
set.seed(2)
cop <- copula::gumbelCopula(param = 2, dim = 4)
data <- copula::rCopula(n = 1000, copula = cop)
stdfEmp(apply(data,2,rank), k = 50)
```

`stdfEmpCorr` *Bias-corrected empirical stable tail dependence function*

Description

Returns the bias-corrected stable tail dependence function in dimension `d`, evaluated in a point `cst`.

Usage

```
stdfEmpCorr(
  ranks,
  k,
  cst = rep(1, ncol(ranks)),
  tau = 5,
  k1 = (nrow(ranks) - 10)
)
```

Arguments

ranks	A n x d matrix, where each column is a permutation of the integers 1:n, representing the ranks computed from a sample of size n.
k	An integer between 1 and $n - 1$; the threshold parameter in the definition of the empirical stable tail dependence function.
cst	The value in which the tail dependence function is evaluated: defaults to rep(1,d).
tau	The parameter of the power kernel. Defaults to 5.
k1	An integer between 1 and n; defaults to n - 10.

Details

The values for k1 and tau are chosen as recommended in Beirlant et al. (2016). This function might be slow for large n.

Value

A scalar between $\max(x_1, \ldots, x_d)$ and $x_1 + \cdots + x_d$.

References

Einmahl, J.H.J., Kiriliouk, A., and Segers, J. (2018). A continuous updating weighted least squares estimator of tail dependence in high dimensions. Extremes 21(2), 205-233.

Beirlant, J., Escobar-Bach, M., Goegebeur, Y., and Guillou, A. (2016). Bias-corrected estimation of stable tail dependence function. Journal of Multivariate Analysis, 143, 453-466.

See Also

stdfEmp

Examples

```
## Simulate data from the Gumbel copula
set.seed(2)
cop <- copula::gumbelCopula(param = 2, dim = 4)
data <- copula::rCopula(n = 1000, copula = cop)
stdfEmpCorr(apply(data,2,rank), k = 50)
```

---

stdfEmpInt *Integrated empirical stable tail dependence function*

Description

Analytical implementation of the integral of the bivariate stable tail dependence function over the unit square.

Usage

```
stdfEmpInt(ranks, k)
```

Arguments

`ranks`	A n x 2 matrix, where each column is a permutation of the integers `1:n`, representing the ranks computed from a sample of size `n`.
`k`	An integer between 1 and $n-1$; the threshold parameter in the definition of the empirical stable tail dependence function.#' @return A scalar.

Value

A positive scalar.

References

Einmahl, J.H.J., Kiriliouk, A., Krajina, A., and Segers, J. (2016). An Mestimator of spatial tail dependence. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 78(1), 275-298.

Examples

```
ranks <- cbind(sample(1:20), sample(1:20))
stdfEmpInt(ranks, k = 5)
```

`tailDepFun` *tailDepFun*

Description

The package `tailDepFun` provides functions implementing two rank-based minimal distance estimation methods for parametric tail dependence models for distributions attracted to a max-stable law. The estimators, referred to as the pairwise M-estimator and the weighted least squares estimator, are described in Einmahl et al. (2016a) and Einmahl et al. (2016b). Extensive examples to illustrate the use of the package can be found in the accompanying vignette.

Details

Currently, this package allows for estimation of the Brown-Resnick process, the Gumbel (or logistic) model and max-linear models (possibly on a directed acyclic graph). The main functions of this package are `EstimationBR`, `EstimationGumbel` and `EstimationMaxLinear`, but several other functions are exported as well: `stdfEmpInt` returns the integral of the bivariate empirical stable tail dependence function over the unit square, and `stdfEmp` and `stdfEmpCorr` return the (bias-corrected) empirical stable tail dependence function. The functions `AsymVarBR`, `AsymVarGumbel`, `AsymVarMaxLinear` return the asymptotic covariance matrices of the estimators. An auxiliary function to select a regular grid of indices in which to evaluate the stable tail dependence function is exported as well, `selectGrid`. Finally, two datasets are available: `dataKNMI` (Einmahl et al., 2016) and `dataEUROSTOXX` (Einmahl et al., 2018).

References

Einmahl, J.H.J., Kiriliouk, A., Krajina, A., and Segers, J. (2016). An Mestimator of spatial tail dependence. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 78(1), 275-298.

Einmahl, J.H.J., Kiriliouk, A., and Segers, J. (2018). A continuous updating weighted least squares estimator of tail dependence in high dimensions. Extremes 21(2), 205-233.

Examples

```
## get a list of all help files of user-visible functions in the package
help(package = tailDepFun)
```

Index