Package ‘graphicalExtremes’

December 2, 2022

Title Statistical Methodology for Graphical Extreme Value Models

Version 0.2.0

Maintainer Sebastian Engelke <sebastian.engelke@unige.ch>

Description Statistical methodology for sparse multivariate extreme value models. Methods are provided for exact simulation and statistical inference for multivariate Pareto distributions on graphical structures as described in the paper 'Graphical Models for Extremes' by Engelke and Hitz (2020) <doi:10.1111/rssb.12355>.

Depends R (>= 3.6.0)

License GPL-3

Encoding UTF-8

LazyData true

LazyDataCompression bzip2

Suggests testthat (>= 2.1.0), knitr, rmarkdown, dplyr, ggplot2, bookdown, maps

RoxygenNote 7.2.2

Imports igraph (>= 1.2.4.1), mvtnorm (>= 1.0.10), Rdpack, stats (>= 3.6.0), utils, corpcor, osqp, glmnet, glassoFast, edmcr

URL https://github.com/sebastian-engelke/graphicalExtremes

BugReports https://github.com/sebastian-engelke/graphicalExtremes/issues

RdMacros Rdpack

VignetteBuilder knitr

NeedsCompilation no

Author Sebastian Engelke [aut, cre],
Adrien S. Hitz [aut],
Nicola Gnecco [aut],
Manuel Hentschel [aut]

Repository CRAN

Date/Publication 2022-12-02 12:20:04 UTC

R topics documented:

`check_Gamma_and_graph` *Check input graph and Gamma matrix*

Description

Checks and converts the (incomplete) Gamma matrix and graph given for a HR graphical model.

Usage

```
check_Gamma_and_graph(Gamma, graph = NULL, graph_type = "general")
```

Arguments

`Gamma`	A Gamma matrix or vector of entries corresponding to the edges of `graph`
`graph`	A graph object or `NULL` if the graph structure is specified by `NA` in the Gamma matrix
`graph_type`	Passed to `check_graph()`.

Value

A list consisting of

`Gamma`	The Gamma matrix given as input or implied by the input
`graph`	The graph given as input or implied by the input

Throws an error if the input is not valid.

See Also

Other Input checks: `check_graph()`

check_graph — *Check input graph*

Description

Checks that the input graph is a valid graph for an extremal graphical model. If necessary, converts the graph to an undirected graph. Removes vertex labels if present.

Usage

```
check_graph(
  graph,
  graph_type = c("general", "decomposable", "block", "tree"),
  check_connected = TRUE,
  nVertices = NULL
)
```

Arguments

graph	An [igraph::graph] object.
graph_type	"general", "decomposable", "block", "tree". The required type of graph.
check_connected	Whether to check if the graph is connected.
nVertices	The number of vertices required in the graph.

Value

The given graph, if necessary converted to undirected. If the graph is not valid an error is thrown.

See Also

Other Input checks: check_Gamma_and_graph()

chi2Gamma — *Transformation of extremal correlation χ to the Huesler–Reiss variogram Γ*

Description

Transforms the extremal correlation χ into the Gamma matrix from the definition of a Huesler–Reiss distribution.

Usage

```
chi2Gamma(chi)
```

Arguments

chi	Numeric or matrix, with entries between 0 and 1.

Details

The formula for transformation from chi to Γ that is applied element-wise is

$$\Gamma = (2\Phi^{-1}(1 - 0.5\chi))^2,$$

where Φ^{-1} is the inverse of the standard normal distribution function. This is the inverse of Gamma2chi().

Value

Numeric or matrix. The Γ parameters in the Huesler–Reiss distribution.

---

complete_Gamma *Completion of Gamma matrices*

Description

Given a graph and a (partial) variogram matrix Gamma, returns a full variogram matrix that agrees with Gamma in entries corresponding to edges of graph and whose corresponding precision matrix, obtained by Gamma2Theta(), has zeros in entries corresponding to non-edges of graph. For results on the existence and uniqueness of this completion, see Hentschel et al. (2022).

Usage

```
complete_Gamma(Gamma, graph = NULL, ...)
```

Arguments

Gamma	Numeric $d \times d$ variogram matrix.
graph	NULL or igraph::graph object. If NULL, the graph is implied by non-edge entries in Gamma being NA. Must be connected, undirected.
...	Further arguments passed to complete_Gamma_general() if graph is not decomposable

Details

If graph is decomposable, Gamma only needs to be specified on the edges of the graph, other entries are ignored. If graph is not decomposable, the graphical completion algorithm requires a fully specified (but non-graphical) variogram matrix Gamma to begin with. If necessary, this initial completion is computed using edmcr::npf().

Value

Completed $d \times d$ Gamma matrix.

References

Hentschel M, Engelke S, Segers J (2022). "Statistical Inference for Hüsler-Reiss Graphical Models Through Matrix Completions." doi:10.48550/ARXIV.2210.14292, https://arxiv.org/abs/2210.14292.

See Also

Gamma2Theta()

Other Matrix completions: complete_Gamma_decomposable(), complete_Gamma_general_demo(), complete_Gamma_general_split(), complete_Gamma_general()

Examples

```
## Block graph:
Gamma <- rbind(
  c(0, .5, NA, NA),
  c(.5, 0, 1, 1.5),
  c(NA, 1, 0, .8),
  c(NA, 1.5, .8, 0)
)

complete_Gamma(Gamma)

## Alternative representation of the same completion problem:
my_graph <- igraph::graph_from_adjacency_matrix(rbind(
  c(0, 1, 0, 0),
  c(1, 0, 1, 1),
  c(0, 1, 0, 1),
  c(0, 1, 1, 0)
), mode = "undirected")
Gamma_vec <- c(.5, 1, 1.5, .8)
complete_Gamma(Gamma_vec, my_graph)

## Decomposable graph:
G <- rbind(
c(0, 5, 7, 6, NA),
c(5, 0, 14, 15, NA),
c(7, 14, 0, 5, 5),
c(6, 15, 5, 0, 6),
c(NA, NA, 5, 6, 0)
)

complete_Gamma(G)

## Non-decomposable graph:
G <- rbind(
c(0, 5, 7, 6, 6),
c(5, 0, 14, 15, 13),
c(7, 14, 0, 5, 5),
c(6, 15, 5, 0, 6),
c(6, 13, 5, 6, 0)
```

```
)
g <- igraph::make_ring(5)

complete_Gamma(G, g)
```

---

complete_Gamma_decomposable *Completion of decomposable Gamma matrices*

Description

Given a decomposable graph and incomplete variogram matrix Gamma, returns the full Gamma matrix implied by the conditional independencies.

Usage

```
complete_Gamma_decomposable(Gamma, graph = NULL)
```

Arguments

Gamma	A variogram matrix that is specified on the edges of graph and the diagonals. All other entries are ignored (if graph is specified), or should be NA to indicate non-edges in graph.
graph	NULL or a decomposable [igraph::graph] object. If NULL, the structure of NA entries in Gamma is used instead.

Value

A complete variogram matrix that agrees with Gamma on the entries corresponding to edges in graph and the diagonals. The corresponding Θ matrix produced by Gamma2Theta() has zeros in the remaining entries.

See Also

Other Matrix completions: complete_Gamma_general_demo(), complete_Gamma_general_split(), complete_Gamma_general(), complete_Gamma()

complete_Gamma_general *Non-decomposable completion of variogram matrices*

Description

Given a non-decomposable graph, and (non-graphical) variogram matrix Gamma, modifies Gamma in non-edge entries, such that the resulting matrix is a variogram matrix with graphical structure described by graph.

Usage

```
complete_Gamma_general(
  Gamma,
  graph,
  N = 10000,
  tol = get_large_tol(),
  check_tol = 100
)
```

Arguments

Gamma	Numeric $d \times d$ variogram matrix.
graph	igraph::graph() object.
N	Maximum number of iterations.
tol	Numeric scalar. Tolerance to be used when completing submatrices.
check_tol	Numeric/integer scalar. How often to check the tolerance when completing submatrices.

Value

A completed $d \times d$ variogram matrix.

See Also

Other Matrix completions: complete_Gamma_decomposable(), complete_Gamma_general_demo(), complete_Gamma_general_split(), complete_Gamma()

complete_Gamma_general_demo

DEMO-VERSION: Completion of non-decomposable Gamma matrices

Description

Given a graph and variogram matrix Gamma, returns the full Gamma matrix implied by the conditional independencies. DEMO VERSION: Returns a lot of details and allows specifying the graph list that is used. Is way slower than other functions.

Usage

```
complete_Gamma_general_demo(Gamma, graph, N = 1000, tol = 0, gList = NULL)
```

Arguments

Gamma	A complete variogram matrix (without any graphical structure).
graph	An igraph::graph object.
N	The maximal number of iterations of the algorithm.
tol	The tolerance to use when checking for zero entries in Theta.
gList	A list of graphs to be used instead of the output from make_sep_list().

Value

A nested list, containing the following details. The "error term" is the maximal absolute value of Theta in a non-edge entry.

graph, N, tol: As in the input

gList: As in the input or computed by make_sep_list().

Gamma0, Theta0, err0: Initial Gamma, Theta, and error term.

iterations: A nested list, containing the following infos for each performed iteration:

- n Number of the iteration
- t Corresponding index in gList
- g The graph used
- Gamma, Theta, err The value of Gamma, Theta, and error term after the iteration

See Also

Other Matrix completions: complete_Gamma_decomposable(), complete_Gamma_general_split(), complete_Gamma_general(), complete_Gamma()

complete_Gamma_general_split *Non-decomposable completion of variogram matrices*

Description

Given a non-decomposable graph, and (non-graphical) variogram matrix Gamma, modifies Gamma in non-edge entries, such that the resulting matrix is a variogram matrix with graphical structure described by graph. Does so by splitting graph at complete separators into smaller subgraphs, and calling complete_Gamma_general for each subgraph/submatrix, using multiple cores if available.

Usage

```
complete_Gamma_general_split(
  Gamma,
  graph,
  N = 10000,
  sub_tol = get_large_tol() * 0.001,
  check_tol = 100,
  mc_cores_overwrite = NULL,
  final_tol = get_large_tol()
)
```

Arguments

Gamma	Numeric $d \times d$ variogram matrix.
graph	igraph::graph() object.
N	Maximum number of iterations.
sub_tol	Numeric scalar. Tolerance to be used when completing submatrices. Should be smaller than final_tol.
check_tol	Numeric/integer scalar. How often to check the tolerance when completing submatrices.
mc_cores_overwrite	NULL or numeric/integer scalar. Maximal number of cores to use.
final_tol	Numeric scalar. Check convergence of the final result with this tolerance. Skipped if this value is < 0.

Value

A completed $d \times d$ variogram matrix.

See Also

Other Matrix completions: complete_Gamma_decomposable(), complete_Gamma_general_demo(), complete_Gamma_general(), complete_Gamma()

danube *Upper Danube basin dataset*

Description

A dataset containing river discharge data for tributaries of the Danube.

Usage

```
danube
```

Format

A named `list` with four entries

- `data_clustered` A numeric matrix, containing pre-processed discharge data for each gauging station
- `data_raw` A numeric matrix, containing daily (raw) discharge data for each gauging station
- `info` A data frame, containing information about each gauging station
- `flow_edges` A two-column numeric matrix. Each row contains the indices (in `info`) of a pair of gauging stations that are directly connected by a river.

Details

To obtain the matrix `data`, the daily discharge data from the summer months of 1960 to 2010, given in `dailyData`, was declustered, yielding between seven and ten observations per year. Each row corresponds to one observation from this declustered time series, the non-unique rownames indicate which year an observation is from. Each column corresponds to one of the gauging stations, with column indices in `data` corresponding to row indices in `info`. See (Asadi et al. 2015) for details on the preprocessing and declustering.

`info` is a data frame containing the following information for each of the gauging stations or its corresponding catchment area:

- `RivNames` Name of the river at the gauging station
- `Lat`, `Long` Coordinates of the gauging station
- `Lat_Center`, `Long_Center` Coordinates of the center of the catchment corresponding to the gauging station
- `Alt` Mean altitude of the catchment
- `Area` Area of the catchment corresponding to the gauging station
- `Slope` Mean slope of the catchment
- `PlotCoordX`, `PlotCoordY` X-Y-coordinates which can be used to arrange the gauging stations when plotting a flow graph.

Source

Bavarian Environmental Agency https://www.gkd.bayern.de.

References

Asadi P, Davison AC, Engelke S (2015). "Extremes on river networks." *The Annals of Applied Statistics*, **9**(4), 2023 – 2050. doi:10.1214/15AOAS863.

See Also

flights, `vignette('graphicalExtremes')`

Examples

```
g <- igraph::graph_from_edgelist(danube$flow_edges)
loc <- as.matrix(danube$info[,c('PlotCoordX', 'PlotCoordY')])
plot(g, layout = loc)
```

---

data2mpareto — *Data standardization to multivariate Pareto scale*

Description

Transforms the `data` matrix empirically to the multivariate Pareto scale.

Usage

```
data2mpareto(data, p, na.rm = FALSE)
```

Arguments

`data`	Numeric $n \times d$ matrix, where `n` is the number of observations and `d` is the dimension.
`p`	Numeric between 0 and 1. Probability used for the quantile to threshold the data.
`na.rm`	Logical. If rows containing NAs should be removed.

Details

The columns of the `data` matrix are first transformed empirically to standard Pareto distributions. Then, only the observations where at least one component exceeds the `p`-quantile of the standard Pareto distribution are kept. Those observations are finally divided by the `p`-quantile of the standard Pareto distribution to standardize them to the multivariate Pareto scale.

If `na.rm` is `FALSE`, missing entries are left as such during the transformation of univariate marginals. In the thresholding step, missing values are considered as `-Inf`.

Value

Numeric $m \times d$ matrix, where `m` is the number of rows in the original `data` matrix that are above the threshold.

Examples

```
n <- 20
d <- 4
p <- .8
G <- cbind(
  c(0, 1.5, 1.5, 2),
  c(1.5, 0, 2, 1.5),
  c(1.5, 2, 0, 1.5),
  c(2, 1.5, 1.5, 0)
)

set.seed(123)
my_data <- rmstable(n, "HR", d = d, par = G)
data2mpareto(my_data, p)
```

---

eglearn — *Learning extremal graph structure*

Description

Fits an extremal graph structure using the neighborhood selection approach (see Meinshausen and Bühlmann (2006)) or graphical lasso (see Friedman et al. (2008)).

Usage

```
eglearn(
  data,
  p = NULL,
  rholist = c(0.1, 0.15, 0.19, 0.205),
  reg_method = c("ns", "glasso"),
  complete_Gamma = FALSE
)
```

Arguments

data	Numeric $n \times d$ matrix, where n is the number of observations and d is the dimension.
p	Numeric between 0 and 1 or NULL. If NULL (default), it is assumed that the data are already on multivariate Pareto scale. Else, p is used as the probability in the function data2mpareto() to standardize the data.
rholist	Numeric vector of non-negative regularization parameters for the lasso. Default is rholist = c(0.1, 0.15, 0.19, 0.205). For details see glasso::glassopath().
reg_method	One of "ns", "glasso", for neighborhood selection and graphical lasso, respectively. Default is reg_method = "ns". For details see Meinshausen and Bühlmann (2006), Friedman et al. (2008).
complete_Gamma	Whether you want to try fto complete Gamma matrix. Default is complete_Gamma = FALSE.

Value

List made of:

graph
: A list of [igraph::graph] objects representing the fitted graphs for each rho in rholist.

Gamma
: A list of numeric estimated $d \times d$ variogram matrices Γ corresponding to the fitted graphs, for each rho in rholist. If complete_Gamma = FALSE or the underlying graph is not connected, it returns NULL.

rholist
: The list of penalty coefficients.

graph_ic
: A list of [igraph::graph] objects representing the optimal graph according to the aic, bic, and mbic information criteria. If reg_method = "glasso", it returns a list of NA.

Gamma_ic
: A list of numeric $d \times d$ estimated variogram matrices Γ corresponding to the aic, bic, and mbic information criteria. If reg_method = "glasso", complete_Gamma = FALSE, or the underlying graph is not connected, it returns a list of NA.

---

emp_chi — *Empirical estimation of extremal correlation matrix χ*

Description

Estimates empirically the matrix of bivariate extremal correlation coefficients χ.

Usage

```
emp_chi(data, p = NULL)

emp_chi_pairwise(data, p = NULL, verbose = FALSE)
```

Arguments

data
: Numeric $n \times d$ matrix, where n is the number of observations and d is the dimension.

p
: Numeric scalar between 0 and 1 or NULL. If NULL (default), it is assumed that the data are already on multivariate Pareto scale. Else, p is used as the probability in data2mpareto() to standardize the data.

verbose
: Print verbose progress information

Details

emp_chi_pairwise calls emp_chi for each pair of observations. This is more robust if the data contains many NAs, but can take rather long.

Value

Numeric matrix $d \times d$. The matrix contains the bivariate extremal coefficients χ_{ij}, for $i, j = 1, ..., d$.

Examples

```
n <- 100
d <- 4
p <- .8
Gamma <- cbind(
  c(0, 1.5, 1.5, 2),
  c(1.5, 0, 2, 1.5),
  c(1.5, 2, 0, 1.5),
  c(2, 1.5, 1.5, 0)
)

set.seed(123)
my_data <- rmstable(n, "HR", d = d, par = Gamma)
emp_chi(my_data, p)
```

---

emp_chi_multdim *Empirical estimation of extremal correlation χ*

Description

Estimates the d-dimensional extremal correlation coefficient χ empirically.

Usage

```
emp_chi_multdim(data, p = NULL)
```

Arguments

data	Numeric $n \times d$ matrix, where n is the number of observations and d is the dimension.
p	Numeric scalar between 0 and 1 or NULL. If NULL (default), it is assumed that the data are already on multivariate Pareto scale. Else, p is used as the probability in data2mpareto() to standardize the data.

Value

Numeric scalar. The empirical d-dimensional extremal correlation coefficient χ for the data.

Examples

```
n <- 100
d <- 2
p <- .8
G <- cbind(
  c(0, 1.5),
  c(1.5, 0)
)
```

```
set.seed(123)
my_data <- rmstable(n, "HR", d = d, par = G)
emp_chi_multdim(my_data, p)
```

---

emp_vario *Estimation of the variogram matrix Γ of a Huesler–Reiss distribution*

Description

Estimates the variogram of the Huesler–Reiss distribution empirically.

Usage

```
emp_vario(data, k = NULL, p = NULL)

emp_vario_pairwise(data, k = NULL, p = NULL, verbose = FALSE)
```

Arguments

data	Numeric $n \times d$ matrix, where n is the number of observations and d is the dimension.
k	Integer between 1 and d. Component of the multivariate observations that is conditioned to be larger than the threshold p. If NULL (default), then an average over all k is returned.
p	Numeric between 0 and 1 or NULL. If NULL (default), it is assumed that the data are already on multivariate Pareto scale. Else, p is used as the probability in the function data2mpareto() to standardize the data.
verbose	Print verbose progress information

Details

emp_vario_pairwise calls emp_vario for each pair of observations. This is more robust if the data contains many NAs, but can take rather long.

Value

Numeric $d \times d$ matrix. The estimated variogram of the Huesler–Reiss distribution.

emst *Fitting extremal minimum spanning tree*

Description

Fits an extremal minimum spanning tree, where the edge weights are:

- negative maximized log-likelihoods of the bivariate Huesler–Reiss distributions, if `method = "ML"`. See Engelke and Hitz (2020) for details.
- empirical extremal variogram, if `method = "vario"`. See Engelke and Volgushev (2022) for details.
- empirical extremal correlation, if `method = "chi"`. See Engelke and Volgushev (2022) for details.

Usage

```
emst(data, p = NULL, method = c("vario", "ML", "chi"), cens = FALSE)
```

Arguments

data	Numeric $n \times d$ matrix, where n is the number of observations and d is the dimension.
p	Numeric between 0 and 1 or `NULL`. If `NULL` (default), it is assumed that the `data` are already on multivariate Pareto scale. Else, `p` is used as the probability in the function `data2mpareto()` to standardize the `data`.
method	One of `"vario", "ML", "chi"`. Default is `method = "vario"`.
cens	Logical. This argument is considered only if `method = "ML"`. If `TRUE`, then censored likelihood contributions are used for components below the threshold. By default, `cens = FALSE`.

Value

List consisting of:

graph	An `igraph::graph` object. The fitted minimum spanning tree.
Gamma	Numeric $d \times d$ estimated variogram matrix Γ corresponding to the fitted minimum spanning tree.

References

Engelke S, Hitz AS (2020). "Graphical models for extremes (with discussion)." *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **82**, 871–932.

Engelke S, Volgushev S (2022). "Structure learning for extremal tree models." *J. R. Stat. Soc. Ser. B Stat. Methodol.*. doi:10.1111/rssb.12556, Forthcoming, https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/rssb.12556.

Examples

```
## Fitting a 4-dimensional HR minimum spanning tree
my_graph <- igraph::graph_from_adjacency_matrix(
  rbind(
    c(0, 1, 0, 0),
    c(1, 0, 1, 1),
    c(0, 1, 0, 0),
    c(0, 1, 0, 0)
  ),
  mode = "undirected"
)
n <- 100
Gamma_vec <- c(.5, 1.4, .8)
complete_Gamma(Gamma = Gamma_vec, graph = my_graph) ## full Gamma matrix

set.seed(123)
my_data <- rmpareto_tree(n, "HR", tree = my_graph, par = Gamma_vec)
my_fit <- emst(my_data, p = NULL, method = "ML", cens = FALSE)
```

---

emtp2 — *Performs Gaussian likelihood optimization under Laplacian matrix constraints.*

Description

This function implements a block descent algorithm to find the maximum of the Gaussian log-likelihood under the constraint that the concentration matrix is a Laplacian matrix. See Röttger et al. (2021) for details.

Usage

```
emtp2(Gamma, tol = 1e-06, verbose = TRUE, initial_point = TRUE)
```

Arguments

`Gamma`	conditionally negative semidefinite matrix. This will be typically the empirical variogram matrix.
`tol`	The convergence tolerance. The algorithm terminates when the sum of absolute differences between two iterations is below `tol`.
`verbose`	if TRUE (default) the output will be printed.
`initial_point`	if TRUE (default), the algorithm will construct an initial point before the iteration steps.

Value

A list consisting of:

`G_emtp2`	The optimal value of the variogram matrix
`it`	The number of iterations

References

Röttger F, Engelke S, Zwiernik P (2021). “Total positivity in multivariate extremes.” doi:10.48550/ARXIV.2112.14727, https://arxiv.org/abs/2112.14727.

---

fast_diag — *Fast computation of diag(y %% M %% t(y))*

Description

Fast computation of diag(y *%% M %%* t(y))

Usage

```
fast_diag(y, M)
```

Arguments

y	Numeric matrix
M	Numeric matrix

Value

Numeric vector

---

fillFixedParams — *Helper function to combine par with fixed params (in init)*

Description

Helper function to combine par with fixed params (in init)

Usage

```
fillFixedParams(par, init, fixParams)
```

Arguments

par	Numeric vector. The parameters that are optimized
init	Numeric vector. The initial parameters (including the ones optimized over)
fixParams	Numeric or logical vector. Positions of fixed parameters in the full parameter vector.

`fitInInterval` *Fit value(s) in interval*

Description

Fit value(s) in interval, all arguments are recycled where necessary.

Usage

```
fitInInterval(x, xMin = -Inf, xMax = Inf)
```

Arguments

`x`	Numeric vector
`xMin`	Numeric vector
`xMax`	Numeric vector

Value

Numeric vector

`flightCountMatrixToConnectionList` *Convert flight counts to connection list*

Description

Convert a numeric matrix containing flight counts between ariports to a data frame containing a list of connections.

Usage

```
flightCountMatrixToConnectionList(nFlightsPerConnection, directed = TRUE)
```

Arguments

`nFlightsPerConnection`	A square, numeric matrix with identical column- and row-names. Each entry represents the number of flights from the airport indexing the row to the airport indexing the column in some arbitrary time period.
`directed`	Logical scalar. Whether flights A->B and B->A should be considered separately.

Value

A data frame with columns `departureAirport`, `arrivalAirport`, `nFlights`. Each row represents one connection with >=1 flights in the input matrix.

flights *Flights delay data*

Description

A dataset containing daily total delays of major U.S. airlines. The raw data was obtained from the U.S. Bureau of Transportation Statistics, and pre-processed as described in Hentschel et al. (2022). *Note: The CRAN version of this package contains only data from 2010-2013. The full dataset is available in the Github version of this package.*

Usage

```
flights
```

Format

A named `list` with three entries:

`airports` A `data.frame`, containing information about US airports

`delays` A numeric matrix, containing daily aggregated delays at the airports in the dataset

`flightCounts` A numeric array, containing yearly flight numbers between airports in the dataset

Details

`flightCounts` is a three-dimensional array, containing the number of flights in the dataset between each pair of airports, aggregated on a yearly basis. Each entry is the total number of flights between the departure airport (row) and destination airport (column) in a given year (dimension 3). This array does not contain any `NA`s, even if an airport did not operate at all in a given year, which is simply indicated by zeros.

`delays` is a three-dimensional array containing daily total positive delays, in minutes, of incoming and outgoing flights respectively. Each column corresponds to an airport in the dataset and each row corresponds to a day. The third dimension has length two, `'arrivals'` containing delays of incoming flights and `'departures'` containing delays of outgoing flights. Zeros indicate that there were flights arriving/departing at that airport on a given day, but none of them had delays. `NA`s indicate that there were no flights arriving/departing at that airport on that day at all.

`airports` is a data frame containing the following information about a number of US airports. Some entries are missing, which is indicated by `NA`s.

`IATA` 3-letter IATA code

`Name` name of the airport

`City` main city served by the airport

`Country` country or territory where the airport is located (mostly `"United States"`)

`ICAO` 4-letter ICAO code

`Latitude` latitude of the airport, in decimal degrees

`Longitude` longitude of the airport, in decimal degrees

`Altitude` altitude of the airport, in feet

`Timezone` timezone of the airport, in hours offset from UTC

`DST` Daylight savings time used at the airport. 'A'=US/Canada, 'N'=None.

`Timezone2` name of the timezone of the airport

Source

Raw delays data:

- https://www.bts.dot.gov/browse-statistical-products-and-data/bts-publications/airline-service-quality-performance-234-time

Fields/Forms used in the raw data:

- https://esubmit.rita.dot.gov/ViewReports.aspx
- https://esubmit.rita.dot.gov/On-Time-Form1.aspx
- https://esubmit.rita.dot.gov/On-Time-Form3A.aspx

Airports (includes license information):

- https://openflights.org/data.html

References

Hentschel M, Engelke S, Segers J (2022). "Statistical Inference for Hüsler-Reiss Graphical Models Through Matrix Completions." doi:10.48550/ARXIV.2210.14292, https://arxiv.org/abs/2210.14292.

See Also

danube, flightCountMatrixToConnectionList, plotFlights

Examples

```
## Not run:

# Get total number of flights in the dataset:
totalFlightCounts <- apply(flights$flightCounts, c(1,2), sum)

# Get number of flights for specific years in the dataset:
flightCounts_08_09 <- apply(flights$flightCounts[,,c('2008', '2009')], c(1,2), sum)

# Get list of connections:
connections <- flightCountMatrixToConnectionList(flights$flightCounts)
connections_08 <- flightCountMatrixToConnectionList(flights$flightCounts[,,'2008'])

# Get total delays (arriving + departing):
totalDelays <- apply(flights$delays, c(1,2), sum)

## End(Not run)
```

fmpareto_graph_HR *Parameter fitting for Huesler–Reiss graphical models*

Description

Fits the parameter matrix (variogram) of a multivariate Huesler–Reiss Pareto distirubtion with a given graphical structure, using maximum-likelihood estimation or the empirical variogram.

Usage

```
fmpareto_graph_HR(
  data,
  graph,
  p = NULL,
  method = c("vario", "ML"),
  handleCliques = c("average", "full", "sequential"),
  ...
)
```

Arguments

data	Numeric $n \times d$ matrix, where n is the number of observations and d is the number of dimensions.
graph	Undirected, connected [igraph::graph] object with d vertices, representing the graphical structure of the fitted Huesler–Reiss model.
p	Numeric between 0 and 1 or NULL. If NULL (default), it is assumed that the data is already on a multivariate Pareto scale. Else, p is used as the probability in the function data2mpareto() to standardize the data.
method	One of c('vario', 'ML'), with 'vario' as default, indicating the method to be used for parameter estimation. See details.
handleCliques	How to handle cliques and separators in the graph. See details.
...	Arguments passed to fmpareto_HR_MLE(). Currently cens, maxit, optMethod, and useTheta are supported.

Details

If handleCliques='average', the marginal parameter matrix is estimated for each maximal clique of the graph and then combined into a partial parameter matrix by taking the average of entries from overlapping cliques. Lastly, the full parameter matrix is computed using complete_Gamma().

If handleCliques='full', first the full parameter matrix is estimated using the specified method and then the non-edge entries are adjusted such that the final parameter matrix has the graphical structure indicated by graph.

If handleCliques='sequential', graph must be decomposable, and method='ML' must be specified. The parameter matrix is first estimated on the (recursive) separators and then on the rest of the cliques, keeping previously estimated entries fixed.

If `method='ML'`, the computational cost is mostly influenced by the total size of the graph (if `handleCliques='full'`) or the size of the cliques, and can already take a significant amount of time for modest dimensions (e.g. `d=3`).

Value

The estiamted parameter matrix.

`Gamma2chi` *Transformation of the Huesler–Reiss variogram Γ to extremal correlation χ*

Description

Transforms the `Gamma` matrix from the definition of a Huesler–Reiss distribution into the corresponding extremal correlation χ.

Usage

```
Gamma2chi(Gamma)
```

Arguments

`Gamma`	Numeric or matrix, with positive entries.

Details

The formula for transformation from `Gamma` to χ that is applied element-wise is

$$\chi = 2 - 2\Phi(\sqrt{\Gamma}/2),$$

where Φ is the standard normal distribution function. This is the inverse of `chi2Gamma()`.

Value

Numeric or matrix. The extremal correlation coefficient.

Gamma2graph *Transformation of matrix to graph object*

Description

Transforms a Γ or Θ matrix to an igraph::graph object for the corresponding Huesler–Reiss extremal graphical model.

Usage

```
Gamma2graph(Gamma, tol = get_large_tol())

Theta2graph(Theta, tol = get_large_tol())
```

Arguments

Gamma	Numeric $d \times d$ variogram matrix.
tol	Numeric scalar, entries in the precision matrix with absolute value smaller than this are considered to be zero.
Theta	Numeric $d \times d$ precision matrix.

Value

Graph object from igraph package. An undirected graph.

See Also

get_large_tol()

Other MatrixTransformations: Gamma2Sigma(), Gamma2Theta(), Sigma2Gamma(), Theta2Gamma()

Examples

```
Gamma <- cbind(
  c(0, 1.5, 1.5, 2),
  c(1.5, 0, 2, 1.5),
  c(1.5, 2, 0, 1.5),
  c(2, 1.5, 1.5, 0)
)

Gamma2graph(Gamma)
```

`Gamma2Sigma` *Transformation of Γ matrix to Σ or Σ^k matrix*

Description

Transforms the `Gamma` matrix from the definition of a Huesler–Reiss distribution to the corresponding Σ or Σ^k matrix.

Usage

```
Gamma2Sigma(Gamma, k = NULL, full = FALSE)
```

Arguments

`Gamma`	Numeric $d \times d$ variogram matrix.
`k`	`NULL` (default) or an integer between `1` and `d`. Indicates which matrix Σ, or Σ^k should be produced.
`full`	Logical. If true, then the kth row and column in Σ^k are included and the function returns a $d \times d$ matrix. By default, `full = FALSE`.

Details

Every $d \times d$ `Gamma` matrix in the definition of a Huesler–Reiss distribution can be transformed into a $(d-1) \times (d-1)$ Σ^k matrix, for any `k` from `1` to `d`. The inverse of Σ^k contains the graph structure corresponding to the Huesler–Reiss distribution with parameter matrix `Gamma`. If `full = TRUE`, then Σ^k is returned as a $d \times d$ matrix with additional kth row and column that contain zeros. For details see Engelke and Hitz (2020) and Hentschel et al. (2022).

Value

Numeric Σ^k matrix of size $(d-1) \times (d-1)$ if `full = FALSE`, and Σ of size $d \times d$ if `full = TRUE`.

References

Engelke S, Hitz AS (2020). "Graphical models for extremes (with discussion)." *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **82**, 871–932.

Hentschel M, Engelke S, Segers J (2022). "Statistical Inference for Hüsler-Reiss Graphical Models Through Matrix Completions." doi:10.48550/ARXIV.2210.14292, `https://arxiv.org/abs/2210.14292`.

See Also

Other MatrixTransformations: `Gamma2Theta()`, `Gamma2graph()`, `Sigma2Gamma()`, `Theta2Gamma()`

Examples

```
Gamma <- cbind(
  c(0, 1.5, 1.5, 2),
  c(1.5, 0, 2, 1.5),
  c(1.5, 2, 0, 1.5),
  c(2, 1.5, 1.5, 0)
)
Gamma2Sigma(Gamma, k = 1, full = FALSE)
```

`Gamma2Theta` *Transformation of Γ matrix to Θ matrix*

Description

Transforms the variogram matrix (Γ) from the definition of a Huesler–Reiss distribution to the corresponding precision matrix (Θ or Θ^k).

Usage

```
Gamma2Theta(Gamma, k = NULL)
```

Arguments

`Gamma`	Numeric $d \times d$ variogram matrix.
`k`	`NULL` or integer between 1 and d. If this is `NULL`, the $d \times d$ matrix Θ is produced, otherwise the specified $(d-1) \times (d-1)$ matrix Θ^k.

Details

Every $d \times d$ `Gamma` matrix in the definition of a Huesler–Reiss distribution can be transformed into a $d \times d$ Θ matrix, which contains the graph structure corresponding to the Huesler–Reiss distribution with parameter matrix `Gamma`.

Value

Numeric Σ^k matrix of size $(d-1) \times (d-1)$ if `full = FALSE`, and of size $d \times d$ if `full = TRUE`.

References

There are no references for Rd macro `\insertAllCites` on this help page.

See Also

Other MatrixTransformations: `Gamma2Sigma()`, `Gamma2graph()`, `Sigma2Gamma()`, `Theta2Gamma()`

Examples

```
Gamma <- cbind(
  c(0, 1.5, 1.5, 2),
  c(1.5, 0, 2, 1.5),
  c(1.5, 2, 0, 1.5),
  c(2, 1.5, 1.5, 0)
)
Gamma2Theta(Gamma)
```

---

generate_random_cactus

Generate a random cactus graph

Description

Generates a random cactus graph (mostly useful for benchmarking).

Usage

```
generate_random_cactus(d, cMin = 2, cMax = 6)
```

Arguments

d	Number of vertices in the graph
cMin	Minimal size of each block (last block might be smaller)
cMax	Maximal size of each block

See Also

Other Example generations: generate_random_Gamma(), generate_random_chordal_graph(), generate_random_connected_graph(), generate_random_graphical_Gamma(), generate_random_integer_Gamma(), generate_random_model(), generate_random_spd_matrix(), generate_random_tree()

---

generate_random_chordal_graph

Generate a random chordal graph

Description

Generates a random chordal graph by starting with a (small) complete graph and then adding new cliques until the specified size is reached. The sizes of cliques and separators can be specified.

Usage

```
generate_random_chordal_graph(
  d,
  cMin = 2,
  cMax = 6,
  sMin = 1,
  sMax = 4,
  block_graph = FALSE,
  ...
)
```

Arguments

d	Number of vertices in the graph
cMin	Minimal size of cliques (last clique might be smaller if necessary)
cMax	Maximal size of cliques
sMin	Minimal size of separators
sMax	Maximal size of separators
block_graph	Force `sMin == sMax == 1` to produce a block graph
...	Ignored, only allowed for compatibility

Value

An `[igraph::graph]` object

See Also

Other Example generations: `generate_random_Gamma()`, `generate_random_cactus()`, `generate_random_connected_gr`
`generate_random_graphical_Gamma()`, `generate_random_integer_Gamma()`, `generate_random_model()`, `generate_random_spd_matrix()`, `generate_random_tree()`

---

generate_random_connected_graph

Generate a random connected graph

Description

Generates a random connected graph. First tries to generate an Erdoes-Renyi graph, if that fails, falls back to producing a tree and adding random edges to that tree.

Usage

```
generate_random_connected_graph(
  d,
  m = NULL,
  p = 2/(d + 1),
  maxTries = 1000,
  ...
)
```

Arguments

d	Number of vertices in the graph
m	Number of edges in the graph (specify this or p)
p	Probability of each edge being in the graph (specify this or m)
maxTries	Maximum number of tries to produce a connected Eroes-Renyi graph
...	Ignored, only allowed for compatibility

Value

An [igraph::graph] object

See Also

Other Example generations: generate_random_Gamma(), generate_random_cactus(), generate_random_chordal_grap
generate_random_graphical_Gamma(), generate_random_integer_Gamma(), generate_random_model(),
generate_random_spd_matrix(), generate_random_tree()

---

generate_random_Gamma *Generate a random Gamma matrix*

Description

Generates a valid Gamma matrix with a given dimension

Usage

```
generate_random_Gamma(d, ...)
```

Arguments

d	Size of the matrix
...	Further arguments passed to generate_random_spd_matrix()

See Also

Other Example generations: generate_random_cactus(), generate_random_chordal_graph(),
generate_random_connected_graph(), generate_random_graphical_Gamma(), generate_random_integer_Gamma(),
generate_random_model(), generate_random_spd_matrix(), generate_random_tree()

`generate_random_graphical_Gamma`
Generate a random Gamma matrix for a given graph

Description

Generates a valid Gamma matrix with conditional independence structure specified by a graph

Usage

```
generate_random_graphical_Gamma(graph, ...)
```

Arguments

`graph`	An [`igraph::graph`] object
`...`	Furhter arguments passed to `generate_random_spd_matrix()`

See Also

Other Example generations: `generate_random_Gamma()`, `generate_random_cactus()`, `generate_random_chordal_grap`
`generate_random_connected_graph()`, `generate_random_integer_Gamma()`, `generate_random_model()`, `generate_random_spd_matrix()`, `generate_random_tree()`

`generate_random_integer_Gamma`
Generate a random Gamma matrix containing only integers

Description

Generates a random variogram Matrix by producing a $(d-1) \times (d-1)$ matrix `B` with random integer entries between `-b` and `b`, computing `S = B %*% t(B)`, and passing this `S` to `Sigma2Gamma()`. This process is repeated with an increasing `b` until a valid Gamma matrix is produced.

Usage

```
generate_random_integer_Gamma(d, b = 2, b_step = 1)
```

Arguments

`d`	Number of rows/columns in the output matrix
`b`	Initial `b` used in the algorithm described above
`b_step`	By how much `b` is increased in each iteration

Value

A numeric $d \times d$ variogram matrix with integer entries

See Also

Other Example generations: generate_random_Gamma(), generate_random_cactus(), generate_random_chordal_graph(), generate_random_connected_graph(), generate_random_graphical_Gamma(), generate_random_model(), generate_random_spd_matrix(), generate_random_tree()

Examples

```
generate_random_integer_Gamma(5, 2, 0.1)
```

---

generate_random_model *Generate random Huesler–Reiss Models*

Description

Generates a random connected graph and Gamma matrix with conditional independence structure corresponding to that graph.

Usage

```
generate_random_model(d, graph_type = "general", ...)
```

Arguments

d	Number of vertices in the graph
graph_type	"tree", "block", "decomposable", "complete", or "general"
...	Further arguments passed to functions generating the graph and Gamma matrix

See Also

Other Example generations: generate_random_Gamma(), generate_random_cactus(), generate_random_chordal_graph(), generate_random_connected_graph(), generate_random_graphical_Gamma(), generate_random_integer_Gamma(), generate_random_spd_matrix(), generate_random_tree()

Examples

```
set.seed(1)
d <- 12

generate_random_model(d, 'tree')
generate_random_model(d, 'block')
generate_random_model(d, 'decomposable')
generate_random_model(d, 'general')
generate_random_model(d, 'complete')
```

---

generate_random_spd_matrix
Generate a random symmetric positive definite matrix

Description

Generates a random $d \times d$ symmetric positive definite matrix. This is done by generating a random $d \times d$ matrix `B`, then computing `B %*% t(B)`, and then normalizing the matrix to approximately single digit entries.

Usage

```
generate_random_spd_matrix(d, bMin = -10, bMax = 10, ...)
```

Arguments

`d`	Number of rows/columns
`bMin`	Minimum value of entries in `B`
`bMax`	Maximum value of entries in `B`
`...`	Ignored, only allowed for compatibility

See Also

Other Example generations: `generate_random_Gamma()`, `generate_random_cactus()`, `generate_random_chordal_grap`
`generate_random_connected_graph()`, `generate_random_graphical_Gamma()`, `generate_random_integer_Gamma()`,
`generate_random_model()`, `generate_random_tree()`

---

generate_random_tree *Generate a random tree*

Description

Generates a random tree from a random Pruefer sequence

Usage

```
generate_random_tree(d)
```

Arguments

`d`	Number of vertices in the graph

Value

An [`igraph::graph`] object

See Also

Other Example generations: generate_random_Gamma(), generate_random_cactus(), generate_random_chordal_graph(), generate_random_connected_graph(), generate_random_graphical_Gamma(), generate_random_integer_Gamma(), generate_random_model(), generate_random_spd_matrix()

---

get_mc_cores *Number of cores to be used in parallel computations*

Description

Helper function that returns the number of cores to be used in parallel computations. Will always be 1 on windows. On other systems, this value can be set using `setOption('graphicalExtremes.mc.cores', ...)`.

Usage

```
get_mc_cores(overwrite = NULL)
```

Arguments

overwrite	Use this value (if it is valid and not on windows)

Value

An integer to be used as number of cores

See Also

graphicalExtremes-package

---

get_small_tol *Tolerances to be used in computations*

Description

Helper function that returns the tolerance to be used in internal computations.

Usage

```
get_small_tol(overwrite = NULL)

get_large_tol(overwrite = NULL)
```

Arguments

overwrite	NULL or numeric scalar. If specified, use this value instead of the option value.

Details

There are two different tolerances used in the package, for details see `graphicalExtremes-package`. The default values for these tolerances can be set using the options `"graphicalExtremes.tol.small"` and `"graphicalExtremes.tol.large"`.

Value

A non-negative numerical scalar

See Also

`graphicalExtremes-package`

`graphs_equal` *Graph equality*

Description

Produce true if two graphs have same vertices and edges (labelled)

Usage

```
graphs_equal(g1, g2)
```

Arguments

g1	igraph::graph
g2	igraph::graph

Value

`logical` indicating if the graphs are equal

`loglik_HR` *Compute Huesler–Reiss log-likelihood, AIC, and BIC*

Description

Computes (censored) Huesler–Reiss log-likelihood, AIC, and BIC values.

Usage

```
loglik_HR(data, p = NULL, graph = NULL, Gamma, cens = FALSE)
```

Arguments

data	Numeric $n \times d$ matrix. It contains observations following a multivariate HR Pareto distribution.
p	Numeric between 0 and 1 or `NULL`. If `NULL` (default), it is assumed that the `data` are already on multivariate Pareto scale. Else, `p` is used as the probability in the function `data2mpareto()` to standardize the `data`.
graph	An `[igraph::graph]` object or `NULL`. The `graph` must be undirected and connected. If no graph is specified, the complete graph is used.
Gamma	Numeric $n \times d$ matrix. It represents a variogram matrix Γ.
cens	Boolean. If true, then censored log-likelihood is computed. By default, `cens = FALSE`.

Value

Numeric vector `c("loglik"=..., "aic"=..., "bic"=...)` with the evaluated log-likelihood, AIC, and BIC values.

---

partialMatrixToGraph *Transformation of a partial matrix to a graph*

Description

Creates a graph that has edges in entries corresponding to non-NA entries in Gamma.

Usage

```
partialMatrixToGraph(Matrix)
```

Arguments

Matrix	A matrix with NA entries

Value

An `igraph::graph` object

plotFlights *Plot flight data*

Description

Plotting function to visualize the flight connections from the flights dataset. This function requires the package ggplot2 to be installed.

Usage

```
plotFlights(
  airportIndices = NULL,
  airports_sel = NULL,
  connections_sel = NULL,
  graph = NULL,
  plotAirports = TRUE,
  plotConnections = TRUE,
  returnGGPlot = FALSE,
  useAirportNFlights = FALSE,
  useConnectionNFlights = FALSE,
  minNFlights = 0,
  map = "state",
  vertexColors = NULL,
  vertexShapes = NULL,
  xyRatio = NULL,
  clipMap = FALSE,
  useLatex = FALSE,
  edgeAlpha = 0.2
)
```

Arguments

airportIndices	The indices of the airports (w.r.t. airports_sel) to include.
airports_sel	The airports to plot. Might be further subset by arguments airportIndeices, graph. If NULL, then flights$aiports will be used.
connections_sel	A three columns data frame as output by flightCountMatrixToConnectionList(). If NULL, then flights$nFlights will be used to construct one.
graph	An optional igraph::graph object, containing a flight graph to plot. Vertices should either match the selected airports in number and order, or be named with the corresponding IATA codes of the airports they represent.
plotAirports	Logical. Whether to plot the airports specified.
plotConnections	Logical. Whether to plot the connections specified.
returnGGPlot	If TRUE, a ggplot2::ggplot object is returned and not plotted immediately.

useAirportNFlights
: Logical. Whether to vary the size of the circles representing airports in the plot, according to the number of flights at that airport.

useConnectionNFlights
: Logical. Whether to vary the size of the edges representing connections in the plot, according to the number of flights on that connection.

minNFlights
: Numeric scalar. Only plot connections with at least this many flights.

map
: String or NULL. What map to use as the background image. Is passed to ggplot2::map_data().

vertexColors
: Optional vector, named with IATA codes, to be used as colors for the vertices/airports.

vertexShapes
: Optional vector, named with IATA codes, to be used as shapes for the vertices/airports. Is coerced to character.

xyRatio
: Approximate X-Y-ratio (w.r.t. distance on the ground) of the area shown in the plot.

clipMap
: Logical or numeric scalar. Whether to ignore the map image when determining the axis limits of the plot. If it is a positive scalar, the plot limits are extended by that factor.

useLatex
: Whether to format numbers etc. as latex code (useful when plotting to tikz).

edgeAlpha
: Numeric scalar between 0 and 1. The alpha value to be used when plotting edges/connections.

Value

If returnGGPlot is TRUE, a ggplot2::ggplot object, otherwise NULL.

Examples

```
# Plot all airports in the dataset
plotFlights(plotConnections = FALSE, map = 'world')

# Plot a selection of airports
plotFlights(c('JFK', 'SFO', 'LAX'), useConnectionNFlights = TRUE, useAirportNFlights = TRUE)

# Plot airports with a custom connections graph
IATAs <- c('ACV', 'BFL', 'EUG', 'SFO', 'MRY')
graph <- igraph::make_full_graph(length(IATAs))
plotFlights(IATAs, graph=graph, clipMap = 1.5)
```

---

rmpareto *Sampling of a multivariate Pareto distribution*

Description

Simulates exact samples of a multivariate Pareto distribution.

Usage

```
rmpareto(n, model = c("HR", "logistic", "neglogistic", "dirichlet"), d, par)
```

Arguments

`n`	Number of simulations.
`model`	The parametric model type; one of: • `HR` (default), • `logistic`, • `neglogistic`, • `dirichlet`.
`d`	Dimension of the multivariate Pareto distribution.
`par`	Respective parameter for the given `model`, that is, • Γ, numeric $d \times d$ variogram matrix, if `model` = `HR`. • $\theta \in (0, 1)$, if `model` = `logistic`. • $\theta > 0$, if `model` = `neglogistic`. • α, numeric vector of size `d` with positive entries, if `model` = `dirichlet`.

Details

The simulation follows the algorithm in Engelke and Hitz (2020). For details on the parameters of the Huesler–Reiss, logistic and negative logistic distributions see Dombry et al. (2016), and for the Dirichlet distribution see Coles and Tawn (1991).

Value

Numeric $n \times d$ matrix of simulations of the multivariate Pareto distribution.

References

Coles SG, Tawn JA (1991). "Modelling extreme multivariate events." *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **53**, 377–392.

Dombry C, Engelke S, Oesting M (2016). "Exact simulation of max-stable processes." *Biometrika*, **103**, 303–317.

Engelke S, Hitz AS (2020). "Graphical models for extremes (with discussion)." *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **82**, 871–932.

Examples

```
## A 4-dimensional HR distribution
n <- 10
d <- 4
G <- cbind(
  c(0, 1.5, 1.5, 2),
  c(1.5, 0, 2, 1.5),
```

```
  c(1.5, 2, 0, 1.5),
  c(2, 1.5, 1.5, 0)
)

rmpareto(n, "HR", d = d, par = G)

## A 3-dimensional logistic distribution
n <- 10
d <- 3
theta <- .6
rmpareto(n, "logistic", d, par = theta)

## A 5-dimensional negative logistic distribution
n <- 10
d <- 5
theta <- 1.5
rmpareto(n, "neglogistic", d, par = theta)

## A 4-dimensional Dirichlet distribution
n <- 10
d <- 4
alpha <- c(.8, 1, .5, 2)
rmpareto(n, "dirichlet", d, par = alpha)
```

`rmpareto_tree` *Sampling of a multivariate Pareto distribution on a tree*

Description

Simulates exact samples of a multivariate Pareto distribution that is an extremal graphical model on a tree as defined in Engelke and Hitz (2020).

Usage

```
rmpareto_tree(n, model = c("HR", "logistic", "dirichlet")[1], tree, par)
```

Arguments

`n` Number of simulations.

`model` The parametric model type; one of:
- `HR` (default),
- `logistic`,
- `dirichlet`.

`tree` Graph object from `igraph` package. This object must be a tree, i.e., an undirected graph that is connected and has no cycles.

`par` Respective parameter for the given `model`, that is,

- Γ, numeric $d \times d$ variogram matrix, where only the entries corresponding to the edges of the `tree` are used, if `model = HR`. Alternatively, can be a vector of length `d-1` containing the entries of the variogram corresponding to the edges of the given `tree`.
- $\theta \in (0, 1)$, vector of length `d-1` containing the logistic parameters corresponding to the edges of the given `tree`, if `model = logistic`.
- a matrix of size $(d-1) \times 2$, where the rows contain the parameters vectors α of size 2 with positve entries for each of the edges in `tree`, if `model = dirichlet`.

Details

The simulation follows the algorithm in Engelke and Hitz (2020). For details on the parameters of the Huesler–Reiss, logistic and negative logistic distributions see Dombry et al. (2016), and for the Dirichlet distribution see Coles and Tawn (1991).

Value

Numeric $n \times d$ matrix of simulations of the multivariate Pareto distribution.

References

Coles SG, Tawn JA (1991). “Modelling extreme multivariate events.” *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **53**, 377–392.

Dombry C, Engelke S, Oesting M (2016). “Exact simulation of max-stable processes.” *Biometrika*, **103**, 303–317.

Engelke S, Hitz AS (2020). “Graphical models for extremes (with discussion).” *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **82**, 871–932.

Examples

```
## A 4-dimensional HR tree model

my_tree <- igraph::graph_from_adjacency_matrix(rbind(
  c(0, 1, 0, 0),
  c(1, 0, 1, 1),
  c(0, 1, 0, 0),
  c(0, 1, 0, 0)
),
mode = "undirected"
)
n <- 10
Gamma_vec <- c(.5, 1.4, .8)
set.seed(123)
rmpareto_tree(n, "HR", tree = my_tree, par = Gamma_vec)

## A 4-dimensional Dirichlet model with asymmetric edge distributions

alpha <- cbind(c(.2, 1, .5), c(1.5, .6, .8))
```

```
rmpareto_tree(n, model = "dirichlet", tree = my_tree, par = alpha)
```

---

rmstable *Sampling of a multivariate max-stable distribution*

Description

Simulates exact samples of a multivariate max-stable distribution.

Usage

```
rmstable(n, model = c("HR", "logistic", "neglogistic", "dirichlet")[1], d, par)
```

Arguments

n	Number of simulations.
model	The parametric model type; one of: • HR (default), • logistic, • neglogistic, • dirichlet.
d	Dimension of the multivariate Pareto distribution.
par	Respective parameter for the given model, that is, • Γ, numeric $d \times d$ variogram matrix, if model = HR. • $\theta \in (0, 1)$, if model = logistic. • $\theta > 0$, if model = neglogistic. • α, numeric vector of size d with positive entries, if model = dirichlet.

Details

The simulation follows the extremal function algorithm in Dombry et al. (2016). For details on the parameters of the Huesler–Reiss, logistic and negative logistic distributions see Dombry et al. (2016), and for the Dirichlet distribution see Coles and Tawn (1991).

Value

Numeric $n \times d$ matrix of simulations of the multivariate max-stable distribution.

References

Coles SG, Tawn JA (1991). "Modelling extreme multivariate events." *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **53**, 377–392.

Dombry C, Engelke S, Oesting M (2016). "Exact simulation of max-stable processes." *Biometrika*, **103**, 303–317.

Examples

```
## A 4-dimensional HR distribution
n <- 10
d <- 4
G <- cbind(
  c(0, 1.5, 1.5, 2),
  c(1.5, 0, 2, 1.5),
  c(1.5, 2, 0, 1.5),
  c(2, 1.5, 1.5, 0)
)

rmstable(n, "HR", d = d, par = G)

## A 3-dimensional logistic distribution
n <- 10
d <- 3
theta <- .6
rmstable(n, "logistic", d, par = theta)

## A 5-dimensional negative logistic distribution
n <- 10
d <- 5
theta <- 1.5
rmstable(n, "neglogistic", d, par = theta)

## A 4-dimensional Dirichlet distribution
n <- 10
d <- 4
alpha <- c(.8, 1, .5, 2)
rmstable(n, "dirichlet", d, par = alpha)
```

`rmstable_tree` *Sampling of a multivariate max-stable distribution on a tree*

Description

Simulates exact samples of a multivariate max-stable distribution that is an extremal graphical model on a tree as defined in Engelke and Hitz (2020).

Usage

```
rmstable_tree(n, model = c("HR", "logistic", "dirichlet")[1], tree, par)
```

Arguments

`n`	Number of simulations.
`model`	The parametric model type; one of: • `HR` (default),

- `logistic`,
- `dirichlet`.

`tree` Graph object from `igraph` package. This object must be a tree, i.e., an undirected graph that is connected and has no cycles.

`par` Respective parameter for the given `model`, that is,

- Γ, numeric $d \times d$ variogram matrix, where only the entries corresponding to the edges of the `tree` are used, if `model = HR`. Alternatively, can be a vector of length `d-1` containing the entries of the variogram corresponding to the edges of the given `tree`.
- $\theta \in (0, 1)$, vector of length `d-1` containing the logistic parameters corresponding to the edges of the given `tree`, if `model = logistic`.
- a matrix of size $(d-1) \times 2$, where the rows contain the parameter vectors α of size 2 with positve entries for each of the edges in `tree`, if `model = dirichlet`.

Details

The simulation follows a combination of the extremal function algorithm in Dombry et al. (2016) and the theory in Engelke and Hitz (2020) to sample from a single extremal function. For details on the parameters of the Huesler–Reiss, logistic and negative logistic distributions see Dombry et al. (2016), and for the Dirichlet distribution see Coles and Tawn (1991).

Value

Numeric $n \times d$ matrix of simulations of the multivariate max-stable distribution.

References

Coles SG, Tawn JA (1991). "Modelling extreme multivariate events." *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **53**, 377–392.

Dombry C, Engelke S, Oesting M (2016). "Exact simulation of max-stable processes." *Biometrika*, **103**, 303–317.

Engelke S, Hitz AS (2020). "Graphical models for extremes (with discussion)." *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **82**, 871–932.

Examples

```
## A 4-dimensional HR tree model

my_tree <- igraph::graph_from_adjacency_matrix(rbind(
  c(0, 1, 0, 0),
  c(1, 0, 1, 1),
  c(0, 1, 0, 0),
  c(0, 1, 0, 0)
),
mode = "undirected"
)
```

```
n <- 10
Gamma_vec <- c(.5, 1.4, .8)
rmstable_tree(n, "HR", tree = my_tree, par = Gamma_vec)

## A 4-dimensional Dirichlet model with asymmetric edge distributions

alpha <- cbind(c(.2, 1, .5), c(1.5, .6, .8))
rmstable_tree(n, model = "dirichlet", tree = my_tree, par = alpha)
```

Sigma2Gamma *Transformation of Σ and Σ^k matrix to Γ matrix*

Description

Transforms the Σ^k matrix from the definition of a Huesler–Reiss distribution to the corresponding Γ matrix.

Usage

```
Sigma2Gamma(Sigma, k = NULL, full = FALSE)
```

Arguments

Sigma	Numeric $(d-1)\times(d-1)$ covariance matrix Σ^k from the definition of a Huesler–Reiss distribution. Numeric $d \times d$ covariance matrix if `full = TRUE`, see `full` parameter.
k	Integer between `1` (the default value) and `d`. Indicates which matrix Σ^k is given as `S`.
full	Logical. If true, then the kth row and column in Σ^k are included and the function returns a $d \times d$ matrix. By default, `full = FALSE`.

Details

For any `k` from `1` to `d`, the Σ^k matrix of size $(d-1) \times (d-1)$ in the definition of a Huesler–Reiss distribution can be transformed into a the corresponding $d \times d$ Γ matrix. If `full = TRUE`, then Σ^k must be a $d \times d$ matrix with kth row and column containing zeros. For details see Engelke and Hitz (2020). This is the inverse of function of `Gamma2Sigma()`.

Value

Numeric $d \times d$ Γ matrix.

References

Engelke S, Hitz AS (2020). "Graphical models for extremes (with discussion)." *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **82**, 871–932.

See Also

Other MatrixTransformations: `Gamma2Sigma()`, `Gamma2Theta()`, `Gamma2graph()`, `Theta2Gamma()`

Examples

```
Sigma1 <- rbind(
  c(1.5, 0.5, 1),
  c(0.5, 1.5, 1),
  c(1, 1, 2)
)
Sigma2Gamma(Sigma1, k = 1, full = FALSE)
```

`Theta2Gamma` *Transformation of Γ matrix to Θ matrix*

Description

Transforms a precision matrix (Θ or Θ^k) to the corresponding variogram matrix.

Usage

```
Theta2Gamma(Theta, k = NULL)
```

Arguments

`Theta`	Numeric $d \times d$ matrix (if `k` is `NULL`) or $(d-1) \times (d-1)$ matrix (if `k` is a number).
`k`	`NULL` or integer between 1 and d. If this is `NULL` the input `Theta` is interpreted as a $d \times d$ precision matrix Θ, otherwise as Θ^k.

Value

The $d \times d$ variogram matrix implied by `Theta`.

See Also

Other MatrixTransformations: `Gamma2Sigma()`, `Gamma2Theta()`, `Gamma2graph()`, `Sigma2Gamma()`

`Zmatrix` *Computes the Z-matrix*

Description

Copied from the R package "golazo" with kind permission by Piotr Zwiernik piotr.zwiernik@utoronto.ca. This function outputs the Z matrix, that is, the unique ultrametric matrix dominating S. This matrix is used to connstruct a starting point in the GOLAZO algorithm when L=0 but U has strictly positive (off-diagonal entries).

Usage

```
Zmatrix(S)
```

Arguments

`S`	a covariance matrix

Index