Package: ReturnCurves (via r-universe)

March 7, 2025

Type Package

Title Estimation of Return Curves

Version 1.0.1

Date 2025-02-05

Maintainer Lídia André <lidia.andre@unamur.be>

Description Estimates the p-probability return curve proposed by Murphy-Barltrop et al. (2023) <doi:10.1002/env.2797>. Implements pointwise and smooth estimation of the angular dependence function introduced by Wadsworth and Tawn (2013) <doi:10.3150/12-BEJ471>.

License GPL (>= 3)

Depends R (>= 2.10)

Imports evd, ismev, Rdpack, mathjaxr, ggplot2 (>= 3.4.4), gridExtra, methods, stats, shiny, tools, DT, shinydashboard, openair

Encoding UTF-8

RoxygenNote 7.3.2

RdMacros Rdpack, mathjaxr

Collate 'adf_est.R' 'adf_gof.R' 'adf_properties.R' 'bootstrap.R' 'composite_likelihood.R' 'heffernantawn.R' 'marginal_transform.R' 'min_projection_hill.R' 'rc_exp.R' 'rc_est.R' 'rc_gof.R' 'rc_uncertainty.R' 'ReturnCurves-package.R' 'airdata.R' 'runShiny.R' 'marggpd.R'

Suggests knitr, rmarkdown

VignetteBuilder knitr

LazyData true

URL https://github.com/lidiamandre/ReturnCurves

NeedsCompilation no

Author Lídia André [aut, cre], Callum Murphy-Barltrop [aut]

Repository CRAN

Date/Publication 2025-02-05 17:40:02 UTC

Config/pak/sysreqs make libicu-dev libjpeg-dev libpng-dev libx11-dev zlib1g-dev

Contents

`ReturnCurves-package` *Estimation of Return Curves*

Description

Implements the estimation of the p-probability return curve (Murphy-Barltrop et al. 2023), as well as a pointwise and smooth estimation of the angular dependence function (Wadsworth and Tawn 2013).

Available functions

`adf_est`: Estimation of the Angular Dependence Function (ADF)

`adf_gof`: Goodness of fit of the Angular Dependence Function estimates

`airdata`: Air pollution data

`marggpd`: Assessing the Marginal Tail Fits

`margtransf`: Marginal Transformation

`rc_est`: Return Curve estimation

`rc_gof`: Goodness of fit of the Return Curve estimates

`rc_unc`: Uncertainty of the Return Curve estimates

`runShiny`: Complementary Shiny app for the ReturnCurves package

Author(s)

Maintainer: Lídia André <`l.andre@lancaster.ac.uk`>

Authors:

- Callum Murphy-Barltrop <`callum.murphy-barltrop@tu-dresden.de`>

References

Murphy-Barltrop CJR, Wadsworth JL, Eastoe EF (2023). "New estimation methods for extremal bivariate return curves." *Environmetrics*, **34**(5). ISSN 1099095X, doi:10.1002/env.2797.

Wadsworth JL, Tawn JA (2013). "A new representation for multivariate tail probabilities." *Bernoulli*, **19**(5B), 2689-2714. ISSN 13507265, doi:10.3150/12BEJ471.

Examples

```
library(ReturnCurves)

data(airdata)

n <- dim(airdata)[1]

# Marginal Transformation
margdata <- margtransf(airdata)

head(margdata@dataexp)

# Return Curves estimation

prob <- 1/n

retcurve <- rc_est(margdata = margdata, p = prob, method = "hill")

head(retcurve@rc)

# ADF estimation
lambda <- adf_est(margdata = margdata, method = "hill")

head(lambda@adf)
```

`adf_est` *Estimation of the Angular Dependence Function (ADF)*

Description

Estimation of the angular dependence function $\lambda(\omega)$ introduced by Wadsworth and Tawn (2013).

Usage

```
adf_est(
  margdata,
  w = NULL,
  method = c("hill", "cl"),
  q = 0.95,
  qalphas = rep(0.95, 2),
```

```
  k = 7,
  constrained = FALSE,
  tol = 1e-04,
  par_init = rep(0, k - 1)
)
```

Arguments

margdata	An S4 object of class `margtransf.class`. See `margtransf` for more details.
w	Sequence of rays between `0` and `1`. Default is `NULL`, where a pre-defined grid is used.
method	String that indicates which method is used for the estimation of the angular dependence function. Must either be `"hill"`, to use the Hill estimator (Hill 1975), or `"cl"` to use the smooth estimator based on Bernstein-Bezier polynomials estimated by composite maximum likelihood.
q	Marginal quantile used to define the threshold u_ω of the min-projection variable T^1 at ray ω $\left(t^1_\omega = t_\omega - u_\omega \mid t_\omega > u_\omega\right)$, and/or Hill estimator (Hill 1975). Default is `0.95`.
qalphas	A vector containing the marginal quantile used for the Heffernan and Tawn conditional extremes model (Heffernan and Tawn 2004) for each variable, if `constrained = TRUE`. Default is `rep(0.95, 2)`.
k	Polynomial degree for the Bernstein-Bezier polynomials used for the estimation of the angular dependence function with the composite likelihood method (Murphy-Barltrop et al. 2024). Default is `7`.
constrained	Logical. If `FALSE` (default) no knowledge of the conditional extremes parameters is incorporated in the angular dependence function estimation.
tol	Convergence tolerance for the composite maximum likelihood procedure. Success is declared when the difference of log-likelihood values between iterations does not exceed this value. Default is `0.0001`.
par_init	Initial values for the parameters β of the Bernstein-Bezier polynomials used for estimation of the angular dependence function with the composite likelihood method (Murphy-Barltrop et al. 2024). Default is `rep(0, k-1)`.

Details

The angular dependence function $\lambda(\omega)$ can be estimated through a pointwise estimator, obtained with the Hill estimator, or via a smoother approach, obtained using Bernstein-Bezier polynomials and estimated via composite likelihood methods.

Knowledge of the conditional extremes framework introduced by Heffernan and Tawn (2004) can be incorporated by setting `constrained = TRUE`. Let $\alpha^1_{x|y} = \alpha_{x|y}/(1+\alpha_{x|y})$ and $\alpha^1_{y|x} = 1/(1+\alpha_{y|x})$ with $\alpha_{x|y}$ and $\alpha_{y|x}$ being the conditional extremes parameters. After obtaining $\hat{\alpha}_{x|y}$ and $\hat{\alpha}_{y|x}$ via maximum likelihood estimation, $\lambda(\omega) = \max\{\omega, 1-\omega\}$ for $\omega \in [0, \hat{\alpha}^1_{x|y}) \cup (\hat{\alpha}^1_{y|x}, 1]$ and is estimated as before for $\omega \in [\hat{\alpha}^1_{x|y}, \hat{\alpha}^1_{y|x}]$. For more details see Murphy-Barltrop et al. (2024).

Value

An object of S4 class `adf_est.class`. This object returns the arguments of the function and two extra slots:

- `interval`: A vector containing the maximum likelihood estimates from the conditional extremes model, $\hat{\alpha}^1_{x|y}$ and $\hat{\alpha}^1_{y|x}$, if `constrained = TRUE`. If `constrained = FALSE`, then `c(0, 1)` is returned; we note that this has no meaningful interpretation as the estimation is performed in an unconstrained interval.
- `adf`: A vector containing the estimates of the angular dependence function.

References

Heffernan JE, Tawn JA (2004). "A conditional approach for multivariate extreme values (with discussion)." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **66**(3), 497-546. doi:10.1111/j.14679868.2004.02050.x, https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9868.2004.02050.x.

Hill BM (1975). "A Simple General Approach to Inference About the Tail of a Distribution." *The Annals of Statistics*, **3**(5), 1163 – 1174. doi:10.1214/aos/1176343247.

Murphy-Barltrop CJR, Wadsworth JL, Eastoe EF (2024). "Improving estimation for asymptotically independent bivariate extremes via global estimators for the angular dependence function." 2303.13237.

Wadsworth JL, Tawn JA (2013). "A new representation for multivariate tail probabilities." *Bernoulli*, **19**(5B), 2689-2714. ISSN 13507265, doi:10.3150/12BEJ471.

Examples

```
library(ReturnCurves)

data(airdata)

n <- dim(airdata)[1]

margdata <- margtransf(airdata)

lambda <- adf_est(margdata = margdata, method = "hill")

plot(lambda)

# To see the the S4 object's slots
str(lambda)

# To access the estimates of the ADF
lambda@adf

# If constrained = T, the MLE estimates for the conditional extremes model
# can be accessed as
lambda@interval
```

adf_gof *Goodness of fit of the Angular Dependence function estimates*

Description

Assessment of the goodness of fit of the angular dependence function estimates $\lambda(\omega)$ following the procedure of Murphy-Barltrop et al. (2024).

Usage

```
adf_gof(adf, ray, blocksize = 1, nboot = 250, alpha = 0.05)
```

Arguments

`adf`	An S4 object of class `adf_est.class`. See `adf_est` for more details.
`ray`	Ray ω to be considered on the goodness of fit assessment.
`blocksize`	Size of the blocks for the block bootstrap procedure. If `1` (default), then a standard bootstrap approach is applied.
`nboot`	Number of bootstrap samples to be taken. Default is `250` samples.
`alpha`	Significance level to compute the $(1-\alpha)\%$ tolerance intervals. Default is `0.05`.

Details

Define the min-projection variable as $t_{\omega}^{1} = t_{\omega} - u_{\omega} | t_{\omega} > u_{\omega}$, then variable $\lambda(\omega) T_{\omega}^{1} \sim Exp(1)$ as $u_{\omega} \to \infty$ for all $\omega \in [0,1]$.

Let F_{E}^{-1} denote the inverse of the cumulative distribution function of a standard exponential variable and $T_{(i)}^{1}$ denote the i-th ordered increasing statistic, $i = 1, \dots, n$. Function `plot` shows a QQ plot between the model and empirical exponential quantiles, i.e. points $\left(F_{E}^{-1}\left(\frac{i}{n+1}\right), T_{(i)}^{1}\right)$, along with the line $y = x$. Uncertainty is obtained via a (block) bootstrap procedure and shown by the grey region on the plot. A good fit is shown by agreement of model and empirical quantiles, i.e. points should lie close to the line $y = x$. In addition, line $y = x$ should mainly lie within the $(1-\alpha)\%$ tolerance intervals.

We note that, if the grid for ω used to estimate the Angular Dependence Function (ADF) does not contain `ray`, then the closest ω in the grid is used to assess the goodness-of-fit of the ADF.

Value

An object of S4 class `adf_gof.class`. This object returns the arguments of the function and an extra slot `gof` which is a list containing:

`model`	A vector containing the model quantiles.
`empirical`	A vector containing the empirical quantiles.
`lower`	A vector containing the lower bound of the tolerance interval.
`upper`	A vector containing the upper bound of the tolerance interval.

Note

It is recommended to assess the goodness-of-fit of $\lambda(\omega)$ for a few values of ω.

References

Murphy-Barltrop CJR, Wadsworth JL, Eastoe EF (2024). "Improving estimation for asymptotically independent bivariate extremes via global estimators for the angular dependence function." 2303.13237.

Examples

```
library(ReturnCurves)

data(airdata)

n <- dim(airdata)[1]

margdata <- margtransf(airdata)

lambda <- adf_est(margdata = margdata, method = "hill")

# blocksize to account for temporal dependence
gof <- adf_gof(adf = lambda, ray = 0.4, blocksize = 10)

plot(gof)

# To see the the S4 object's slots
str(gof)

# To access the list of vectors
gof@gof
```

---

airdata *Air pollution data*

Description

Air pollution data from Marylebone, London (UK), between December and February of 1998 to 2005. It contains daily measurements of air pollutant concentrations of NOx and PM10. The dataset is a subset of the data from the `openair` package and consists of `1427` observations.

Format

Data frame with 1427 observations (rows) and 2 variables (columns):

nox Vector containing daily measurements of NOx, in parts per billion.

pm10 Vector containing daily measurements of PM10, in ug/m3.

Source

airdata is a subset of the data provided in the openair R package.

---

marggpd *Assessing the Marginal Tail Fits*

Description

Assessment of the marginal tail fits for each margin following the marginal transformation procedure margtransf.

Usage

```
marggpd(margdata, blocksize = 1, nboot = 250, alpha = 0.05)
```

Arguments

margdata	An S4 object of class margtransf.class. See margtransf for more details.
blocksize	Size of the blocks for the block bootstrap procedure. If 1 (default), then a standard bootstrap approach is applied.
nboot	Number of bootstrap samples to be taken. Default is 250 samples.
alpha	Significance level to compute the $(1-\alpha)\%$ tolerance intervals. Default is 0.05.

Details

Let $X_{(i)}^{GPD}$ denote the i-th ordered increasing statistic $(i = 1, \dots, n)$ of the exceedances, i.e., $X^{GPD} = (X - u \mid X > u)$, n_{exc} denote the sample size of these exceedances, and F_{GPD}^{-1} denote the inverse of the cumulative distribution function of a generalised Pareto distribution (GPD). Function plot shows QQ plots between the model and empirical GPD quantiles for both variables, i.e, for the first variable points $\left(F_{GPD}^{-1}\left(\frac{i}{n_{exc}+1}\right) + u, X_{(i)}^{GPD} + u\right)$, along with the line $y = x$.

Uncertainty on the empirical quantiles is obtained via a (block) bootstrap procedure and shown by the grey region on the plot. A good fit is shown by agreement of model and empirical quantiles, i.e. points should lie close to the line $y = x$. In addition, line $y = x$ should mainly lie within the $(1-\alpha)\%$ tolerance intervals.

Value

An object of S4 class marggpd.class. This object returns the arguments of the function and an extra slot marggpd which is a list containing:

model	A list containing the model quantiles for each variable.
empirical	A list containing the empirical quantiles for each variable.
lower	A list containing the lower bounds of the tolerance intervals for each variable.
upper	A list containing the upper bounds of the tolerance intervals for each variable.

Examples

```
library(ReturnCurves)

data(airdata)

n <- dim(airdata)[1]

margdata <- margtransf(airdata)

# blocksize to account for temporal dependence
marggpd <- marggpd(margdata = margdata, blocksize = 10)

plot(marggpd)

# To see the the S4 object's slots
str(marggpd)

# To access the list of lists
marggpd@marggpd
```

`margtransf` *Marginal Transformation*

Description

Marginal transformation of a bivariate random vector to standard exponential margins following Coles and Tawn (1991). Variables within each margin are assumed identically distributed.

Usage

```
margtransf(data, qmarg = rep(0.95, 2), constrainedshape = TRUE)
```

Arguments

`data`	A matrix containing the data on the original margins.
`qmarg`	A vector containing the marginal quantile used to fit the Generalised Pareto Distribution (GPD) for each variable. Default is `rep(0.95, 2)`.
`constrainedshape`	Logical. If `TRUE` (Default), the estimated shape parameter of the Generalised Pareto Distribution (GPD) is constrained to lie strictly above `-1`.

Details

Given a threshold value u, each stationary random vector is transformed by using the empirical cumulative distribution function (cdf) below u, and a Generalise Pareto Distribution (GPD) fit above u.

The option to constrain $\xi > -1$ is included as $\xi \leq -1$ implies that the fitted upper endpoint of the distribution's support is the maximum data point. This situation is rarely encountered in practice.

Value

An object of S4 class `margtransf.class`. This object returns the arguments of the function, a slot `parameters` containing a matrix with the shape and scale parameters of the Generalised Pareto Distribution (GPD) for each variable, a slot `thresh` containing a vector with the threshold u above which the GPD is fitted, and a slot `dataexp` containing a matrix with the data on standard exponential margins.

The `plot` function takes an object of S4 class `margtransf.class`, and a `which` argument specifying the type of plot desired (see **Examples**):

`"hist"`	Plots the marginal distributions of the two variables on original and standard exponential margins.
`"ts"`	Plots the time series of the two variables on original and standard exponential margins.
`"joint"`	Plots the joint distribution of the two variables on original and standard exponential margins.
`"all"`	Plots all the above mentioned plots (default).

References

Coles SG, Tawn JA (1991). "Modelling Extreme Multivariate Events." *Journal of the Royal Statistical Society. Series B (Methodological)*, **53**(2), 377–392. ISSN 00359246, doi:10.1111/j.2517-6161.1991.tb01830.x.

Examples

```
library(ReturnCurves)

data(airdata)

n <- dim(airdata)[1]

margdata <- margtransf(airdata)

# Plots the marginal distributions of X and Y on original vs standard exponential margins
plot(margdata, which = "hist")

# Plots the time series of X and Y on original vs standard exponential margins
plot(margdata, which = "ts")

# Plots the joint distribution of X and Y on original vs standard exponential margins
plot(margdata, which = "joint")

# Plots all the available plots
plot(margdata, which = "all")

# To see the the S4 object's slots
str(margdata)

# To access the matrix with the data on standard exponential margins
margdata@dataexp
```

rc_est *Estimation of the Return Curve*

Description

Estimation of the p-probability return curve following Murphy-Barltrop et al. (2023).

Usage

```
rc_est(
  margdata,
  w = NULL,
  p,
  method = c("hill", "cl"),
  q = 0.95,
  qalphas = rep(0.95, 2),
  k = 7,
  constrained = FALSE,
  tol = 0.001,
  par_init = rep(0, k - 1)
)
```

Arguments

margdata	An S4 object of class `margtransf.class`. See `margtransf` for more details.
w	Sequence of rays between `0` and `1`. Default is `NULL`, where a pre-defined grid is used.
p	Curve survival probability. Must be $p < 1 - q$ and $p < 1 - q_\alpha$.
method	String that indicates which method is used for the estimation of the angular dependence function. Must either be `"hill"`, to use the Hill estimator (Hill 1975), or `"cl"` to use the smooth estimator based on Bernstein-Bezier polynomials estimated by composite maximum likelihood.
q	Marginal quantile used to define the threshold u_ω of the min-projection variable T^1 at ray ω $\left(t^1_\omega = t_\omega - u_\omega \| t_\omega > u_\omega\right)$, and/or Hill estimator (Hill 1975). Default is `0.95`.
qalphas	A vector containing the marginal quantile used for the Heffernan and Tawn conditional extremes model (Heffernan and Tawn 2004) for each variable, if `constrained = TRUE`. Default is `rep(0.95, 2)`.
k	Polynomial degree for the Bernstein-Bezier polynomials used for the estimation of the angular dependence function with the composite likelihood method (Murphy-Barltrop et al. 2024). Default is `7`.
constrained	Logical. If `FALSE` (default) no knowledge of the conditional extremes parameters is incorporated in the angular dependence function estimation.

`tol` Convergence tolerance for the composite maximum likelihood procedure. Success is declared when the difference of log-likelihood values between iterations does not exceed this value. Default is `0.0001`.

`par_init` Initial values for the parameters β of the Bernstein-Bezier polynomials used for estimation of the angular dependence function with the composite likelihood method (Murphy-Barltrop et al. 2024). Default is `rep(0, k-1)`.

Details

Given a probability p and a joint survival function $Pr(X > x, Y > y)$, the p-probability return curve is defined as

$$RC(p) := \left\{(x, y) \in R^2 : Pr(X > x, Y > y) = p\right\}.$$

This method focuses on estimation of $RC(p)$ for small p near 0, so that (X, Y) are in the tail of the distribution.

$Pr(X > x, Y > y)$ is estimated using the angular dependence function $\lambda(\omega)$ introduced by Wadsworth and Tawn (2013). More details on how to estimate $\lambda(\omega)$ can be found in `adf_est`.

The return curve estimation $\hat{RC}(p)$ is done on standard exponential margins and then back transformed onto the original margins.

Value

An object of S4 class `rc_est.class`. This object returns the arguments of the function and extra slot `rc`

`interval:` A vector containing the maximum likelihood estimates from the conditional extremes model, $\hat{\alpha}^1_{x|y}$ and $\hat{\alpha}^1_{y|x}$, if `constrained = TRUE`. If `constrained = FALSE`, then `c(0, 1)` is returned; we note that this has no meaningful interpretation as the estimation is performed in an unconstrained interval.

`rc:` A matrix with the estimates of the Return Curve.

References

Heffernan JE, Tawn JA (2004). "A conditional approach for multivariate extreme values (with discussion)." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **66**(3), 497-546. doi:10.1111/j.14679868.2004.02050.x, https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9868.2004.02050.x.

Hill BM (1975). "A Simple General Approach to Inference About the Tail of a Distribution." *The Annals of Statistics*, **3**(5), 1163 – 1174. doi:10.1214/aos/1176343247.

Murphy-Barltrop CJR, Wadsworth JL, Eastoe EF (2023). "New estimation methods for extremal bivariate return curves." *Environmetrics*, **34**(5). ISSN 1099095X, doi:10.1002/env.2797.

Murphy-Barltrop CJR, Wadsworth JL, Eastoe EF (2024). "Improving estimation for asymptotically independent bivariate extremes via global estimators for the angular dependence function." 2303.13237.

Wadsworth JL, Tawn JA (2013). "A new representation for multivariate tail probabilities." *Bernoulli*, **19**(5B), 2689-2714. ISSN 13507265, doi:10.3150/12BEJ471.

Examples

```
library(ReturnCurves)

data(airdata)

n <- dim(airdata)[1]

prob <- 10/n

margdata <- margtransf(airdata)

retcurve <- rc_est(margdata = margdata, p = prob, method = "hill")

plot(retcurve)

# To see the the S4 object's slots
str(retcurve)

# To access the return curve estimation
retcurve@rc

# If constrained = T, the MLE estimates for the conditional extremes model
# can be accessed as
retcurve@interval
```

`rc_gof` *Goodness of fit of the Return Curve estimates*

Description

Assessment of the goodness-of-fit of the return curve estimates following the approach of Murphy-Barltrop et al. (2023).

Usage

```
rc_gof(retcurve, blocksize = 1, nboot = 250, nangles = 150, alpha = 0.05)
```

Arguments

`retcurve`	An S4 object of class `rc_est.class`. See `rc_est` for more details.
`blocksize`	Size of the blocks for the block bootstrap procedure. If `1` (default), then a standard bootstrap approach is applied.
`nboot`	Number of bootstrap samples to be taken. Default is `250` samples.

`nangles` Number of angles m in the interval $(0, \pi/2)$ (Murphy-Barltrop et al. 2023). Default is `150` angles.

`alpha` Significance level to compute the $(1-\alpha)\%$ confidence intervals. Default is `0.05`.

Details

Given a return curve RC(p), the probability of lying in a survival region is p. Let

$$\mathbf{\Theta} := \left\{ \frac{\pi(m+1-j)}{2(m+1)} \mid 1 \leq j \leq m \right\}$$

be a set of angles decreasing from near $\pi/2$ to 0. For each angle $\theta_j \in \mathbf{\Theta}$, and corresponding point in the estimated return curve $\{(\hat{x}_{\theta_j}, \hat{y}_{\theta_j})\}$, the empirical probability $\hat{p}_j$ of lying in the survival region is given by the proportion of points in the region $(\hat{x}_{\theta_j}, \infty) \times (\hat{y}_{\theta_j}, \infty)$.

Thus, for each angle $\theta_j \in \mathbf{\Theta}$, a (block) bootstrap procedure to the original data set is applied, and the empirical probabilities $\hat{p}_j$ estimated. Then, the median and $(1-\alpha)\%$ pointwise confidence intervals are obtained for each θ_j. Function `plot` shows the median of $\hat{p}_j$, the confidence intervals and the true probability p; ideally, this value should be contained in the confidence region.

We note that due to the use of empirical probabilities, the value of p should be within the range of the data and not too extreme.

Value

An object of S4 class `rc_gof.class`. This object returns the arguments of the function and an extra slot `gof` which is a list containing:

`median` A vector containing the median of the empirical probability of lying in a survival region.

`lower` A vector containing the lower bound of the confidence interval.

`upper` A vector containing the upper bound of the confidence interval.

References

Murphy-Barltrop CJR, Wadsworth JL, Eastoe EF (2023). "New estimation methods for extremal bivariate return curves." *Environmetrics*, **34**(5). ISSN 1099095X, doi:10.1002/env.2797.

Examples

```
library(ReturnCurves)

data(airdata)

n <- dim(airdata)[1]

prob <- 10/n

margdata <- margtransf(airdata)

rc_orig <- rc_est(margdata = margdata, p = prob, method = "hill")
```

```
# blocksize to account for temporal dependence
gof <- rc_gof(retcurve = rc_orig, blocksize = 10)

plot(gof)

# To see the the S4 object's slots
str(gof)

# To access the list of vectors
gof@gof
```

`rc_unc` *Uncertainty of the Return Curve estimates*

Description

Uncertainty assessment of the return curve estimates following the procedure of Murphy-Barltrop et al. (2023).

Usage

```
rc_unc(retcurve, blocksize = 1, nboot = 250, nangles = 150, alpha = 0.05)
```

Arguments

`retcurve`	An S4 object of class `rc_est.class`. See `rc_est` for more details.
`blocksize`	Size of the blocks for the block bootstrap procedure. If `1` (default), then a standard bootstrap approach is applied.
`nboot`	Number of bootstrap samples to be taken. Default is `250` samples.
`nangles`	Number of angles m in the interval $(0, \pi/2)$ (Murphy-Barltrop et al. 2023). Default is `150` angles.
`alpha`	Significance level to compute the $(1-\alpha)\%$ confidence intervals. Default is `0.05`.

Details

Define a set of angles

$$\boldsymbol{\Theta} := \left\{ \frac{\pi(m+1-j)}{2(m+1)} \mid 1 \leq j \leq m \right\}$$

decreasing from near $\pi/2$ to 0, and let $L_\theta := \left\{(x, y) \in R^2_+ \,|\, \tan(\theta) = y/x\right\}$ denote the line segment intersecting the origin with gradient $\tan(\theta) > 0$. For each $\theta \in \boldsymbol{\Theta}$, L_θ intersects the estimated $\hat{RC}(p)$ exactly once, i.e. $\{(\hat{x}_\theta, \hat{y}_\theta)\} := \hat{RC}(p) \cap L_\theta$. Uncertainty of the return curve is then quantified by the distribution of $\hat{d}_\theta := (\hat{x}^2_\theta + \hat{y}^2_\theta)^{1/2}$ via a (block) bootstrap procedure.

This procedure is as follows; for $k = 1, \ldots,$ `nboot`:

1. (Block) bootstrap the original data set;
2. For each $\theta \in \mathbf{\Theta}$, obtain $\hat{d}_{\theta,k}$ for the corresponding return curve point estimate.

Full details can be found in Murphy-Barltrop et al. (2023)

Value

An object of S4 class `rc_unc.class`. This object returns the arguments of the function and an extra slot `unc` which is a list containing:

`median`	A vector containing the median estimates of the return curve.
`mean`	A vector containing the mean estimates of the return curve.
`lower`	A vector containing the lower bound of the confidence interval.
`upper`	A vector containing the upper bound of the confidence interval.

The `plot` function takes an object of S4 class `rc_unc.class`, and a `which` argument specifying the type of plot desired (see **Examples**):

`"rc"`	Plots the estimated Return Curve and its uncertainty (default).
`"median"`	Plots the median estimates of the Return Curve and its uncertainty.
`"mean"`	Plots the mean estimates of the Return Curve and its uncertainty.
`"all"`	Plots the estimated Return Curve, the median and mean estimates of the Return Curve together, and the associated uncertainty.

References

Murphy-Barltrop CJR, Wadsworth JL, Eastoe EF (2023). "New estimation methods for extremal bivariate return curves." *Environmetrics*, **34**(5). ISSN 1099095X, doi:10.1002/env.2797.

Examples

```
library(ReturnCurves)

data(airdata)

n <- dim(airdata)[1]

prob <- 10/n

margdata <- margtransf(airdata)

rc_orig <- rc_est(margdata = margdata, p = prob, method = "hill")


# Set nboot = 50 for an illustrative example
# blocksize to account for temporal dependence
unc <- rc_unc(rc_orig, blocksize = 10)

# Plots the estimated Return Curve
plot(unc, which = "rc")
```

```
# Plots the median estimates of the Return Curve
plot(unc, which = "median")

# Plots the mean estimates of the Return Curve
plot(unc, which = "mean")

# Plots the estimated Return Curve and its the median and mean estimates
plot(unc, which = "all")

# To see the the S4 object's slots
str(unc)

# To access the list of vectors
unc@unc
```

---

runShiny *Complementary Shiny app for the ReturnCurves package*

Description

Launches the R Shiny app complementary to the ReturnCurves-package.

Usage

```
runShiny()
```

Index