Package 'TLMoments'

December 4, 2019

Type Package

Title Calculate TL-Moments and Convert Them to Distribution Parameters

Version 0.7.5

Date 2019-12-04

Author Jona Lilienthal

Maintainer Jona Lilienthal <lilienthal@statistik.tu-dortmund.de>

Description Calculates empirical TL-moments (trimmed L-moments) of arbitrary order and trimming, and converts them to distribution parameters.

License GPL (>= 2)

Depends R (>= 2.10), Rcpp (>= 0.12.12)

Imports hypergeo, ggplot2, stats

Suggests evd, knitr, magrittr, lmom, lmomco, Lmoments, rmarkdown

VignetteBuilder knitr

LinkingTo Rcpp

Encoding UTF-8

RoxygenNote 7.0.2

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-12-04 12:20:02 UTC

R topics documented:

`as.parameters` *Converting to parameters-objects*

Description

Convert vector, matrix, list, or data.frame to parameters-objects.

Usage

```
as.parameters(..., distr = NULL)

## S3 method for class 'numeric'
as.parameters(..., distr)

## S3 method for class 'matrix'
as.parameters(x, distr, ...)

## S3 method for class 'list'
as.parameters(x, distr, ...)

## S3 method for class 'data.frame'
as.parameters(x, formula, distr, ...)
```

Arguments

...	parameters of distribution. This can be named vectors or lists, matrices, or data.frames. See examples below.
distr	character giving the distribution. Function of name q\"distr\" has to be available.
x	numeric vector, matrix, list, or data.frame of parameters.
formula	if x is data.frame a formula has to be given.

Value

object of class parameters, see parameters help page.

Methods (by class)

- numeric: as.parameters for numeric data vectors
- matrix: as.parameters for numeric data matrices
- list: as.parameters for numeric data lists
- data.frame: as.parameters for numeric data.frames

See Also

parameters

Examples

```
# Vector input:
as.parameters(loc = 3, scale = 2, shape = .4, distr = "gev")
as.parameters(c(loc = 3, scale = 2, shape = .4), distr = "gev")

# Names can be shortened if unambiguous:
as.parameters(l = 3, sc = 2, sh = .4, distr = "gev")
as.parameters(m = 3, s = 1, distr = "norm")

# Wrong or ambiguous names lead to errors!
## Not run:
as.parameters(l = 3, s = 2, s = .4, distr = "gev")
as.parameters(loc2 = 3, scale = 2, shape = .4, distr = "gev")

## End(Not run)

# If no names are given, a warning is given and they are guessed for gev, gpd, gum, and ln3.
as.parameters(3, 2, .4, distr = "gev")
as.parameters(c(3, 2, .4), distr = "gev")
## Not run:
as.parameters(3, 2, .2, .4, distr = "gev") #=> doesn't work

## End(Not run)

# Matrix input:
# Parameters in matrices must have either matching rownames or colnames!
```

```
as.parameters(cbind(loc = 10, scale = 4, shape = seq(0, .4, .1)), distr = "gev")
as.parameters(rbind(loc = 10, scale = 4, shape = seq(0, .4, .1)), distr = "ln3")

# If no names are given, a guess is made based on number of rows
# or cols according to distribution (and a warning is given).
as.parameters(matrix(1:9, nr = 3), distr = "gev")
as.parameters(matrix(1:8, nc = 2), distr = "gum")


# The same principles apply for list input and data.frames:

# List input:
as.parameters(list(list(mean = 2, sd = 1), list(mean = 0, sd = 1)), distr = "norm")
as.parameters(list(c(m = 2, s = 1), c(m = 0, s = 1)), distr = "norm")
as.parameters(list(c(loc = 2, scale = 1), c(0, 1)), distr = "gum")
## Not run:
as.parameters(list(c(loc = 2, scale = 1), c(0, 1, 2)), distr = "gum")

## End(Not run)

# Dataframe input:
xdat <- data.frame(station = c(1, 2), mean = c(2, 0), sd = c(1, 1))
as.parameters(xdat, cbind(mean, sd) ~ station, distr = "norm")
as.parameters(xdat, . ~ station, distr = "norm")
as.parameters(xdat, cbind(mean, sd) ~ ., distr = "norm")

xdat <- data.frame(station = c(1, 2), m = c(2, 0), s = c(1, 1))
as.parameters(xdat, cbind(m, s) ~ station, distr = "norm")
## Not run:
as.parameters(xdat, cbind(m, s) ~ station, distr = "gev")

## End(Not run)

###

# Results of as.parameters can be used in the normal TLMoments-scheme:
# they can be transfered to quantiles or to TLMoments.

xdat <- data.frame(station = c(1, 2), mean = c(2, 0), sd = c(1, 1))
quantiles(as.parameters(xdat, cbind(mean, sd) ~ ., distr = "norm"), c(.99))

# quantile estimation
p <- as.parameters(loc = 3, scale = 2, shape = .4, distr = "gev")
quantiles(p, c(.9, .95))
p <- as.parameters(cbind(loc = 10, scale = 4, shape = seq(0, .4, .1)), distr = "gev")
quantiles(p, c(.9, .95))
p <- as.parameters(list(list(mean = 2, sd = 1), list(mean = 0, sd = 1)), distr = "norm")
quantiles(p, c(.95, .975))

# With magrittr
library(magrittr)
as.parameters(loc = 3, scale = 2, shape = .4, distr = "gev") %>% quantiles(c(.9, .99))
```

`as.PWMs` *Convert to PWMs-object*

Description

Convert vector, matrix, list, or data.frame to PWMs-objects.

Usage

```
as.PWMs(x, ..., order = NULL)

## S3 method for class 'numeric'
as.PWMs(x, order = seq_along(x) - 1, ...)

## S3 method for class 'matrix'
as.PWMs(x, order = row(x)[, 1] - 1, ...)

## S3 method for class 'list'
as.PWMs(x, order = seq_along(x[[1]]) - 1, ...)

## S3 method for class 'data.frame'
as.PWMs(x, formula, order = NULL, ...)
```

Arguments

`x`	vector or matrix of PWMs.
`...`	additional arguments.
`order`	integer, corresponding order to given PWMs. If NULL, order is set to 0:(length(x)-1).
`formula`	if `x` is data.frame. See examples.

Value

object of class PWMs, see PWMs help page.

Methods (by class)

- `numeric`: as.PWMs for numeric data vectors
- `matrix`: as.PWMs for numeric data matrices
- `list`: as.PWMs for numeric data lists
- `data.frame`: as.PWMs for numeric data.frames

See Also

PWMs

Examples

```
xmat <- cbind(c(0.12, .41, .38, .33), c(.05, 0.28, .25, .22))
xvec <- xmat[, 1]
xlist <- lapply(1:ncol(xmat), function(i) xmat[, i])
xdat <- data.frame(
 station = letters[1:3],
 season = c("S", "W", "S"),
 b0 = c(.12, .15, .05),
 b1 = c(.41, .33, .28),
 b2 = c(.38, .18, .25)
)

as.PWMs(xvec)
as.PWMs(xvec[-2], order = c(0, 2, 3))

as.PWMs(xmat)
as.PWMs(xmat[-2, ], order = c(0, 2, 3))

as.PWMs(xlist)

as.PWMs(xdat, cbind(b0, b1, b2) ~ station)
as.PWMs(xdat, . ~ station + season)
as.PWMs(xdat, cbind(b0, b2) ~ station, order = c(0, 2))

p <- as.PWMs(xdat, cbind(b0, b1, b2) ~ station)
TLMoments(p)

(p <- as.PWMs(xdat, cbind(b0, b1) ~ station))
#parameters(TLMoments(p), "gev") # => error
#parameters(TLMoments(p), "gpd") # => error
parameters(TLMoments(p), "gpd", u = 10)

(p <- as.PWMs(xdat, cbind(b0, b2) ~ station, order = c(0, 2)))
#TLMoments(p) # => error
```

`as.TLMoments` *Convert to TLMoments-object*

Description

Convert vector, matrix, list, or data.frame of TL-moments or TL-moment ratios or a PWMs-object to a TLMoments-object in order to be used with TLMoments-functions. The first position of a vector or the first row of a matrix is always used as the L1-moment. The `ratios` argument determines if the following positions or rows are used as TL-moments oder TL-moments ratios. The trimming has to be given using the `leftrim` and `rightrim` arguments.

Usage

```
as.TLMoments(x, ..., leftrim, rightrim, ratios)

## S3 method for class 'numeric'
as.TLMoments(x, leftrim = 0L, rightrim = 0L, ratios = FALSE, ...)

## S3 method for class 'matrix'
as.TLMoments(x, leftrim = 0L, rightrim = 0L, ratios = FALSE, ...)

## S3 method for class 'list'
as.TLMoments(x, leftrim = 0L, rightrim = 0L, ratios = FALSE, ...)

## S3 method for class 'data.frame'
as.TLMoments(x, formula, leftrim = 0L, rightrim = 0L, ratios = FALSE, ...)
```

Arguments

x	vector or matrix of TL-moments (or TL-moment ratios if ratios is TRUE) or a PWMs-object. The first position or row is always used as the L1-moment, if vector or matrix are given.
...	additional arguments.
leftrim, rightrim	integer, order of trimmed L-moments.
ratios	boolean, if TRUE the non-first positions or rows of x give L-moment ratios, if FALSE (default) they give L-moments. If ratios are used and the first position or row is NA, L1 is assumed to be 1!
formula	if x is data.frame. See examples.

Value

object of class TLMoments, see PWMs help page.

Methods (by class)

- numeric: as.TLMoments for numeric data vectors
- matrix: as.TLMoments for numeric data matrices
- list: as.TLMoments for numeric data lists
- data.frame: as.TLMoments for numeric data.frames

See Also

TLMoments

Examples

```
### Vector or matrix as input
xmat <- cbind(c(1, .2, .05), c(1, .2, NA), c(1.3, NA, .1))
xvec <- xmat[, 1]
xlist <- lapply(1:ncol(xmat), function(i) xmat[, i])
xdat <- data.frame(
 station = rep(letters[1:3], each = 1),
 season = c("S", "W", "S"),
 L1 = c(1, 1, 1.3),
 L2 = c(.2, .2, .3),
 L3 = c(.05, .04, .1)
)

as.TLMoments(xvec, rightrim = 1)
as.TLMoments(xmat, rightrim = 1)
as.TLMoments(xlist, rightrim = 1)
as.TLMoments(xdat, cbind(L1, L2, L3) ~ station)
as.TLMoments(xdat, .~station+season)
as.TLMoments(xdat, cbind(L1, L2, L3) ~ .)

parameters(as.TLMoments(xvec, rightrim = 0), "gev")
#lmomco::lmom2par(lmomco::vec2lmom(c(1, .2, .25)), "gev")$para

xmat <- cbind(c(NA, .2, -.05), c(NA, .2, .2))
xvec <- xmat[, 1]

as.TLMoments(xvec, ratios = TRUE)
as.TLMoments(xmat, ratios = TRUE)
parameters(as.TLMoments(xvec, ratios = TRUE), "gev")
#lmomco::lmom2par(lmomco::vec2lmom(c(1, .2, -.05)), "gev")$para

xmat <- cbind(c(10, .2, -.05), c(10, .2, .2))
xvec <- xmat[, 1]

as.TLMoments(xvec, ratios = TRUE)
as.TLMoments(xmat, ratios = TRUE)
parameters(as.TLMoments(xvec, ratios = TRUE), "gev")
#lmomco::lmom2par(lmomco::vec2lmom(c(10, .2, -.05)), "gev")$para
```

`est_cov` *Covariance matrix of PWMs, TLMoments, parameters, or quantiles*

Description

Calculation of the empirical or theoretical covariance matrix of objects of the classes `PWMs`, `TLMoments`, `parameters`, or `quantiles`.

Usage

```
est_cov(x, ...)

## S3 method for class 'PWMs'
est_cov(x, select = attr(x, "order"), ...)

## S3 method for class 'TLMoments'
est_cov(x, select = attr(x, "order"), ...)

## S3 method for class 'parameters'
est_cov(x, select = c("loc", "scale", "shape"), ...)

## S3 method for class 'quantiles'
est_cov(x, select = attr(x, "p"), ...)
```

Arguments

x	object of PWMs, TLMoments, parameters, or quantiles constructed using the same-named functions.
...	additional arguments given to the sub-functions: distr and np.cov (see details).
select	numeric oder character vector specifying a subset of the covariance matrix. If not specified the full covariance matrix is returned.

Details

Covariance matrices of PWMs and TLMoments are calculated without parametric assumption by default. Covariance matrices of parameters and quantiles use parametric assumption based on their stored distribution attribute (only GEV at the moment). Parametric (GEV) calculation can be enforced by specifying distr=\"gev\", non-parametric calculation by using np.cov=TRUE.

Value

numeric matrix (if x is of class PWMs, parameters, or quantiles) or a list of two matrices (lambdas and ratios, if x is of class TLMoments).

See Also

PWMs, TLMoments, parameters, quantiles

Examples

```
### 1: PWMs:

xvec <- rgev(100, shape = .1)
xmat <- cbind(rgev(100, shape = .1), rgev(100, shape = .3))

# Covariance estimation of PWMs normally without parametric assumption:
est_cov(PWMs(xvec))
est_cov(PWMs(xvec), select = 0:1)
```

```
est_cov(PWMs(xmat))
est_cov(PWMs(xmat), select = 3)
est_cov(PWMs(xmat[, 1, drop = FALSE]), select = 2:3)

# Parametric assumptions (only GEV by now) can be used:
est_cov(PWMs(xvec), distr = "gev")
est_cov(PWMs(xvec), distr = "gev", select = c(1, 3))

## Not run:
cov(t(replicate(100000,
  as.vector(PWMs(cbind(rgev(100, shape = .1), rgev(100, shape = .3)), max.order = 1)))
))

## End(Not run)


### 2. TLMoments:

xvec <- rgev(100, shape = .1)
xmat <- cbind(rgev(100, shape = .1), rgev(100, shape = .3))

# Covariance estimation of TLMoments normally without parametric assumption:
est_cov(TLMoments(xvec))
est_cov(TLMoments(xvec, rightrim = 1))
est_cov(TLMoments(xvec), select = 3:4)

# Parametric assumptions (only GEV by now) can be used:
est_cov(TLMoments(xvec), distr = "gev")

# Matrix inputs
est_cov(TLMoments(xmat))
est_cov(TLMoments(xmat), select = 3:4)
est_cov(TLMoments(xmat[, 1, drop = FALSE]), select = 3:4)

# Covariance of theoretical TLMoments only with parametric assumption:
est_cov(as.TLMoments(c(14, 4, 1)), distr = "gev", set.n = 100)
est_cov(as.TLMoments(c(14, 4, 1), rightrim = 1), distr = "gev", set.n = 100)

# Regionalized TLMoments
est_cov(regionalize(TLMoments(xmat), c(.75, .25)))
est_cov(regionalize(TLMoments(xmat), c(.75, .25)), distr = "gev", select = 3:4)


### 3. Parameters:

xvec <- rgev(100, shape = .1)
xmat <- cbind(rgev(100, shape = .1), rgev(100, shape = .3))

# Covariance estimation of parameters normally with parametric assumption:
est_cov(parameters(TLMoments(xvec), "gev"))
est_cov(parameters(TLMoments(xvec, rightrim = 1), "gev"))
est_cov(parameters(TLMoments(xvec, rightrim = 1), "gev"), select = c("scale", "shape"))
```

```
# A nonparametric estimation can be enforced with np.cov:
est_cov(parameters(TLMoments(xvec), "gev"), np.cov = TRUE)
est_cov(parameters(TLMoments(xvec, rightrim = 1), "gev"), np.cov = TRUE)

# Matrix inputs
est_cov(parameters(TLMoments(xmat), "gev"))
est_cov(parameters(TLMoments(xmat), "gev"), select = "shape")
est_cov(parameters(TLMoments(xmat[, 1]), "gev"), select = "shape")

# Theoretical values (leftrim and/or rightrim have to be specified)
para <- as.parameters(loc = 10, scale = 5, shape = .2, distr = "gev")
est_cov(para, set.n = 100)
est_cov(para, rightrim = 1, set.n = 100)

## Not run:
var(t(replicate(10000, parameters(TLMoments(rgev(100, 10, 5, .2)), "gev"))))

## End(Not run)
## Not run:
var(t(replicate(10000, parameters(TLMoments(rgev(100, 10, 5, .2), rightrim = 1), "gev"))))

## End(Not run)

# Parameter estimates from regionalized TLMoments:
est_cov(parameters(regionalize(TLMoments(xmat), c(.75, .25)), "gev"))


### 4. Quantiles:

xvec <- rgev(100, shape = .2)
xmat <- cbind(rgev(100, shape = .1), rgev(100, shape = .3))

# Covariance estimation of parameters normally with parametric assumption:
q <- quantiles(parameters(TLMoments(xvec), "gev"), c(.9, .95, .99))
est_cov(q)
est_cov(q, select = c("0.9", "0.99"))
est_cov(q, select = .95)

# A nonparametric estimation can be enforced with np.cov:
est_cov(q, np.cov = TRUE)

# Matrix inputs
param <- parameters(TLMoments(xmat, 0, 1), "gev")
q <- quantiles(param, c(.9, .95, .99))
est_cov(q)
est_cov(q, select = .99)
param <- parameters(TLMoments(xmat[, 1, drop = FALSE], 0, 1), "gev")
q <- quantiles(param, c(.9, .95, .99))
est_cov(q, select = .99)

# Theoretical values
q <- quantiles(as.parameters(loc = 10, scale = 5, shape = .3, distr = "gev"), c(.9, .99))
est_cov(q)
```

```
est_cov(q, leftrim = 0, rightrim = 1)
est_cov(q, leftrim = 0, rightrim = 1, set.n = 100)

# Quantile estimates from regionalized TLMoments:
param <- parameters(regionalize(TLMoments(xmat), c(.75, .25)), "gev")
est_cov(quantiles(param, c(.9, .99)))
```

`est_paramcov` *Estimate the covariance matrix of parameter estimations*

Description

Internal function. Use est_cov. Description not done yet.

Usage

```
est_paramcov(x, distr = "", leftrim = 0L, rightrim = 0L, ...)

## S3 method for class 'numeric'
est_paramcov(x, distr, leftrim = 0L, rightrim = 0L, np.cov = FALSE, ...)

## S3 method for class 'matrix'
est_paramcov(
  x,
  distr,
  leftrim = 0L,
  rightrim = 0L,
  np.cov = FALSE,
  reg.weights = NULL,
  ...
)

## S3 method for class 'parameters'
est_paramcov(
  x,
  distr = attr(x, "distribution"),
  leftrim = attr(x, "source")$trimmings[1],
  rightrim = attr(x, "source")$trimmings[2],
  set.n = NA,
  ...
)
```

Arguments

x	numeric vector or matrix containing data OR an object of parameters.

distr	character indicating the distribution from which the parameters are calculated. If x is parameters-object, distr does not need to be specified.
leftrim, rightrim	lower and upper trimming parameter used for parameter calculation, have to be non-negative integers.
...	additional arguments.
np.cov	boolean, if TRUE no parametric assumptions are used to calculate the covariance matrix (default FALSE).
reg.weights	numeric vector of weights for regionalized TLMoments.
set.n	hypothetical data length n if theoretical values are given.

Value

numeric matrix

Examples

```
### Numeric vectors
x <- rgev(500, shape = .2)

parameters(TLMoments(x), "gev")
est_paramcov(x, "gev", 0, 0)
#cov(t(replicate(10000, parameters(TLMoments(rgev(500, shape = .2)), "gev"))))

parameters(TLMoments(x, rightrim = 1), "gev")
est_paramcov(x, "gev", 0, 1)
#cov(t(replicate(10000,
#   parameters(TLMoments(rgev(500, shape = .2), rightrim = 1), "gev")
#)))

parameters(TLMoments(x, rightrim = 2), "gev")
est_paramcov(x, "gev", 0, 2)
#cov(t(replicate(10000,
#  parameters(TLMoments(rgev(500, shape = .2), rightrim = 2), "gev")
#)))

### Numeric matrices
x <- matrix(rgev(600, shape = .2), nc = 3)

parameters(TLMoments(x), "gev")
est_paramcov(x, "gev", 0, 0)
#cov(t(replicate(5000,
#  as.vector(parameters(TLMoments(matrix(rgev(600, shape = .2), nc = 3)), "gev")))
#))

### parameters-object
x <- as.parameters(loc = 3, scale = 2, shape = .4, distr = "gev")
est_paramcov(x)
est_paramcov(x, leftrim = 0, rightrim = 0)
est_paramcov(x, leftrim = 0, rightrim = 0, set.n = 100)
```

```
# distr-argument can be neglected
```

`est_pwmcov` *Estimate the covariance matrix of PWM estimations*

Description

Internal function. Use est_cov. Description not done yet.

Usage

```
est_pwmcov(x, order = 0:3, distr = NULL, distr.trim = c(0, 0))

## S3 method for class 'numeric'
est_pwmcov(x, order = 0:3, distr = NULL, distr.trim = c(0, 0))

## S3 method for class 'matrix'
est_pwmcov(x, order = 0:3, distr = NULL, distr.trim = c(0, 0))
```

Arguments

`x`	numeric vector or matrix of data.
`order`	numeric vector giving the orders that are returned.
`distr`	character of length 1 which indicates a distribution if a parametric assumption should be used.
`distr.trim`	integer vector of length 2 indicating the trimming used to calculate parameters if a parametric assumption is used (i.e. `distr` is set).

Value

numeric matrix

Examples

```
### Numeric vectors
x <- rgev(500, shape = .2)
est_pwmcov(x)
est_pwmcov(x, distr = "gev")

### Numeric matrices
x <- matrix(rgev(600, shape = .2), nc = 3)
est_pwmcov(x, order = 0:2)
est_pwmcov(x, order = 0:2, distr = "gev")
```

est_quancov *Estimate the covariance matrix of quantile estimations*

Description

Internal function. Use est_cov. Description not done yet.

Usage

```
est_quancov(x, distr = "", p = NULL, leftrim = 0L, rightrim = 0L, ...)

## S3 method for class 'numeric'
est_quancov(x, distr, p, leftrim = 0L, rightrim = 0L, np.cov = FALSE, ...)

## S3 method for class 'matrix'
est_quancov(
  x,
  distr,
  p,
  leftrim = 0L,
  rightrim = 0L,
  np.cov = FALSE,
  reg.weights = NULL,
  ...
)

## S3 method for class 'quantiles'
est_quancov(
  x,
  distr = attr(x, "distribution"),
  p = attr(x, "p"),
  leftrim = attr(x, "source")$trimmings[1],
  rightrim = attr(x, "source")$trimmings[2],
  set.n = NA,
  ...
)
```

Arguments

x	numeric vector or matrix containing data.
distr	character of length 1 giving the distribution if parametric assumption should be used.
p	quantile levels from which the covariance should be calculated.
leftrim, rightrim	lower and upper trimming parameter used for parameter calculation, have to be non-negative integers.

...	additional arguments.
np.cov	boolean, if TRUE no parametric assumptions are used to calculate the covariance matrix (default FALSE).
reg.weights	numeric vector of weights for regionalized TLMoments.
set.n	hypothetical data length n if theoretical values are given.

Value

numeric matrix

Examples

```
### Numeric vectors
x <- rgev(500, shape = .2)

quantiles(parameters(TLMoments(x), "gev"), c(.9, .95, .99))
est_quancov(x, "gev", c(.9, .95, .99), 0, 0)
#cov(t(replicate(5000,
#  quantiles(parameters(TLMoments(rgev(500, shape = .2)), "gev"), c(.9, .95, .99))
#)))

quantiles(parameters(TLMoments(x, rightrim = 1), "gev"), c(.9, .95, .99))
est_quancov(x, "gev", c(.9, .95, .99), 0, 1)
#cov(t(replicate(5000,
#  quantiles(
#    parameters(TLMoments(rgev(500, shape = .2), rightrim = 1), "gev"),
#    c(.9, .95, .99)
#  )
#)))

### Numeric matrices
x <- matrix(rgev(600, shape = .2), nc = 3)

quantiles(parameters(TLMoments(x), "gev"), c(.9, .95, .99))
est_quancov(x, "gev", c(.9, .95, .99), 0, 0)

est_quancov(x, "gev", .9, 0, 0)
#cov(t(replicate(5000,
# quantiles(
#   parameters(TLMoments(matrix(rgev(600, shape = .2), nc = 3)),
#  "gev"), .9)
#  )
#))

### quantiles object
q <- quantiles(as.parameters(loc = 3, scale = 2, shape = .4, distr = "gev"), c(.9, .99))
est_quancov(q)
est_quancov(q, leftrim = 0, rightrim = 0)
est_quancov(q, leftrim = 0, rightrim = 0, set.n = 10)
```

`est_tlmcov` *Estimate the covariance matrix of TL-moments estimations*

Description

Internal function. Use est_cov. Description not done yet.

Usage

```
est_tlmcov(
  x,
  leftrim = 0L,
  rightrim = 0L,
  order = 1:3,
  distr = NULL,
  lambda.cov = TRUE,
  ratio.cov = TRUE,
  ...
)

## S3 method for class 'numeric'
est_tlmcov(
  x,
  leftrim = 0L,
  rightrim = 0L,
  order = 1:3,
  distr = NULL,
  lambda.cov = TRUE,
  ratio.cov = TRUE,
  ...
)

## S3 method for class 'matrix'
est_tlmcov(
  x,
  leftrim = 0L,
  rightrim = 0L,
  order = 1:3,
  distr = NULL,
  lambda.cov = TRUE,
  ratio.cov = TRUE,
  reg.weights = NULL,
  ...
)

## S3 method for class 'TLMoments'
est_tlmcov(
```

```
  x,
  leftrim = attr(x, "leftrim"),
  rightrim = attr(x, "rightrim"),
  order = attr(x, "order"),
  distr = NULL,
  lambda.cov = TRUE,
  ratio.cov = TRUE,
  set.n = NA,
  ...
)
```

Arguments

x	numeric vector or matrix containing data OR an object of TLMoments.
leftrim, rightrim	integer indicating lower and upper trimming parameters, have to be non-negative integers.
order	numeric vector giving the orders that are returned (default is first three L-moments).
distr	character of length 1 giving the distribution if parametric assumption should be used.
lambda.cov	boolean, if TRUE (default) TL-moment estimation covariance matrix is calculated.
ratio.cov	boolean, if TRUE (default) TL-moment-ratio estimation covariance matrix is calculated.
...	additional arguments.
reg.weights	numeric vector of weights for regionalized TLMoments.
set.n	hypothetical data length n if theoretical values are given.

Value

a list of numeric matrices (if lambda.cov and ratio.cov are TRUE (default)), or a single matrix.

Examples

```
### Numeric vectors
x <- rgev(500, loc = 10, scale = 5, shape = .1)

est_tlmcov(x)
est_tlmcov(x, order = 2:3)
est_tlmcov(x, rightrim = 1, order = 4:5)
# cov(t(replicate(10000,
#   TLMoments(rgev(500, loc = 10, scale = 5, shape = .1))$lambdas)
# ))
# cov(t(replicate(10000,
#   TLMoments(rgev(500, loc = 10, scale = 5, shape = .1))$ratios)
# ))

est_tlmcov(x, ratio.cov = FALSE)
```

```
est_tlmcov(x, lambda.cov = FALSE)

est_tlmcov(x, distr = "gev")

est_tlmcov(x, leftrim = 0, rightrim = 1)
# cov(t(replicate(10000,
#  TLMoments(rgev(500, loc = 10, scale = 5, shape = .1), 0, 1, 3)$lambdas
# )))
# cov(t(replicate(10000,
#  TLMoments(rgev(500, loc = 10, scale = 5, shape = .1), 0, 1, 3)$ratios
# )))

### Numeric matrices
x <- matrix(rgev(600), nc = 3)

est_tlmcov(x)
est_tlmcov(x, order = 3:4)
# cov(t(replicate(10000,
#   as.vector(TLMoments(matrix(rgev(600), nc = 3))$lambdas[3:4, ])
# )))
# cov(t(replicate(10000,
#   as.vector(TLMoments(matrix(rgev(600), nc = 3))$ratios[3:4, ])
# )))

est_tlmcov(x, ratio.cov = FALSE)
est_tlmcov(x, lambda.cov = FALSE)

TLMoments:::est_tlmcov(x, order = 2:3, distr = "gev")
# cov(t(replicate(10000,
#   as.vector(TLMoments(matrix(rgev(600), nc = 3))$lambdas[2:3, ])
# )))
# cov(t(replicate(10000,
#   as.vector(TLMoments(matrix(rgev(600), nc = 3))$ratios[2:3, ])
# )))

### TLMoments-object (theoretical calculation)
tlm <- TLMoments(as.parameters(loc = 10, scale = 5, shape = .1, distr = "gev"), 0, 1)
est_tlmcov(tlm, distr = "gev", set.n = 100)
est_tlmcov(tlm, distr = "gev", set.n = 100, ratio.cov = FALSE)
est_tlmcov(tlm, distr = "gev", set.n = 100, lambda.cov = FALSE)
```

`parameters` *Converting TL-moments to distribution parameters*

Description

Converts TL-moments (or PWMs) to distribution parameters. By now, conversions for gev, gumbel, gpd, and ln3 are available. Important trimming options are calculated through known formulas (see references for some of them), other options are calculated through a numerical optimization.

Usage

```
parameters(x, distr, ...)

## S3 method for class 'PWMs'
parameters(x, distr, ...)

## S3 method for class 'TLMoments'
parameters(x, distr, ...)
```

Arguments

x	object returned by TLMoments (or PWMs, in which case TLMoments with no trimming is used).
distr	character object defining the distribution. Supported types are "gev", "gum", "gpd", and "ln3".
...	additional arguments.

Value

numeric vector, matrix, list, or data.frame of parameter estimates with class `parameters`. The object contains the following attributes:

- `distribution`: a character indicating the used distribution
- `source`: a list with background information (used function, data, n, formula, trimmings; mainly for internal purposes)

The attributes are hidden in the print-function for a clearer presentation.

Methods (by class)

- `PWMs`: parameters for PWMs-object
- `TLMoments`: parameters for TLMoments-object

References

Elamir, E. A. H. (2010). Optimal choices for trimming in trimmed L-moment method. Applied Mathematical Sciences, 4(58), 2881-2890.

Fischer, S., Fried, R., & Schumann, A. (2015). Examination for robustness of parametric estimators for flood statistics in the context of extraordinary extreme events. Hydrology and Earth System Sciences Discussions, 12, 8553-8576.

Hosking, J. R. (1990). L-moments: analysis and estimation of distributions using linear combinations of order statistics. Journal of the Royal Statistical Society. Series B (Methodological), 105-124.

Hosking, J. R. M. (2007). Some theory and practical uses of trimmed L-moments. Journal of Statistical Planning and Inference, 137(9), 3024-3039.

See Also

PWMs, TLMoments, quantiles, summary.parameters, as.parameters. Built-in distributions: pgev, pgum, pgpd, pln3.

Examples

```
xmat <- matrix(rgev(100, shape = .2), nc = 4)
xvec <- xmat[, 3]
xlist <- lapply(1L:ncol(xmat), function(i) xmat[, i])
xdat <- data.frame(
 station = rep(letters[1:2], each = 50),
 season = rep(c("S", "W"), 50),
 hq = as.vector(xmat)
)

# TLMoments-objects or PWMs-objects can be used. However, in case of PWMs
# simply the TLMoments(., leftrim = 0, rightrim = 0)-variant is used.

parameters(PWMs(xvec), "gev")
tlm <- TLMoments(xvec, leftrim = 0, rightrim = 0)
parameters(tlm, "gev")

tlm <- TLMoments(xmat, leftrim = 1, rightrim = 1)
parameters(tlm, "gum")

tlm <- TLMoments(xlist)
parameters(tlm, "gpd")

tlm <- TLMoments(xdat, hq ~ station, leftrim = 0, rightrim = 2)
parameters(tlm, "gev")

tlm <- TLMoments(xdat, hq ~ station + season, leftrim = 0, rightrim = 2)
parameters(tlm, "gev")


# If no explicit formula is implemented, it is tried to calculate
# parameters numerically. The attribute source$param.computation.method
# indicates if this is the case.

tlm <- TLMoments(rgum(200, loc = 5, scale = 2), leftrim = 1, rightrim = 4)
parameters(tlm, "gum")

tlm <- TLMoments(rgev(200, loc = 10, scale = 5, shape = .4), leftrim = 2, rightrim = 2)
parameters(tlm, "gev")

tlm <- TLMoments(rln3(200, loc = 3, scale = 1.5, shape = 2), leftrim = 0, rightrim = 1)
parameters(tlm, "ln3")

# Numerical calculation is A LOT slower:
## Not run:
system.time(replicate(500,
  parameters(TLMoments(rgum(100, loc = 5, scale = 2), 1, 1), "gum")
```

```
))[3]
system.time(replicate(500,
  parameters(TLMoments(rgum(100, loc = 5, scale = 2), 1, 2), "gum")
))[3]

## End(Not run)

# Using magrittr
library(magrittr)

TLMoments(rgpd(500, loc = 10, scale = 3, shape = .3), rightrim = 0) %>%
 parameters("gpd")

TLMoments(rgpd(500, loc = 10, scale = 3, shape = .3), rightrim = 0) %>%
 parameters("gpd", u = 10)

TLMoments(rgpd(500, loc = 10, scale = 3, shape = .3), rightrim = 1) %>%
 parameters("gpd")

TLMoments(rgpd(500, loc = 10, scale = 3, shape = .3), rightrim = 2) %>%
 parameters("gpd")
```

---

pgev *Generalized Extreme Value distribution*

Description

Cumulative distribution function, density function, quantile function and generation of random variates of the generalized extreme value distribution.

Usage

```
pgev(q, loc = 0, scale = 1, shape = 0)

dgev(x, loc = 0, scale = 1, shape = 0)

qgev(p, loc = 0, scale = 1, shape = 0)

rgev(n, loc = 0, scale = 1, shape = 0)
```

Arguments

loc, scale, shape	location, scale, and shape parameter of the generalized extreme value distribution. All must be of length one.
x, q, p	numeric vector of values, quantiles, or probabilites.
n	numeric, number of random variates.

See Also

pgum, pgpd, pln3

---

pgpd *Generalized Pareto distribution*

Description

Cumulative distribution function, density function, quantile function and generation of random variates of the generalized Pareto distribution.

Usage

```
pgpd(q, loc = 0, scale = 1, shape = 0)

dgpd(x, loc = 0, scale = 1, shape = 0)

qgpd(p, loc = 0, scale = 1, shape = 0)

rgpd(n, loc = 0, scale = 1, shape = 0)
```

Arguments

loc, scale, shape	location, scale, and shape parameter of the generalized Pareto distribution. All must be of length one.
x, q, p	numeric vector of values, quantiles, or probabilites.
n	numeric, number of random variates.

See Also

pgev, pgum, pln3

---

pgum *Gumbel distribution*

Description

Cumulative distribution function, density function, quantile function and generation of random variates of the Gumbel distribution.

Usage

```
pgum(q, loc = 0, scale = 1)

dgum(x, loc = 0, scale = 1)

qgum(p, loc = 0, scale = 1)

rgum(n, loc = 0, scale = 1)
```

Arguments

loc, scale	location and scale parameter of the Gumbel distribution. All must be of length one.
x, q, p	numeric vector of values, quantiles, or probabilites.
n	numeric, number of random variates.

See Also

pgev, pgpd, pln3

---

pln3 *Three-parameter lognormal distribution*

Description

Cumulative distribution function, density function, quantile function and generation of random variates of the three-parameter lognormal distribution.

Usage

```
pln3(q, loc = 0, scale = 1, shape = 1)

dln3(x, loc = 0, scale = 1, shape = 1)

qln3(p, loc = 0, scale = 1, shape = 1)

rln3(n, loc = 0, scale = 1, shape = 1)
```

Arguments

loc, scale, shape	location, scale, and shape parameter of the three-parameter lognormal distribution. All must be of length one.
x, q, p	numeric vector of values, quantiles, or probabilites.
n	numeric, number of random variates.

See Also

pgev, pgum, pgpd

---

plot.TLMoments *L-Moment-ratio diagram*

Description

Generates a ggplot2 object containing a scatterplot of TL skewness and TL kurtosis as well as the theoretical curves and points of several distributions (for now: GEV, GPD, LN3, GUM, EXP, NORM).

Usage

```
## S3 method for class 'TLMoments'
plot(x, ...)

## S3 method for class 'numeric'
plot.TLMoments(x, distr = "all", add_center = FALSE, use_internal = TRUE, ...)

## S3 method for class 'matrix'
plot.TLMoments(x, distr = "all", add_center = TRUE, use_internal = TRUE, ...)

## S3 method for class 'list'
plot.TLMoments(x, distr = "all", add_center = TRUE, use_internal = TRUE, ...)

## S3 method for class 'data.frame'
plot.TLMoments(x, distr = "all", add_center = TRUE, use_internal = TRUE, ...)
```

Arguments

x	object of TLMoments.
...	additional arguments, not used at the moment.
distr	character indicating the plotted theoretical distributions, see details.
add_center	boolean, if TRUE (default, except for vector TLMoments) the center of all TL-moment ratios is printed as a cross.
use_internal	boolean, if TRUE (default) internal pre-calculated values (if available) are used to print curves and points.

Details

distr: this can either be a vector containing the abbreviations of the theoretical distributions (gev, gpd, ln3, pe3, glo, gum, exp, or norm) or one of the shortcuts \"all\" (default), \"only-lines\", or \"only-points\" that indicate all distributions, all distributions displayed as lines (i.e. gev, gpd, ln3, pe3, glo), or all distributions displayed as points (ie. gum, exp, norm), respectively.

Values of theoretical distributions are pre-calculated for several trimmings. If other trimmings are selected this results in a (small) delay for calculation.

Value

A ggplot object.

Methods (by class)

- `numeric`: plot.TLMoments for numeric vector
- `matrix`: plot.TLMoments for numeric matrix
- `list`: plot.TLMoments for numeric list
- `data.frame`: plot.TLMoments for numeric data.frame

Examples

```
## Not run:
xmat <- matrix(rgev(1000, shape = .1), nc = 10)
xvec <- xmat[, 3]
xlist <- lapply(1L:ncol(xmat), function(i) xmat[, i])
xdat <- data.frame(
 station = rep(letters[1:10], each = 100),
 season = rep(c("S", "W"), 50),
 hq = as.vector(xmat)
)

library(ggplot2)
plot(TLMoments(xvec))
plot(TLMoments(xlist), distr = c("gev", "gum"), add_center = FALSE)
plot(TLMoments(xmat), distr = "only-points")
plot(TLMoments(xmat), distr = "only-lines") + scale_colour_viridis_d()
plot(TLMoments(xmat, 0, 1))
plot(TLMoments(xmat, 0, 1)) + coord_cartesian(xlim = c(-.05, .4), ylim = c(.05, .2))
plot(TLMoments(xdat, hq ~ station, 1, 0))

plot(TLMoments(xmat), add_center = FALSE)
plot(TLMoments(xmat), use_internal = FALSE)
plot(TLMoments(xmat, 2, 3))

## End(Not run)
```

PWM *Probability weighted moments*

Description

Calculates empirical probability weighted moments of specific order(s).

Usage

```
PWM(x, order = 0, na.rm = FALSE)
```

Arguments

x	numeric vector of data.
order	integer, order of probability weighted moment, can be a set of {0,1,...}.
na.rm	logical, indicates if NAs should be removed.

Value

numeric vector, empirical PWM of orders order of x.

See Also

PWMs

Examples

```
PWM(rnorm(25))
PWM(rnorm(25), order = 2)
PWM(rnorm(25), order = c(0, 2, 4))
```

PWMs *Probability weighted moments*

Description

Calculates probability weighted moments up to a specific order. Note that PWMs start with order 0. Acceptable input types are numeric vectors, matrices, lists, and data.frames.

Usage

```
PWMs(x, ...)

## S3 method for class 'numeric'
PWMs(x, max.order = 4L, na.rm = FALSE, ...)

## S3 method for class 'matrix'
PWMs(x, max.order = 4L, na.rm = FALSE, ...)

## S3 method for class 'list'
PWMs(x, max.order = 4L, na.rm = FALSE, ...)

## S3 method for class 'data.frame'
PWMs(x, formula, max.order = 4L, na.rm = FALSE, ...)

## S3 method for class 'TLMoments'
PWMs(x, ...)
```

Arguments

x	numeric vector or matrix, list, or data.frame of data OR an object of TLMoments.
...	additional arguments.
max.order	integer, maximal order of PWMs.
na.rm	logical, indicates if NAs should be removed.
formula	if x is of type data.frame a formula has to be submitted.

Value

numeric vector, matrix, list, or data.frame consisting of the PWMs and with class PWMs. The object contains the following attributes:

- order: a integer vector with corresponding PWM orders
- source: a list with background information (used function, data, n, formula; mainly for internal purposes)

The attributes are hidden in the print-function for a clearer presentation.

References

Greenwood, J. A., Landwehr, J. M., Matalas, N. C., & Wallis, J. R. (1979). Probability weighted moments: definition and relation to parameters of several distributions expressable in inverse form. Water Resources Research, 15(5), 1049-1054.

Examples

```
# Generating data sets:
xmat <- matrix(rnorm(100), nc = 4)
xvec <- xmat[, 3]
xlist <- lapply(1L:ncol(xmat), function(i) xmat[, i])
xdat <- data.frame(
 station = rep(letters[1:2], each = 50),
 season = rep(c("S", "W"), 50),
 hq = as.vector(xmat)
)

# Calculating PWMs from data:
PWMs(xvec)
PWMs(xmat)
PWMs(xlist)
PWMs(xdat, formula = hq ~ station)
PWMs(xdat, formula = hq ~ season)
PWMs(xdat, formula = hq ~ .)
PWMs(xdat, formula = . ~ station + season)

# Calculating PWMs from L-moments:
PWMs(TLMoments(xvec))
PWMs(TLMoments(xmat))
PWMs(TLMoments(xlist))
```

```
PWMs(TLMoments(xdat, hq ~ station))
PWMs(TLMoments(xdat, hq ~ season))
PWMs(TLMoments(xdat, hq ~ .))
PWMs(TLMoments(xdat, . ~ station + season))

# In data.frame-mode invalid names are preceded by "."
xdat <- data.frame(
 beta0 = rep(letters[1:2], each = 50),
 beta1 = as.vector(xmat)
)
PWMs(xdat, formula = beta1 ~ beta0)
```

`quantiles` *Calculating quantiles from distribution parameters*

Description

Calculates quantiles from distribution parameters received by parameters or from a named vector.

Usage

```
quantiles(x, p, distr = attr(x, "distribution"))
```

Arguments

`x`	object returned by parameters or a named vector. In the latter you have to specify the `distr`-argument.
`p`	numeric vector giving the quantiles to calculate.
`distr`	character object defining the distribution. Supported types are "gev", "gum" and "gpd". You do not need to set this, if `x` is from parameters.

Value

numeric vector, matrix, list, or data.frame of the quantiles and with class `quantiles`. The object contains the following attributes:

- `distribution`: a character indicating the used distribution
- p: a vector with the calculated quantiles
- `source`: a list with background information (used function, data, n, formula, trimmings; mainly for internal purposes)

The attributes are hidden in the print-function for a clearer presentation.

See Also

`PWMs`, `TLMoments`, `parameters`, `summary.quantiles`

Examples

```
# Generating data sets:
xmat <- matrix(rnorm(100), nc = 4)
xvec <- xmat[, 3]
xlist <- lapply(1L:ncol(xmat), function(i) xmat[, i])
xdat <- data.frame(
 station = rep(letters[1:2], each = 50),
 season = rep(c("S", "W"), 50),
 hq = as.vector(xmat)
)

# Calculating quantiles from parameters-object
tlm <- TLMoments(xvec, leftrim = 0, rightrim = 1)
quantiles(parameters(tlm, "gev"), c(.9, .99))
tlm <- TLMoments(xmat, leftrim = 1, rightrim = 1)
quantiles(parameters(tlm, "gum"), c(.9, .95, .999))
tlm <- TLMoments(xlist)
quantiles(parameters(tlm, "gpd"), .999)
tlm <- TLMoments(xdat, hq ~ station, leftrim = 2, rightrim = 3)
quantiles(parameters(tlm, "gev"), seq(.1, .9, .1))
tlm <- TLMoments(xdat, hq ~ station + season, leftrim = 0, rightrim = 2)
quantiles(parameters(tlm, "gum"), seq(.1, .9, .1))

# Distribution can be overwritten (but parameters have to fit)
tlm <- TLMoments(xvec, leftrim = 0, rightrim = 1)
params <- parameters(tlm, "gev")
quantiles(params, c(.9, .99))
quantiles(params[1:2], c(.9, .99), distr = "gum")
evd::qgumbel(c(.9, .99), loc = params[1], scale = params[2])


# Using magrittr
library(magrittr)
rgev(50, shape = .3) %>%
  TLMoments(leftrim = 0, rightrim = 1) %>%
  parameters("gev") %>%
  quantiles(c(.99, .999))

# Calculating quantiles to given parameters for arbitrary functions
quantiles(c(mean = 10, sd = 3), c(.95, .99), "norm")
qnorm(c(.95, .99), mean = 10, sd = 3)

# These give errors:
#quantiles(c(loc = 10, scale = 5, shape = .3), c(.95, .99), "notexistingdistribution")
#quantiles(c(loc = 10, scale = 5, shpe = .3), c(.95, .99), "gev") # wrong arguments
```

`regionalize` *Calculation of regionalized TL-moments*

Description

regionalize takes the result of TLMoments and calculates a weighted mean of TL-moments and TL-moment ratios.

Usage

```
regionalize(x, ...)

## S3 method for class 'numeric'
regionalize(x, ...)

## S3 method for class 'matrix'
regionalize(x, w = attr(x, "source")$n, reg.lambdas = TRUE, ...)

## S3 method for class 'data.frame'
regionalize(x, w = attr(x, "source")$n, reg.lambdas = TRUE, ...)

## S3 method for class 'list'
regionalize(x, w = attr(x, "source")$n, reg.lambdas = TRUE, ...)
```

Arguments

x	object returned by TLMoments.
...	additional arguments, not used at the moment.
w	numeric vector giving the weights. Default: Sample lengths of corresponding data. Internally scaled so that it adds up to 1.
reg.lambdas	logical, if TRUE (default) regionalization is based upon TL-moments. If false it's based on TL-moment-ratios.

Value

list of two dimensions: lambdas/ratios are numeric vectors consisting of the regionalized TL-moments/TL-moment-ratios. The list has the class TLMoments. The object contains the following attributes:

- leftrim: a numeric giving the used leftrim-argument
- rightrim: a numeric giving the used rightrim-argument
- order: a integer vector with corresponding TL-moment orders
- source: a list with background information (used function, data, n, formula, computation.method; mainly for internal purposes)

Examples

```
xmat <- matrix(rgev(100), nc = 4)
xvec <- xmat[, 3]
xlist <- lapply(1L:ncol(xmat), function(i) xmat[, i])
xdat <- data.frame(
 station = rep(letters[1:2], each = 50),
```

```
 season = rep(c("S", "W"), 50),
 hq = as.vector(xmat)
)

regionalize(TLMoments(xmat))
regionalize(TLMoments(xlist))
regionalize(TLMoments(xdat, hq ~ station))
# For numeric vector TLMoments, nothing happens:
regionalize(TLMoments(xvec))

tlm <- TLMoments(xmat)
regionalize(tlm)
regionalize(tlm, reg.lambdas = FALSE)

parameters(regionalize(tlm), "gev")
parameters(regionalize(tlm, reg.lambdas = FALSE), "gev")

quantiles(parameters(regionalize(tlm), "gev"), c(.99, .999))
quantiles(parameters(regionalize(tlm, reg.lambdas = FALSE), "gev"), c(.99, .999))


# With magrittr
library(magrittr)
matrix(rgev(200, shape = .3), nc = 5) %>%
 TLMoments(rightrim = 1) %>%
 regionalize %>%
 parameters("gev") %>%
 quantiles(c(.99, .999))
```

`returnParameters` *returnParameters*

Description

Sets attributions to parameters objects and returns them. This function is for internal use.

Usage

```
returnParameters(out, distribution, ...)
```

Arguments

`out`	-
`distribution`	-
`...`	-

Value

An object of class parameters.

returnPWMs *returnPWMs*

Description

Sets attributes to PWMs objects and returns them. This function is for internal use.

Usage

```
returnPWMs(out, order, ...)
```

Arguments

out	-
order	-
...	-

Value

An object of class PWMs.

returnQuantiles *returnQuantiles*

Description

Sets attributions to quantiles objects and returns them. This function is for internal use.

Usage

```
returnQuantiles(out, distribution, p, ...)
```

Arguments

out	-
distribution	-
p	-
...	-

Value

An object of class quantiles.

returnTLMoments *returnTLMoments*

Description

Sets attributions to TLMoments objects and returns them. This function is for internal use.

Usage

```
returnTLMoments(out, leftrim, rightrim, order, ...)
```

Arguments

out	-
leftrim	-
rightrim	-
order	-
...	-

Value

An object of class TLMoments.

summary.parameters *Summary parameters*

Description

Calculating and printing of summary statistics to a given parameters-object.

Usage

```
## S3 method for class 'parameters'
summary(object, ci.level = 0.9, ...)
```

Arguments

object	object of parameters.
ci.level	numeric vector of length 1 giving the confidence level (default is 0.9).
...	additional arguments submitted to est_cov.

Value

A summary.parameters-object, a list with dimensions

- param
- ci.level
- ci
- cov

It is printed with print.summary.parameters.

See Also

parameters, est_cov

Examples

```
x <- cbind(rgev(100, shape = .2), rgev(100, shape = .2))

p <- parameters(TLMoments(x[, 1]), "gev")
summary(p)
summary(p, select = c("scale", "shape"))

p <- parameters(TLMoments(x[, 1], rightrim = 1), "gev")
summary(p)

p <- parameters(TLMoments(x), "gev")
summary(p)
summary(p, select = "shape")

p <- as.parameters(loc = 10, scale = 5, shape = .3, distr = "gev")
summary(p)
summary(p, rightrim = 1, set.n = 250)
```

---

summary.PWMs *Summary PWMs*

Description

Calculating and printing of summary statistics to a given PWMs-object.

Usage

```
## S3 method for class 'PWMs'
summary(object, ci.level = 0.9, ...)
```

Arguments

object	object of PWMs.
ci.level	numeric vector of length 1 giving the confidence level (default is 0.9).
...	additional arguments submitted to est_cov.

Value

A summary.PWMs-object, a list with dimensions

- pwm
- ci.level
- ci
- cov

It is printed with print.summary.PWMs.

See Also

PWMs, est_cov

Examples

```
x <- cbind(rgev(100, shape = .2), rgev(100, shape = .2))

summary(PWMs(x[, 1]))
summary(PWMs(x[, 1]), distr = "gev")
summary(PWMs(x[, 1]), distr = "gev", select = 1:2)

summary(PWMs(x))
summary(PWMs(x), select = 1:2)

## Not run:
summary(as.PWMs(c(15, 4, .5)))

## End(Not run)
```

---

summary.quantiles *Summary quantiles*

Description

Calculating and printing of summary statistics to a given quantiles-object.

Usage

```
## S3 method for class 'quantiles'
summary(object, ci.level = 0.9, ...)
```

Arguments

object	object of quantiles.
ci.level	numeric vector of length 1 giving the confidence level (default is 0.9).
...	additional arguments submitted to est_cov.

Value

A summary.quantiles-object, a list with dimensions

- q
- ci.level
- ci
- cov

It is printed with print.summary.quantiles.

See Also

quantiles, est_cov

Examples

```
x <- cbind(rgev(100, shape = .2), rgev(100, shape = .2))

q <- quantiles(parameters(TLMoments(x[, 1]), "gev"), c(.9, .95, .99))
summary(q)
summary(q, select = c(.9, .99))

q <- quantiles(parameters(TLMoments(x[, 1], rightrim = 1), "gev"), .95)
summary(q)

q <- quantiles(parameters(TLMoments(x), "gev"), c(.9, .95, .99))
summary(q)
summary(q, select = .95)

q <- quantiles(as.parameters(loc = 10, scale = 5, shape = .3, distr = "gev"), c(.9, .99))
summary(q)
summary(q, rightrim = 1, set.n = 250)
```

---

summary.TLMoments *Summary TLMoments*

Description

Calculating and printing of summary statistics to a given TLMoments-object.

Usage

```
## S3 method for class 'TLMoments'
summary(object, ci.level = 0.9, ...)
```

Arguments

object	object of TLMoments.
ci.level	numeric vector of length 1 giving the confidence level (default is 0.9).
...	additional arguments submitted to est_cov.

Value

A summary.TLMoments-object, a list with dimensions

- tlm
- ci.level
- lambda.ci
- lambda.cov
- ratio.ci
- ratio.cov

It is printed with print.summary.TLMoments.

See Also

TLMoments, est_cov

Examples

```
tlm <- TLMoments(rgev(100, shape = .2))
summary(tlm)

tlm <- TLMoments(rgev(100, shape = .2), rightrim = 1)
summary(tlm, select = 3:4)

tlm <- TLMoments(rgev(100, shape = .2), max.order = 2, rightrim = 1)
summary(tlm)

tlm <- TLMoments(matrix(rgev(100, shape = .2), nc = 2))
summary(tlm, select = 3:4)

tlm <- TLMoments(matrix(rgev(100, shape = .2), nc = 2), max.order = 3)
summary(tlm, ci = .95, distr = "gev")

tlm <- as.TLMoments(c(15, 5, 1.3))
summary(tlm, distr = "gev", set.n = 100)
```

TLMoment *Trimmed L Moments*

Description

Calculates empirical Trimmed L-moments of specific order(s) and trimming.

Usage

```
TLMoment(
  x,
  order = 1L,
  leftrim = 0L,
  rightrim = 0L,
  na.rm = FALSE,
  computation.method = "auto"
)
```

Arguments

x	numeric vector of data.
order	integer, order of Trimmed Moment, has to be greater than 1.
leftrim, rightrim	integer indicating trimming parameters, have to be greater than 0.
na.rm	logical, indicates if NAs should be removed.
computation.method	character, indicating if the computation is performed via PWMs, direct, recursive, or recurrence (see References Hosking & Balakrishnan, 2015). Possible values are auto (default, automatically choose appropriate method), pwm, direct, recursive, or recurrence. Only if empirical moments are calculated.

Value

numeric vector, empirical TL(leftrim,rightrim)-moments of orders order of x

References

Elamir, E. A., & Seheult, A. H. (2003). Trimmed L-moments. Computational Statistics & Data Analysis, 43(3), 299-314.

Hosking, J. R. (1990). L-moments: analysis and estimation of distributions using linear combinations of order statistics. Journal of the Royal Statistical Society. Series B (Methodological), 105-124.

Hosking, J. R. M. (2007). Some theory and practical uses of trimmed L-moments. Journal of Statistical Planning and Inference, 137(9), 3024-3039.

Hosking, J. R. M., & Balakrishnan, N. (2015). A uniqueness result for L-estimators, with applications to L-moments. Statistical Methodology, 24, 69-80.

See Also

TLMoments

Examples

```
x <- rnorm(100)
TLMoment(x, order = 1)
TLMoment(x, order = 2, leftrim = 0, rightrim = 1)
TLMoment(x, order = c(1, 2, 3), leftrim = 2, rightrim = 2)
TLMoment(x, order = c(1, 3, 2), leftrim = 2, rightrim = 2)
```

TLMoments *Trimmed L-moments*

Description

Calculates empirical or theoretical Trimmed L-moments and -ratios up to a specific order. If empirical moments should be calculated, acceptable input types are numeric vectors, matrices, lists, data.frames. TLMoments is type-preservative, so the input type is also the output type. If theoretical moments should be calculated, the input type has to be of class parameters or PWMs, so an object returned by parameters, as.parameters or PWMs, as.PWMs.

Usage

```
TLMoments(x, ...)

## S3 method for class 'numeric'
TLMoments(
  x,
  leftrim = 0L,
  rightrim = 0L,
  max.order = 4L,
  na.rm = FALSE,
  computation.method = "auto",
  ...
)

## S3 method for class 'matrix'
TLMoments(
  x,
  leftrim = 0L,
  rightrim = 0L,
  max.order = 4L,
  na.rm = FALSE,
  computation.method = "auto",
  ...
)
```

```
## S3 method for class 'list'
TLMoments(
  x,
  leftrim = 0L,
  rightrim = 0L,
  max.order = 4L,
  na.rm = FALSE,
  computation.method = "auto",
  ...
)

## S3 method for class 'data.frame'
TLMoments(
  x,
  formula,
  leftrim = 0L,
  rightrim = 0L,
  max.order = 4L,
  na.rm = FALSE,
  computation.method = "auto",
  ...
)

## S3 method for class 'PWMs'
TLMoments(x, leftrim = 0L, rightrim = 0L, ...)

## S3 method for class 'parameters'
TLMoments(
  x,
  leftrim = attr(x, "source")$trimmings[1],
  rightrim = attr(x, "source")$trimmings[2],
  max.order = 4L,
  ...
)
```

Arguments

x	numeric data in form of vector, matrix, list, or data.frame OR an object of class parameters or PWMs.
...	additional arguments.
leftrim	integer indicating lower trimming parameter, has to be greater than 0.
rightrim	integer indicating upper trimming parameter, has to be greater than 0.
max.order	integer, maximum order of Trimmed L-moments/ratios, has to be greater than 1.
na.rm	logical, indicates if NAs should be removed. Only if empirical moments are calculated.

`computation.method`
: character, indicating if the computation is performed via PWMs, direct, recursive, or recurrence (see References Hosking & Balakrishnan, 2015). Possible values are `auto` (default, automatically choose appropriate method), `pwm`, `direct`, `recursive`, or `recurrence`. Only if empirical moments are calculated.

`formula`
: if `x` is data.frame. See examples.

Value

list of two dimensions: `lambdas`/`ratios` are a numeric vector, matrix, list, or data.frame consisting of the TL-moments/TL-moment-ratios. The list has the class `TLMoments`. The object contains the following attributes:

- `leftrim`: a numeric giving the used leftrim-argument
- `rightrim`: a numeric giving the used rightrim-argument
- `order`: a integer vector with corresponding TL-moment orders
- `source`: a list with background information (used function, data, n, formula, computation method; mainly for internal purposes)

The attributes are hidden in the print-function for a clearer presentation.

Methods (by class)

- `numeric`: TLMoments for numeric vector of data
- `matrix`: TLMoments for numeric matrix of data
- `list`: TLMoments for numeric list of data
- `data.frame`: TLMoments for numeric data.frame of data
- `PWMs`: TLMoments for PWMs-object
- `parameters`: TLMoments for parameters-object

References

Elamir, E. A., & Seheult, A. H. (2003). Trimmed L-moments. Computational Statistics & Data Analysis, 43(3), 299-314.

Hosking, J. R. M. (1990). L-moments: analysis and estimation of distributions using linear combinations of order statistics. Journal of the Royal Statistical Society. Series B (Methodological), 105-124.

Hosking, J. R. M. (2007). Some theory and practical uses of trimmed L-moments. Journal of Statistical Planning and Inference, 137(9), 3024-3039.

Hosking, J. R. M., & Balakrishnan, N. (2015). A uniqueness result for L-estimators, with applications to L-moments. Statistical Methodology, 24, 69-80.

See Also

`PWMs`, `parameters`, `quantiles`, `summary.TLMoments`, `as.TLMoments`

Examples

```
# Generating data sets:
xmat <- matrix(rnorm(100), nc = 4)
xvec <- xmat[, 3]
xlist <- lapply(1L:ncol(xmat), function(i) xmat[, i])
xdat <- data.frame(
 station = rep(letters[1:2], each = 50),
 season = rep(c("S", "W"), 50),
 hq = as.vector(xmat)
)

# Calculating TL-moments from data:
TLMoments(xvec, leftrim = 0, rightrim = 1)
TLMoments(xmat, leftrim = 1, rightrim = 1)
TLMoments(xlist, max.order = 7)
TLMoments(xdat, hq ~ station, leftrim = 0, rightrim = 2)
TLMoments(xdat, hq ~ season, leftrim = 0, rightrim = 2)
TLMoments(xdat, hq ~ ., leftrim = 0, rightrim = 2)

# Calculating TL-moments from PWMs:
TLMoments(PWMs(xvec))
TLMoments(PWMs(xmat), rightrim = 1)
TLMoments(PWMs(xlist), leftrim = 1, rightrim = 1)
TLMoments(PWMs(xdat, hq ~ station), leftrim = 0, rightrim = 2)
TLMoments(PWMs(xdat, hq ~ station + season), leftrim = 0, rightrim = 2)
TLMoments(as.PWMs(cbind(c(0.12, .41, .38, .33), c(.05, 0.28, .25, .22))), 0, 1)

# Calculating TL-moments from parameters:
(tlm <- TLMoments(xmat, leftrim = 0, rightrim = 1))
TLMoments(parameters(tlm, "gev"))

(tlm <- TLMoments(xdat, hq ~ station, leftrim = 0, rightrim = 2))
TLMoments(parameters(tlm, "gev"))

p <- as.parameters(loc = 3, scale = 2, shape = .4, distr = "gev")
TLMoments(p, rightrim = 1)

p <- as.parameters(cbind(loc = 10, scale = 4, shape = seq(0, .4, .1)), distr = "gev")
TLMoments(p, max.order = 6)

p <- as.parameters(list(
 list(loc = 3, scale = 2, shape = .4),
 list(loc = 3, scale = 2, shape = .2)
), distr = "gev")
TLMoments(p)

p <- as.parameters(data.frame(
 station = letters[1:2],
 loc = c(2, 3),
 scale = c(2, 2),
 shape = c(.4, .2)
), .~station, distr = "gev")
```

```
TLMoments(p)
```

Index