Package 'IDF'

October 4, 2017

Type Package

Title Estimation and Plotting of IDF Curves

Version 1.1

Date 2017-09-29

Author Christoph Ritschel, Carola Detring, Sarah Joedicke

Maintainer Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

Description Intensity-duration-frequency (IDF) curves are a widely used analysis-tool in hydrology to assess extreme values of precipitation [e.g. Mailhot et al., 2007, <doi:10.1016/j.jhydrol.2007.09.019>]. The package 'IDF' provides a function to read precipitation data from German weather service (DWD) 'webwerdis' <http://www.dwd.de/EN/ourservices/webwerdis/webwerdis.html> files and Berlin station data from 'Stadtmessnetz' <http://www.geo.fu-berlin.de/en/met/service/stadtmessnetz/index.html> files, and additionally IDF parameters can be estimated also from a given data.frame containing a precipitation time series. The data is aggregated to given levels yearly intensity maxima are calculated either for the whole year or given months. From these intensity maxima IDF parameters are estimated on the basis of a duration-dependent generalised extreme value distribution [Koutsoyannis et al., 1998, <doi:10.1016/S0022-1694(98)00097-3>]. IDF curves based on these estimated parameters can be plotted.

Depends stats4, evd, ismev

License GPL (>= 2)

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-10-04 10:55:15 UTC

R topics documented:

`dgev.d` *Density function of modified generalized extreme value distribution*

Description

The function `dgev.d` is a modified version of the function `dgev` for different durations `d` developed by Koutsoyiannis et al. (1998).

Usage

```
dgev.d(q, mu = 0, sigma = 1, xi = 0, theta = 0, eta = 1, d = 1,
  log = FALSE)
```

Arguments

`q`	Vector of quantiles
`mu`	location value
`sigma`	scale value
`xi`	shape value
`theta`	value defining the curvature of the IDF
`eta`	value defining the slope of the IDF
`d`	vector of durations
`log`	`logical` option to use logarithmic parameter values, default=FALSE

Value

dgev.d gives the density function

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

See Also

`dgev`

fit.fun *Fitting function to optimize IDF model parameters*

Description

The function fit.fun fits IDF model parameters mu,sigma,xi,theta,eta to a set of given observations obs, typically a series of yearly maxima at different durations d. Options for using logarithmic parameter values and debugging are given. Also the optim parameters method and upper,lower can be defined.

Usage

```
fit.fun(obs, dur, mu = 1, sigma = 1, xi = 0.5, theta = 1, eta = 1,
  use.log = F, DEBUG = F, method = "Nelder-Mead", upper = Inf,
  lower = -Inf)
```

Arguments

obs	vector of yearly intensity maxima at different durations. Order: Y1D1, Y2D1,...,YnD1,Y1D2,...YnD2,Y1
dur	vector of durations with same length as obs. Order: n x D1, n x D2, ... n x Dk
mu	location value
sigma	scale value
xi	shape value
theta	value defining the curvature of the IDF
eta	value defining the slope of the IDF
use.log	logical value for usage of logarithmic values, default is FALSE
DEBUG	logical value for usage of debugging, if TRUE the input parameters and the value of negative log-likelihood are printed on console for each iteration during optimization.
method	character defining the method to be used in optim, preferences are: "Nelder-Mead", "BFGS", "L-BFGS-B"e
upper	vector specifying the upper boundary of parameters for "L-BFGS-B" method
lower	vector specifying the lower boundary of parameters for "L-BFGS-B" method

Value

$min value of negative log-likelihood at optimization minimum

$par vector of IDF parameters at optimization minimum

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

IDF.fit *Fitting IDF model parameters to observations at different durations*

Description

The function IDF.fit fits the IDF model parameters mu,sigma,xi,eta,theta to a data.frame of observations data with temporal inforamtion (at least years) and values of precipitation at a given temporal resoultion. This precipitation time series gets aggregated at given aggregation levels. agg.lev and yearly maxima of intensity are caluclated for a specific month or the whole year/dataset. The starting values of the IDF model parameters can be determined by the user as well as specific options to use during optimization. Logartihmic transformation, debugging, the optimization method, and an option to plot the IDF curves.

Usage

```
IDF.fit(data, agg.lev = c(2, 3, 6, 12, 24, 48, 72, 96), month = "all",
  moving.sum = "FALSE", theta.init = 0, use.log = FALSE, DEBUG = FALSE,
  method = "Nelder-Mead", upper = Inf, lower = -Inf, plot = FALSE,
  probs = c(0.5, 0.9, 0.99), cols = c(rgb(1, 0, 0, 1), rgb(0, 1, 0, 1),
  rgb(0, 0, 1, 1)), station.name = "Berlin", data.name = "obs")
```

Arguments

data	a data,frame, preferably generated by function IDF.read. It should at least contain a $RR and $year element for the function tow work properly.
agg.lev	a vector of aggregation levels used to fit the IDF curves.
month	integer value specifying the month to be used for estimating the IDF parameters. Type "all" for all months or if the whole time series should be fitted.
moving.sum	logical specifying if moving sum filtering should be applied for time series aggregation.
theta.init	inital value defining the curvature of the IDF, default is zero, it is not recommended to change it
use.log	logical value for usage of logarithmic values, default is FALSE
DEBUG	logical value for usage of debugging, if TRUE the input parameters and the value of negative
method	character defining the method to be used in optim, preferences are: "Nelder-Mead", "BFGS", "L-BFGS-B"e
upper	vector specifying the upper boundary of parameters for "L-BFGS-B" method
lower	vector specifying the lower boundary of parameters for "L-BFGS-B" method
plot	logical option of creating a plot of IDF curves with estimated parameters.
probs	a vector of probabilities for which the IDF curves are calculated
cols	a vector of colors for the seperate IDF curves, needs same length as probs
station.name	character overall naming of the IDF plot, e.g. name of location or model name
data.name	character naming the data points, e.g. obs or model name

Value

$ints vector of sorted intensities for selected aggregation levels

$durs vector of sorted aggregation levels

$min minimum value of negative log-likelihood during optimization

$par vector of estimated IDF model parameters mu,sigma,xi,theta,eta at minimum value of negative log-likelihood.

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

Examples

```
RR <- rgamma(10*30*24,shape=1)
year <- sort(rep(1:(10),30*24))
data <- data.frame(RR,year)
fit <- IDF.fit(data)
pars <- fit$par
```

IDF.nll *Negativ log-likelihood of modified GEV*

Description

The function IDF.nll calculates the negative log-likelihood for a given set of model parameters mu,sigma,xi,theta,eta, given observations x and given durations d. Options for the usage of logartihmic values use.log and a debugging function DEBUG are available.

Usage

```
IDF.nll(mu = 0, sigma = 1, xi = 0, theta = 0, eta = 1, x, d,
  use.log = F, DEBUG = F)
```

Arguments

mu	location value
sigma	scale value
xi	shape value
theta	value defining the curvature of the IDF
eta	value defining the slope of the IDF
x	vector of observations at different durations d
d	vector of durations
use.log	logical value for usage of logarithmic values, default is FALSE
DEBUG	logical value for usage of debugging, if TRUE the input parameters and the value of negative log-likelihood are printed on console.

Value

retruns weightes negative log-likelihood by number of observatons uesd

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

IDF.plot *Plotting IDF curves*

Description

The function IDF.plot plots a set of IDF curves with given IDF model parameters pars for several probability levels probs at given durations dur. The colors of the curves can be defined with parameter cols (need to have same length as probs). The station.name will be printed in the legend.

Usage

```
IDF.plot(pars, probs = c(0.5, 0.9, 0.99), dur = c(0.5, 1, 2, 3, 6, 12, 24,
  48, 72, 96), cols = c(rgb(1, 0, 0, 1), rgb(0, 1, 0, 1), rgb(0, 0, 1, 1)),
  st.name = "Berlin-Dahlem", dt.name = "obs", ints = NA, ds = NA)
```

Arguments

pars	a vector of IDF model parameters mu,sigma,xi,eta,theta
probs	a vector of probabilities for which the IDF curves are calculated
dur	a vector of durations at which the IDF curves are calculated
cols	a vector of colors for the seperate IDF curves, needs same length as probs
st.name	character overall naming of the IDF plot, e.g. name of location or model name
dt.name	character naming the data points, e.g. obs or model name
ints	vector of observational intensities (surted by durations)
ds	vector of durations (same length as intensities)

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

Examples

```
RR <- rgamma(10*30*24,shape=1)
year <- sort(rep(1:(10),30*24))
data <- data.frame(RR,year)
fit <- IDF.fit(data)
param <- fit$par
IDF.plot(pars=param,st.name="example",dt.name="rgamma")
```

IDF.read *Reading precipitation data*

Description

The function IDF.read reads a file in table format and creates a data.frame from it and adds some attributes (station information, aggregation time, data source). The only data values used are: date, precipitation The data.frame will have the following format: | year | mon | day | hour | min | RR | |——+——+——+——+——+——-+ | | | | | | |

Usage

```
IDF.read(file, type)
```

Arguments

file	a character string naming the file from which the data is to be read.
type	a character string defining the type of data to be read: either "stadtmessnetz" or "webwerdis", depending on if the data comes from the Stadtmessnetz Berlin or WebWerdis. If type = "webwerdis", the data will be read, then sorted, formatted and missing lines added, while if type = "stadtmessnetz", the data will just be read and formatted. Both source types have a different layout in the original file.

Details

This function is designed to prepare a data file for doing an estimation on IDF parameters in function IDF.fit. The time given in the data is the end time, so the precipitation was measured up to that time.

Value

Liste a data.frame of date and time information and precipitation values for each time step

Author(s)

Sarah Joedicke <sarah.joedicke@fu-berlin.de>

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

See Also

read.table, IDF.fit

qgev.d *Quantile function of modified generalized extreme value distribution*

Description

The function qgev.d is a modified version of the function qgev for different durations d developed by Koutsoyiannis et al. (1998).

Usage

```
qgev.d(p, mu = 0, sigma = 1, xi = 0, theta = 0, eta = 1, d = 1,
  lower.tail = TRUE)
```

Arguments

p	Vector of probabilities
mu	location value
sigma	scale value
xi	shape value
theta	value defining the curvature of the IDF
eta	value defining the slope of the IDF
d	vector of durations
lower.tail	logical if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]

Value

qgev.d gives the quantile function

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

See Also

qgev

`rgev.d` *Random generation for the modified generalized extreme value distribution*

Description

The function `rgev.d` is a modified version of the function `rgev` for different durations `d` developed by Koutsoyiannis et al. (1998).

Usage

```
rgev.d(n, mu = 0, sigma = 1, xi = 0, theta = 0, eta = 1, d = 1)
```

Arguments

`n`	Number of observations
`mu`	location value
`sigma`	scale value
`xi`	shape value
`theta`	value defining the curvature of the IDF
`eta`	value defining the slope of the IDF
`d`	vector of durations

Value

rgev.d generates random derivates

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

See Also

`rgev`

TS.acc *Accumulation of a time series*

Description

TS.acc accumulates a given time series x at a given accumulation level acc.val. Minimum value for acc.val is 2 [unit time]. Option for using moving sum is given.

Usage

```
TS.acc(x,acc.val,moving.sum="FALSE")
```

Arguments

x	vector of a time series
acc.val	value specifying the accumulation level, minimum value is 2
moving.sum	logical 'TRUE' means moving sum will be applied

Value

x.acc TS.acc returns a vector of an accumulated time series

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

Carola Detring <carola.detring@met.fu-berlin.de>

Examples

```
TS <- rgamma(n=1000,shape=1)
acc.2 <- TS.acc(TS,acc.val=2)

acc.24 <- TS.acc(TS,acc.val=24,moving.sum=TRUE)
```

Index