Package: mevr (via r-universe)

November 3, 2024

Version 1.2.3

Date 2024-10-04

Title Fitting the Metastatistical Extreme Value Distribution MEVD

Maintainer Harald Schellander <harald.schellander@geosphere.at>

Depends R (>= 4.1), dplyr, rlang

Encoding UTF-8

Imports graphics, stats, EnvStats, parallel, foreach, doParallel, bamlss, mgcv, mmand, data.table

Description Extreme value analysis with the metastatistical extreme value distribution MEVD (Marani and Ignaccolo, 2015, <doi:10.1016/j.advwatres.2015.03.001>) and some of its variants. In particular, analysis can be performed with the simplified metastatistical extreme value distribution SMEV (Marra et al., 2019, <doi:10.1016/j.advwatres.2019.04.002>) and the temporal metastatistical extreme value distribution TMEV (Falkensteiner et al., 2023, <doi:10.1016/j.wace.2023.100601>). Parameters can be estimated with probability weighted moments, maximum likelihood and least squares. The data can also be left-censored prior to a fit. Density, distribution function, quantile function and random generation for the MEVD, SMEV and TMEV are included. In addition, functions for the calculation of return levels including confidence intervals are provided. For a description of use cases please see the provided references.

License GPL-3

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

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

RemoteUrl https://github.com/haraldschellander/mevr

RemoteRef HEAD

RemoteSha 4daaabbbf5d4b1def32ba575bc60fa3707adc038

Contents

censored_weibull_fit *Fit Weibull distribution to censored data*

Description

Finds the optimal left-censoring threshold(s) at which the data series should be censored to make sure that the observations in the tail are likely sampled from a Weibull distribution

Usage

```
censored_weibull_fit(x, thresholds, warn)
```

Arguments

`x`	A tibble which is most commonly a result of function `weibull_tail_test`.
`thresholds`	A numeric or vector of quantiles which shal be tested as optimal threshold for left-censoring.
`warn`	If `TRUE` which is the default, warnings about censoring are given.

Value

A tibble with the optimal threshold itself and the Weibull scale and shape parameters obtained from the censored sample.

Examples

```
data("dailyrainfall")
wbtest <- weibull_tail_test(dailyrainfall)
censored_weibull_fit(wbtest, 0.9)
```

---

dailyrainfall *Daily rainfall data*

Description

A dataset containing daily rainfall intended to be used with the package `mevr`

Usage

```
data(dailyrainfall)
```

Format

The dataset contains real world daily rainfall observations from a station in the northern Alps. The series contains values from 1971 to 1985 and are assumed to be Weibull distributed. This data series is intended to be used as is as input data for the package `mevr` to fit the metastatistical extreme value distribution and its variants with different estimation methods.

The dataset is a dataframe with two columns, dates and val:

dates Days of class `Date` in the format YYYY-MM-DD

val Rainfall observations corresponding to the date in the row. The value is the 24 hour sum from the morning hours of day-1 to the morning hours of day.

Examples

```
## Load example data
data(dailyrainfall)

## explore dataset
head(dailyrainfall)
hist(dailyrainfall$val)
plot(dailyrainfall$val, type = "o")
```

dmev *The Metastatistical Extreme Value Distribution*

Description

Density, distribution function, quantile function and random generation for the MEV distribution with shape parameter 'w', scale parameter 'c'. Parameter 'n' refers either to the mean number of wet days per year in case of the simplified MEV, or to the number of wet days for each year.

Usage

```
dmev(x, w, c, n)

pmev(q, w, c, n)

qmev(p, w, c, n)

rmev(N, w, c, n)
```

Arguments

x, q	numeric vector or single values of quantiles for dmev and pmev.
w, c	vector or single values of shape and scale parameter of the MEV distribution. If a vector, w and c must have the same length as n.
n	Either mean number of wet events per year for the SMEV, or a vector for yearly MEVD calculations, i.e. one value per year (see details). If a vector, n must have the same length as w and c.
p	vector or single value of probabilities for qmev.
N	Number of observations to sample from the MEVD or SMEV.

Value

dmev gives the density function, pmev gives the distribution function, qmev gives the quantile function and rmev provides random realizations of the SMEV and MEVD.

Functions

- pmev(): distribution quantile function
- qmev(): quantile function
- rmev(): random generation function

Examples

```
# SMEV
dmev(1200:1300, 0.7, 20, 110)
pmev(1200:1300, 0.7, 70, 110)
qmev(1 - 1 / seq(5,50,5), 0.7, 70, 110)

# MEVD: 50-year event of 10 years Weibull series
w <- rnorm(10, 0.8, 0.1) # shape parameter
c <- rnorm(10, 200, 30) # scale parameter
n <- rnorm(10, 200, 50) # number of wet days
qmev(1 - 1 / 50, w, c, n)

# rl-plot
rp <- seq(5, 50, 5)
rl <- qmev(1 - 1 / rp, w, c, n)
pp <- (1:length(rp)) / (length(rp) + 1)
plot(pp, rl, type = "o")
```

dtmev *The non-stationary Metastatistical Extreme Value Distribution*

Description

Quantile function for the TMEV distribution with a Weibull parent distribution.

Usage

```
dtmev(x, data)

ptmev(q, data)

qtmev(p, data)
```

Arguments

x, q	Numeric vector or single value of probabilities for dtmev.
data	A data frame with at least columns c, w and year. Can be taken from the output of the fitted TMEV object, i.e. x$data (see ftmev).
p	Numeric vector or single value of probabilities for qtmev.

Value

dtmev gives the density function, ptmev gives the distribution function, and qtmev gives the quantile function of the TMEV.

Functions

- ptmev(): distribution quantile function
- qtmev(): distribution quantile function

Examples

```
data(dailyrainfall)
fit <- ftmev(dailyrainfall)
rp <- pp.weibull(fit$maxima)
rl <- qtmev(1 - 1 / rp, fit$data)
plot(rp, sort(fit$maxima), main = "TMEV", ylab = "return level", xlab = "return period (years)")
lines(rp, rl, type = "l")
```

`event_separation` *Detect Rainfall Events with Specified Separation Time (Dry Time) and Minimum Rain Threshold*

Description

The ‘event_separation‘ function identifies rainfall events in time-series data based on a minimum rainfall threshold and a separation time between events. It uses morphological operations to detect and separate events, and it can ignore short events that do not meet a minimum duration criterion. Based on Marra...

Usage

```
event_separation(
  data,
  separation_in_min = 360,
  time_resolution = 10,
  ignore_event_duration = 30,
  min_rain = 0.1
)
```

Arguments

`data` A data frame with two columns, observation date and values.

- ‘groupvar‘: A time variable (e.g., POSIXct or Date) indicating the timestamp of each observation.
- ‘val‘: A numeric vector containing the observed rainfall values.

`separation_in_min` Numeric. The minimum dry period (in minutes) that separates two distinct rainfall events. Default is 360 minutes (6 hours). Values less than `min_rain` count as zero. Note that this should depend on local climatological conditions.

`time_resolution` Numeric. The time step resolution of the data (in minutes). For example, if observations are recorded every 10 minutes, ‘time_resolution‘ should be 10. Default is 10 minutes.

`ignore_event_duration`
: Numeric. The minimum duration (in minutes) of rainfall required for an event to be considered valid. Events shorter than this will be ignored. Default is 30 minutes.

`min_rain`
: Numeric. The minimum rainfall value (e.g., in mm) required for a data point to be considered part of an event. Default is 0.1.

Details

The function works by first dilating the rainfall data using a structuring element defined by the separation time, followed by erosion to remove noise. It then detects events based on the separation time and the minimum rainfall threshold, returning the time intervals where events occurred. Events that span more than two years or are shorter than the ‘ignore_event_duration‘ are ignored.

Value

A list with the following elements:

- ‘data‘: A modified version of the input data with an additional column ‘is_event‘, indicating whether each time step is part of a detected event (1 for event, 0 for no event). This is useful in the next step of the `event_separation`.
- ‘fromto‘: A tibble with two columns, ‘from‘ and ‘to‘, representing the indices of the start (‘from‘) and end (‘to‘) of each detected rainfall event.
- ‘ts_res‘: The time resolution used in the function (in minutes).

Examples

```
## Not run:
# Example data
mock_data <- data.frame(
  groupvar = seq.POSIXt(Sys.time(), by = "10 min", length.out = 100),
  val = c(rep(0, 10), runif(20, 0, 1), rep(0, 70))
)

# Detect rainfall events with a separation of 6 hours and a minimum rain threshold of 0.1
result <- event_separation(mock_data, separation_in_min = 360, time_resolution = 10, min_rain = 0.1)
print(result$fromto)  # View detected events

## End(Not run)
```

`fmev` *Fitting the Metastatistical Extreme Value Distribution (MEVD)*

Description

Fit the MEVD distribution to rainfall observations with different estimation methods.

Usage

```
fmev(
  data,
  threshold = 0,
  method = c("pwm", "mle", "ls"),
  censor = FALSE,
  censor_opts = list(),
  warn = TRUE
)
```

Arguments

`data` The data to which the MEVD should be fitted to. `data` must be a data.frame with two columns. The first column must contain dates of class `Date`, the second or last column must contain the rainfall values corresponding to datums in the rows. No negative values are allowed. NA values are removed with a warning.

`threshold` A numeric that is used to define wet days as values > threshold. $data <= threshold$ is set to NA.

`method` Character string describing the method that is used to estimate the Weibull parameters c and w. Possible options are probability weighted moments (`method='pwm'`), maximum likelihood (`method='mle'`) or least squares (`method='ls'`). The `default` is `pwm`. (see details).

`censor` If `censor=TRUE`, the data series will be left-censored to assure that the observed maxima are samples from a weibull tail. Defaults to `censor=FALSE`.

`censor_opts` An empty list which can be populated with components `thresholds`, `mon`, `nrtrials` and `R`. They give the range of quantiles used as left-censoring threshold, the month with which the block starts, the number of trials used to achieve a weibull fit to the left-censored sample, and the number of synthetic samples used for the test statistics, respectively. See also `weibull_tail_test`.

`warn` If `TRUE` which is the default, warnings about censoring are given.

Details

With the aim of weakening the requirement of an asymptotic assumption for the GEV distribution, a metastatistical approach was proposed by Marani and Ignaccolo (2015). The MEVD is defined in terms of the distribution of the statistical parameters describing "ordinary" daily rainfall occurrence and intensity. The MEVD accounts for the random process of event occurrence in each block and the possibly changing probability distribution of event magnitudes across different blocks, by recognizing the number of events in each block, n, and the values of the shape and scale parameters w and C of the parent Weibull distribution to be realisations of stochastic variables. The MEVD can then be written as

$$F = \frac{1}{T}\sum_{j=1}^{T}\prod_{k\in A_j}\left(1 - e^{-\left(\frac{x}{C(j,k)}\right)^{w(j,k)}}\right)$$

for $w > 0$ and $C > 0$. With T fully recorded years, yearly C and w can be estimated by fitting a Weibull distribution to the values x of this year, and n is the number of ordinary events per year, i.e. all rainfall events larger than a threshold.

If the probability distribution of daily rainfall is assumed to be time-invariant, the MEVD can be simplified to

$$F = [1 - exp(-x/C)^{w}]^{n}$$

with single values for the shape and scale parameters w and C. n is then the mean number of wet days at this location (Marra et al., 2019; Schellander et al., 2019).

As is shown e.g. Schellander et al., 2019, probability weighted moments should be preferred over maximum likelihood for the estimation of the Weibull parameters w and C. Therefore `method = 'pwm'` is the default.

The MEVD can also be used for sub-daily precipitation (Marra et al., 2019). In that case n has to be adapted accordingly to the 'mean number of wet events' per year.

This function returns the parameters of the fitted MEVD distribution as well as some additional fitting results and input parameters useful for further analysis.

Value

A list of class `mevr` with the fitted Weibull parameters and other helpful ingredients.

`c` vector of Weibull scale parameters of the MEVD, each component refers to one year.

`w` vector of Weibull shape parameters of the MEVD, each component refers to one year.

`n` Number of wet events per year. Wet events are defined as rainfall > `threshold`.

`params` A named vector of the fitted parameters.

`maxima` Maximum values corresponding to each year.

`data` $data >= threshold$ used to fit the MEVD and additional components which may be useful for further analysis.

`years` Vector of years as YYYY.

`threshold` The chosen threshold.

`method` Method used to fit the MEVD.

`type` The type of distribution ("MEVD")

Author(s)

Harald Schellander, Alexander Lieb

References

Marani, M. and Ignaccolo, M. (2015) 'A metastatistical approach to rainfall extremes', Advances in Water Resources. Elsevier Ltd, 79(Supplement C), pp. 121-126. doi: 10.1016/j.advwatres.2015.03.001.

Schellander, H., Lieb, A. and Hell, T. (2019) 'Error Structure of Metastatistical and Generalized Extreme Value Distributions for Modeling Extreme Rainfall in Austria', Earth and Space Science, 6, pp. 1616-1632. doi: 10.1029/2019ea000557.

See Also

fsmev, ftmev

Examples

```
data(dailyrainfall)
fit <- fmev(dailyrainfall, method = "mle")
fit
plot(fit)
```

fsmev *Fitting the simplified Metastatistical Extreme Value Distribution (SMEV)*

Description

Fit the SMEV distribution to rainfall observations with different estimation methods.

Usage

```
fsmev(
  data,
  threshold = 0,
  method = c("pwm", "mle", "ls"),
  censor = FALSE,
  censor_opts = list(),
  warn = TRUE,
  sd = FALSE,
  sd.method = "boot",
  R = 502
)
```

Arguments

`data` The data to which the SMEV should be fitted to. `data` must be a data.frame with two columns. The first column must contain dates of class `Date`, the second or last column must contain the rainfall values corresponding to datums in the rows. No negative values are allowed. NA values are removed with a warning.

`threshold` A numeric that is used to define wet days as values > threshold. $data <= threshold$ is set to NA.

`method` Character string describing the method that is used to estimate the Weibull parameters c and w. Possible options are probability weighted moments (`method='pwm'`), maximum likelihood (`method='mle'`) or least squares (`method='ls'`). The `default` is pwm. (see details).

`censor` If `censor=TRUE`, the data series will be left-censored to assure that the observed maxima are samples from a weibull tail. Defaults to `censor=FALSE`.

censor_opts An empty list which can be populated with components thresholds, mon, nrtrials and R. They give the range of quantiles used as left-censoring threshold, the month with which the block starts, the number of trials used to achieve a weibull fit to the left-censored sample, and the number of synthetic samples used for the test statistics, respectively. See also weibull_tail_test.

warn If TRUE which is the default, warnings about censoring are given.

sd If sd=TRUE, confidence intervals of the SMEV distribution are calculated (see details).

sd.method Currently only a non parametric bootstrap technique can be used to calculate SMEV confidence intervals with sd.method='boot'. The default is sd=FALSE.

R The number of samples drawn from the SMEV distribution to calculate the confidence intervals with sd.method='boot'

Details

The SMEV was introduced by (Marra et al., 2019) as a simplified version of the MEVD (Marani and Ignaccolo, 2015) with the assumption of a stationary parent Weibull distribution as

$$F = [1 - exp(-x/C)^w]^n$$

for $w > 0$ and $C > 0$ being the Weibull shape and scale parameter and $n > 0$ being the mean number of wet days over all years. Wet days are defined as rainfall events > threshold. As it was shown by e.g. Schellander et al., 2019, probability weighted moments should be preferred over maximum likelihood for the estimation of the Weibull parameters w and C. Therefore method = 'pwm' is the default.

Confidence intervals of the SMEV distribution can be calculated using a non parametric bootstrap technique. Note that this very slow.

This function returns the parameters of the fitted SMEV distribution as well as some additional fitting results and input parameters useful for further analysis.

Value

A list of class mevr with components:

c Single value of the Weibull scale parameter of the SMEV.

w Single value of the Weibull shape parameter of the SMEV.

n Mean number of wet events, averaged over all years. Wet events are defined as rainfall > threshold.

params A named vector of the fitted parameters.

maxima Maximum values corresponding to each year.

std Standard error of fitted parameters (if sd=TRUE).

varcov Covariance matrix of fitted parameters (if sd=TRUE).

data $data >= threshold$ used to fit the SMEV and additional components which may be useful for further analysis.

years Vector of years as YYYY.

`threshold` The chosen threshold.

`method` Method used to fit the MEVD.

`censor` `TRUE` when the data-series was left-censored and `FALSE` otherwise.

`type` The type of distribution ("SMEV")

`rejected` If `censor=TRUE`, `rejected=TRUE` when the Weibull tail assumption is rejected and `rejected=FALSE` otherwise. If `censor=FALSE` this value is not returned.

Author(s)

Harald Schellander, Alexander Lieb

References

Marra, F. et al. (2019) ’A simplified MEV formulation to model extremes emerging from multiple nonstationary underlying processes’, Advances in Water Resources. Elsevier Ltd, 127(April), pp. 280-290. doi: 10.1016/j.advwatres.2019.04.002.

See Also

`fmev`, `ftmev`

Examples

```
data(dailyrainfall)

fit <- fsmev(dailyrainfall)
fit
plot(fit)

# left censor data prior to fitting
set.seed(123)
sample_dates <- seq.Date(from = as.Date(”2000-01-01”), to = as.Date(”2020-12-31”), by = 1)
sample_data <- data.frame(dates = sample_dates, val = sample(rnorm(length(sample_dates))))
d <- sample_data |>
  filter(val >= 0 & !is.na(val))

fit <- fsmev(d)
fit_c <- fsmev(d,
               censor = TRUE,
               censor_opts = list(thresholds = c(seq(0.5, 0.9, 0.1), 0.95),
                                  mon = 1,
                                  nrtrials = 2,
                                  R = 100))

rp <- 2:100
rl <- return.levels.mev(fit, return.periods = rp)
rl_c <- return.levels.mev(fit_c, return.periods = rp)
plot(sort(pp.weibull(fit$maxima)), sort(fit$maxima),
  xlab = ”Return period (a)”, ylab = ”daily rain (mm)”)
lines(rl$rp, rl$rl)
lines(rl_c$rp, rl_c$rl, col = "red")
legend("bottomright", legend = c("std", "censored"),
  col = c("black", "red"), lty = c(1, 2), lwd = c(1, 1.5), bty = "n")
```

---

ftmev *Fitting the temporal Metastatistical Extreme Value Distribution (TMEV)*

Description

Fit the temporal MEVD distribution TMEV to rainfall observations with a cyclic spline to account for seasonality.

Usage

```
ftmev(
  data,
  threshold = 0,
  minyears = 10,
  day_year_interaction = FALSE,
  verbose = FALSE,
  yday_ti_shape_k = 10,
  yday_ti_scale_k = 10,
  year_ti_shape_k = 10,
  year_ti_scale_k = 10
)
```

Arguments

`data` The data to which the TMEV should be fitted to. `data` must be a data.frame with two columns. The first column must contain dates of class `Date`, the second or last column must contain the rainfall values corresponding to datums in the rows. No negative values are allowed. NA values are removed with a warning.

`threshold` A numeric that is used to define wet days as values > threshold. $data <= threshold$ is set to NA.

`minyears` Minimum number of available years for fitting a cyclic spline to the non-stationary data series (see details).

`day_year_interaction` Logical. Should an additional year vs day of the year interaction be used for the calculation of the temporal trend in seasonality? (see details). Default is `FALSE`.

`verbose` Logical. If `TRUE`, verbose output of the temporal fitting process is shown during runtime.

`yday_ti_shape_k` A numeric that is used to set the dimension of the bases used to represent the smooth term ti() for yday in the formula for shape.

`yday_ti_scale_k`
: A numeric that is used to set the dimension of the bases used to represent the smooth term ti() for yday in the formula for scale.

`year_ti_shape_k`
: A numeric that is used to set the dimension of the bases used to represent the smooth term ti() for year in the formula for shape (only used when day_year_interaction == TRUE).

`year_ti_scale_k`
: A numeric that is used to set the dimension of the bases used to represent the smooth term ti() for year in the formula for scale (only used when day_year_interaction == TRUE).

Details

With the aim of exploiting the full temporal information for parameter estimation, Falkensteiner et al., (2023) introduced the TMEV, which is an explicitly non-stationary formulation of the MEVD (Marani and Ignaccolo, 2015). Adopting a Weibull distribution for ordinary rainfall events, the assumption of yearly constant coefficients is relaxed by allowing the Weibull parameters to fluctuate with time. The TMEV can then be written as

$$F = \frac{1}{T}\sum_{j=1}^{T}\prod_{k\in A_j}\left(1 - e^{-\left(\frac{x}{C(j,k)}\right)^{w(j,k)}}\right)$$

with $w > 0$ and $C > 0$ being the Weibull shape and scale parameters, and $A_j \subseteq (1, ..., 366)$ being the wet days in year j. The temporal and the superimposed seasonal dependence on w and c is modeled with a cyclic seasonal effect on the day of the year.

Technically this is accomplished by fitting a cyclic spline to the daily rainfall values. The following formula is used for the fitting procedure of both the Weibull scale and shape parameter with the function `bamlss` from package `bamlss`:

$$parameter = x \sim s(year) + ti(yday, bs = "cc", k = 10)$$

The first effect models the long-term temporal trend of the parameter with a thin-plate spline. The second effect models the superimposed seasonal fluctuations of the parameter with the 'day of the year' with a cyclic cubic regression spline and 10 knots, to ensure a smooth transition between December and January. The number of knots (k) in the above equation can be set separately for the year and yday effect as well as separately for the shape and scale parameter of the Weibull distribution. This can be done by overwriting the parameters `yday_ti_shape_k`, `yday_ti_scale_k`, `year_ti_shape_k`, `year_ti_scale_k` in the call to `ftmev`. Note that these values depend on many factors, such as the structure of the data, the TMEV is fitted to. Please refer to the documentation of the packages `bamlss` and, in particular `mgcv`.

For data series with lengths < 10 years, the first temporal effect is changed to a simple linear time trend.

For trend analysis, an additional interaction term can be added to the model formula. The following term models the relationship between the seasonality as day of the year and the year itself with a combination of a thin plate and a cyclic cubic spline:

$$ti(year, yday, bs = c(”tp”, ”cc”), d = c(1, 1), k = c(year_t i_k, yday_t i_k))$$

This function returns the parameters of the fitted TMEV distribution as well as some additional fitting results and input parameters useful for further analysis.

Value

A list of class `mevr` with components:

`c` Vector of Weibull scale parameters of the TMEV, each row refers to one event, which is a day for daiyl rainfall.

`w` Vector of Weibull shape parameters of the TMEV, each row refers to one event, which is a day for daiyl rainfall.

`n` Number of wet events per year. Wet events are defined as rainfall > `threshold`.

`maxima` Maximum values corresponding to each year.

`data` A data frame with the data used to fit the TMEV and the fitted Weibull parameters `c` and `w`.

`years` Vector of years as YYYY.

`threshold` The chosen threshold.

`x` The fitted `bamlss` object.

`type` The type of distribution ("TMEV").

`minyears` The minimum number of years used to fit the TMEV as provided.

Author(s)

Marc-Andre Falkensteiner, Harald Schellander

References

Marani, M. and Ignaccolo, M. (2015) ’A metastatistical approach to rainfall extremes’, Advances in Water Resources. Elsevier Ltd, 79(Supplement C), pp. 121-126. doi: 10.1016/j.advwatres.2015.03.001.

Falkensteiner, M., Schellander, H., Hell, T. (2023) ’Accounting for seasonality in the metastatistical extreme value distribution’, (Weather and Climate Extremes, 42, 2023, https://doi.org/10.1016/j.wace.2023.100601).

See Also

`fmev`, `fsmev`

Examples

```
data(dailyrainfall)
fit <- ftmev(dailyrainfall)
plot(fit, type = ”rl”)

# temporal trend of the Weibull parameters
pred <- predict(fit)
pred_year <- predict(fit, term = ”year”)
```

```
boxplot(c.pred ~ year, data = pred)
with(pred_year, lines(year - 1970, c.pred.year, type = "b", pch = 20, col = "red"))
```

ordinary_events *Identifies ordinary rainfall events by calculating the maximum within rainfall events defined with event_separation*

Description

The ‘ordinary_events‘ function calculates the rolling sum of values in the provided data over a specified duration and identifies the events with the highest sum within the defined time intervals. It is useful for detecting significant events based on aggregated values over time.

Usage

```
ordinary_events(x, duration, na.rm = TRUE)
```

Arguments

x — A list containing at least the following elements (usually the output of function event_separation):

- ‘data‘: A data frame or data table with at least two columns: ‘val‘ (the values to sum) and ‘groupvar‘ (timestamps or another grouping variable).
- ‘ts_res‘: The time resolution of the data in minutes (i.e., how many minutes each time step represents).
- ‘fromto‘: A data frame with two columns: ‘from‘ and ‘to‘, representing the start and end indices of the time intervals to analyze.

duration — Numeric. The duration in minutes for which maxima shall be calculated.

na.rm — Logical. Removes lines with NA values from x when na.rm = TRUE.

Value

Returns a tibble with individual rainfall events that can be used as input for functions fsmev, fmev, ftmev.

Examples

```
## Not run:
# Example usage
x <- list(
  data = data.frame(val = runif(100),
  groupvar = seq.POSIXt(Sys.time(),
                        by = "10 min",
                        length.out = 100)
                        ),
  ts_res = 10,
  fromto = data.frame(from = c(1, 51), to = c(50, 100))
)
duration <- 30
result <- ordinary_events(x, duration)
print(result)

## End(Not run)
```

plot.mevr *Plot graphs of MEVD, SMEV or TMEV fit*

Description

A return level plot, qq-plot, pp-plot and a histogram with the fitted density is produced

Usage

```
## S3 method for class 'mevr'
plot(
  x,
  q = c(2, 10, 20, 30, 50, 75, 100, 150, 200),
  ci = FALSE,
  type = c("all", "rl", "qq", "pp", "hist"),
  ...
)
```

Arguments

`x` An object of classmevr, whose type argument is one of MEVD, SMEV or TMEV

`q` vector of return periods, $q > 1$.

`ci` if `ci=TRUE`, confidence intervals will be computed.

`type` if omitted a panel with a return level plot (`type='rl'`, a density plot (`type='hist'`), a qq-plot (`type='qq'`) and a probability plot (`tpe='pp'`) are shown.

`...` Further parameters may also be supplied as arguments. See e.g. plot.

Value

No return value, only a plot is produced.

Examples

```
data(dailyrainfall)

# fit a simplified MEVD
fit <- fsmev(dailyrainfall)
fit
```

```
plot(fit)

# fit MEVD
fit <- fmev(dailyrainfall, method = "ls")
fit
plot(fit)
```

`pp.weibull` *Weibull plotting position*

Description

Calculates the Weibull plotting position for the given maxima

Usage

```
pp.weibull(x)
```

Arguments

`x`	Numeric vector of block maxima

Value

A numeric vector of Weibull plotting positions corresponding to the given maxima `x`

Examples

```
data(dailyrainfall)
fit <- fsmev(dailyrainfall)
rp <- pp.weibull(fit$maxima)
rl <- return.levels.mev(fit, return.periods = rp)
plot(rp, sort(fit$maxima), xlab = "Return period (years)", ylab = "Return level", main = fit$type)
lines(rp, rl$rl)
```

`predict.mevr` *TMEV prediction*

Description

Takes a `mevr` object where the TMEV has been fitted to rainfall data and calculates `bamlss` predictions for the distributional parameters and the model terms. Basically a wrapper to the corresponding function `predict.bamlss`

Usage

```
## S3 method for class 'mevr'
predict(object, newdata, term, ...)
```

Arguments

`object` Object of class `mevr`, fitted with the TMEV.

`newdata` A data frame with the model covariates (year, yday) at which predictions are required. Note that depending on argument term, only covariates that are needed by the corresponding model terms need to be supplied. If not supplied, predictions are made on the data supplied by the fitted object `x`.

`term` Character of the model terms for which predictions shall be calculated. Can only be `"year"` or `"yday"`. If not specified, predictions for all terms are calculated.

`...` Arguments passed to prediction functions that are part of a bamlss.family object, i.e., the objects has a $predict() function that should be used instead.

Details

See also the details of `ftmev` for an explanation of the model terms used to fit the temporal trend of the Weibull parameters. The basis dimensions yday_ti_shape_k, yday_ti_scale_k, year_ti_shape_k, year_ti_scale_k are taken from the fitting process, i.e. the call to `ftmev`.

Value

A data.frame with the supplied covariables and the predicted parameters.

See Also

`ftmev`, `predict.bamlss`

Examples

```
data(dailyrainfall)

# restrict for the sake of speed
idx <- which(as.POSIXlt(dailyrainfall$date)$year + 1900 < 1976)
data <- dailyrainfall[idx, ]

f <- ftmev(data, minyears = 5)
predict(f, term = "year")
```

`print.mevr` *Print method for object of class mevr*

Description

Print nicely formatted output of the fit to the MEVD and its variants

Usage

```
## S3 method for class 'mevr'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`x`	Object of class `mevr`, fitted with MEVD, SMEV or TMEV.
`digits`	Number of digits.
`...`	Additional parameters.

Value

A nicely formatted output of the fitting results.

Examples

```
data(dailyrainfall)

# fit a simplified MEVD
fit <- fsmev(dailyrainfall)
print(fit)
```

`return.levels.mev` *Return Levels for the MEVD/SMEV/TMEV extreme value distributions*

Description

Calculate return levels for a MEVD, SMEV or TMEV extreme value distributions from an object of class `mevr`.

Usage

```
return.levels.mev(
  x,
  return.periods = c(2, 10, 20, 30, 50, 75, 100, 150, 200),
  ci = FALSE,
  alpha = 0.05,
  method = "boot",
  R = 502,
  ncores = 2L
)
```

Arguments

`x`	An object of class `mevr`, either fitted with the MEVD, SMEV or TMEV
`return.periods`	A vector of return periods in years, excluding 1.
`ci`	If `ci=TRUE`, confidence intervals are calculated depending on the type of distribution (only for MEVD or SMEV).
`alpha`	Number between zero and one giving the `1 - alpha` confidence level. Defaults to `alpha=0.05`.

`method` Character string giving the method for confidence interval calculation. Option `method='boot'` employs a parametric bootstrap that simulates data from the fitted model, and then fits the chosen MEVD type to each simulated data set to obtain a sample of parameters or return levels (very slow).

`R` The number of bootstrap iterations.

`ncores` Number of cores used for parallel computing of confidence intervals. Defaults to 2.

Details

Note that bootstraping the confidence intervals is very slow.

Value

A list with return levels, chosen return periods and, if `ci=TRUE`, `alpha/2` and `1 - alpha/2` confidence intervals.

Examples

```
data(dailyrainfall)

fit <- fmev(dailyrainfall)
return.levels.mev(fit)
plot(fit)
```

`weibull_tail_test` *Weibull tail test*

Description

This functions provides a way to test if observed rainfall maxima from a data series are likely samples from a parent distribution with a Weibull tail. The concept and the code is based on the paper Marra F, W Amponsah, SM Papalexiou, 2023. Non-asymptotic Weibull tails explain the statistics of extreme daily precipitation. Adv. Water Resour., 173, 104388, https://doi.org/10.1016/j.advwatres.2023.104388. They also provide the corresponding Matlab code (https://zenodo.org/records/7234708).

Usage

```
weibull_tail_test(
  data,
  threshold = 0,
  mon = 1,
  cens_quant = 0.9,
  p_test = 0.1,
  R = 500
)
```

Arguments

data A data.frame

threshold A numeric that is used to define wet days as values > threshold.

mon This month defines the block whose maxima will be tested. The block goes from month-YYYY-1 to month-YYYY.

cens_quant The quantile at which the tail test should be performed. Must be a single numeric.

p_test A numeric defining the 1 - p_test confidence band. This function tests the ratio of observed block maxima below p_test and above 1 - p_test. See details.

R The number of synthetic samples.

Details

Null-Hyothesis: block maxima are samples from a parent distribution with Weibull tail (tail defined by a given left-censoring threshold). If the fraction of observed block maxima outside of the interval defined by p_test is larger than p_test the null hypothesis is rejected.

Value

A tibble with the test outcome and other useful results:

is_rejected outcome of the test (TRUE means that the assumption of Weibull tails for the given left-censoring threshold is rejected).

p_out fraction of block maxima outside of the Y = 1 - p_out confidence interval

p_hi fraction of block maxima above the Y = 1 - p_out confidence interval

p_lo fraction of block maxima below the Y = 1 - p_out confidence interval

scale scale parameter of the Weibull distribution describing the non-censored samples

shape shape parameter of the Weibull distribution describing the non-censored samples

quant the quantile used as left-censoring threshold

Examples

```
data("dailyrainfall")
weibull_tail_test(dailyrainfall)

# generate data
set.seed(123)
sample_dates <- seq.Date(from = as.Date("2000-01-01"), to = as.Date("2010-12-31"), by = 1)
sample_data <- data.frame(dates = sample_dates, val = sample(rnorm(length(sample_dates))))
d <- sample_data |>
  filter(val >= 0 & !is.na(val))
fit_uncensored <- fsmev(d)

# censor the data
thresholds <- c(seq(0.1, 0.9, 0.1), 0.95)
p_test <- 0.1
res <- lapply(thresholds, function(x) {
  weibull_tail_test(d, cens_quant = x, p_test = p_test, R = 200)
})
res <- do.call(rbind, res)

# find the optimal left-censoring threshold
cens <- censored_weibull_fit(res, thresholds)
cens$optimal_threshold
cens$quantile

# plot return levels censored vs uncensored
rp <- c(2:100)
rl_uncensored <- return.levels.mev(fit_uncensored, return.periods = rp)$rl
rl_censored <- qmev(1 - 1/rp, cens$shape, cens$scale, fit_uncensored$n)
plot(rp, rl_uncensored, type = "l", log = "x", ylim = c(0, max(rl_censored, rl_uncensored)),
     ylab = "return level", xlab = "return period (a)")
points(pp.weibull(fit_uncensored$maxima), sort(fit_uncensored$maxima))
lines(rp, rl_censored, type = "l", col = "red")
legend("bottomright", legend = c("uncensored",
        paste0("censored at ", round(cens$optimal_threshold, 1), "mm")),
        col = c("black", "red"), lty = c(1, 1))
```

Index