Package ‘lfstat’

May 8, 2026

Type Package

Title Calculation of Low Flow Statistics for Daily Stream Flow Data

Version 0.9.13

Date 2025-09-04

BugReports https://github.com/mundl/lfstat/issues

Description
The ``Manual on Low-flow Estimation and Prediction'' (Gustard & Demuth (2009, ISBN:978-92-63-11029-9)), published by the World Meteorological Organisation, gives a comprehensive summary on how to analyse stream flow data focusing on low-flows. This packages provides functions to compute the described statistics and produces plots similar to the ones in the manual.

License GPL (>= 2)

Language en-GB

LazyLoad yes

Depends R (>= 4.1.0), lmom, xts

Imports lmomRFA, dygraphs, zoo, latticeExtra, plyr, scales, lattice, grDevices, graphics, stats, utils

Suggests testthat

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Tobias Gauster [ctb, cre],
Gregor Laaha [aut] (ORCID: <https://orcid.org/0000-0002-6793-9640>),
Daniel Koffler [aut]

Maintainer Tobias Gauster <t.gauster@gmail.com>

Repository CRAN

Date/Publication 2025-09-04 06:40:31 UTC

Contents

`lfstat-package` *Calculation of Low Flow Statistics for Daily Stream Flow Data*

Description

The "Manual on Low-flow Estimation and Prediction" (Gustard & Demuth (2009, ISBN:978-92-63-11029-9)), published by the World Meteorological Organisation, gives a comprehensive summary on how to analyse stream flow data focusing on low-flows. This packages provides functions to compute the described statistics and produces plots similar to the ones in the manual.

Details

Create an object of class `lfobj` **(Low-Flow-Objects):** The package calculates indices and makes graphics for low flow analysis. It brings its own class `'lfobj'`, a special data.frame format with columns `'day'`, `'month'`, `'year'`, `'flow'`, `'hyear'` and possibly `'baseflow'`.

`'day'`, `'month'` and `'year'` refer to the date, `'flow'` is the measured runoff (unit-independent), `'baseflow'` the calculated base flow.

`'hyear'` refers to the hydrological year. When creating the `'lfobj'` you define the month where the stations hydrological year starts. If annual indices are calculated or single years are plotted, the `'hyear'` is taken.

Basically there are to options to create an low flow object:

If you have special data format, e.g. GRDC, you can use the function `readlfdata`, see `?readlfdata` to see which formats are currently supported. Otherwise you can use `createlfobj`. You can apply it for new data in one of two ways: 1) You create a data.frame with columns: `'day'`, `'month'`, `'year'` and `'flow'`. 2) You create a time-series (`'ts'`) from `'flow'` and give the start date of the series when calling `'createlfobj'`.

Preparation: lfstat does not need to know the unit of the flow, but you might want it to appear in your plots. You can use `setlfunit` to define how units are labelled in your graphics. Examples are given in `'?setlfunit'`.

Please check for NA-values using `lfnacheck`, indices and plots are made as if series were complete. See the manual on how to deal with missing values and, if reasonable, use `lfnainterpolate`.

Available Indices:

- `meanflow`
- `Qxx`
- `MAM` (mean annual minima)
- `BFI`
- `recession` (recession constant)

- streamdef (Streamflow deficit)
- tyears (Extreme value - T-years event)
- seasratio
- seasindex
- multistationsreport

Plots:

- hydrograph
- recessionplot (Diagnosis for recession)
- fdc (Flow-duration-curve)
- sbplot (seasonal bar chart)
- seglenplot (select recession length for recession)
- streamdefplot (Streamflow deficit)
- rfa (Regional frequency analysis)
- dmcurve (Double mass curve)

Index of help topics:

```
BFI                     Base Flow Index
MAM                     Mean Annual Minimum
Qxx                     Qxx, Q95, Q90, Q70
apply.seasonal          Apply an aggregation function seasonally.
as.lfobj                Coerce to class "lfobj"
as.xts.lfobj            Convert Object To Class "xts"
baseflow                Calculate the base flow of a river
bfplot                  Base Flow Plot
check_distribution      Checks if a Distribution is suited
createlfobj             Create an low flow object for further Low Flow
                        Analysis
dmcurve                 Double Mass Curve
ev_return_period        Estimate the return period for given quantiles
evfit                   Fit an extreme value distribution to
                        observations
evquantile              Estimating populations quantiles of extreme
                        values
fdc                     Flow Duration Curve
fill_na                 Interpolation NA values in a vector
find_droughts           Identifying Low Flow Periods
flowunit                Set and retrieve unit of the discharge
gringorten              Gringorten Plotting Positions
hydrograph              Hydrograph
hyear_start             Extract or guess the Start of a Hydrological
                        Year
lfnacheck               Low flow object check for missing values.
lfnainterpolate         Interpolate missing values
lfstat-package          Calculation of Low Flow Statistics for Daily
                        Stream Flow Data
```

ma	Simple Moving Average
meanflow	Mean flow
multistationsreport	Report for several stations
ngaruroro	Daily stream flow data used for low flow analysis
plot.deficit	Plot time series of deficits
pooling	Pooling Procedures of Low Flow Events
readlfdata	Reads data sheets
recession	Recession Constant
recessionplot	Recession diagnostic plot
reversing	Reversed functions for several Extreme Value Distributions
rfa	Regional Frequency Analysis
rfaplot	Regional Frequency Analysis
rpline	Highlight quantiles/return periods
sbplot	Seasonal Bar Chart
seasindex	Seasonality Index
season	Attribute dates to seasons
seasratio	Seasonality Ratio
seglenplot	Bar chart of recession length
setlfunit	Define the unit to use in low flow plots
streamdef	Streamflow Deficit
streamdefplot	Streamflow Deficit Plot
summary.deficit	Object Summaries
trace_value	Draw Paths to Points perpendicular to Coordinate Axis
tyears	Calculate Low-Flow Quantiles for given Return Periods
tyearsS	Calculate Low-Flow Quantiles for given Return Periods
vary_threshold	Create varying thresholds
water_year	Compute the water year

Author(s)

Tobias Gauster [ctb, cre], Gregor Laaha [aut] (ORCID: <https://orcid.org/0000-0002-6793-9640>), Daniel Koffler [aut]

Maintainer: Tobias Gauster <t.gauster@gmail.com>

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p. https://library.wmo.int/idurl/4/32176

`apply.seasonal` *Apply an aggregation function seasonally.*

Description

Similar to the functions `apply.daily`, `apply.monthly`, `apply.yearly` etc. from the **xts** package.

Usage

```
apply.seasonal(x, varying, fun = function(x) min(x, na.rm = TRUE),
                aggregate = NULL, replace.inf = TRUE, origin = 1)
```

Arguments

`x`	an object of class 'xts' or 'zoo'. The time series which should get aggregated.
`varying`	a character vector of length one of a possibly named vector of class 'Date' or coercible to 'Date'. Valid character vectors are "daily", "weekly", "monthly" or "constant". If of class 'Date', the elements are considered as start points of a season. See Examples.
`fun`	the function used for aggregating all elements of a season.
`aggregate`	possibly a function used for aggregating per season.
`replace.inf`	should non-finite values introduced by `fun` be omitted?
`origin`	The start of the hydrological year. If set to 1 (the default) aggregation is carried out using the calendar year.

Value

a matrix with every (hydrological) year being a row and every column being a season.

Examples

```
data(ngaruroro)
ng <- as.xts(ngaruroro)

year <- water_year(time(ng), origin = "Sept")
ng10 <- ng[year %in% 1991:2000, ]

# computes the annual minima (AM)
apply.seasonal(ng10, varying = "yearly", origin = 9)

# computes the mean annual minima (MAM)
apply.seasonal(ng10, varying = "yearly", aggregate = mean, origin = 9)

# computes monthly minima (AM)
apply.seasonal(ng10, varying = "monthly", origin = 9)

# computes minima for summer and winter separately
```

```
# winter starts in September
seasons <- as.Date(c("1999-09-01", "1999-11-04"))
names(seasons) <- c("winter", "summer")
apply.seasonal(ng10$discharge, varying = seasons, origin = 9)
```

---

as.lfobj *Coerce to class* 'lfobj'

Description

Functions to check if object is of class 'fobj' or coerce it if possible. Currently, only methods for 'zoo' and 'xts' exist.

Usage

```
as.lfobj(x, ...)
is.lfobj(x)
## S3 method for class 'xts'
as.lfobj(x, ...)
## S3 method for class 'zoo'
as.lfobj(x, ...)
```

Arguments

x	any R object.
...	additional arguments to be passed to or from methods.

Value

An object of class 'lfobj'.

See Also

createlfobj

Examples

```
data(ngaruroro)
is.lfobj(ngaruroro)

# coerce zoo object to class \code{'lfobj'}
z1 <- zoo(1:10, order.by = seq(Sys.Date(), length.out = 10, by = "days"))
as.lfobj(z1, hyearstart = 5)

# coerce xts object to class \code{'lfobj'}
xts1 <- xts(1:10, order.by = seq(Sys.Date(), length.out = 10, by = "days"))
as.lfobj(xts1, hyearstart = 5)
```

as.xts.lfobj — *Convert Object To Class* 'xts'

Description

Conversion function to coerce data objects of classes 'lfobj' to class 'xts'.

Usage

```
## S3 method for class 'lfobj'
as.xts(x, ...)
```

Arguments

x	an object of class 'lfobj'.
...	additional parameters or attributes

Value

An S3 object of class 'xts'.

See Also

as.xts

Examples

```
data(ray)
r <- as.xts(ray)

# attributes of the lfobject are retained
attr(ray, "lfobj")
xtsAttributes(r)
```

baseflow — *Calculate the base flow of a river*

Description

Given a stream flow hydrograph of flows (regular time series), the base flow is separated. The minima of a period (default block.len = 5) is calculated and turning points are identified. At turning points the base flow equals the actual flow, in between, linear interpolation is carried out.

Usage

```
baseflow(x, tp.factor = 0.9, block.len = 5)
```

Arguments

x	numeric vector containing flows
tp.factor	numeric vector of length one. Towards high flows, allow the central value of three consecutive minima only to be of a factor (1 - tp.factor) higher than the surrounding values
block.len	numeric vector of length one.

Value

A numeric vector of length(x). It contains NAs as until the first turning point, the base flow cannot be determined.

References

Tallaksen, L. M. and Van Lanen, H. A. J. 2004 Hydrological Drought: Processes and Estimation Methods for Streamflow and Groundwater. *Developments in Water Science* **48**, Amsterdam: Elsevier.

Examples

```
## reproducing Tallaksen and van Lanen (2004)
## Example 5.3 Base Flow Index"

data(ray)
ray <- as.xts(ray)

# calculate base flow and plot it
ray$baseflow <- baseflow(ray$discharge)
ray96 <- ray[format(time(ray), "%Y") == "1996", ]
plot(ray96$discharge, type = "l")
lines(ray96$baseflow, col = 2)

# aggregated base flows for river Ray
# these are mean flow totals per day, not per year as written
# in Tallaksen and van Lanen (2004)
round(colSums(ray96[, c("discharge", "baseflow")]), 2)
```

BFI *Base Flow Index*

Description

Calculates the base flow index of an object of class 'lfobj'.

Usage

```
BFI(lfobj, year = "any",breakdays = NULL, yearly = FALSE)
```

Arguments

`lfobj`	An object of class `'lfobj'`
`year`	The year for which the BFI should be computed. If `hyearstart != 1` the BFI is calculated for the hydrological, `year = "any"` means the whole series should be taken.
`breakdays`	A vector of break days if the BFI should be calculated for different seasons.
`yearly`	If TRUE, the BFI is calculated for each hydrological year separately.

Details

If `'breakdays'` is a single day, e.g. "01/06", the start of the hydrological year is taken as the second break day. If more than two seasons are to be specified, a vector of all break days is needed.

Value

A length one vector giving the base flow index for the whole series or the specified year. If yearly is true, a vector of the annual base flow indices is returned. If break days are specified, the values are separated per season.

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

`bfplot`

Examples

```
data(ngaruroro)
BFI(ngaruroro)
BFI(ngaruroro, breakdays = c("01/11","01/05"))
BFI(ngaruroro, year = 1991)
bfplot(ngaruroro, year = 1991)
```

bfplot — *Base Flow Plot*

Description

Visualizes the hydrograph versus the base flow hydrograph.

Usage

```
bfplot(lfobj,
       year = "any",
       col = "green",
       bfcol = "blue",
       ylog = FALSE)
```

Arguments

lfobj	An object of class "lfobj"
year	The hydrological year for which the BFI should be computed. If "any" the whole series is plotted.
col	Colour of flow
bfcol	Colour of base flow
ylog	Log y-axis?

Value

No return value, called for side effects (plotting).

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

BFI

Examples

```
data(ngaruroro)
# Plot starts in December, as ngaruroro's hyearstart = 12
bfplot(ngaruroro, year = 1991)
```

check_distribution *Checks if a Distribution is suited*

Description

Most distributions are used for modelling either minima or maxima. Sometimes a better fit can be achieved by reversing the distribution. This functions helps to decide if the reversed distribution is advisable.

Usage

```
check_distribution(extreme = c("minimum", "maximum"), distribution,
                   def = list(minimum = c(),
                              maximum = c("gev")))
```

Arguments

`extreme`	character vector, describing the kind of extreme value to be fitted. Either `'minimum'` or `'maximum'`.
`distribution`	character vector of length one. Distribution chosen by the user.
`def`	a list of length two, containing the elements `'minimum'` and `'maximum'`.

Value

a character vector as long as `distribution` containing the optimal choice for the given `distributions` under the constraints of `def`.

Examples

```
# Using the Weibull distribution for minimum values is a good choice
check_distribution(extreme = "minimum", distribution = "wei")

# ... whereas the GEV is meant for maxima.
# Therefore the reversed distribution is suggested.
check_distribution(extreme = "minimum", distribution = "gev")
```

createlfobj *Create an low flow object for further Low Flow Analysis*

Description

Generic function for creating a low flow object (class `'lfobj'`). Low flow objects can be created from a time series of daily flow, a data.frame with columns "flow", "day", "month" and "year".

Usage

```
createlfobj(x, ...)


## S3 method for class 'data.frame'
createlfobj(x, hyearstart = NULL, baseflow = TRUE,
            meta = list(),...)

## S3 method for class 'ts'
createlfobj(x,
            startdate,
            dateformat = "%d/%m/%Y",
            ...)

## S3 method for class 'lfobj'
createlfobj(x, hyearstart = NULL, baseflow = NULL,
            meta = NULL,...)
```

Arguments

`x`	An object out of which an object of class `'lfobj'` should be created
`hyearstart`	integer between 1 and 12, indicating the start of the hydrological year.
`baseflow`	logical, should the base flow curve be calculated? Needed, if you want to apply functions `'bfplot'` or `'BFI'` later on.
`meta`	A list of meta-information
`startdate`	start of the time-series
`dateformat`	Format of the start date
`...`	Additional arguments, passed on to `createlfobj.data.frame`.

Details

`'hyearstart'` defines the starting month of the hydrological year. If `'hyearstart'` is greater then 6.5, the hydrological year starts earlier then the actual date, e.g. `hyearstart = 10`, then the 1st of October 2011 is part of the hydrological year 2012. If `hyearstart = 4`, then the 31st of March 2011 is part of the hydrological year 2010.

When creating an object of class `lfobj` with the aforementioned functions, eventually `createlfobj.data.frame` is called.

Value

An object of class `'lfobj'`.

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

readlfdata

Examples

```
# Creating a lfobj from a timeseries
# Some sample data:
somevalues <- rexp(365)

# Convert to time series:
time <- ts(somevalues)

# Lets say our data contains values from one hydrological year (Oct-Sep)
# starting on 1. Oct. 1992:
myriver <- createlfobj(time, startdate = "01/10/1992",hyearstart = 10)

# Add meta-data
createlfobj(myriver, meta = list(river = "myriver"))
```

---

dmcurve *Double Mass Curve*

Description

Calculates the double mass curve of two object of class 'lfobj'.

Usage

```
dmcurve(x, y, year = "any", namex = substitute(x), namey = substitute(y),
    na.rm = TRUE)
```

Arguments

x	An object of class "lfobj"
y	An object of class "lfobj"
year	The year for which the double mass curve should be calculated
namex	character - Label of the x-Axis in the double mass curve
namey	character - Label of the y-Axis in the double mass curve
na.rm	Remove NAs?

Value

No return value, called for side effects (plotting).

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

Examples

```
data(ngaruroro)
n1 <- subset(ngaruroro, year %in% 1985:1989)
n2 <- subset(ngaruroro, year %in% 1990:1995)
dmcurve(n1,n2, namex = "'Ngaruroro 1985 - 1989'", namey = "'Ngaruroro 1990
- 1995'")
```

---

evfit — *Fit an extreme value distribution to observations*

Description

Fits an extreme value distribution using L-moments to the values provided. In the presence of zero flow observations a mixed distribution is fitted.

Usage

```
evfit(x, distribution, zeta = NULL, check = TRUE,
      extreme = c("minimum", "maximum"))
```

Arguments

`x`	numeric vector. Data which is an extreme value distribution is fitted to.
`distribution`	A character vector of distributions to fit. Basically all distributions provided by Hosking's `lmom-package` and their reversed counterparts can be chosen. See `reversing`.
`zeta`	numeric vector of length one for manually setting a lower bound. Only a few distributions allow for a lower bound, namely `'gpa'`, `'ln3'`, `'wak'` and `'wei'`. The default value of `NULL` results in not bounding the distribution, therefore the parameter `zeta` is estimated.
`check`	logical, should `check_distribution` get called?
`extreme`	character vector of length one. Can be either `'minimum'` or `'maximum'`. Helps to choose a correct distribution.

Details

This function is vectorized over `distribution`.

According to paragraph 7.4.2 of the WNO manual, special care has to be taken in the presence of zero flow observations. A cdf called G(x) is fitted to the non-zero values of the original time series.

If a distribution is fitted which allows for finite lower bound (`zeta`), and `zeta` is estimated being negative, estimation is repeated constraining `zeta = 0`. If this behavior is not desired, the parameter `zeta` has to be set explicitly.

Value

An object of class `'evfit'` containing the L-moments and the estimated parameters is returned. Objects of class `'evfit'` are basically a list with the following elements:

`values`	the values `x` used for fitting.
`freq.zeros`	a character vector of length one. Frequency of zero flow observations.
`is.censored`	logical, if the censored time was used for fitting.
`parameters`	a list as long as `distribution` containing the estimated parameters for each distribution.
`lmom`	sample L-moments of the censored series (only containing non-zero values).
`extreme`	character vector of length one, indicating what kind of extreme value was fitted.
`T_Years_Event`	optional. If quantiles have been computed they a stored in a matrix with return periods in rows and distributions in columns.

See Also

There are methods for printing summarizing objects of class `'evfit'`.

`evfit`

Examples

```
data("ngaruroro")
ng <- as.xts(ngaruroro)
minima <- as.vector(apply.yearly(ng$discharge, min, na.rm = TRUE))
evfit(x = minima, distribution = c("wei", "gevR"),
      extreme = "minimum")
```

`evquantile` *Estimating populations quantiles of extreme values*

Description

Computes population quantiles for given return periods. Estimation is done using L-moments.

Usage

```
evquantile(fit, return.period = NULL)
```

Arguments

fit — object of class `evfit`, possibly created with `evfit()`.

return.period — numeric vector of return periods

Details

This function is vectorized over `return.period`.

Value

A matrix containing the low-flow quantiles, with rows corresponding to return periods columns to distributions.

Examples

```
data("ngaruroro")

# using tyears is a fast way to produce an object of class evfit
y <- tyears(ngaruroro, dist = "wei", event = 100, plot = TRUE)

# computing quantiles for given return periods
rp <- c(1.42, 5, 10)
evquantile(y, return.period = rp)
rpline(y, return.period = rp, suffix = c("a", "m\u00B3"))
```

---

ev_return_period — *Estimate the return period for given quantiles*

Description

For discharges of interest, estimate the corresponding return period.

Usage

```
ev_return_period(x, fit)
```

Arguments

x — numeric vector containing the quantiles

fit — object of class `evfit` describing the underlying distribution, possibly created with `evfit()`.

Value

a numeric vector of return periods.

See Also

evfit

Examples

```
data("ngaruroro")
ng <- as.xts(ngaruroro)

# yearly minima
minima <- apply.yearly(ng$discharge, min, na.rm = TRUE)

# fit a Weibull distribution
fit <- evfit(x = as.vector(minima), distribution = "wei")

# compute return periods
minima$rp <- round(ev_return_period(minima, fit), 2)

print(minima)
plot(discharge ~ rp, data = minima,
     xlab = "Flow in m^3/s", ylab = "Return period in years")
```

---

fdc *Flow Duration Curve*

Description

Plots the flow duration curve for a given low flow object.

Usage

```
fdc(lfobj, year = "any", breakdays = NULL, colors = TRUE,
    xnorm = FALSE, ylog = TRUE, legend = TRUE, separate = FALSE,
    ...)
```

Arguments

lfobj	An object of class "lfobj"
year	numeric - The year for which the flow duration curve should be computed. If hyearstart != 1 the BFI is calculated for the hydrological year, any = "any" means the whole series should be taken.
breakdays	A vector of break days if the BFI should be calculated for different seasons.
colors	logical - If 'breakdays' are specified, should the different flow duration curves are displayed in different colours?
xnorm	logical - should the x-axis be normalized?
ylog	logical - The the logarithm of the y-axis?
legend	logical - Should a legend be plotted?
separate	logical - Should a separate plot be drawn for every season?
...	Graphical parameters handed to plot

Details

If breakdays is a single day, e.g. "01/06", the start of the hydrological year is taken as the second break day. If more than two seasons are to be specified, a vector of all break days is needed.

Value

A vector of quantiles.

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

ecdf

Examples

```
data(ngaruroro)
fdc(ngaruroro,year = 1991)
```

---

fill_na — *Interpolation NA values in a vector*

Description

This function is a tiny wrapper around approx which allows to contain the maximum number of NA values in a row that will be filled by interpolation. This is useful to obtain regular time series.

Usage

```
fill_na(x, max.len = Inf, ...)
```

Arguments

x	a vector, possibly containing NA values
max.len	an integer vector of length one, constraining the number of of consecutive NA observations which will get replaced with interpolated values
...	further arguments, passed on to approx.

Value

a vector

See Also

approx, na.approx

Examples

```
x <- 1:20
x[c(2, 3, 6, 11:15)] <- NA
fill_na(x, max.len = 2)
```

---

find_droughts *Identifying Low Flow Periods*

Description

A streamflow deficit is defined as an event where discharges are below a given threshold.

Usage

```
find_droughts(x, threshold = vary_threshold, varying = "constant", interval = "day", ...)
```

Arguments

x	an object which can be coerced to class 'xts'. Either with a single column or with a column named 'discharge'. Units of 'x' are retrieved from the attributes. If it fails, an error is raised.
threshold	The threshold can either be a constant value, a time series with the same length as x or a function (taking a single argument) returning either of these. Furthermore threshold can be a character vector of length one specifying a quantile of x like threshold = 'Q80' as a shortcut of threshold = function(x) quantile(x, 0.2, na.rm = TRUE). See vary_threshold for varying a threshold in time.
varying	if varying is a character vector of length one, values of "constant", "daily", "weekly" and "monthly" are allowed. If a vector of class POSIX is provided, a seasonal varying threshold is computed, where the times provided define the start of the season. Only the day of the year is taken, the year itself doesn't have a meaning.
interval	A character string, containing one of "day", "week", "month", "quarter" or "year" as accepted by seq.Date.
...	if threshold is a function, these additional arguments are passed on to the function

Value

an object of class 'deficit', which is basically an 'xts' object with the columns

discharge	discharges as provided with x
threshold	the threshold
def.increase	The increase of the deficit volume in m^3 per day.
event.no	an event id. If an event is numbered "0" this period not considered as a streamflow deficit.

See Also

There are summary and plot methods, see summary.deficit and plot.deficit.

pooling, summary.deficit, plot.deficit

Examples

```
data(ray)
ray <- as.xts(ray)["1970::1979", ]
r <- find_droughts(ray)
head(r)
summary(r)


plot(r)


# threshold is to low, because there are many days with
# zero flow observations
# provide threshold as a constant value
r <- find_droughts(ray, threshold = 0.02)
head(r)
summary(r)

plot(r)



# provide threshold as a function
r <- find_droughts(ray,
                   threshold = function(x) quantile(x, 0.2, na.rm = TRUE))
head(r)
summary(r)
```

flowunit	*Set and retrieve unit of the discharge*

Description

In order to compute deficit volumes time series of discharges (either of class 'lfobj' or 'xts') summary.deficit needs to be aware of the unit. Units are stored in the attributes of the time series. flowunit(x) retrieves the current unit from the attributes, flowunit(x) <- value sets a new one.

Usage

```
flowunit(x)
## S3 method for class 'xts'
flowunit(x)
## S3 method for class 'lfobj'
flowunit(x)

flowunit(x) <- value
## S3 replacement method for class 'xts'
flowunit(x) <- value
## S3 replacement method for class 'lfobj'
flowunit(x) <- value
```

Arguments

x	The time series, either of class 'lfobj' or 'xts'.
value	a valid character string of length one that can be interpreted as flow unit. See details.

Details

Currently, just a few functions like summary.deficit and lfstat:::plot.deficit_dygraph make use of the unit stored as an attribute.

Usually flow units are of dimension $\mathsf{L}^3\mathsf{T}^{-1}$. Currently a length l can be on of c("metre", "cm", "centimetre" "litre"), whereas time T can be one in c("days", "hours", "mins", "secs"), possibly abbreviated. The numerator of the fraction (everything before the literal "/") is interpreted as the length (superscripts like "^3" are discarded), the denominator as time. E.g. valid units would be "cm^3/s", "m^3/day" or "litre/sec".

Value

A character vector of length one, containing the currently used discharge unit.

Examples

```
data(ray)
ray <- as.xts(ray)["1970::1970", ]

# currently discharges are in cubic metres per second
flowunit(ray)

# calculating deficit volumes, for fixed threshold 0.001 m^3/s
(s <- summary(find_droughts(ray, threshold = 0.001)))

# multiplying the discharge by 1000 converts is to litre per second
ray$discharge <- ray$discharge * 1000

# changing the unit accordingly, yields the same volumes
flowunit(ray) <- "l/s"
(ss <- summary(find_droughts(ray, threshold = 1)))

identical(s$volume, ss$volume)
```

`gringorten` *Gringorten Plotting Positions*

Description

Computes the Gringorten Plotting position.

Usage

```
gringorten(x)
```

Arguments

`x`	numeric vector

Value

numeric vector in `[0, 1]`, giving the corresponding plotting positions.

Examples

```
y <- rnorm(10)
pp <- gringorten(y)
pp

plot(pp ~ y, ylim = c(0, 1))
```

hydrograph *Hydrograph*

Description

Plots the hydrograph for a given period.

Usage

```
hydrograph(lfobj, startdate = NULL, enddate = NULL, amin = FALSE,...)
```

Arguments

`lfobj`	An object of class `"lfobj"`
`startdate`	Begin of hydrograph, date or hydrological year
`enddate`	End of hydrograph, date or hydrological year
`amin`	logical, mark annual minima?
`...`	Additional arguments handed to "plot" - please note that some changes e.g. tick-marks on x-axis are not possible

Details

Start date and end date can be `NULL` (first/last date in a low flow object), a date in format `"dd/mm/yyyy"` (e.g. "01/10/1971") or a year `"yyyy"` (e.g 1961).

Value

Plot of hydrograph

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

`bfplot`

Examples

```
data(ngaruroro)
# Full period
hydrograph(ngaruroro)

# Hydrological year 1981 and 1982 with annual minima
hydrograph(ngaruroro, startdate = 1981, enddate = 1982, amin = TRUE)

# From 01/01/1981 to 31/03/1981
hydrograph(ngaruroro, startdate = "01/01/1981", enddate = "31/03/1981")

# Log - yaxis
hydrograph(ngaruroro, startdate = "01/01/1981", enddate = "31/03/1981",log = "y")
```

---

hyear_start *Extract or guess the Start of a Hydrological Year*

Description

Retrieve the start of a hydrological year either from the attributes or from the column 'hyear' of an object of class 'lfobj'.

Usage

```
hyear_start(x, abbreviate = FALSE)

## S3 method for class 'data.frame'
hyear_start(x, abbreviate = FALSE)

## S3 method for class 'xts'
hyear_start(x, abbreviate = FALSE)

hyear_start(x) <- value
## S3 replacement method for class 'xts'
hyear_start(x) <- value
## S3 replacement method for class 'lfobj'
hyear_start(x) <- value
```

Arguments

x	object of which the start of the hydrological year should be determined.
abbreviate	logical. Should the names be abbreviated?
value	numeric vector of length one. Month in which the hydrological year starts.

Details

If a valid start of an hydrological year is found in the attributes, it is returned. Otherwise if a column 'hyear' exists, it is used. If this is note possible the integer number one is returned (for January) and a warning is issued.

Value

a vector of length one, either of type character (`abbreviate = TRUE`) or numeric.

See Also

`water_year`

Examples

```
data(ngaruroro)
hyear_start(ngaruroro)

data(ray)
hyear_start(ray, abbreviate = TRUE)
```

`lfnacheck` *Low flow object check for missing values.*

Description

Looks for NAs in a low flow object.

Usage

```
lfnacheck(lfobj)
```

Arguments

`lfobj`	An object of class `"lfobj"`

Value

A list with the total number of NAs, the percentage, the NAs for every year and the durations of NA-series.

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

`bfplot`

Examples

```
data(ngaruroro)
lfnacheck(ngaruroro)
```

`lfnainterpolate` *Interpolate missing values*

Description

If a low flow object contains missing values, the missing values are replaced by connecting the last available value before the break and the first after the break by a straight line.

Usage

```
lfnainterpolate(lfobj)
```

Arguments

`lfobj`	An object of class "`lfobj`"

Value

`lfobj`	An object of class "`lfobj`"

with interpolated missing values

Warning

Check carefully in advance if interpolation is a reasonable choice for filling the hydrograph

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

`bfplot`

Examples

```
data(ngaruroro)

# Part of the ngaruroro series with missing data
hydrograph(ngaruroro, startdate = "1/7/1987", enddate = "1/9/1987",amin = FALSE)

ngaruroroint <- lfnainterpolate(ngaruroro)

# The completed hydrograph
hydrograph(ngaruroroint, startdate = "1/7/1987", enddate = "1/9/1987",amin = FALSE)
```

`ma` *Simple Moving Average*

Description

Smoothing a time series with moving averages using the `filter` function.

Usage

```
ma(x, n, sides = "past")
```

Arguments

`x`	numeric vector to be smoothed
`n`	numeric vector of length one determining the width of the smoothing window
`sides`	one of `'past'`, `'center'` or `'future'` indicating the side of the filter.)

Value

a vector as long as x, but smoothed. Possibly with NAs.

See Also

`filter`

Examples

```
ma(1:10, n = 3, sides = 2)    # centred around lag 0
ma(1:10, n = 3)               # past values
```

MAM *Mean Annual Minimum*

Description

Computes the Mean Annual Minimum (MAM-n) for any given n.

Usage

```
MAM(lfobj, n = 7, year = "any", breakdays = NULL, yearly = FALSE)
```

Arguments

`lfobj`	An object of class `"lfobj"`
`n`	Mean Annual minimum for n-days, e.g. n=7 computes MAM7
`year`	The year for which the BFI should be computed. If `hyearstart != 1` the BFI is calculated for the hydrological year! `'any'` means the whole series should be taken. If a vector of years is given, all this years are included in the calculation.
`breakdays`	A vector of break days if the BFI should be calculated for different seasons.
`yearly`	If TRUE, the BFI is calculated for each hydrological year separately.

Details

If `breakdays` is a single day, e.g. `"01/06"`, the start of the hydrological year is taken as the second break day. If more than two seasons are to be specified, a vector of all break days is needed.

Value

A length one vector giving the BFI for the whole series or the specified year. If yearly is true, a vector of the annual BFIs is returned. If break days are specified, separated values for every season are given.

Warning

At the moment there is no check for seasonal overlap. E.g. The MAM7 of 1991 and 1992 could take the same days for calculation if the are in $n/2 - days$ range. This problem could be avoided by choosing a "meaningful" `hyearstart` and `breakdays`, usually dates out of the low flow seasons.

Note

The annual minima can be calculated by setting `n = 1` and `yearly = TRUE`.

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

meanflow,Q95

Examples

```
data(ngaruroro)
MAM(ngaruroro)
MAM(ngaruroro, n=1) #Mean annual minimum
MAM(ngaruroro, year = c(1991,1995)) #Taking values from 1991 and 1995
MAM(ngaruroro, year = 1991:1995) #Taking values from 1991 to 1995 (1991,1992,...,1995)
MAM(ngaruroro, breakdays = c("01/11","01/05"))
MAM(ngaruroro, year = 1991)
```

---

meanflow *Mean flow*

Description

Calculates the mean flow of an object of class 'lfobj'.

Usage

```
meanflow(lfobj, year = "any", monthly = FALSE, yearly = FALSE,
breakdays = NULL, na.rm = TRUE)
```

Arguments

lfobj	An object of class "lfobj"
year	The year for which the mean flow should be computed. If hyearstart != 1 the mean flow is calculated for the hydrological year! "any" means the whole series should be taken.
monthly	logical - Should the mean flow be calculated separately for every month?.
yearly	logical - If TRUE, the mean flow is calculated for each hydrological year separately.
breakdays	A vector of break days if the mean flow should be calculated for different seasons.
na.rm	Should missing values be ignored?

Details

If 'breakdays' is a single day, e.g. "01/06", the start of the hydrological year is taken as the second break day. If more than two seasons are to be specified, a vector of all break days is needed.

Value

A length one vector giving the mean flow for the whole series or the specified year. If yearly is true, a vector of the annual mean flows is returned. If break days are specified, the values are separated per season.

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

MAM

Examples

```
data(ngaruroro)
meanflow(ngaruroro)
meanflow(ngaruroro, breakdays = c("01/11","01/05"))
meanflow(ngaruroro, year = 1991)
```

---

multistationsreport *Report for several stations*

Description

Calculates indices for several stations at once.

Usage

```
multistationsreport(...,indices = c("meanflow", "Q95", "MAM1", "MAM7",
    "MAM10", "MAM30", "MAM90", "baseflowindex", "recession"),
    recessionmethod = "MRC", recessionseglength = 7, recessionthreshold = 70,
    recessiontrimIRS = 0.1,lflist = NULL)
```

Arguments

`...`	Objects of class "lfobj"
`indices`	A vector of indices to calculate
`recessionmethod`	See `recession`
`recessionseglength`	See `recession`

recessionthreshold
: See recession

recessiontrimIRS
: See recession

lflist
: Alternative give a list containing low flow objects.

Value

A `data.frame` containing the calculated indices.

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

meanflow, Q95,MAM,BFI,recession

Examples

```
data(ngaruroro)
multistationsreport(ngaruroro, indices = c("meanflow", "MAM7"))

seventies <- subset(ngaruroro, hyear %in% 1970:1979)
eighties <- subset(ngaruroro, hyear %in% 1980:1989)
nineties <- subset(ngaruroro, hyear %in% 1990:1999)

multistationsreport(seventies, eighties, nineties)
```

---

ngaruroro — *Daily stream flow data used for low flow analysis*

Description

This data set provides the streamflow records for the rivers Ngaruroro (New Zealand) and Ray (UK). They are provided as a low flow object (class `'lfobj'`) as used in the package lfstat. The user might want to perform analysis with shorter time series. The data set `ng` just contains the eighties (hydrological year 1980 – 1989) of the Ngaruroro discharges.

Usage

```
data(ngaruroro)
data(ng)
data(ray)
```

Format

A low flow object, createlfobj

Source

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

Examples

```
data(ngaruroro)
hyear_start(ngaruroro)
plot(ngaruroro)

data(ray)
hyear_start(ray)
attr(ray, "lfobj")
```

`plot.deficit` *Plot time series of deficits*

Description

Plot method for objects of class deficit.

Usage

```
## S3 method for class 'deficit'
plot(x, type = "dygraph", ...)
```

Arguments

x	object of class deficit
type	if type = "dygraph" interactive time series plotting is done using the **dygraphs** JavaScript library. Otherwise plot.xts is called.
...	further arguments, passed on to the subsequent plot function, e.g. step = FALSE.

Value

An interactive dygraph plot or an xts plot, depending on argument 'type'.

See Also

dygraph

Examples

```
data(ray)
r <- find_droughts(ray, threshold = 0.02)
plot(r["1970::1970", ])

plot(r["1970::1970", ], step = FALSE)
```

---

pooling *Pooling Procedures of Low Flow Events*

Description

Several pooling procedures can be applied to reduce the number of dependent droughts.

Usage

```
pool_ic(x, tmin = 5, ratio = 0.1)
pool_it(x, tmin = 5)
pool_ma(x, n = 10)
pool_sp(x)
```

Arguments

x	an object of class deficit, e.g. as produced by find_droughts.
tmin	numeric vector of length one interpreted as the number of days between two droughts to be considered independent events. Two droughts are pooled if their inter-event time is less than tmin.
ratio	numeric vector of length. Specifies the minimum ratio of inter-event volume and precedent drought volume. Two droughts are pooled if the critical ratio is deceeded.
n	numeric vector of length one determining the width of the smoothing window

Details

The inter-event criterion (pool_ic) pools subsequent drought events if the inter-event time is less than tmin and the ratio of the drought volume and the inter-event volume is less than a given ratio. The function pool_it is simply a wrapper around pool_ic(..., ratio = Inf).

Pooling by a moving average (pool_ma) simply smooths the time series before finding drought events.

Using the Sequent Peak algorithm (pool_sp), a drought lasts until its cumulative deficit volume is zero again.

Value

an object of class deficit (inherited from xts), with an additional column event.orig.

See Also

find_droughts, summary.deficit

Examples

```
data(ngaruroro)
ng <- as.xts(ngaruroro)
ng <- ng["1986::1990", ]

drought <- find_droughts(ng)


ic <- pool_ic(drought)
summary(ic)

ma <- pool_ma(drought)
summary(ma)

sp <- pool_sp(drought)
summary(sp)
plot(sp)
```

Qxx *Qxx, Q95, Q90, Q70*

Description

Calculates the quantiles of an object of class 'lfobj'.

Usage

```
Qxx(lfobj, Qxx, year = "any", monthly = FALSE, yearly = FALSE,
breakdays = NULL, na.rm = TRUE)

Q95(lfobj, year = "any", monthly = FALSE, yearly = FALSE,
breakdays = NULL, na.rm = TRUE)

Q90(lfobj, year = "any", monthly = FALSE, yearly = FALSE,
breakdays = NULL, na.rm = TRUE)

Q70(lfobj, year = "any", monthly = FALSE, yearly = FALSE,
breakdays = NULL, na.rm = TRUE)
```

Arguments

lfobj	An object of class 'lfobj'
Qxx	The quantile to calculate, e.g. 70 would refer to Q70

`year`	The year for which the Q95 should be computed. If `hyearstart != 1` the Q95 is calculated for the hydrological year, `year = "any"` means the whole series should be taken.
`monthly`	logical - Should the Q95 be calculated separately for every month?.
`yearly`	logical - If TRUE, the Q95 is calculated for each hydrological year separately.
`breakdays`	A vector of break days if the Q95 should be calculated for different seasons.
`na.rm`	Should NAs be ignored?

Details

If `breakdays` is a single day, e.g. "01/06", the start of the hydrological year is taken as the second break day. If more than two seasons are to be specified, a vector of all break days is needed.

Value

A length one vector giving the Q95 for the whole series or the specified year. If yearly is true, a vector of the annual Q95s is returned. If break days are specified, the values are separated per season.

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

MAM

Examples

```
data(ngaruroro)
Q95(ngaruroro)
Q95(ngaruroro, breakdays = c("01/11","01/05"))
Q95(ngaruroro, year = 1991)
# Calculate Q99
Qxx(ngaruroro, Qxx = 99)
```

readlfdata *Reads data sheets*

Description

Reads data sheets of different formats directly and returns objects of class 'lfobj'.

Usage

```
readlfdata(file, type = c("GRDC", "HZB", "LFU", "TU"), lfobj = TRUE,
           readmeta = TRUE, encoding = NULL, ...)
```

Arguments

file	The name of the file which the data are to be read from.
type	The style of the sheet, currently the following formats are accepted: 'GRDC', 'HZB' (Austria), 'LFU' (Germany, Bavaria), 'TU' (Technical University Vienna)
lfobj	logical, should a low flow object be created?
readmeta	logical, should meta information from data sheets be saved?
encoding	The name of the encoding to be assumed. See the Encoding section of connections.
...	Handed to 'createlfobj', could be 'hyearstart', 'baseflow' or 'meta', if 'readmeta' is 'FALSE'.

Value

A 'lfobj' or 'data.frame' depending on 'lfobj'.

Note

If you like other file formats (national standards) to be includes, send some examples with a remark how NAs are marked to the author

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

createlfobj

Examples

```
# Finding the filename of the sample file on your computer
fn <- system.file("samplesheets/9104020.day", package = "lfstat")
grdc <- readlfdata(fn, type = "GRDC", baseflow = FALSE, hyearstart = 1)
head(grdc)


fn <- system.file("samplesheets/kloesterle.dat", package = "lfstat")
hzb <- readlfdata(fn, type = "HZB", baseflow = FALSE, hyearstart = 1)
head(hzb)

fn <- system.file("samplesheets/oberammergau.dat", package = "lfstat")
lfu <- readlfdata(fn, type = "LFU", baseflow = FALSE, hyearstart = 1)
head(lfu)
```

---

recession *Recession Constant*

Description

Does recession analysis using either the MRC (Master recession curve) or IRS (individual recession segments) method.

Usage

```
recession(lfobj,
          method = c("MRC", "IRS"),
          seglength,
          threshold,
          peaklevel = 0.95,
          seasonbreakdays = NULL,
          thresbreaks = c("fixed", "monthly","seasonal"),
          thresbreakdays = NULL,
          plotMRC = TRUE,
          trimIRS = 0,
          na.rm = TRUE)
```

Arguments

`lfobj`	An object of class `'lfobj'`
`method`	`'MRC'` or `'IRS'`
`seglength`	The length of the duration segments - see the WNO-manual and use `seglenplot` to choose a good value.
`threshold`	The threshold level (70 means Q70)
`peaklevel`	A level between 0 and 1 or a logical vector, see details.

seasonbreakdays
: A vector of break days. Needed if the recession constant should be calculated individually for different seasons, see details.

thresbreaks
: 'fixed' uses a fixed threshold level, 'monthly' calculates the threshold for every month separately, 'seasonal' calculates thresholds for every season defined using 'thresbreakdays'.

thresbreakdays
: Needed if thresbreaks = 'seasonal' to define the periods for which separate thresholds should be calculated, see details

plotMRC
: logical, if TRUE and method = 'MRC' a plot like figure 5.4 in the manual is given.

trimIRS
: Should a trimmed mean be used for calculating the IRS-constant? (0 means no, 0.1 means trim by 10 %)

na.rm
: Should NAs in the series be ignored?

Details

For recession analysis it is necessary to define flood discharge peaks in the hydrograph. Argument peaklevel defines a day to be a discharge peak, if $peaklevel * flow > flow[daybefore]$ and $peaklevel * flow > flow[dayafter]$. Use recessionplot to find a good level or hand a logical vector where TRUE means rain peak.

If 'thresbreakdays' or 'seasonbreakdays' is a single day, e.g. '01/06', the start of the hydrological year is taken as the second break day. If more than two seasons are to be specified, a vector of all break days is needed.

Value

The overall recession rate in days. If seasons are defined a rate for every season is calculated.

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

seglenplot, recessionplot

Examples

```
data(ngaruroro)
recession(ngaruroro,method = "MRC",seglen = 7,threshold = 70)
```

recessionplot *Recession diagnostic plot*

Description

Helps to define the peak level of a low flow object and visualises recession periods.

Usage

```
recessionplot(lfobj,
              peaklevel = 0.95,
              plot = TRUE,
              peakreturn = FALSE,
              thresplot = TRUE,
              threscol = "blue",
              threshold = 70,
              thresbreaks = c("fixed","monthly","seasonal"),
              thresbreakdays = c("01/06","01/10"),
              recessionperiod = TRUE,
              recessioncol = "darkblue",
              seglength = 7,
              ...)
```

Arguments

`lfobj`	A object of class 'lfobj'
`peaklevel`	A level between 0 and 1 or a logical vector, see details.
`plot`	Should a plot be made
`peakreturn`	Should a logical with rain peaks be returned
`thresplot`	Should the threshold be plotted
`threscol`	Colour of threshold in plot
`threshold`	Threshold level (70 refers to Q70)
`thresbreaks`	"fixed" uses a fixed threshold level, "monthly" calculates the threshold for every month separately, "seasonal" calculates thresholds for every season defined using 'thresbreakdays'.
`thresbreakdays`	Needed if thresbreaks = 'seasonal' to define the periods for which separate thresholds should be calculated, see details
`recessionperiod`	Should recession periods be marked
`recessioncol`	Colour of recession period marks
`seglength`	The minimum number of days to be marked as recession period
`...`	Further arguments handed to hydrograph

.

Details

For recession analysis it is necessary to define flood discharge peaks in the hydrograph. The peak level defines a day to be a discharge peak, if `peaklevel * flow > flow[day before]` and `peaklevel * flow > flow[day after]`.

This function can be used to check different values of the peak level.

Value

If `'peakreturn = TRUE'`: A logical vector giving rain peaks as `TRUE`

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

`recession`

Examples

```
data(ngaruroro)
# To few points identified as peak flood discharge
recessionplot(ngaruroro, peaklevel = .5, start = 1991, end = 1991)

# To many
recessionplot(ngaruroro, peaklevel = .999, start = 1991, end = 1991)

# Good choice?
recessionplot(ngaruroro, peaklevel = .92, start = 1991, end = 1991)

# Getting peakdays for 1991
peak <- recessionplot(ngaruroro, peaklevel = .92, plot = FALSE, peakreturn = TRUE)
rain1991 <- subset(ngaruroro, subset = (hyear == 1991) & peak, select = c(day, month, year))
```

`reversing` *Reversed functions for several Extreme Value Distributions*

Description

As several Extreme Value distributions are parameterized for high extreme values, reversed functions for minima (e.g. low flow statistics) are derived. Reversing is done by fitting to the negated data (`-x`), subtracting probabilities from one (`1 - f`) and computing the negated probabilities.

Usage

```
cdf_ev(distribution, x, para)
pel_ev(distribution, lmom, ...)
qua_ev(distribution, f, para)
```

Arguments

`distribution`	character vector of length one containing the name of the distribution. The family of the chosen distribution must be supported by the package **lmom**. See `lmom`. For example `distribution = "gev"` directly uses the functions from package **lmom**, whereas `distribution = "gevR"` performs reversing.
`x`	Vector of quantiles.
`f`	Vector of probabilities.
`para`	Numeric vector containing the parameters of the distribution, in the order zeta, beta, delta (location, scale, shape).
`lmom`	Numeric vector containing the L-moments of the distribution or of a data sample. E.g. as returned by `samlmu(x)`.
`...`	parameters like bound, passed on to the estimating function. E.g. in case of `dist = 'wei'` to `pelwei`.

Value

`'cdf_ev'` gives the distribution function; `'qua_ev'` gives the quantile function.

See Also

`lmom`, `cdfgev`, `cdfgev`, `pel-functions`.

Examples

```
data("ngaruroro")
ng <- as.xts(ngaruroro)
minima <- as.vector(apply.yearly(ng$discharge, min, na.rm = TRUE))

# Weibull distribution and reversed GEV give the same results
distr <- "wei"
qua_ev(distr, seq(0, 1, 0.1), para = pel_ev(distr, samlmu(minima)))

distr <- "gevR"
qua_ev(distr, seq(0, 1, 0.1), para = pel_ev(distr, samlmu(minima)))
```

rfa *Regional Frequency Analysis*

Description

This function uses J.R.M. Hosking's package produce an object of class 'rfd', containing the specification of the regional frequency distribution.

Usage

```
rfa(lflist, n = 7, event = 100, dist = c("wei", "gev", "ln3", "gum", "pe3"))
```

Arguments

lflist	A list of 'lfobj's.
n	MAM-n is used (e.g. n=7 means MAM7).
event	A value for T, e.g. event = 100 means the 100 years extreme low flow event.
dist	A vector of distribution to fit, the names are according to Hosking's in his **lmom** package. Can be an of "wei", "gev", "ln3", "gum", "pe3".

Value

An object of class "rfd", containing the specification of the regional frequency distribution: It is a list with the following elements:

dist	The character string dist.
para	Vector containing the parameters of the fitted regional distribution.
qfunc	The quantile function of distribution dist. It is a function that takes a single argument, a vector of probabilities, and returns a vector of quantiles.
rmom	The regional average L-moments.
index	Index flood values at each site. This is a named vector whose values are the index flood values at each site, from regdata[[3]], and whose names are the site names, from regdata[[1]].

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p. https://library.wmo.int/idurl/4/32176

See Also

regfit and lmom-package which this function wraps.

Examples

```
data(ngaruroro)
# Toy example to get some more "rivers"
seventies <- subset(ngaruroro, hyear %in% 1970:1979)
eighties <- subset(ngaruroro, hyear %in% 1980:1989)
nineties <- subset(ngaruroro, hyear %in% 1990:1999)

toyrfa <- rfa(list(seventies,eighties,nineties), n=3,dist = "gev")

require(lmomRFA)
regquant(c(1/1000,1/100),toyrfa)
sitequant(1/100,toyrfa)
```

`rfaplot` *Regional Frequency Analysis*

Description

This function uses J.R.M. Hosking's package **lmom** to produce a L-moment diagram.

Usage

```
rfaplot(lflist, n = 7, ...)
```

Arguments

`lflist`	A list of `'lfobj'`s.
`n`	MAM-n is used (e.g. n=7 means MAM7).
`...`	Arguments passed on to `lmom::lmrd`

`x` Numeric vector of L-skewness values.
Alternatively, if argument `y` is omitted, `x` can be an object that contains both L-skewness and L-kurtosis values. It can be a vector with elements named `"t_3"` and `"t_4"` (or `"tau_3"` and `"tau_4"`), a matrix or data frame with columns named `"t_3"` and `"t_4"` (or `"tau_3"` and `"tau_4"`), or an object of class `"regdata"` (as defined in package **lmomRFA**).

`y` Numeric vector of L-kurtosis values.

`distributions` Indicates the three-parameter distributions whose L-skewness–L-kurtosis relations are to be plotted as lines on the diagram. The following distribution identifiers are recognized, in upper or lower case:

`GLO`	generalized logistic
`GEV`	generalized extreme-value
`GPA`	generalized Pareto
`GNO`	generalized normal
`PE3`	Pearson type III
`WAK.LB`	lower bound of L-kurtosis for given L-skewness,

for the Wakeby distribution.

`ALL.LB` lower bound of L-kurtosis for given L-skewness, for all distributions.

The argument should be either a character vector each of whose elements is one of the above abbreviations or a character string containing one or more of the abbreviations separated by blanks. The specified L-skewness–L-kurtosis curves will be plotted.
If no three-parameter distributions are to be plotted, specify `distributions` to be `FALSE` or the empty string, `""`.

`twopar` Two-parameter distributions whose (L-skewness, L-kurtosis) values are to be plotted as points on the diagram. The following distribution identifiers are recognized, in upper or lower case:

`E` or `EXP`	exponential
`G` or `GUM`	Gumbel
`L` or `LOG`	logistic
`N` or `NOR`	normal
`U` or `UNI`	uniform

The argument should be either a character vector each of whose elements is one of the above abbreviations or a character string containing one or more of the abbreviations separated by blanks. L-skewness–L-kurtosis points for the specified distributions will be plotted and given one-character labels.
The default is to plot the two-parameter distributions that are special cases of the three-parameter distributions specified in argument `distributions`. Thus for example if `distributions="GPA PE3"`, the default for `twopar` is `"EXP NOR UNI"`: NOR is a special case of PE3, UNI of GPA, EXP of both GPA and PE3.
If no two-parameter distributions are to be plotted, specify `twopar` to be `FALSE` or the empty string, `""`.

`xlim` x axis limits. Default: `c(0, 0.6)`, expanded if necessary to cover the range of the data.

`ylim` y axis limits. Default: `c(0, 0.4)`, expanded if necessary to cover the range of the data.

`pch` Plotting character to be used for the plotted (L-skewness, L-kurtosis) points.

`cex` Symbol size for plotted points, like graphics parameter `cex`.

`col` Vector specifying the colors. If it is of length 1 and `x` is present, it will be used for the plotted points. Otherwise it will be used for the lines on the plot. For the default colors for the lines, see the description of argument `lty` below.

`lty` Vector specifying the line types to be used for the lines on the plot.
By default, colors and line types are matched to the distributions given in argument `distributions`, as follows:

`GLO`	blue, solid line
`GEV`	green, solid line

GPA	red, solid line
GNO	black, solid line
PE3	cyan, solid line
WAK.LB	red, dashed line
ALL.LB	black, dashed line

The green and cyan colors are less bright than the standard "green" and "cyan"; they are defined to be "#00C000" and "#00E0E0", respectively.

lwd Vector specifying the line widths to be used for the lines on the plot.

legend.lmrd Controls whether a legend, identifying the L-skewness–L-kurtosis relations of the three-parameter distributions, is plotted. Either logical, indicating whether a legend is to be drawn, or a list specifying arguments to the legend function. Default arguments include bty="n", which must be overridden if a legend box is to be drawn; other arguments set by default are x, y, legend, col, lty, and lwd.
Not used if distributions is FALSE.

xlegend,ylegend x and y coordinates of the upper left corner of the legend. Default: coordinates of the upper left corner of the plot region, shifted to the right and downwards, each by an amount equal to 1% of the range of the x axis.
Not used if distributions is FALSE or if legend.lmrd is FALSE.

xlab X axis label.

ylab Y axis label.

Value

A list, returned invisibly, describing what was plotted. Useful for customization of the legend, as in one of the examples below. List elements:

lines	List containing elements describing the plotted distribution curves (if any). Each element is a vector with the same length as distributions. List elements distributions, col.lines, lty, lwd.
twopar	Character vector containing the 1-character symbols for the two-parameter distributions that were plotted.
points	List containing elements describing the plot (if any) of the data points. List elements col.pts, pch, cex.

If any of the above items was not plotted, the corresponding list element is NULL.

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p. https://library.wmo.int/idurl/4/32176

See Also

lmrd and lmom-package which this function wraps.

Examples

```
data(ngaruroro)
# Toy example to get some more "rivers"
seventies <- subset(ngaruroro, hyear %in% 1970:1979)
eighties <- subset(ngaruroro, hyear %in% 1980:1989)
nineties <- subset(ngaruroro, hyear %in% 1990:1999)

rfaplot(list(seventies, eighties, nineties), n = 3)
```

---

rpline *Highlight quantiles/return periods*

Description

Draw a Line in an extreme value plot corresponding to a given return period.

Usage

```
rpline(fit, return.period = NULL, log = TRUE, ...)
```

Arguments

`fit`	object of class `evfit`, possibly created with `evfit()`.
`return.period`	numeric vector of return periods
`log`	logical. If `TRUE` it is assumed that probabilities were plotted on a double logarithmic scale.
`...`	other arguments, passed on to `trace_value`

Details

Computes the corresponding quantiles and draws lines and labels.

Value

This function is used for its side effects

Examples

```
data("ngaruroro")
y <- tyears(ngaruroro, dist = "wei", event = 100, plot = TRUE)
rp <- c(1.42, 5, 10)
rpline(y, return.period = rp, suffix = c("a", "m\u00B3"))
```

sbplot *Seasonal Bar Chart*

Description

Plots a seasonal bar chart for daily streamflow data

Usage

```
sbplot(lfobj, hyearorder = TRUE)
```

Arguments

lfobj	A low flow object, as created with `createlfobj`
hyearorder	logical, if TRUE the bars are plotted according to the hydrological year, if FALSE they start with January.

Value

An object of class `trellis`, see `barchart`.

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

`createlfobj`

Examples

```
data(ngaruroro)
sbplot(ngaruroro)

# Starting with january
sbplot(ngaruroro, hyearorder = FALSE)
```

`seasindex` *Seasonality Index*

Description

Calculates the seasonality index.

Usage

```
seasindex(lfobj,
          Q = 95,
          na.rm = TRUE)
```

Arguments

`lfobj`	An object of class `"lfobj"`
`Q`	Which quantile to use (standard = Q95)
`na.rm`	Should missing values be ignored?

Value

A list describing the arrow

`theta`	Angle in radians
`D`	Julian Date
`r`	Length

Author(s)

Daniel Koffler and Gregor Laaha

References

Laaha, G. and Blöschl, G. (2006), Seasonality indices for regionalizing low flows. Hydrol. Process., 20

Laaha, G. Process Based Regionalisation of Low Flows, Band 198 von Wiener Mitteilungen, Inst. für Wasserbau u. Ingenieurhydrologie, Techn. Univ. Wien, 2006, ISBN 3852340896

See Also

`seasindex`

Examples

```
data(ngaruroro)

# Start of the hydrological year (01/12) is taken as second break day
seasindex(ngaruroro)
```

season *Attribute dates to seasons*

Description

Based on a vector of breaks (start dates) dates are classified into seasons.

Usage

```
season(x, start = c(winter = as.Date("2005-12-01"),
                    spring = as.Date("2005-03-01"),
                    summer = as.Date("2005-06-01"),
                    autumn = as.Date("2005-09-01")))
```

Arguments

x	Vector of dates to be classified into seasons. Methods for class 'numeric', 'Date' and 'POSIXct' exist. If input is numeric it is assumed to be the day of the year (see strptime '%j').
start	Possibly named vector of starts of a season. If the vector is unnamed generic names are used and a warning is risen.

Value

Factor of classifications of seasons.

See Also

link{apply.seasonal}

Examples

```
# input vector is of class Date
times <- seq(from = Sys.Date(), to = Sys.Date() + 500, by = 20)
season(times)

# input vector is numeric (the day of the year)
n <- as.numeric(format(times, "%j"))
season(n)

identical(season(times), season(n))
```

seasratio *Seasonality Ratio*

Description

Calculates the seasonality ratio for two seasons.

Usage

```
seasratio(lfobj,
          breakdays,
          Q = 95)
```

Arguments

lfobj	An object of class "lfobj"
breakdays	One or two dates defining the summer/winter season
Q	Which quantile to use (standard = Q95)

Details

If 'breakdays' is a single day, e.g. "01/06", the start of the hydrological year is taken as the second break day. If other seasons are to be specified, a vector of two break days is needed.

Value

The seasonality ratio.

Author(s)

Daniel Koffler and Gregor Laaha

References

Laaha, G. and Blöschl, G. (2006), Seasonality indices for regionalizing low flows. Hydrol. Process., 20

See Also

seasindex

Examples

```
data(ngaruroro)

# Start of the hydrological year (01/12) is taken as second break day
seasratio(ngaruroro, breakdays = "01/07")

# Two breakdays
seasratio(ngaruroro, breakdays = c("01/03","01/09"))
```

---

seglenplot *Bar chart of recession length*

Description

Plots a bar chart to find a good value for argument 'seglength' when using recession.

Usage

```
seglenplot(lfobj,
           threslevel = 70,
           thresbreaks = c("fixed","monthly","seasonal"),
           thresbreakdays = NULL,
           rainpeaklevel = 0.95,
           na.rm = TRUE)
```

Arguments

lfobj	An object of class 'lfobj'
threslevel	The threshold level (70 means Q70)
thresbreaks	'fixed' uses a fixed threshold level, 'monthly' calculates the threshold for every month separately, 'seasonal' calculates thresholds for every season defined using 'thresbreakdays'.
thresbreakdays	Needed if thresbreaks = 'seasonal' to define the periods for which separate thresholds should be calculated, see details
rainpeaklevel	A level between 0 and 1 or a logical vector, see details.
na.rm	Should NAs in the series be ignored?

Details

For recession analysis it is necessary to define flood discharge peaks (rain peaks) in the hydrograph. Argument rainpeaklevel defines a day to be a discharge peak, if rainpeaklevel * flow > flow[day before] and rainpeaklevel * flow > flow[day after].

If 'thresbreakdays' or 'seasonbreakdays' is a single day, e.g. '01/06', the start of the hydrological year is taken as the second break day. If more than two seasons are to be specified, a vector of all break days is needed.

Value

A bar chart

Warning

Other then in the manual, we implemented a bar chart instead of a histogram. To save space, empty bars are not plotted!

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

recession

Examples

```
data(ngaruroro)
seglenplot(ngaruroro)
```

---

setlfunit — *Define the unit to use in low flow plots*

Description

Sets the option for the unit in plots.

Usage

```
setlfunit(string = "")
```

Arguments

`string`	String of the unit

Details

The unit string should be readable for the R-function `expression`, for common units see example below.

Value

No return value, called for side effects. For the current R session a unit for discharges is set.

Warning

No calculation on data is done by setting this string.

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

Examples

```
data(ngaruroro)
# Default: no unit
bfplot(ngaruroro, year = 1991)

# The plot does not change, just the y-label does!
setlfunit("m^3/s")
bfplot(ngaruroro,year = 1991)

# Some possible labels:
setlfunit("m^3/s")
setlfunit("m^{3}*s^{-1}")
setlfunit("scriptscriptstyle(frac(m^3,s))")
setlfunit("l/s")
setlfunit("l*s^{-1}")
setlfunit("scriptscriptstyle(frac(l,s))")
setlfunit("m^3/s/km^2")
setlfunit("m^3*s^{-1}*km^{-2}")
setlfunit("scriptscriptstyle(frac(m^3,s%.%km^2))")
setlfunit("l/s/km^2")
setlfunit("l*s^{-1}*km^{-2}")
setlfunit("scriptscriptstyle(frac(l,s%.%km^2))")
```

---

streamdef *Streamflow Deficit*

Description

Calculates the streamflow deficit. Deprecated, use find_droughts instead.

Usage

```
streamdef(lfobj,
          pooling = c("none", "MA", "IT", "IC"),
          threslevel = 70,
          thresbreaks = c("fixed","monthly","daily","seasonal"),
```

```
          breakdays = c("01/06","01/10"),
          MAdays = 7,
          tmin = 5,
          IClevel = 0.1,
          mindur = 0,
          minvol = 0,
          table = c("all", "volmax", "durmax"),
          na.rm = TRUE)
```

Arguments

`lfobj`	An object of class "`lfobj`"
`pooling`	The pooling procedure used, "MA" stands for moving average, "IT" is the inter event time and "IC" is Lena Tallaksen's inter event time and volume criterion.
`threslevel`	The threshold level, 70 means that Q70 should be used as threshold
`thresbreaks`	The periods for which separated thresholds should be used, `'fixed'` uses a constant threshold, `'monthly'` uses monthly breaks, `'daily'` takes daily threshold levels. If `'seasonal'` is specified, you can enter the break days manually using `'breakdays'`.
`breakdays`	A vector of break days if `thresbreaks = "seasonal"`. Please enter the break days using the format "
`MAdays`	If pooling = "MA" this is the number of days that should be averaged
`tmin`	Defines the number of days that low flow events must be separated within the "IT" or "IC" method.
`IClevel`	The ratio between inter-event excess volume in the "IC" method
`mindur`	The minimal duration of a low flow event in "IC" and "IT" method
`minvol`	The minimal deficit in a low flow period in "IC" and "IT" method
`table`	Should the output be a table of "`all`" deficit, "`volmax`" annual volume maxima or "`durmax`" annual duration maxima
`na.rm`	Should NAs be removed?

Details

When method `'MA'` is applied, the first and last `MAdays/2` are not averaged, their original value is taken instead!

Value

A data frame containing characteristics of all low flow periods.

`d`	The duration of the low flow event
`v`	The drought volume (negative Values, as it is a deficit)
`mi`	The drought magnitude, i.e. the (positive) ratio between deficit volume and deficit duration
`Qmin`	The minimum flow of the low flow period

startyear	Year of the start of the low flow period
startmonth	Month of the start of the low flow period
startday	Day of the start of the low flow period

Please note that when using the "IT" method the end date of the low flow period is not necessarily start date + duration.

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

streamdefplot, createlfobj, find_droughts

Examples

```
data(ngaruroro)
ng <- subset(ngaruroro, hyear > 1980)

# Full Table
streamdef(ng, pooling = "MA", MAdays = 6)

# Annual Volume-Maxima only
streamdef(ng, pooling = "MA", MAdays = 6,table = "volmax")
```

---

streamdefplot *Streamflow Deficit Plot*

Description

Gives a plot for a given hydrological year that shows deficit duration, occurrence and volume.

Usage

```
streamdefplot(lfobj, year, threslevel = 70, thresbreaks = c("fixed",
                  "monthly", "daily", "seasonal"), breakdays =
                  c("01/06", "01/10"))
```

Arguments

- `lfobj` An object of class `'lfobj'`
- `year` The hydrological year that should be plotted
- `threslevel` The threshold level, 70 means that Q70 should be used as threshold
- `thresbreaks` The periods for which separated thresholds should be used, `'fixed'` uses a constant threshold, `'monthly'` uses monthly breaks, `'daily'` takes daily threshold levels. If `'seasonal'` is specified, you can enter the break days manually using `'breakdays'`.
- `breakdays` A vector of break days if `thresbreaks = 'seasonal'`. Please enter the break days using the format `'%d/%m'`, e.g. c(`'01/03'`, `'01/09'`) uses the first of March an the first of September as break days.

Value

No return value, called for side effects (plotting).

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

`streamdef`

Examples

```
data(ngaruroro)
streamdefplot(ngaruroro, year = 1991)
```

---

summary.deficit *Object Summaries*

Description

Summarizes an object of class deficit. For every drought event the start, end as well as the drought volume and duration is listed.

Usage

```
## S3 method for class 'deficit'
summary(object, drop_minor = c(volume = "0.5%", duration = 5), ...)
```

Arguments

`object`	an object of class deficit, as produced by `find_droughts`.
`drop_minor`	a vector of length one or two, determining the filtering of minor droughts. If `drop_minor` is of length one and its value is zero, no filtering is applied. Also a numeric of character vector of length two with the named elements `volume` and `duration` is accepted. If a value contains the percentage (%) sign this percentage of the maximum duration or volume is used a the filter criterion.
`...`	currently ignored.

Value

a data.frame where each row corresponds to an event. There are summarizing columns

`event.no`	the event id
`start`	the starting day of the drought event
`time`	the day which the event is attributed to. Usually identical with column `start`, unless the object `x` is the result of the Sequent Peak Algorithm.
`volume`	the volume of the drought event in cubic meters
`duration`	the duration of the drought event in days
`dbt`	days below threshold. Number of days the discharge is lower than the given threshold.
`qmin`	the minimum discharge
`tqmin`	date of the minimum discharge

Examples

```
data(ray)
ray <- as.xts(ray)["1970::1970", ]
r <- find_droughts(ray, threshold = 0.02)
summary(r)      # minor events got filtered

summary(r, drop_minor = 0)      # no filtering
summary(r, drop_minor = c("volume" = 10000, "duration" = 5))
summary(r, drop_minor = c("volume" = "10%", "duration" = 5))
```

`trace_value` *Draw Paths to Points perpendicular to Coordinate Axis*

Description

To depict the distances in `x` and `y` direction to a point, draw lines and labels.

Usage

```
trace_value(x, y, digits = 0, annotate = TRUE, lab.x = x, lab.y = y, prefix = "",
            suffix = "", cex = 0.75, col = "blue", lty = 2, ...)
```

Arguments

x	numeric vector of x coordinates
y	numeric vector of y coordinates
digits	vector of length one or two, giving the number of digits used for rounding the label of the x and y coordinate.
annotate	logical, should the lines get annotated with labels?
lab.x	character vector of length one. Label of the x coordinate.
lab.y	character vector of length one. Label of the y coordinate.
prefix	vector of length one or two, text printed before the label of the x and y coordinate.
suffix	vector of length one or two, text printed after the label of the x and y coordinate.
cex	character expansion factor
col	colour used for text and lines
lty	line type
...	other graphical parameters, passed on to lines, points and text.

Details

This function is vectorised over x and y.

Value

No return value, called for side effects (plotting).

Examples

```
x <- c(-2, 3)
curve(sin, -2*pi, 2*pi, xname = "t")
trace_value(x, sin(x), digits = c(0, 1))
```

---

tyears *Calculate Low-Flow Quantiles for given Return Periods*

Description

Fits an extreme value distribution using L-moments to the minima of a time series of discharges and subsequently estimates quantiles (the so called T-years event) for given return periods. In the presence of zero flow observations a mixed distribution is fitted.

Usage

```
tyears(lfobj, event = 1/probs, probs = 0.01,
       dist = "wei", check = TRUE, zeta = zetawei, zetawei = NULL,
       plot = TRUE, col = 1, log = TRUE, legend = TRUE,
       rp.axis = "top", rp.lab = "Return period",
       freq.axis = TRUE,
       freq.lab = expression(paste("Frequency " *(italic(F)),
                                   " = Non-Exceedance Probability P ",
                                   (italic(X) <= italic(x)))),
       xlab = expression("Reduced variate,  " * -log(-log(italic(F)))),
       ylab = "Quantile",
       hyearstart = hyear_start(lfobj),
       n = NULL)
```

Arguments

`lfobj`	An object of class 'lfobj' or an object which can be coerced to class 'xts'. Either with a single column or with a column named 'discharge'.
`event`	numeric vector specifying the return periods. E.g. `event = 100` will yield the 100 years extreme low flow event.
`probs`	Alternate way to specify the return period of the event.
`dist`	A character vector of distributions to fit. Basically all distributions provided by Hosking's `lmom-package` and their reversed counterparts can be chosen.
`check`	logical, should `check_distribution` get called?
`zeta`	numeric vector of length one for manually setting a lower bound. Only a few distributions allow for a lower bound, namely 'gpa', 'ln3', 'wak' and 'wei'. The default value of `NULL` results in not bounding the distribution, therefore the parameter `zeta` is estimated.
`zetawei`	same as `zeta`
`plot`	logical. If `TRUE`, sample observations as well as estimated quantile functions are plotted.
`col`	numeric or character vector of length one or as long as `dist`, specifying the colour used for plotting.
`log`	logical. If `TRUE` probabilities will be plotted on a double logarithmic scale.
`legend`	logical, should a legend be added to the plot?
`rp.axis`	vector of length one, specifying if and how an additional scale bar for the return periods is drawn. Possible choices are 'bottom', 'top' and 'none'. Alternatively, the position of the scale bar can be specified as an real number between 0 and 1, indicating the y-position of the legend.
`rp.lab`	character vector, text above the scale bar for return periods
`freq.axis`	logical, should an additional abscissa showing the probabilities be drawn on top of the plot?
`freq.lab`	character vector, text above the probability axis
`xlab`	character vector, a label for the x axis

ylab	character vector, a label for the y axis
hyearstart	vector of length one, providing the start of the hydrological year. This is evaluated by water_year. The default is, to retrieve the values stored in the attributes of the lfobj.
n	Argument 'n' is deprecated and ignored. To apply a moving average, do it prior to calling 'tyears'. See section Examples.

Details

This function is vectorised over dist and event.

According to paragraph 7.4.2 of the WNO manual, special care has to be taken in the presence of zero flow observations. A cdf called G(x) is fitted to the non-zero values of the original time series

If a distribution is fitted which allows for finite lower bound (zeta), and zeta is estimated being negative, estimation is repeated constraining zeta = 0. If this behavior is not desired, the parameter zeta has to be set explicitly.

Value

An object of class 'evfit', see evfit.

Author(s)

Daniel Koffler and Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

There are methods for printing summarizing objects of class 'evfit'.

evfit

Examples

```
data("ngaruroro")
ng <- subset(ngaruroro, hyear %in% 1964:2000)

# vector of return periods
rp <- c(1.5, 5, 10, 100)

# Fitting some distributions for the low flows (annual minima)
# and estimating the quantile for arbitrary return periods
y <- tyears(ng, dist = c("gum", "wei", "ln3", "pe3"), event = rp,
            plot = FALSE)

# print()ing the object shows just the return periods
y
```

```
# but y is actually a list
str(y)

# there is a summary method, returning L-moments and estimated parameters
summary(y)

plot(y)


# fitting just one distribution, with annotated quantiles
z <- tyears(ng, dist = c("gevR"), event = rp)
rpline(y, return.period = rp, suffix = c("a", "m\u00B3"))


# applying a moving average before fitting
ng2 <- ng
ng2$flow <- ma(ng2$flow, n = 4)
tyears(ng2, dist = c("gum", "wei", "ln3", "pe3"), event = rp,
       plot = FALSE)
```

`tyearsS` *Calculate Low-Flow Quantiles for given Return Periods*

Description

Fits an extreme value distribution using L-moments to the dry spells of a time series of discharges and subsequently estimates quantiles (the so called T-years event) for given return periods. In the presence of zero flow observations a mixed distribution is fitted.

Usage

```
tyearsS(lfobj, event = 1/probs, probs = 0.01, pooling = NULL,
        dist = "wei", check = TRUE, zeta = NULL,
        plot = TRUE, col = 1, log = TRUE, legend = TRUE,
        rp.axis = "bottom", rp.lab = "Return period", freq.axis = TRUE,
        freq.lab = expression(paste("Frequency " *(italic(F)),
                                    " = Non-Exceedance Probability P ",
                                    (italic(X) <= italic(x)))),
        xlab = expression("Reduced variate,  " * -log(-log(italic(F)))),
        ylab = "Quantile",
        variable = c("volume", "duration"), aggr = "max",
        hyearstart = hyear_start(lfobj), ...)
```

Arguments

`lfobj` An object of class `'lfobj'` or an object which can be coerced to class `'xts'`. Either with a single column or with a column named 'discharge'.

event	numeric vector specifying the return periods. E.g. event = 100 will yield the 100 years extreme low flow event.
probs	Alternate way to specify the return period of the event.
pooling	a pooling function, see pooling.
dist	A character vector of distributions to fit. Basically all distributions provided by Hosking's lmom-package and their reversed counterparts can be chosen.
check	logical, should check_distribution get called?
zeta	numeric vector of length one for manually setting a lower bound. Only a few distributions allow for a lower bound, namely 'gpa', 'ln3', 'wak' and 'wei'. The default value of NULL results in not bounding the distribution, therefore the parameter zeta is estimated.
plot	logical. If TRUE, sample observations as well as estimated quantile functions are plotted.
col	numeric or character vector of length one or as long as dist, specifying the colour used for plotting.
log	logical. If TRUE probabilities will be plotted on a double logarithmic scale.
legend	logical, should a legend be added to the plot?
rp.axis	vector of length one, specifying if and how an additional scale bar for the return periods is drawn. Possible choices are 'bottom', 'top' and 'none'. Alternatively, the position of the scale bar can be specified as an real number between 0 and 1, indicating the y-position of the legend.
rp.lab	character vector, text above the scale bar for return periods
freq.axis	logical, should an additional abscissa showing the probabilities be drawn on top of the plot?
freq.lab	character vector, text above the probability axis
xlab	character vector, a label for the x axis
ylab	character vector, a label for the y axis
variable	character vector of length one. Either 'v' to calculate volumes or 'd' for durations.
aggr	function like max or sum used for aggregating volumes or durations of a hydrological year.
hyearstart	vector of length one, providing the start of the hydrological year. This is evaluated by water_year. The default is, to retrieve the values stored in the attributes of the lfobj.
...	arguments passed on to find_droughts, e.g. threshold.

Details

This function is vectorised over dist and event.

According to paragraph 7.4.2 of the WNO manual, special care has to be taken in the presence of zero flow observations. A cdf called G(x) is fitted to the non-zero values of the original time series

If a distribution is fitted which allows for finite lower bound (zeta), and zeta is estimated being negative, estimation is repeated constraining zeta = 0. If this behavior is not desired, the parameter zeta has to be set explicitly.

Value

An object of class 'evfit', see evfit.

Author(s)

Gregor Laaha

References

Gustard, A. & Demuth, S. (2009) (Eds) Manual on Low-flow Estimation and Prediction. Operational Hydrology Report No. 50, WNO-No. 1029, 136p.

See Also

There are methods for printing summarizing objects of class 'evfit'.

evfit

Examples

```
data("ngaruroro")
rp <- c(1.3, 3, 5, 35)
sumD <- tyearsS(ngaruroro, event = rp, dist = "wei",
                variable = "d", aggr = sum)
sumD
summary(sumD)
```

---

vary_threshold *Create varying thresholds*

Description

Helper function to easily create a daily, weekly, monthly or seasonal varying threshold.

Usage

```
vary_threshold(x, varying = "constant",
                fun = function(x) quantile(x, probs = 0.05, na.rm = TRUE), ...)
```

Arguments

x	an object which can be coerced to class 'xts'. Either with a single column or with a column named 'discharge'.
varying	if varying is a character vector of length one, values of "constant", "daily", "weekly" and "monthly" are allowed. If a vector of class POSIX is provided, a seasonal varying threshold is computed, where the times provided define the start of the season. Only the day of the year is taken, the year itself doesn't have a meaning.

fun — a function accepting a single argument and returning either a vector of length one or a vector as long as x.

... — additional arguments, passed on to fun

Value

a vector as long as x.

Examples

```
data(ngaruroro)
ng <- as.xts(ngaruroro)["1983::1985", ]
r <- find_droughts(ng, varying = "monthly")
plot(r)

thr1 <- vary_threshold(ng, varying = "weekly", fun = mean, na.rm = TRUE)
plot(thr1)

thr2 <- vary_threshold(ng, varying = "monthly", fun = mean, na.rm = TRUE)
lines(thr2, col = 2)
```

---

water_year — *Compute the water year*

Description

Given a date, compute the corresponding water year (hydrological year).

Usage

```
water_year(x, origin = "din", as.POSIX = FALSE,
           assign = c("majority", "start", "end"), ...)
```

Arguments

x — a vector, implicit coercion to class 'POSIXlt' is performed.

origin — a vector of length one specifying the month in which the hydrological year starts. Four different ways of defining the beginning of a hydrological year are supported: a character string like 'din' or 'usgs' representing a definition of an institution (see Details), an integer number between 1 and 12, a character string of the month name (possibly abbreviated) or POSIX/Date object from which only the month is taken.

as.POSIX — logical, if TRUE return value is of class POSIXct. Otherwise a factor is returned.

assign — a character vector of length one, deciding how a hydrological year is labelled. Depending on the climate, the hydrological year can start earlier or later than the calendar year. Usually the hydrological year "equals" the calendar year for the longest period of months they have in common. Alternatively a water year can also be designated by the calendar year in which it starts or ends.

... — arguments, passed on to as.POSIXlt, e.g. such as format

Details

Currently, it is only supported to start a hydrological year on the 1st of a month.

There are abbreviations for a few established definitions:

	start	**description**
'din'	1st of November	DIN 4049 (default), as used in Austria and Germany
'usgs'	1st of October	USGS, the United States Geological Survey
'swiss'	1st of October	as defined by the Swiss "Bundesamt f. Energie" (BFE)
'glacier'	1st of September	Widely used in glaciology

Its convenient to have the water year as a factor with levels even for year without observations. For example, otherwise years without observations don't appear after aggregation.

Value

a factor representing the hydrological year.

Examples

```
# generating monthly sequence
x <- seq(from = as.Date("1992-01-01"),
         by = "months", length.out = 12)

# specifying the beginning with a decimal number
water_year(x, origin = 10)

# using a month name
water_year(x, origin = "Jul")      # can be abbreviated
water_year(x, origin = "july")     # case insensitive

# using an POSIX or Date object
water_year(x, origin = as.Date("2012-08-22"))     # only month is taken
water_year(x, origin = as.POSIXct("2012-08-22"))

# or by specifying an institution
water_year(x, origin = "usgs")
```

Index