Package 'UKFE'

October 10, 2025

Type Package

Title UK Flood Estimation

Version 1.0.2

Maintainer Anthony Hammond <agqhammond@gmail.com>

Description Functions to implement the methods of the Flood Estimation Handbook (FEH), associated updates and the revitalised flood hydrograph model (ReFH). Currently the package uses NRFA peak flow dataset version 13. Aside from FEH functionality, further hydrological functions are available. Most of the methods implemented in this package are described in one or more of the following: ``Flood Estimation Handbook'', Centre for Ecology & Hydrology (1999, ISBN:0 948540 94 X). ``Flood Estimation Handbook Supplementary Report No. 1'', Kjeldsen (2007, ISBN:0 903741 15 7). ``Regional Frequency Analysis - an approach based on L-moments'', Hosking & Wallis (1997, ISBN: 978 0 521 01940 8). ``Proposal of the extreme rank plot for extreme value analysis: with an emphasis on flood frequency studies'', Hammond (2019, <doi:10.2166/nh.2019.157>). ``Making better use of local data in flood frequency estimation'', Environment Agency (2017, ISBN: 978 1 84911 387 8). ``Sampling uncertainty of UK design flood estimation'' , Hammond (2021, <doi:10.2166/nh.2021.059>). ``Improving the FEH statistical procedures for flood frequency estimation'', Environment Agency (2008, ISBN: 978 1 84432 920 5). ``Low flow estimation in the United Kingdom'', Institute of Hydrology (1992, ISBN 0 948540 45 1). Wallingford HydroSolutions, (2016, <http://software.hydrosolutions.co.uk/winfap4/Urban-Adjustment-Procedure-Technical-Note.pdf>). Data from the UK National River Flow Archive (<https://nrfa.ceh.ac.uk/>, terms and conditions: <https://nrfa.ceh.ac.uk/help/costs-terms-and-conditions>).

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

URL https://github.com/agqhammond/ukfe

BugReports https://github.com/agqhammond/ukfe/issues

Imports xml2, methods, sf

Suggests testthat (>= 3.0.0), knitr, rmarkdown, rnrfa, lmomco, evd, dplyr, nsRFA, ismev, lmom, tibble, purrr

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Anthony Hammond [aut, cre],
Environment Agency [ctb],
JBA Consulting [ctb]

Repository CRAN

Date/Publication 2025-10-10 15:30:10 UTC

Contents

`AddGauge` *Add an AMAX sample*

Description

This function allows the user to add an AMAX sample and associated catchment descriptors for use with the FEH process.

Usage

```
AddGauge(CDs, AMAX, ID)
```

Arguments

`CDs`	catchment descriptor object imported using the CDsXML function.
`AMAX`	Either a data.frame with date (or POSIXct) in the first column and a numeric vector in the second (the AMAX). Or an AMAX sample (a numeric vector).
`ID`	This is a user supplied identification number for the AMAX.

Details

The function provides the necessary AMAX sample statistics and data.frame for adding catchment descriptors to the NRFAData data.frame. The user must then add these outputs using the rbind function (see example). The AMAX could be read in or pasted in by the user or imported using the AMImport function. Once they are added they can be used in the current R session. If a new session is started (rather than a saved workspace) the added AMAX would need to be added again.

Value

A list object. The first element is a data.frame which is a row of statistics and descriptors to be added to the NRFAData data.frame. The second element is the AMAX sample formatted to be added to the AMSP data.frame

Author(s)

Anthony Hammond

Examples

```
# Read in AMAX and catchment descriptors
## Not run:
am_add <- AMImport(r"{D:\NRFAPeakFlow_v12_1_0\suitable-for-neither\027003.am}")
cds_add <- CDsXML(r"{D:\NRFAPeakFlow_v12_1_0\suitable-for-neither\027003.xml}")

## End(Not run)

# Apply the function and add the results to the necessary data frames
## Not run:
gauge_27003 <- AddGauge(cds_add, am_add, ID = "27003")

## End(Not run)

# Append the descriptors and stats (element[[1]]) to NRFAData
## Not run:
nrfa_data <- rbind(NRFAData, gauge_27003[[1]])

## End(Not run)

# Append the AMAX series (element[[2]]) to AMSP
## Not run:
amsp <- rbind(AMSP, gauge_27003[[2]])

## End(Not run)
```

AggDayHour — *Aggregate a time series*

Description

Aggregates time series data, creating hourly data from 15-minute data for example.

Usage

```
AggDayHour(x, func, Freq = "Day", hour = 9)
```

Arguments

x	a data.frame with POSIXct in the first column and numeric vector in the second.
func	the function used for aggregation; mean, max, or sum, for example.
Freq	Choices are "Day", or "Hour".
hour	An integer between 0 and 23. This is used if "Day" is chosen in the Freq argument to determine when the day starts.

Details

The function can be used with a data.frame with POSIXct in the first column and a variable in the second. You can choose the level of aggregation in hours, or you can choose daily. In the daily case you can choose which hour of the day to start the aggregation. For example, you might want mean flows from 09:00 rather than midnight. You can also choose the function used to aggregate the data. For example, you might want "sum" for rainfall, and "mean" for flow. When aggregating hourly the aggregation starts at whatever hour is in the first row of x and the associated time stamps will reflect this.

Value

A data.frame with POSIXct in the first column (unless daily is chosen, then it's Date class), and the aggregated variable in the second column

Author(s)

Anthony Hammond

Examples

```
# Create a data frame with a normally distributed variable at
# a 15 minute sampling rate
ts_seq <- seq(
  as.POSIXct("2000-01-01 00:00:00", tz = "Europe/London"),
  as.POSIXct("2001-01-01 00:00:00", tz = "Europe/London"),
  by = 60 * 15
)
```

```
ts_df <- data.frame(DateTime = ts_seq, Var = rnorm(length(ts_seq), 10, 2))

# Aggregate to an hourly sampling rate, taking the maximum of each hour
hourly <- AggDayHour(ts_df, func = max, Freq = "Hour")

# Aggregate with the mean at a daily scale
daily <- AggDayHour(ts_df, func = mean, Freq = "Day")
```

AMImport *Import an annual maximum (AMAX) sample from NRFA peak flow .AM files*

Description

Imports the peak flows and dates from from NRFA peak flow .AM files, excluding the rejected years

Usage

```
AMImport(x)
```

Arguments

x	the file path for the .AM file

Details

File paths for importing data require forward slashes. On some operating systems, such as windows, the copy and pasted file paths will have backward slashes and would need to be changed accordingly.

Value

A data.frame with columns; Date and Flow

Author(s)

Anthony Hammond

Examples

```
# Import an AMAX sample and display the first six rows in the console
## Not run:
am_4003 <- AMImport("C:/Data/NRFAPeakFlow_v11/Suitable for QMED/4003.AM")

## End(Not run)
## Not run:
head(am_4003)

## End(Not run)
```

AMplot *Plot of the annual maximum sample*

Description

Provides a barplot for an annual maximum sample

Usage

```
AMplot(x, ylab = "Discharge (m3/s)", xlab = "Hydrological year", main = NULL)
```

Arguments

x	A data.frame with at least two columns. The first a date column and the second the annual maximum (AM) sequence. A third column with the station id can be applied which is then used for the default plot title.
ylab	Label for the y axis (character string).
xlab	Label for the x axis (character string).
main	Title for the plot (character string). The default is 'Annual maximum sample:', where : is followed by an ID number if this is included in a third column of the dataframe x.

Details

When used with a GetAM object or any data.frame with dates/POSIXct in the first column, the date-times are daily or sub-daily. Therefore, although it's an annual maximum (AM) sequence, some bars may be closer together depending on the number of days between them.

Value

A barplot of the annual maximum sample

Author(s)

Anthony Hammond

Examples

```
# Get an AMAX sample and plot
AMplot(GetAM(58002))
```

AMSP — *National River Flow Archive (NRFA) annual maximum data for sites suitable for pooling*

Description

A data.frame with three columns; Date, Flow, id. NRFA Peak Flow Dataset - Version (14?)

Usage

```
AMSP
```

Format

A data frame with 26539 rows and 3 columns

Date Date

Flow Annual maximum peak flow, m3/s

id Station identification number

Source

https://nrfa.ceh.ac.uk/data/peak-flow-dataset

AnnualStat — *Annual statistics extraction*

Description

Extracts annual statistics (default maximums) from a data.frame which has dates (or POSIXct) in the first column and variable in the second.

Usage

```
AnnualStat(
  x,
  Stat = max,
  Truncate = TRUE,
  Mon = 10,
  Hr = 9,
  Sliding = FALSE,
  N = 24,
  ...
)
```

Arguments

- `x` — a data.frame with dates (or POSIXct) in the first column and variable in the second
- `Stat` — A user chosen function to extract statistics, for example mean. The default is max. User supplied functions could also be used.
- `Truncate` — Logical argument with a default of TRUE. If TRUE, then x is truncated to be within the first and last occurrence of the chosen month and time. If FALSE truncation is not done and results from partial years will be included.
- `Mon` — Choice of month as a numeric, from 1 to 12. The default is 10 which means the year starts October 1st.
- `Hr` — Choice of hour to start the year (numeric from 0 to 23). The default is 9.
- `Sliding` — Logical argument with a default of FALSE. This can be applied if you want the statistic over a sliding period. For example, deriving maximum annual rainfall totals over a 24 hour period, rather than the maximum daily totals. The number of periods (timesteps) is chosen with the N argument. If for example you want the annual maximum sum of rainfall over a 24 hour period, and you have 15minute data, the Stat input would be sum, and N would be 96 (because there are 96 15 minute periods in 24 hours).
- `N` — Number of timesteps to slide over - used in conjunction with Sliding. The default is 24, make sure to adjust this depending on the duration of interest and the sampling rate of the input data.
- `...` — further arguments for the stat function. Such as na.rm = TRUE.

Details

The statistics are extracted based on the UK hydrological year by default (start month = 10). Month can be changed using the Mon argument. A year is from Mon-Hr to Mon-(Hr-1). For example, the 2018 hydrological year with Hr = 9 would be from 2018-10-01 09:00:00 to 2019-10-01 08:00:00. If Hr = 0, then it would be from 2018-10-01 00:00:00 to 2019-09-30 23:00:00. Data before the first occurrence of the 'start month' and after and including the last occurrence of the 'start month' is not included in the calculation of the statistic.

Value

a data.frame with columns; DateTime and Result. By default Result is the annual maximum sample, but will be any statistic used as the Stat argument.

Author(s)

Anthony Hammond

Examples

```
# Extract the Thames AMAX daily mean flow and display the first six rows
thames_am <- AnnualStat(ThamesPQ[, c(1, 3)])
head(thames_am)
```

```
# Extract the annual rainfall totals
thames_annual_p <- AnnualStat(ThamesPQ[, 1:2], Stat = sum)

# Extract maximum five day rainfall totals from the Thames rainfall series
thames_5day_am <- AnnualStat(ThamesPQ[, 1:2], Stat = sum, Sliding = TRUE, N = 5)
```

ARF *Areal reduction factor (ARF)*

Description

The results of applying, to a rainfall depth, the ratio of the rainfall over an area to the rainfall depth of the same duration at a representative point in the area.

Usage

```
ARF(Depth, Area, D)
```

Arguments

`Depth`	depth of rainfall
`Area`	catchment area in km2
`D`	duration in hours

Details

The ARF and it's use is detailed in the Flood Estimation Handbook (1999), volume 2. The DDF model is calibrated on point rainfall and the areal reduction factor converts it to a catchment rainfall for use with a rainfall runoff model such as ReFH (see details for ReFH function). The ReFH model includes a design rainfall profile for winter and summer but the depth duration frequency (DDF) model is calibrated on annual maximum peaks as opposed to seasonal peaks. A seasonal correction factor (SCF) is necessary to convert the DDF estimate to a seasonal one. The final depth, therefore is; Depth = DDFdepth x ARF x SCF.

Value

the rainfall depth or rainfall return period

Author(s)

Anthony Hammond

Examples

```
# Derive the ARF for a depth of 30, an area of 500km^2 and a duration of 12 hours
ARF(30, 500, 12)
```

`BFI` *Baseflow index (BFI)*

Description

Calculates the baseflow index from a daily mean flow series

Usage

```
BFI(Q, x.lim = NULL, y.lim = NULL, PlotTitle = "Baseflow plot", Plot = TRUE)
```

Arguments

`Q`	the daily mean flow series. Numeric vector
`x.lim`	the x axis limits of the plot. Numeric vector of length two. Default is the extent of the data
`y.lim`	the y axis limits of the plot. Numeric vector of length two. Default is the extent of the data
`PlotTitle`	the title of the plot. The default is "Baseflow plot"
`Plot`	a logical argument with a default of TRUE. If TRUE the daily flow is plotted with the baseflow highlighted.

Details

The baseflow index is calculated using the method outlined in Gustard, A. Bullock, A. Dixon, J. M.. (1992). Low flow estimation in the United Kingdom. Wallingford, Institute of Hydrology, 88pp. (IH Report No.108)

Value

the baseflow index and if Plot equals TRUE, a plot showing the flow time series (black) and the associated baseflow (red)

Author(s)

Anthony Hammond

Examples

```
# Calculate the BFI from the daily discharge at Kingston upon Thames,
# which is in column three of the ThamesPQ data
BFI(ThamesPQ[, 3])
```

`Bootstrap` *Bootstrap*

Description

Resampling with replacement to approximate the sampling distribution of a statistic and quantify uncertainty.

Usage

```
Bootstrap(x, Stat, n = 500, Conf = 0.95, ReturnSD = FALSE, ...)
```

Arguments

`x`	a numeric vector. The sample of interest
`Stat`	the function (to calculate the statistic) to be applied to the bootstrapped samples. For example mean, max, or median.
`n`	the number of bootstrapped samples (default 500). i.e. the size of the derived sampling distribution.
`Conf`	the confidence level of the intervals (default 0.95). Must be between 0 and 1.
`ReturnSD`	Logical argument with a default of FALSE. If true the bootstrapped sampling distribution is returned.
`...`	further arguments for the Stat function. For example if you use the GEVAM function you might want to add RP = 50 to derive a sampling distribution for the 50-year quantile.

Details

The bootstrapping procedure resamples from a sample length(x) * n times with replacement. After splitting into n samples of size length(x), the statistic of interest is calculated on each.

Value

If ReturnSD is FALSE a data.frame is returned with one row and three columns; central, lower, and upper. If ReturnSD is TRUE, the sampling distribution is returned.

Author(s)

Anthony Hammond

Examples

```
# Extract an AMAX sample and quantify uncertainty for the Gumbel estimated 50-year flow
am_203018 <- GetAM(203018)
Bootstrap(am_203018$Flow, Stat = GumbelAM, RP = 50)

# Quantify uncertainty for the sample standard deviation at the 90% confidence level
Bootstrap(am_203018$Flow, Stat = sd, Conf = 0.90)

# Return the sampling distribution of the mean and plot an associated histogram
samp_dist <- Bootstrap(am_203018$Flow, Stat = mean, ReturnSD = TRUE)
hist(samp_dist)
```

CDsXML *Import catchment descriptors from .xml files*

Description

Imports catchment descriptors from xml files either from an FEH webservice download or from the Peakflows dataset downloaded from the national river flow archive (NRFA) website

Usage

```
CDsXML(x)
```

Arguments

x	the xml file path

Details

File paths for importing data require forward slashes. On some operating systems, such as windows, the copy and pasted file paths will have backward slashes and would need to be changed accordingly.

Value

A data.frame with columns; Descriptor and Value.

Author(s)

Anthony Hammond

Examples

```
# Import catchment descriptors from an NRFA Peak Flows XML file and display in console
## Not run:
cds_4003 <- CDsXML("C:/Data/NRFAPeakFlow_v11/Suitable for QMED/4003.xml")
cds_4003

## End(Not run)

# Import catchment descriptors from a FEH webserver XML file and display XML in the console
## Not run:
cds_my_site <- CDsXML("C:/Data/FEH_Catchment_384200_458200.xml")
cds_my_site

## End(Not run)
```

ConvertGridRef — *Convert between British National Grid Reference (BNG) and Latitude and Longitude or Irish Grid references.*

Description

Function to convert between BNG easting & northing and Latitude & Longitude (or vice versa). Or to convert between BNG and Irish national grid (or vice versa)

Usage

```
ConvertGridRef(x, fromBNG = TRUE, IGorLatLon = "LatLon")
```

Arguments

`x`	A vector of length 2. Either latitude and longitude (if fromBNG = FALSE) or BNG easting and northing (if fromBNG = TRUE). Or Irish easting and northing if IGorLatLon is set to IG and fromBNG = FALSE.
`fromBNG`	A logical argument with a default of TRUE. When TRUE it converts from BNG easting and northing to latitude and longitude (or to IG easting and northing if IGorLatLon is set to "IG"). When FALSE it converts the other way round.
`IGorLatLon`	This argument allows you to choose between Latitude & Longitude and Irish grid reference. The acceptable options are "LatLon" or "IG". If you choose "IG" you are converting between BNG and IG. If you choose "LatLon", you are converting between BNG and Lat Lon.

Details

To convert to Lat and Lon from BNG, ensure that the fromBNG argument is TRUE. To convert the other way, set fromBNG as FALSE. The same applies for converting between Irish grid and BNG. To convert Irish grid and BNG set the IGorLatLon argument to IG.

Value

A data.frame with the converted grid references. Either latitude and longitude if BNG = TRUE. Or BNG easting and northing if fromBNG = FALSE. Or, IG easting & northing if fromBNG = TRUE and IGorLatLon = "IG".

Author(s)

Anthony Hammond

Examples

```
# Convert a BNG numeric reference to latitude and longitude
ConvertGridRef(c(462899, 187850))

# Convert latitude and longitude to easting and northing
ConvertGridRef(c(51.6, -1), fromBNG = FALSE)
```

DDF — *DDF results from a DDFImport object*

Description

Extracts results from a data frame imported using the DDFImport function

Usage

```
DDF(x, duration, RP = 100)
```

Arguments

`x`	A data frame of DDF13 or DDF22 results imported using the DDFImport function
`duration`	the duration (hrs) for which a rainfall depth estimate is required
`RP`	the return period (years) for which a rainfall depth estimate is required

Details

The .xml files only provide a set number of durations and return periods for DDF13 and DDF22. This function optimises the GEV distribution on the results in order to interpolate across return periods. A linear interpolation is used between durations. The interpolation method may provide results that differ from the FEH webservice in the region of 0.1mm. The result is then rounded to an integer.

Value

A DDF13 or DDF22 estimate of rainfall depth (mm)

Author(s)

Anthony Hammond

Examples

```
# Import DDF13 results from an NRFA Peak Flows XML file
## Not run:
ddf13_4003 <- DDFImport("C:/Data/NRFAPeakFlow_v9/Suitable for QMED/04003.xml", DDFVersion = 13)

## End(Not run)

# Estimate the 20-year, 5-hour depth
## Not run:
DDF(ddf13_4003, duration = 5, RP = 20)

## End(Not run)
```

DDF99 *FEH99 depth duration frequency precipitation model*

Description

Estimation of design rainfall depths, and the rarity of observed rainfall

Usage

```
DDF99(Duration, RP, pars, Depth = NULL, disc = NULL)
```

Arguments

`Duration`	numeric. The duration of interest (in hours)
`RP`	return period
`pars`	a numeric vector of length six. The six catchment parameters for the DDF model in the order of: c, d1, d2, d3, e, f
`Depth`	a user supplied rainfall depth for the duration under question
`disc`	converts from the sliding duration to fixed duration estimate. Choices are "hourly" or "daily"

Details

The depth duration frequency rainfall model is detailed in the Flood Estimation Handbook (1999), volume 2. A note about the discretisation: The user can choose between "daily" or "hourly" for the sliding duration to fixed duration conversion. If the 'Depth' argument is used, it overrides the return period (RP) argument and provides RP as a function of depth. However, if both the 'Depth' and the 'disc' arguments are used, the sliding duration depth is provided as a function of the user input depth. This resulting depth can then be used without the 'disc' argument to determine the sliding duration RP.

Value

the rainfall depth or rainfall return period

Author(s)

Anthony Hammond

Examples

```
# Examples from FEH volume 2
# The parameters for these examples are from FEH v2

# What is the 2-day rainfall with return period 100 years for Norwich?
DDF99(Duration = 48, RP = 100, pars = c(-0.023, 0.273, 0.351, 0.236, 0.309, 2.488))

# What is the 4-hour rainfall with return period 20 years for a typical point in the Lyne catchment?
DDF99(Duration = 4, RP = 20, pars = c(-0.025, 0.344, 0.485, 0.402, 0.287, 2.374))

# How rare was the rainfall of 6th August 1978 at Broughshane, County Antrim?
DDF99(Duration = 5, Depth = 47.7, pars = c(-0.022, 0.412, 0.551, 0.276, 0.261, 2.252))
```

DDF99Pars *DDF99 parameters from .xml files*

Description

Imports the FEH 1999 depth duration frequency parameters from xml files either from an FEH webservice download or from the Peakflows dataset downloaded from the national river flow archive (NRFA) website

Usage

```
DDF99Pars(x)
```

Arguments

x	the xml file path

Details

This function is coded to import DDF99 parameters from xml files from the NRFA or the FEH webserver. File paths for importing data require forward slashes. On some operating systems, such as windows, the copy and pasted file paths will have backward slashes and would need to be changed accordingly.

Value

A list with two elements, each a data frame with columns; parameters and associated values The first data frame is for the catchment average parameters (these still require an ARF adjustment where appropriate) and the second for the 1km2 grid point parameters.

Author(s)

Anthony Hammond

Examples

```
# Import DDF99 parameters from an NRFA Peak Flows XML file and display in console
## Not run:
ddf99_4003 <- DDF99Pars("C:/Data/NRFAPeakFlow_v11/Suitable for QMED/04003.xml")
ddf99_4003

## End(Not run)

# Import DDF99 parameters from a FEH webserver XML file and display in the console
## Not run:
ddf99_my_site <- DDF99Pars("C:/Data/FEH_Catchment_384200_458200.xml")
ddf99_my_site

## End(Not run)
```

DDFExtract — *Derive and plot rainfall Depth Duration Frequency curves.*

Description

Derive and plot rainfall Depth Duration Frequency curves from an input of rainfall data.

Usage

```
DDFExtract(x, Plot = TRUE, main = NULL, Truncate = TRUE)
```

Arguments

x	A data.frame with POSIXct in the first column and rainfall in the second. The data must have an hourly or sub-hourly sampling rate.
Plot	Logical argument with a default of TRUE. If TRUE, the DDF curves are plotted.
main	Title for the plot (character string). The default is no title.
Truncate	Logical argument with a default of TRUE. If TRUE the extraction of annual maximum process truncates the data to incorporate only full hydrological years. If there is significant rainfall within a partial year it will not be included unless Truncate = FALSE. If Truncate = FALSE, ensure that there are at least 92 hours of data available in the partial years or the function will fail.

Details

The function works by extracting the annual maximum sample (by hydrological year - starting Oct 1st) of rainfall for a range of sliding durations (1 hour to 96 hours). It then calculates the median annual maximum rainfall depth (RMED) and a GEV growth curve for each duration. To ensure RMED increases with duration a power curve is fit as a function of duration to provide the final RMED estimates. Then the average growth factor for each return period (across the durations) is assumed.

Value

A dataframe with hours (1 to 96) in the first column then depths associated with a range of return periods (2 to 1000) in the remaining nine columns. If Plot = TRUE, a plot of the DDF curves is also returned.

Author(s)

Anthony Hammond

Examples

```
# Extract all available 15-minute rainfall from the St Ives (Cambridgeshire)
# rain gauge (WISKI ID = 179365).
## Not run:
st_ives <- GetDataEA_Rain(WISKI_ID = "179365", Period = "15Mins")

## End(Not run)

# Apply the DDF function.
## Not run:
DDFExtract(st_ives)

## End(Not run)
```

DDFImport *DDF13 or DDF22 results from .xml files*

Description

Imports the depth duration frequency 2013 or 2022 results from xml files either from an FEH webservice download or from the Peakflows dataset downloaded from the national river flow archive (NRFA) website

Usage

```
DDFImport(x, ARF = FALSE, Plot = TRUE, DDFVersion = 22)
```

Arguments

`x`	the xml file path
`ARF`	logical argument with a default of FALSE. If TRUE, the areal reduction factor is applied to the results. If FALSE, no area reduction factor is applied
`Plot`	logical argument with a default of TRUE. If TRUE the DDF curve is plotted for a few return periods
`DDFVersion`	Version of the DDF model (numeric). either 22 or 13. The default is 22.

Details

This function returns a data-frame of design rainfall estimates. For further durations and return periods, the separate DDF function can be applied with the data-frame as the argument/input.

Value

A data frame of DDF results (mm) with columns for duration and rows for return period. If Plot equals TRUE a DDF plot is also returned.

Author(s)

Anthony Hammond

Examples

```
# Import DDF22 results from an NRFA Peak Flows XML file and display them in console
## Not run:
ddf22_4003 <- DDFImport(r"{C:\Data\NRFAPeakFlow_v11\Suitable for QMED\04003.xml}")
ddf22_4003

## End(Not run)

# Import DDF22 results from a FEH webserver XML file and display them in the console
## Not run:
ddf22_my_site <- DDFImport(r"{C:\Data\FEH_Catchment_384200_458200.xml}")
ddf22_my_site

## End(Not run)
```

DesHydro *Design hydrograph extraction*

Description

Extracts a mean hydrograph from a flow series

Usage

```
DesHydro(
  x,
  Threshold = 0.975,
  EventSep,
  N = 10,
  Exclude = NULL,
  Plot = TRUE,
  main = "Design Hydrograph"
)
```

Arguments

`x`	a dataframe with Date or POSIXct in the first column and the numeric vector of discharge in the second
`Threshold`	The threshold above which the peaks of the hydrograph are first identified. The default is 0.975.
`EventSep`	Number of timesteps to determine individual peak events during the extraction process. For the comparison and averaging process the start and end point of the hydrograph is Peak - EventSep and Peak + EventSep * 1.5.
`N`	number of event hydrographs from which to derive the mean hydrograph. Default is 10. Depending on the length of x, there may be fewer than 10
`Exclude`	An index (single integer or vector of integers up to N) for which hydrographs to exclude if you so wish. This may require some trial and error. You may want to increase N for every excluded hydrograph.
`Plot`	logical argument with a default of TRUE. If TRUE, all the hydrographs from which the mean is derived are plotted along with the mean hydrograph.
`main`	Title for the plot

Details

All the peaks over the threshold (default 0.975th) are identified and separated by a user defined value 'EventSep', which is a number of timesteps (peaks are separated by EventSep * 3). The top N peaks are selected and the hydrographs are then extracted. The hydrograph start is the time of peak minus EventSep. The End of the hydrograph is time of peak plus EventSep times 1.5. All events are scaled to have a peak flow of one, and the mean of these is taken as the scaled design hydrograph.

Value

a list of length three. The first element is a dataframe of the peaks of the hydrographs and the associated dates. The second element is a dataframe with all the scaled hydrographs, each column being a hydrograph. The third element is the averaged hydrograph

Author(s)

Anthony Hammond

Examples

```
# Extract a design hydrograph from the Thames daily mean flow and print the resulting hydrograph
thames_des_hydro <- DesHydro(ThamesPQ[, c(1, 3)], EventSep = 10, N = 10)
```

DeTrend — *Linearly detrend a sample*

Description

Applies a linear detrend to a sample

Usage

```
DeTrend(x)
```

Arguments

x	a numeric vector

Details

Adjusts all the values in the sample, of size n, by the difference between the linearly modelled ith data point and the linearly modelled nth data point.

Value

A numeric vector which is a linearly detrended version of x.

Author(s)

Anthony Hammond

Examples

```
# Get an annual maximum (AM) sample that looks to have a significant trend
am_21025 <- GetAM(21025)

# Plot the resulting AM as a bar plot. Then detrend and compare with another plot
plot(am_21025$Flow, type = "h", ylab = "Discharge (m^3/s)")
am_detrend <- DeTrend(am_21025$Flow)
plot(am_detrend, type = "h", ylab = "Discharge (m^3/s)")
```

DiagPlots *Diagnostic plots for pooling groups*

Description

Provides 10 plots to compare the sites in the pooling group

Usage

```
DiagPlots(x, gauged = FALSE)
```

Arguments

`x`	pooling group derived from the Pool() function
`gauged`	logical argument with a default of FALSE. TRUE adds the top site in the pooling group to the plots in a different colour

Value

ten diagnostic plots for pooling groups

Author(s)

Anthony Hammond

Examples

```
# Form a gauged pooling group and plot the diagnostics
pool_28015 <- Pool(GetCDs(28015))
DiagPlots(pool_28015, gauged = TRUE)

# Form an ungauged pooling group and plot the diagnostics
pool_28015 <- Pool(GetCDs(28015), exclude = 28015)
DiagPlots(pool_28015)
```

DonAdj *Donor adjustment candidates & results*

Description

Provides donor adjustment candidates, descriptors, and results in order of the proximity to the centroid of the subject catchment.

Usage

```
DonAdj(CDs = NULL, x, y, QMEDscd = NULL, alpha = TRUE, rows = 10, d2 = NULL)
```

Arguments

`CDs`	catchment descriptors derived from either GetCDs or CDsXML for the site of interest
`x`	catchment centroid easting (for when CDs isn't used)
`y`	catchment centroid northing (for when CDs isn't used)
`QMEDscd`	QMED estimate for the catchment of interest (for when CDs isn't used)
`alpha`	logical argument with a default of TRUE. If FALSE the exponent of the donor adjustment equation is set to one
`rows`	number of sites provided; default is 10
`d2`	a numeric vector of length two; the two site references for the donor catchments chosen for the two donor case

Details

The donor adjustment method is as outlined in Science Report: SC050050 - Improving the FEH statistical procedures for flood frequency estimation. The method for two donors is outlined in 'Kjeldsen, T. (2019). Adjustment of QMED in ungauged catchments using two donor sites. Circulation - The Newsletter of the British Hydrological Society, 4'. When d2 is NULL this function returns a list of possible donors and associated details - including a default adjusted QMED estimate for the site of interest. The results for single donor adjustment are in the final column headed 'QMED.adj'. If alpha is set to FALSE, the results in the final column are from the same donor equation but with an exponent of 1. The QMEDfse column provides the gauged factorial standard error for the median of the annual maximum sample. It is worth considering this when choosing a donor site (a high value indicates a poor donor). When choosing between two donors, the site with a lower QMEDfse would be an appropriate choice (all else being equal). The QMEDfse is calculated with the QMEDfseSS() function. When d2 is populated (when two donors are used), only the resulting adjusted QMED and alpha values are returned, rather than donor candidate information. Note that the estimates for the single donor and two donor adjustment do not consider urbanisation in the catchment of interest or the donor catchments. Such things can be considered more specifically using the QMED() function.

Value

A data.frame with rownames of site references and columns of catchment descriptors, distance from subect site, and associated results. When two donors are used, only the resulting adjusted QMED is returned

Author(s)

Anthony Hammond

Examples

```
# Get some CDs and output candidate donor sites
cds_54022 <- GetCDs(54022)
DonAdj(cds_54022)
```

```
# Get results with inputs of x,y, and QMEDscd
DonAdj(x = 283261, y = 288067, QMEDscd = 17.931)

# Get a result with two donors
DonAdj(cds_54022, d2 = c(54092, 54091))
```

EncProb *Encounter probabilities*

Description

Calculates the probability of experiencing at least n events with a given return period (RP), over a given number of years

Usage

```
EncProb(n, yrs, RP, dist = "Poisson")
```

Arguments

n	number of events
yrs	number of years
RP	return period of the events
dist	choice of probability distribution. Either "Poisson" or "Binomial"

Details

The choice of binomial or Poisson distributions for calculating encounter probablities is akin to annual maximum (AM) versus peaks over threshold (POT) approaches to extreme value analysis. AM and binomial assume only one "event" can occur in the blocked time period. Whereas Poisson and POT don't make this assumption. In the case of most catchments in the UK, it is rare to have less than two independent "events" per year; in which case the Poisson and POT choices are more suitable. In large catchments, with seasonally distinctive baseflow, there may only be one independent peak in the year. However, the results from both methods converge with increasing magnitude, yielding insignificant difference beyond a 20-year return period.

Value

A probability

Author(s)

Anthony Hammond

Examples

```
# Calculate the probability of exceeding at least one 50-year RP event
# over a 10-year period, using the Poisson distribution
EncProb(n = 1, yrs = 10, RP = 50)

# Calculate the probability of exceeding at least two 100-year RP events
# over a 100-year period, using the Binomial distribution
EncProb(n = 2, yrs = 100, RP = 100, dist = "Binomial")
```

`ERPlot` *Extreme rank plot*

Description

A plot to inspect the distribution of ordered data

Usage

```
ERPlot(x, dist = "GenLog", main = NULL, Pars = NULL, GF = NULL, ERType = 1)
```

Arguments

`x`	numeric vector. A sample for inspection
`dist`	a choice of distribution. The choices are "GenLog" (the default), "GEV", "Kappa3,"Gumbel", and "GenPareto"
`main`	a character string to change the default title, which is the distribution choice.
`Pars`	a vector of parameters for the distribution. In the order of location, scale, & shape (ignoring the latter if Gumbel). If left null the parameters are estimated from x.
`GF`	a vector of length growth curve parameters, in the order of; Lcv, LSkew and Median (ignoring the LSkew if Gumbel).
`ERType`	Either 1, 2. If it is the default 1 then ranks are plotted on the x axis and percentage difference of modelled from observed is plotted on the y axis.

Details

By default this plot compares the percentage difference of simulated results with observed for each rank of the data. Another option (see ERType argument) compares the simulated flows for each rank of the sample with the observed of the same rank. For both plots 500 simulated samples are used. With the second option for each rank they are plotted and the mean of these is highlighted in red. There is a line of perfect fit so you can see how much this "cloud" of simulation differs from the observed. By default the parameters of the distribution for comparison with the sample are estimated from the sample. However, the pars argument can be used to compare the distribution with parameters estimated separately. Similarly the growth factor (GF) parameters, linear coefficient of variation (Lcv) & linear skewness (LSkew) with the median can be entered. In this way the pooling estimated distribution can be compared to the sample. This ERplot is an updated version of that described in Hammond, A. (2019). Proposal of the 'extreme rank plot' for extreme value analysis: with an emphasis on flood frequency studies. Hydrology Research, 50 (6), 1495-1507.

Value

The extreme rank plot as described in the details

Author(s)

Anthony Hammond

Examples

```
# Get an AMAX sample and plot
## Not run:
am_27083 <- GetAM(27083)
ERPlot(am_27083$Flow)

## End(Not run)

# Get a pooled estimate of Lcv and LSkew to use with the GF argument
## Not run:
QuickResults(GetCDs(27083), gauged = TRUE)

## End(Not run)

# Use the resulting Lcv, Lskew and QMED for the GF argument and change the title
## Not run:
ERPlot(am_27083$Flow, main = "Site 27083 pooled comparison", GF = c(0.23, 0.17, 12))

## End(Not run)
```

`EVPlot` *Extreme value plot (frequency and growth curves)*

Description

Plots the extreme value frequency curve or growth curve with observed sample points.

Usage

```
EVPlot(
  x,
  dist = "GenLog",
  scaled = TRUE,
  Title = "Extreme value plot",
  ylabel = NULL,
```

```
  LineName = NULL,
  Unc = TRUE
)
```

Arguments

x	a numeric vector. The sample of interest
dist	a choice of distribution. "GEV", "GenLog", "Kappa3","Gumbel" or "GenPareto". The default is "GenLog"
scaled	logical argument with a default of TRUE. If TRUE the plot is a growth curve (scaled by the QMED). If FALSE, the plot is a frequency curve
Title	a character string. The user chosen plot title. The default is "Extreme value plot"
ylabel	a character string. The user chosen label for the y axis. The default is "Q/QMED" if scaled = TRUE and "Discharge (m3/s)" if scaled = FALSE
LineName	a character string. User chosen label for the plotted curve
Unc	logical argument with a default of TRUE. If TRUE, 95 percent uncertainty intervals are plotted.

Details

The plotting has the option of generalised extreme value (GEV), generalised Pareto (GenPareto), Gumbel, or generalised logistic (GenLog) distributions. The uncertainty is quantified by bootstrapping.

Value

An extreme value plot (frequency or growth curve) with intervals to quantify uncertainty

Author(s)

Anthony Hammond

Examples

```
# Get an AMAX sample and plot the growth curve with the GEV distribution
am_203018 <- GetAM(203018)
EVPlot(am_203018$Flow, dist = "GEV")
```

EVPlotAdd *Add lines and/or points to an extreme value plot*

Description

Functionality to add extra lines or points to an extreme value plot (derived from the EVPlot function).

Usage

```
EVPlotAdd(
  Pars,
  dist = "GenLog",
  Name = "Adjusted",
  MED = NULL,
  xyleg = NULL,
  col = "red",
  lty = 1,
  pts = NULL,
  ptSym = NULL
)
```

Arguments

`Pars`	a numeric vector of length two. The first is the Lcv (linear coefficient of variation) and the second is the Lskew (linear skewness).
`dist`	distribution name with a choice of "GenLog", "GEV", "GenPareto", "Kappa3", and "Gumbel"
`Name`	character string. User chosen name for points or line added (for the legend)
`MED`	The two year return level. Necessary in the case where the EV plot is not scaled
`xyleg`	a numeric vector of length two. They are the x and y position of the symbol and text to be added to the legend.
`col`	The colour of the points of line that have been added
`lty`	An integer. The type of line added
`pts`	A numeric vector. An annual maximum sample, for example. This is for points to be added
`ptSym`	An integer. The symbol of the points to be added

Details

A line can be added using the Lcv and Lskew based on one of four distributions (Generalised extreme value, Generalised logistic, Gumbel, Generalised Pareto). Points can be added as a numeric vector. If a single point is required, the base points() function can be used and the x axis will need to be log(RP-1).

Value

Additional, user specified line or points to an extreme value plot derived from the EVPlot function.

Author(s)

Anthony Hammond

Examples

```
# Get an AMAX sample and plot the growth curve with the GEV distribution
am_203018 <- GetAM(203018)
EVPlot(am_203018$Flow, dist = "GEV")

# Now add a line (dotted and red) for the generalised logistic distribution
# First get the Lcv and Lskew using the L-moments function
pars <- as.numeric(LMoments(am_203018[, 2])[c(5, 6)])
EVPlotAdd(
  Pars = pars, dist = "GenLog", Name = "GenLog",
  xyleg = c(-5.2, 2.65), lty = 3
)

# Now add a line for the Gumbel distribution which is dark green and dashed
EVPlotAdd(
  Pars = pars[1], dist = "Gumbel", Name = "Gumbel",
  xyleg = c(-5.19, 2.5), lty = 3, col = "darkgreen"
)

# Now plot afresh and get another AMAX and add the points
EVPlot(am_203018$Flow, dist = "GEV")
am_27090 <- GetAM(27090)
EVPlotAdd(xyleg = c(-4.9, 2.65), pts = am_27090[, 2], Name = "27090")
```

EVPool — *Extreme value plot for pooling groups*

Description

Plots the extreme value frequency curve or growth curve for gauged or ungauged pooled groups

Usage

```
EVPool(
  x,
  AMAX = NULL,
  gauged = FALSE,
  dist = "GenLog",
  QMED = NULL,
  Title = "Pooled growth curve",
```

```
  UrbAdj = FALSE,
  CDs
)
```

Arguments

x	pooling group derived from the Pool() function
AMAX	the AMAX sample to be plotted in the case of gauged. If NULL, & gauged equals TRUE, the AMAX from the first site in the pooling group is used
gauged	logical argument with a default of FALSE. If FALSE, the plot is the ungauged pooled curve accompanied by the single site curves of the group members. If TRUE, the plot is the gauged curve and single site curve with the observed points added
dist	a choice of distribution. Choices are "GEV", "GenLog", "Kappa3", or "Gumbel". The default is "GenLog"
QMED	a chosen QMED to convert the curve from a growth curve to the frequency curve
Title	a character string. The user chosen plot title. The default is "Pooled growth curve"
UrbAdj	a logical argument with a default of FALSE. If TRUE an urban adjustment is applied to the pooled growth curve
CDs	catchment descriptors derived from either GetCDs or CDsXML. Only necessary if UrbAdj is TRUE

Value

An extreme value plot for gauged or ungauged pooling groups

Author(s)

Anthony Hammond

Examples

```
# Get some CDs, form an ungauged pooling group and apply EVPool
cds_28015 <- GetCDs(28015)
pool_28015 <- Pool(cds_28015, exclude = 28015)
EVPool(pool_28015)

# Do the same for the gauged case, change the title, and convert with a QMED of 9.8
pool_g_28015 <- Pool(cds_28015)
EVPool(pool_g_28015, gauged = TRUE, Title = "Gauged frequency curve - Site 28015", QMED = 9.8)

# Pretend we have an extra AMAX for the gauge. Amend the pooling group Lcv and LSkew
# for the site accordingly, then apply EVPool with the updated AMAX
# Firstly, get the AMAX sample
am_28015 <- GetAM(28015)

# Add an extra AMAX flow of 15 m^3/s
append_28015 <- append(am_28015$Flow, 15)
```

```
# Amend the Lcv and Lskew in the pooling group
pool_g_28015[1, c(16, 17)] <- c(Lcv(append_28015), LSkew(append_28015))

# Now plot gauged with the updated AMAX
EVPool(pool_g_28015, AMAX = append_28015, gauged = TRUE)
```

FlowDurationCurve *Flow duration curve*

Description

A function to plot flow duration curves for a single flow series or flow duration curves from multiple flow series.

Usage

```
FlowDurationCurve(
  x = NULL,
  main = "Flow duration curve",
  CompareCurves = NULL,
  LegNames = NULL,
  Cols = NULL,
  AddQs = NULL,
  ReturnData = FALSE
)
```

Arguments

`x`	a dataframe with date in the first column and numeric (flow) in the second.
`main`	A title for the plot. The default is 'Flow duration curve'.
`CompareCurves`	A user supplied list where each element is a numeric vector (each a flow series). This is useful for when you want to compare curves from multiple flow series'.
`LegNames`	User supplied names for the legend. This only works when the CompareCurves argument is used. The default is Curve1, Curve2...CurveN.
`Cols`	User supplied vector of colours. This only works when the CompareCurves argument is used. The default is the Zissou 1 palette.
`AddQs`	Adds additional flows and associated horizontal plot lines to the plot. It should be a single numeric value or a vector, for example c(25, 75, 100).
`ReturnData`	Logical argument with a default of FALSE. When TRUE, a dataframe is returned with the data from the plot.

Details

The user can input a dataframe of dates (or POSIXct) and flow to return a plot of the flow duration curve for annual, winter and summer periods. Alternatively a list of flow series' (vectors) can be applied for a plot comparing the individual flow duration curves.

Value

If a dataframe of date in the first column and flow in the second is applied with the x argument a plot of the flow duration curves for the winter, summer and annual periods is returned. If a list of flow series is applied with the CompareCurves argument the associated flow duration curves are all plotted together. If ReturnData is TRUE, the plotted data is also returned.

Author(s)

Anthony Hammond

Examples

```
# Plot a flow duration curve for the Thames at Kingston October 2000 to September 2015
FlowDurationCurve(ThamesPQ[, c(1, 3)])

# Add two additional flow lines for the plot
FlowDurationCurve(ThamesPQ[, c(1, 3)], AddQs = c(25, 200))

# Compare flows from the rather wet 2013 water year (rows 4749 and 5114) with the rest of the flow
FlowDurationCurve(
  CompareCurves = list(
    ThamesPQ$Q[-seq(4749, 5114)],
    ThamesPQ$Q[4749:5114]
  ),
  LegNames = c("All but 2013", "Water year 2013")
)
```

`FlowSplit` *Flow splitter*

Description

A function to separate baseflow from runoff.

Usage

```
FlowSplit(
  x,
  BaseQUpper = NULL,
  AdjUp = NULL,
  ylab = "Value",
  xlab = "Time index"
)
```

Arguments

x	A numeric vector (your flow series / hydrograph).
BaseQUpper	Numeric value which is an upper level of baseflow (i.e. the baseflow will not extend above this level). The default is the mean of x. It can be set arbitrarily high so that the baseflow joins all low points/troughs in the hydrograph.
AdjUp	A numeric value between 0 and 0.5. This allows the user to adjust the baseflow up the falling limb/s of the hydrogaph. With 0.05 being a small upward adjustment and 0.49 being a large upward adjustment.
ylab	Label for the y-axis (character string). The default is "value",
xlab	Label for the x-axis (character string). The default is "Time index".

Details

The function is intended for the event scale as opposed to long term flow series. It works by linearly joining all the low points in the hydrograph - also the beginning and end points. Where a low point is any point with two higher points either side. Then any values above the hydrograph (xi) are set as xi. The baseflow point on the falling limb of the hydrograph/s can be raised using the AdjUp argument. The function works for any sampling frequency and arbitrary hydrograph length (although more suited for event scale and sub-annual events in general). This function is not design for deriving long term baseflow index. It could be used for such a purpose but careful consideration would be required for the BaseQUpper argument especially for comparison across river locations. If baseflow index is required the BFI function (with daily mean flow) is more suitable.

Value

A dataframe with the original flow (x) in the first column and the baseflow in the second. A plot of the original flow and the baseflow is also returned.

Author(s)

Anthony Hammond

Examples

```
# We'll extract a wet six month period on the Thames during the 2006-2007 hydrological year
thames_q <- subset(ThamesPQ[, c(1, 3)], Date >= "2006-11-04" & Date <= "2007-05-06")

# Then apply the flow split with default settings
q_split <- FlowSplit(thames_q$Q)

# Now do it with an upper baseflow level of 100 m^3/s
q_split <- FlowSplit(thames_q$Q, BaseQUpper = 100)

# First we'll use the DesHydro function to pick out a reasonable looking event from the Thames flow
q <- DesHydro(ThamesPQ[, c(1, 3)], Plot = FALSE, EventSep = 15)
q <- q$AllScaledHydrographs$hydro7

# Then we'll use our flow split function
FlowSplit(q)
```

```
# Next we will get a single peaked "idealised" hydrograph using the ReFH function
q_refh <- ReFH(GetCDs(15006))
q_refh <- q_refh[[2]]$TotalFlow

# Now use the function with and without an upward adjustment of the baseflow on the falling limb
q_flow_split <- FlowSplit(q_refh)
q_flow_split <- FlowSplit(q_refh, AdjUp = 0.15)
```

GenLogAM *Generalised logistic distribution - estimates directly from sample*

Description

Estimated quantiles as a function of return period (RP) and vice versa, directly from the data

Usage

```
GenLogAM(x, RP = 100, q = NULL)
```

Arguments

x	numeric vector (block maxima sample)
RP	return period (default = 100)
q	quantile (magnitude of variable)

Details

If the argument q is used, it overrides RP and provides RP as a function of q (magnitude of variable) as opposed to q as a function of RP. The parameters are estimated by the method of L-moments, as detailed in 'Hosking J. and Wallis J. 1997 Regional Frequency Analysis: An Approach Based on L-Moments. Cambridge University Press, New York'.

This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

quantile as a function of RP or vice versa.

Author(s)

Anthony Hammond

Examples

```
# Get an annual maximum sample and estimate the 50-year RP
am_27090 <- GetAM(27090)
GenLogAM(am_27090$Flow, RP = 50)

# Estimate the RP for a 600 m^3/s discharge
GenLogAM(am_27090$Flow, q = 600)
```

GenLogEst — *Generalised logistic distribution estimates from parameters*

Description

Estimated quantiles as function of return period (RP) and vice versa, from user input parameters

Usage

```
GenLogEst(loc, scale, shape, q = NULL, RP = 100)
```

Arguments

`loc`	location parameter
`scale`	scale parameter
`shape`	shape parameter
`q`	quantile. magnitude of the variable under consideration
`RP`	return period

Details

If the argument q is used, it overrides RP and provides RP as a function of q (magnitude of variable) as opposed to q as a function of RP. This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

quantile as a function of RP or vice versa

Author(s)

Anthony Hammond

Examples

```
# Get an annual maximum sample, estimate the parameters, and estimate 50-year RP
am_27090 <- GetAM(27090)
GenLogPars(am_27090$Flow)

# Store the parameters in an object
pars <- as.numeric(GenLogPars(am_27090$Flow))

# Get an estimate of 50-year flow
GenLogEst(pars[1], pars[2], pars[3], RP = 50)

# Estimate the RP for a 600 m^3/s discharge
GenLogEst(pars[1], pars[2], pars[3], q = 600)
```

GenLogGF *Generalised logistic distribution growth factors*

Description

Estimated growth factors as a function of return period, with inputs of Lcv & LSkew (linear coefficient of variation & linear skewness)

Usage

```
GenLogGF(lcv, lskew, RP)
```

Arguments

lcv	linear coefficient of variation
lskew	linear skewness
RP	return period

Details

Growth factors are calculated by the method outlined in the Flood Estimation Handbook, volume 3, 1999.

Value

Generalised logistic estimated growth factor

Author(s)

Anthony Hammond

Examples

```
# Estimate the 50-year growth factor from an Lcv and Lskew of 0.17 and 0.04, respectively
GenLogGF(0.17, 0.04, RP = 50)
```

GenLogPars *Generalised logistic distribution parameter estimates*

Description

Estimated parameters from a sample (with Lmoments or maximum likelihood estimation) or from L1 (first L-moment), Lcv (linear coefficient of variation), and LSkew (linear skewness)

Usage

```
GenLogPars(x = NULL, mle = FALSE, L1, LCV, LSKEW)
```

Arguments

x	numeric vector. The sample
mle	logical argument with a default of FALSE. If FALSE the parameters are estimated with Lmoments, if TRUE the parameters are estimated by maximum likelihood estimation.
L1	first Lmoment
LCV	linear coefficient of variation
LSKEW	linear skewness

Details

The L-moment estimated parameters are by the method detailed in 'Hosking J. and Wallis J. 1997 Regional Frequency Analysis: An Approach Based on L-Moments. Cambridge University Press, New York'.

This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

Parameter estimates (location, scale, shape)

Author(s)

Anthony Hammond

Examples

```
# Get an annual maximum sample and estimate the parameters using L-moments
am_27090 <- GetAM(27090)
GenLogPars(am_27090$Flow)

# Estimate parameters using MLE
GenLogPars(am_27090$Flow, mle = TRUE)

# Calculate L-moments and estimate the parameters with L1, Lcv, and Lskew
LMoments(am_27090$Flow)

# Store L-moments in an object
l_pars <- as.numeric(LMoments(am_27090$Flow))[c(1, 5, 6)]
GenLogPars(L1 = l_pars[1], LCV = l_pars[2], LSKEW = l_pars[3])
```

GenParetoEst — *Generalised Pareto distribution estimates from parameters*

Description

Estimated quantiles as function of return period (RP) and vice versa, from user input parameters

Usage

```
GenParetoEst(loc, scale, shape, q = NULL, RP = 100, ppy = 1)
```

Arguments

loc	location parameter
scale	scale parameter
shape	shape parameter
q	quantile. magnitude of the variable under consideration
RP	return period
ppy	peaks per year. Default is one

Details

If the argument q is used, it overrides RP and provides RP as a function of q (magnitude of variable) as opposed to q as a function of RP. The average number of peaks per year argument (ppy) is necessary when ppy is not equal to 1.

This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

quantile as a function of RP or vice versa

Author(s)

Anthony Hammond

Examples

```
# Get a POT sample, estimate the parameters, and estimate the 50-year RP
thames_pot <- POTextract(ThamesPQ[, c(1, 3)], thresh = 0.90)
GenParetoPars(thames_pot$peak)

# Store the parameters in an object
pars <- as.numeric(GenParetoPars(thames_pot$peak))

# Get an estimate of 50-year flow
GenParetoEst(pars[1], pars[2], pars[3], ppy = 1.867, RP = 50)

# Estimate the RP for a 600 m^3/s discharge
GenParetoEst(pars[1], pars[2], pars[3], ppy = 1.867, q = 600)
```

GenParetoGF *Generalised Pareto distribution growth factors*

Description

Estimated growth factors as a function of return period, with inputs of Lcv & LSkew (linear coefficient of variation & linear skewness). The Lcv and LSkew in this case should be calculated from peaks over threshold data and the ppy argument is necessary where the average number of peaks per year is not 1.

Usage

```
GenParetoGF(lcv, lskew, RP, ppy = 1)
```

Arguments

lcv	linear coefficient of variation
lskew	linear skewness
RP	return period
ppy	peaks per year

Details

Growth factors (GF) are calculated by the method outlined in the Flood Estimation Handbook, volume 3, 1999. The average number of peaks per year argument (ppy) is for the function to convert from the peaks over threshold (POT) scale to the annual scale. For example, if there are 3 peaks per year, the probability associated with the 100-yr return period estimate would be 0.01/3 (i.e. an RP of 300 on the POT scale) rather than 0.01.

Value

Generalised Pareto estimated growth factor

Author(s)

Anthony Hammond

Examples

```
# Get POT flow data from the Thames at Kingston (noting the no. peaks per year).
# Then estimate the 100-year growth factor with lcv and lskew estimates
tpot <- POTextract(ThamesPQ[, c(1, 3)], thresh = 0.90)
GenParetoGF(Lcv(tpot$peak), LSkew(tpot$peak), RP = 100, ppy = 1.867)

# Multiply by the median of the POT data for an estimate of the 100-year flood
GenParetoGF(Lcv(tpot$peak), LSkew(tpot$peak), RP = 100, ppy = 1.867) * median(tpot$peak)
```

GenParetoPars *Generalised Pareto distribution parameter estimates*

Description

Estimated parameters from a sample (with Lmoments or maximum likelihood estimation) or from L1 (first L-moment), Lcv (linear coefficient of variation), and LSkew (linear skewness)

Usage

```
GenParetoPars(x = NULL, L1, LCV, LSKEW)
```

Arguments

`x`	numeric vector. The sample
`L1`	first Lmoment
`LCV`	linear coefficient of variation
`LSKEW`	linear skewness

Details

The L-moment estimated parameters are by the method detailed in 'Hosking J. and Wallis J. 1997 Regional Frequency Analysis: An Approach Based on L-Moments. Cambridge University Press, New York'.

This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

Parameter estimates (location, scale, shape)

Author(s)

Anthony Hammond

Examples

```
# Get a peaks over threshold sample and estimate the parameters using L-moments
thames_pot <- POTextract(ThamesPQ[, c(1, 3)], thresh = 0.90)
GenParetoPars(thames_pot$peak)

# Calculate L-moments and estimate the parameters with L1, Lcv, and Lskew
LMoments(thames_pot$peak)

# Store L-moments in an object
l_pars <- as.numeric(LMoments(thames_pot$peak))[c(1, 5, 6)]
GenParetoPars(L1 = l_pars[1], LCV = l_pars[2], LSKEW = l_pars[3])
```

GenParetoPOT — *Generalised Pareto distribution - estimates directly from sample*

Description

Estimated quantiles as function of return period (RP) and vice versa, directly from the data

Usage

```
GenParetoPOT(x, ppy = 1, RP = 100, q = NULL)
```

Arguments

x	numeric vector (peaks over threshold sample)
ppy	peaks per year
RP	return period (default = 100)
q	quantile (magnitude of variable)

Details

If the argument q is used, it overrides RP and provides RP as a function of q (magnitude of variable) as opposed to q as a function of RP. The average number of peaks per year argument (ppy) is for the function to convert from the peaks over threshold (POT) scale to the annual scale. For example, if there are 3 peaks per year, the probability associated with the 100-yr return period estimate would be 0.01/3 (i.e. an RP of 300 on the POT scale) rather than 0.01. The parameters are estimated by the method of L-moments, as detailed in 'Hosking J. and Wallis J. 1997 Regional Frequency Analysis: An Approach Based on L-Moments. Cambridge University Press, New York'.

This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

quantile as a function of RP or vice versa

Author(s)

Anthony Hammond

Examples

```
# Get a POT series and estimate the 50-year RP
thames_pot <- POTextract(ThamesPQ[, c(1, 3)], thresh = 0.90)
GenParetoPOT(thames_pot$peak, ppy = 1.867, RP = 50)

# Estimate the RP for a 600 m^3/s discharge
GenParetoPOT(thames_pot$peak, ppy = 1.867, q = 600)
```

GetAM — *Get an annual maximum sample from the National River Flow Archive sites suitable for pooling*

Description

Extracts the annual maximum peak flow sample and associated dates for the site of interest.

Usage

```
GetAM(ref)
```

Arguments

`ref`	the site reference of interest (numeric)

Value

A data.frame with columns; Date, Flow, and id

Author(s)

Anthony Hammond

Examples

```
# Get an AMAX sample and display it in the console
GetAM(203018)

# Save an AMAX sample as an object
am_203018 <- GetAM(203018)
```

GetCDs — *Get catchment descriptors from the National River Flow Archive sites considered suitable for median annual maximum flow estimation (QMED) and pooling.*

Description

Extracts the catchment descriptors for a site of interest from the National River Flow Archive. If the site is considered suitable for QMED and pooling the CDs are extracted from the QMEDData data.frame. Otherwise they are extracted using the NRFA API.

Usage

```
GetCDs(x)
```

Arguments

x	the site reference of interest (numeric)

Value

A data.frame with columns; Descriptor and Value.

Author(s)

Anthony Hammond

Examples

```
# Get CDs and display in the console
cds_203018 <- GetCDs(203018)
cds_203018
```

GetDataEA_QH *Get flow or level data from the Environment Agency's Hydrology Data Explorer*

Description

Function to extract flow or level data from the Environment Agency's Hydrology Data Explorer.

Usage

```
GetDataEA_QH(
  Lat = 54,
  Lon = -2.25,
  Range = 20,
  RiverName = NULL,
  WISKI_ID = NULL,
  From = NULL,
  To = NULL,
  Type = "flow",
  Period = "DailyMean"
)
```

Arguments

Lat	Latitude (as a decimal) for the centre of the search for gauges. You can convert BNG to Lat and Lon using the GridRefConvert function.
Lon	Longitude (as a decimal) for the centre of the search for gauges. You can convert BNG to Lat and Lon using the GridRefConvert function.
Range	Radius of search when using latitude and longitude inputs (km).
RiverName	Name of the river along which you want to search for gauges. Character string.
WISKI_ID	The WISKI ID for the gauge from which you want to obtain data (character string). Note that sometimes a preceding zero, which is not returned via the API, is needed. If the data extraction fails, this may be the cause and you can resolve it by including the preceding zero in the WISKI_ID.
From	Date for start of data extraction in the format of "2015-12-02". If NULL the start date of the available data is used.
To	Date for the end of data extraction in the format of "2015-12-02". If NULL end date of the available data is used.
Type	The variable to extract, either "flow" or "level"
Period	The sampling rate of the data you want. Either "DailyMax", "DailyMean", "Hourly", "15Mins".

Details

To find gauges you can input either a river name or a latitude and longitude. You can convert BNG to Lat and Lon using the ConvertGridRef function (you can also get lat and lon by left clicking on Google maps). The lat and lon option will provide all flow and level gauges within a specified range (default of 10km). This provides gauged details including the WISKI ID. You can get data from specific gauges using the WISKI_ID. Note that flow gauges also have level data available. You can get data from a date range using the From and To arguments or you can return all data by leaving the From and To as the default (NULL). Lastly, WISKI IDs are sometimes returned without a preceding 0 which might be necessary for the data extraction (oddly, most do have the necessary 0). If data extraction fails try adding a 0 to the beginning of the WISKI ID.

Value

If searching for gauge details with lat and lon or river name, then a list is returned. The first element is a dataframe with flow gauge details and the second is a dataframe of level gauge details. When extracting flow or level data with a WISKI ID then a dataframe with two columns is returned. The first being a Date or POSIXct column/vector and the second is the timeseries of interest.

Author(s)

Anthony Hammond

Examples

```
# Find gauges on the River Tame
## Not run:
GetDataEA_QH(RiverName = "Tame")

## End(Not run)

# Find gauges within 10km of a latitude/longitude coordinate somewhere near the
# centre of Dartmoor
## Not run:
GetDataEA_QH(Lat = 50.6, Lon = -3.9, Range = 10)

## End(Not run)

# Get all available daily maximum flow data from the Bellever gauge on the
# East Dart River
## Not run:
bellever_max <- GetDataEA_QH(WISKI_ID = "SX67F051")

## End(Not run)

# Get 15-minute data from the Bellever for the November 2024 event
## Not run:
bellever_nov_2024 <- GetDataEA_QH(
  WISKI_ID = "SX67F051",
  From = "2024-11-23", To = "2024-11-25", Period = "15Mins"
)
```

```
## End(Not run)
```

GetDataEA_Rain *Get Environment Agency rainfall data (England).*

Description

Extract rainfall data from the Environment Agency's Hydrology Data Explorer.

Usage

```
GetDataEA_Rain(
  Lat = 54,
  Lon = -2,
  Range = 10,
  WISKI_ID = NULL,
  Period = "Daily",
  From = NULL,
  To = NULL
)
```

Arguments

Lat	Latitude (as a decimal) for the centre of the search for gauges. You can convert BNG to Lat and Lon using the GridRefConvert function.
Lon	Longitude (as a decimal) for the centre of the search for gauges. You can convert BNG to Lat and Lon using the GridRefConvert function.
Range	The radius (km) from the point of interest (Lat, Lon) for which the user wants rain gauge information (currently works to a maximum of just over 20km).
WISKI_ID	The WISKI identification (as "character") for the rain gauge of interest
Period	The sampling rate of the rainfall in hours. Either "Daily", "15Mins", "Hourly".
From	The start date of the data extraction in the form of "YYYY-MM-DD". To get data from the first date available leave as NULL.
To	The end date of the data extraction in the form of "YYYY-MM-DD". To get data from the first date available leave as NULL.

Details

The function provides one of two outputs. Either information about available local rain gauges, or the data from a specified gauge (specified by WISKI ID). The process is to find the local information (including WISKI ID) by using the latitude and longitude and range (You can convert BNG to Lat and Lon using the GridRefConvert function). Then use the WISKI ID to get the data. If data requested is not available, for example - outside the date range or not available at the requested sampling rate, an error message is returned stating "no lines available in input". To extract all the available data leave the From and To arguments as Null.

Value

If searching for rain gauge details with the Latitude and Longitude a dataframe of gauges is returned. If extracting rainfall using the WISKI_ID, a dataframe is returned with Date / POSIXct in the first columns and rainfall in the second.

Author(s)

Anthony Hammond

Examples

```
# Get information about available rain gauges
# within a 10km radius of latitude 54.5 and longitude -3.2
## Not run:
GetDataEA_Rain(Lat = 54.5, Lon = -3.2)

## End(Not run)

# Use the WISKI reference ID for the Honister rain gauge
# to get some hourly rain data for December 2015
## Not run:
honister_dec_2015 <- GetDataEA_Rain(
  WISKI_ID = "592463",
  Period = "Hourly", From = "2015-12-01", To = "2015-12-31"
)

## End(Not run)

# Inspect the first few rows and plot the data
## Not run:
head(honister_dec_2015)
plot(honister_dec_2015, type = "h", ylab = "Rainfall (mm)")

## End(Not run)
```

GetDataMetOffice   *Get regional Met Office average temperature or rainfall series (monthly, seasonal, and annual).*

Description

Extracts regional mean temperature or rainfall from the Met Office UK & regional series. The total duration of bright sunshine is also available.

Usage

```
GetDataMetOffice(Variable, Region)
```

Arguments

`Variable`	Either Tmean, Rainfall, or Sunshine
`Region`	One of "UK", "England", "Wales", "Scotland", "Northern_Ireland", "England_and_Wales", "England_N", "England_S", "Scotland_N", "Scotland_E", "Scotland_W", "England_E_and_NE", "England_NW_and_N_Wales", "Midlands", "East_Anglia", "England_SW_and_S_Wales", "England_SE_and_Central_S".

Details

The function returns time series data from the 19th century through to the present month.

Value

A data.frame with 18 columns; year, months, seasons, and annual. Rows then represent each year of the timeseries.

Author(s)

Anthony Hammond

Examples

```
# Get the rainfall time series for the UK
## Not run:
uk_rain <- GetDataMetOffice(Variable = "Rainfall", Region = "UK")

## End(Not run)

# Get the mean temperature data for East Anglia
## Not run:
temp_east_anglia <- GetDataMetOffice(Variable = "Tmean", Region = "East_Anglia")

## End(Not run)
```

`GetDataNRFA` *Get National River Flow Archive data using gauge ID.*

Description

Extracts NRFA data using the API.

Usage

```
GetDataNRFA(ID, Type = "Q")
```

Arguments

ID	ID number of the gauge of interest.
Type	Type of data required. One of "Q", "P", "PQ", "Gaugings", "AMAX", "POT", or "Catalogue".

Details

The function can be used to get daily catchment rainfall or mean flow, or both together (concurrent). It can also be used to get gaugings, AMAX, and POT data. Note that some sites have rejected peak flow years. In which case, if Type = AMAX or POT, the function returns a list, the first element of which is the rejected years, the second is the full AMAX or POT. Lastly if Type = "Catalogue" it will return a dataframe of all the NRFA gauges, associated details, comments, and descriptors.

Value

A data.frame with date in the first columns and variable/s of interest in the remaining column/s. Except for the following circumstances: When Type = "Catalogue", then a large dataframe is returned with all the NRFA gauge metadata. When Type = "AMAX" or "POT" and there are rejected years a list is returned, where the first element is the dataframe of data and the second is rejected year/s (character string).

Author(s)

Anthony Hammond

Examples

```
# Get the concurrent rainfall (P) and mean flow (Q) series for the Tay at Ballathie (site 15006)
## Not run:
ballathie_pq <- GetDataNRFA(15006, "PQ")

## End(Not run)

# Get the gaugings
## Not run:
ballathie_gaugings <- GetDataNRFA(15006, "Gaugings")

## End(Not run)
```

GetDataSEPA_QH *Get flow or level data from the Scottish Environmental Protection Agency.*

Description

Function to extract flow or level data from SEPA.

Usage

```
GetDataSEPA_QH(
  Lat = NULL,
  Lon = NULL,
  RiverName = NULL,
  Type = "Flow",
  StationID = NULL,
  Range = 20,
  From = NULL,
  To = NULL,
  Period = "Daily"
)
```

Arguments

`Lat`	Latitude (as a decimal) for the centre of the search for gauges. You can convert BNG to Lat and Lon using the GridRefConvert function.
`Lon`	Longitude (as a decimal) for the centre of the search for gauges. You can convert BNG to Lat and Lon using the GridRefConvert function.
`RiverName`	Name of the river along which you want to search for gauges. Character string.
`Type`	The variable to extract, either "flow" or "level"
`StationID`	The ID for the gauge from which you want to obtain data (character string)
`Range`	Radius of search when using latitude and longitude inputs (km).
`From`	Date for start of data extraction in the format of "2015-12-02". If NULL the first date of the available data is used.
`To`	Date for the end of data extraction in the format of "2015-12-02". If NULL the present date is used (and most recent available data is returned).
`Period`	The sampling rate of the data you want. Either "Daily", "Hourly", or "15Mins".

Details

To find gauges you can input either a river name or a latitude and longitude. You can convert BNG to Lat and Lon using the ConvertGridRef function. The lat and lon option will provide all flow or level gauges within a specified range (default of 50km). This provides gauged details including the StationID. You can get data from specific gauges using the StationID. Note that flow gauges also have level data available. You can get data from a date range using the From and To arguments. If the From and To arguments are left as NULL the full range of available data are returned.

Value

If searching for gauge details with lat and lon or river name, then a dataframe is returned with necessary information to obtain flow or level data. When extracting flow or level data with a station ID then a dataframe with two columns is returned. The first being a Date or POSIXct column/vector and the second is the timeseries of interest.

Author(s)

Anthony Hammond

Examples

```
# Find gauges on the River Spey
## Not run:
GetDataSEPA_QH(RiverName = "Spey")

## End(Not run)

# Find gauges within 20km of a latitude/longitude coordinate somewhere near the centre of Scotland
# This example causes a fatal session error in this version of the UKFE package
## Not run:
GetDataSEPA_QH(lat = 56, lon = -4, Range = 20)

## End(Not run)

# Get all available daily mean flow data from the Boat o Brig gauge on the Spey
## Not run:
spey_daily <- GetDataSEPA_QH(StationID = "37174")

## End(Not run)

# Get 15-minute data from the Boat o Brig for the highest recorded peak
## Not run:
boatobrig_aug_1970 <- GetDataSEPA_QH(
  StationID = "37174",
  From = "1970-08-16", To = "1970-08-19", Period = "15Mins"
)

## End(Not run)
```

GetDataSEPA_Rain — *Get Scottish Environment Protection Agency (SEPA) hourly rainfall data.*

Description

Extract hourly rainfall data from SEPA's API.

Usage

```
GetDataSEPA_Rain(
  Lat = NULL,
  Lon = NULL,
  Range = 30,
  StationName,
```

```
  From = NULL,
  To = NULL
)
```

Arguments

Lat	Latitude of the point of interest. Provided when the user wants information about available local rain gauges
Lon	Longitude of the point of interest. Provided when the user wants information about available local rain gauges
Range	The radius (km) from the point of interest (Lat, Lon) for which the user wants a list of rain gauges (default is 30).
StationName	The name of the station for which you want rainfall. If you type something other than one of the available stations, the list of stations will be returned.
From	A start date for the data in the form of "YYYY-MM-DD". Default of NULL means the earliest available date is used
To	An end date for the data in the form of "YYYY-MM-DD". The default is the most recent date available.

Details

You can download data using the gauge name and you can find gauges within a given range using the latitude and longitude. If the 'From' date is left as null, the earliest date of available data will be used. If the 'To' date is left as null, the most recent date of available data will be used.

Value

A data.frame with POSIXct in the first column, and rainfall in the second column. Unless the StationName provided is not in the available list, then the available list is returned.

Author(s)

Anthony Hammond

Examples

```
# Get the list of available stations
## Not run:
GetDataSEPA_Rain(StationName = "AnythingButAStationName")

## End(Not run)

# Get rain from the Bannockburn station
## Not run:
bannockburn <- GetDataSEPA_Rain(
  StationName = "Bannockburn",
  From = "1998-10-01", To = "1998-10-31"
)
```

```
## End(Not run)

# Inspect the first few rows and plot the data
## Not run:
head(bannockburn)
plot(bannockburn, type = "h", ylab = "Rainfall (mm)")

## End(Not run)
```

GetQMED *QMED from a gauged site suitable for QMED*

Description

Provides QMED (median annual maximum flow) from a site suitable for QMED, using the site reference. This provides the observed median of the annual maximum sample (excluding rejected observations) for the site of interest. It does not apply any adjustments or updates to account for non-stationarity.

Usage

```
GetQMED(x)
```

Arguments

x	the gauged reference

Value

the median annual maximum

Author(s)

Anthony Hammond

Examples

```
# Get the observed QMED from site 55002
GetQMED(55002)
```

GEVAM — *Generalised extreme value distribution - estimates directly from sample*

Description

Estimated quantiles as function of return period (RP) and vice versa, directly from the data

Usage

```
GEVAM(x, RP = 100, q = NULL)
```

Arguments

`x`	numeric vector (block maxima sample)
`RP`	return period (default = 100)
`q`	quantile (magnitude of variable)

Details

If the argument q is used, it overrides RP and provides RP as a function of q (magnitude of variable) as opposed to q as a function of RP. The parameters are estimated by the method of L-moments, as detailed in 'Hosking J. and Wallis J. 1997 Regional Frequency Analysis: An Approach Based on L-Moments. Cambridge University Press, New York'.

This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

quantile as a function of RP or vice versa.

Author(s)

Anthony Hammond

Examples

```
# Get an annual maximum sample and estimate the 50-year RP
am_27090 <- GetAM(27090)
GEVAM(am_27090$Flow, RP = 50)

# Estimate the RP for a 600 m^3/s discharge
GEVAM(am_27090$Flow, q = 600)
```

GEVEst *Generalised extreme value distribution estimates from parameters*

Description

Estimated quantiles as function of return period (RP) and vice versa, from user input parameters

Usage

```
GEVEst(loc, scale, shape, q = NULL, RP = 100)
```

Arguments

loc	location parameter
scale	scale parameter
shape	shape parameter
q	quantile. magnitude of the variable under consideration
RP	return period

Details

If the argument q is used, it overrides RP and provides RP as a function of q (magnitude of variable) as opposed to q as a function of RP.

This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

quantile as a function of RP or vice versa

Author(s)

Anthony Hammond

Examples

```
# Get an annual maximum sample, estimate the parameters, and estimate the 50-year RP
am_27090 <- GetAM(27090)
GEVPars(am_27090$Flow)

# Store the parameters in an object
pars <- as.numeric(GEVPars(am_27090$Flow))

# Get an estimate of 50-year flow
GEVEst(pars[1], pars[2], pars[3], RP = 50)
```

```
# Estimate the RP for a 600 m^3/s discharge
GEVEst(pars[1], pars[2], pars[3], q = 600)
```

GEVGF *Generalised extreme value distribution growth factors*

Description

Estimated growth factors as a function of return period, with inputs of Lcv & LSkew (linear coefficient of variation & linear skewness)

Usage

```
GEVGF(lcv, lskew, RP)
```

Arguments

lcv	linear coefficient of variation
lskew	linear skewness
RP	return period

Details

Growth factors are calculated by the method outlined in the Flood Estimation Handbook, volume 3, 1999.

Value

Generalised extreme value estimated growth factor

Author(s)

Anthony Hammond

Examples

```
# Estimate the 50-year growth factor from Lcv = 0.17 and Lskew = 0.04
GEVGF(0.17, 0.04, RP = 50)
```

`GEVPars` *Generalised extreme value distribution parameter estimates*

Description

Estimated parameters from a sample (with Lmoments or maximum likelihood estimation) or from L1 (first L-moment), Lcv (linear coefficient of variation), and LSkew (linear skewness)

Usage

```
GEVPars(x = NULL, mle = FALSE, L1 = NULL, LCV = NULL, LSKEW = NULL)
```

Arguments

`x`	numeric vector. The sample
`mle`	logical argument with a default of FALSE. If FALSE the parameters are estimated with Lmoments, if TRUE the parameters are estimated by maximum likelihood estimation
`L1`	first Lmoment
`LCV`	linear coefficient of variation
`LSKEW`	linear skewness

Details

The L-moment estimated parameters are by the method detailed in 'Hosking J. and Wallis J. 1997 Regional Frequency Analysis: An Approach Based on L-moments. Cambridge University Press, New York'.

This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

Parameter estimates (location, scale, shape)

Author(s)

Anthony Hammond

Examples

```
# Get an annual maximum sample and estimate the parameters using L-moments
am_27090 <- GetAM(27090)
GEVPars(am_27090$Flow)

# Estimate parameters using MLE
GEVPars(am_27090$Flow, mle = TRUE)

# Calculate L-moments and estimate the parameters with L1, Lcv, and Lskew
LMoments(am_27090$Flow)

# Store L-moments in an object
l_pars <- as.numeric(LMoments(am_27090$Flow))[c(1, 5, 6)]
GEVPars(L1 = l_pars[1], LCV = l_pars[2], LSKEW = l_pars[3])
```

GoFCompare *Goodness of fit comparison (single sample)*

Description

compares the RMSE of four distribution fits for a single AMAX sample.

Usage

```
GoFCompare(x)
```

Arguments

x	a numeric vector (your AMAX sample)

Details

This function calculates an RMSE fit score for four distributions (GEV, GenLog, Gumbel, & Kappa3). The lowest RMSE is the best fit. It works as follows. For each distribution: Step1. Simulate 500 samples the same size as x. Step2. Calculate the mean across all 500 samples for each rank to create an ordered central estimate. Step3. Calculate the RMSE between the result of step 2 and the ordered x. Step4. Standardise the RMSE by dividing it by the mean of x and multiply it by 100 (RMSE as a percentage of mean). Note that this is not a hypothesis test. It is only for comparing the fit across the distributions.

Value

A list. The first element is a dataframe with four columns and one row of results. Each column has the standardised RMSE associated with one of the four distributions (GEV, GenLog, Gumbel, Kappa3). The second element is a character string stating the distribution with the best fit.

Author(s)

Anthony Hammond

Examples

```
# Get an AMAX sample and then compare the fit
am_15006 <- GetAM(15006)
GoFCompare(am_15006$Flow)
```

GoFComparePool *Goodness of fit comparison (for a pooling group)*

Description

compares the RMSE of four distribution fits for a pooling group.

Usage

```
GoFComparePool(x)
```

Arguments

x	a numeric vector (your AMAX sample)

Details

This function calculates an RMSE fit score for four distributions (GEV, GenLog, Gumbel, & Kappa3). The lowest RMSE is the best fit. It works for pooling groups created using the Pool or PoolSmall function. It uses the same method as GoFCompare (see the associated details of that function). It first standardises the pooled AMAX samples (by dividing them by median) and then treats them as a single large sample. Note that this is not a hypothesis test. It is only for comparing the fit across the distributions.

Value

A list. The first element is a dataframe with four columns and one row of results. Each column has the standardised RMSE associated with one of the four distributions (GEV, GenLog, Gumbel, Kappa3). The second element is a character string stating the distribution with the best fit.

Author(s)

Anthony Hammond

Examples

```
# Get a pooling group and then compare the fit
pool_60009 <- Pool(GetCDs(60009))
GoFComparePool(pool_60009)
```

GumbelAM *Gumbel distribution - estimates directly from sample*

Description

Estimated quantiles as a function of return period (RP) and vice versa, directly from the data

Usage

```
GumbelAM(x, RP = 100, q = NULL)
```

Arguments

x	numeric vector (block maxima sample)
RP	return period (default = 100)
q	quantile (magnitude of variable)

Details

If the argument q is used, it overrides RP and provides RP as a function of q (magnitude of variable) as opposed to q as a function of RP. The parameters are estimated by the method of L-moments, as detailed in 'Hosking J. and Wallis J. 1997 Regional Frequency Analysis: An Approach Based on L-Moments. Cambridge University Press, New York'.

This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

quantile as a function of RP or vice versa.

Author(s)

Anthony Hammond

Examples

```
# Get an annual maximum sample and estimate the 50-year RP
am_27090 <- GetAM(27090)
GumbelAM(am_27090$Flow, RP = 50)

# Estimate the RP for a 600 m^3/s discharge
GumbelAM(am_27090$Flow, q = 600)
```

GumbelEst — *Gumbel distribution estimates from parameters*

Description

Estimated quantiles as function of return period (RP) and vice versa, from user input parameters

Usage

```
GumbelEst(loc, scale, q = NULL, RP = 100)
```

Arguments

`loc`	location parameter
`scale`	scale parameter
`q`	quantile. magnitude of the variable under consideration
`RP`	return period

Details

If the argument q is used, it overrides RP and provides RP as a function of q (magnitude of variable) as opposed to q as a function of RP.

This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

quantile as a function of RP or vice versa

Author(s)

Anthony Hammond

Examples

```
# Get an annual maximum sample, estimate the parameters, and estimate the 50-year RP
am_27090 <- GetAM(27090)
pars <- as.numeric(GumbelPars(am_27090$Flow))
GumbelEst(pars[1], pars[2], RP = 50)

# Estimate the RP for a 600 m^3/s discharge
GumbelEst(pars[1], pars[2], q = 600)
```

GumbelGF — *Gumbel distribution growth factors*

Description

Estimated growth factors as a function of return period, with inputs of Lcv & LSkew (linear coefficient of variation & linear skewness)

Usage

```
GumbelGF(lcv, RP)
```

Arguments

`lcv`	linear coefficient of variation
`RP`	return period

Details

Growth factors are calculated by the method outlined in the Flood Estimation Handbook, volume 3, 1999.

Value

Gumbel estimated growth factor

Author(s)

Anthony Hammond

Examples

```
# Estimate the 50-year growth factor from an Lcv of 0.17
GumbelGF(0.17, RP = 50)
```

GumbelPars — *Gumbel distribution parameter estimates*

Description

Estimated parameters from a sample (with Lmoments or maximum likelihood estimation) or from L1 (first L-moment), Lcv (linear coefficient of variation)

Usage

```
GumbelPars(x = NULL, mle = FALSE, L1, LCV)
```

Arguments

x	numeric vector. The sample
mle	logical argument with a default of FALSE. If FALSE the parameters are estimated with Lmoments, if TRUE the parameters are estimated by maximum likelihood estimation
L1	first Lmoment
LCV	linear coefficient of variation

Details

The L-moment estimated parameters are by the method detailed in 'Hosking J. and Wallis J. 1997 Regional Frequency Analysis: An Approach Based on L-Moments. Cambridge University Press, New York'.

This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

Parameter estimates (location, scale)

Author(s)

Anthony Hammond

Examples

```
# Get an annual maximum sample and estimate the parameters using L-moments
am_27090 <- GetAM(27090)
GumbelPars(am_27090$Flow)

# Estimate parameters using MLE
GumbelPars(am_27090$Flow, mle = TRUE)

# Calculate L-moments and estimate the parameters with L1 and Lcv
pars <- as.numeric(LMoments(am_27090$Flow)[c(1, 5)])
GumbelPars(L1 = pars[1], LCV = pars[2])
```

H2 *Heterogeneity measure (H2) for pooling groups.*

Description

Quantifies the heterogeneity of a pooled group

Usage

```
H2(x, H1 = FALSE)
```

Arguments

x	pooling group derived from the Pool() function
H1	logical with a default of FALSE. If TRUE, the function applies the 'H1' version of the test (see Hosking & Wallis 1997 reference). If FALSE, the default H2 version is applied.

Details

The H2 measure was developed by Hosking & Wallis and can be found in their book 'Regional Frequency Analysis: An Approach Based on L-Moments' (1997). It was also adopted for use by the Flood Estimation Handbook (1999) and is described in volume 3. It works by recreating 500 pooling groups with the same sample sizes, assuming a four parameter Kappa distribution (parameters from the pooled L-moments). L-moment ratios are calculated for each of the 500 simulated pooling groups. The heterogeneity is determined by comparing the variance of L-moment ratios in the observed pooling group with the variance of the L-moment ratios across the simulated pooling groups. The simulations are homogeneous, therefore if the observed pooling group is homogeneous the expectation is that the variance will be similar to the average of the simulated variance.

Value

A vector of two characters; the first representing the H2 score and the second stating a qualitative measure of heterogeneity.

Author(s)

Anthony Hammond

Examples

```
# Get CDs, form a pooling group, and calculate H2
cds_203018 <- GetCDs(203018)
pool_203018 <- Pool(cds_203018)
H2(pool_203018)
```

HydroPlot *Hydrological plot of concurrent discharge and precipitation*

Description

Plots concurrent precipitation and discharge, with precipitation along the top and discharge along the bottom

Usage

```
HydroPlot(
  x,
  main = "Concurrent Rainfall & Discharge",
  ylab = "Discharge (m3/s)",
  From = NULL,
  To = NULL,
  adj.y = 1.5,
  plw = 1,
  qlw = 1.8,
  Return = FALSE
)
```

Arguments

`x`	a data.frame with three columns in the order of date (or POSIXct), precipitation, and discharge
`main`	a character string. The user chosen plot title. The default is "Concurrent Rainfall & Discharge"
`ylab`	User choice for the y label of the plot. The default is "Discharge (m3/s)".
`From`	a starting time for the plot. In the form of a date or POSIXct object. The default is the first row of x
`To`	an end time for the plot. In the form of a date or POSIXct object. The default is the last row of x
`adj.y`	a numeric value to adjust the closeness of the preciptation and discharge in the plot. Default is 1.5. A lower value brings them closer and a larger value further apart
`plw`	a numeric value to adjust the width of the precipitation lines. Default is one. A larger value thickens them and vice versa
`qlw`	a numeric value to adjust the width of the discharge line. Default is 1.8. A larger value thickens them and vice versa
`Return`	a logical argument with a default of FALSE. If TRUE the data-frame of time, precipitation, and flow is returned

Details

The input of x is a dataframe with the first column being time. If the data is sub daily this should be class POSIXct with time as well as date.

Value

A plot of concurrent precipitation and discharge, with the former at the top and the latter at the bottom. If the Return argument equals true the associated data-frame is also returned.

Author(s)

Anthony Hammond

Examples

```
# Plot the Thames precipitation and discharge for the 2013 hydrological year,
# adjusting the y axis to 1.8
HydroPlot(ThamesPQ, From = "2013-10-01", To = "2014-09-30", adj.y = 1.8)
```

Kappa3AM — *Kappa3 distribution - estimates directly from sample*

Description

Estimated quantiles as a function of return period (RP) and vice versa, directly from the data

Usage

```
Kappa3AM(x, RP = 100, q = NULL)
```

Arguments

x	numeric vector (block maxima sample)
RP	return period (default = 100)
q	quantile (magnitude of variable)

Details

If the argument q is used, it overrides RP and provides RP as a function of q (magnitude of variable) as opposed to q as a function of RP. The parameters are estimated by the method of L-moments, as detailed in 'Hosking J. and Wallis J. 1997 Regional Frequency Analysis: An Approach Based on L-moments. Cambridge University Press, New York'. This is the Kappa3 distribution as defined in Kjeldsen, T. (2019), 'The 3-parameter Kappa distribution as an alternative for use with FEH pooling groups.' (Circulation - The Newsletter of the British Hydrological Society, no. 142).

This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

quantile as a function of RP or vice versa.

Author(s)

Anthony Hammond

Examples

```
# Get an annual maximum sample and estimate the 50-year RP
am_27090 <- GetAM(27090)
Kappa3AM(am_27090$Flow, RP = 50)

# Estimate the RP for a 600 m^3/s discharge
Kappa3AM(am_27090$Flow, q = 600)
```

Kappa3Est *Kappa3 distribution estimates from parameters*

Description

Estimated quantiles as function of return period (RP) and vice versa, from user input parameters

Usage

```
Kappa3Est(loc, scale, shape, q = NULL, RP = 100)
```

Arguments

loc	location parameter
scale	scale parameter
shape	shape parameter
q	quantile. magnitude of the variable under consideration
RP	return period

Details

If the argument q is used, it overrides RP and provides RP as a function of q (magnitude of variable) as opposed to q as a function of RP. This is the Kappa3 distribution as defined in Kjeldsen, T (2019), 'The 3-parameter Kappa distribution as an alternative for use with FEH pooling groups.' (Circulation - The Newsletter of the British Hydrological Society, no. 142).

This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

quantile as a function of RP or vice versa

Author(s)

Anthony Hammond

Examples

```
# Get an annual maximum sample, estimate the parameters, and the 50-year RP flow
am_27090 <- GetAM(27090)

# Get the parameters and store in an object
pars <- as.numeric(Kappa3Pars(am_27090$Flow))

# Get an estimate of the 50-year flow
Kappa3Est(pars[1], pars[2], pars[3], RP = 50)

# Estimate the RP for a 600 m^3/s discharge
Kappa3Est(pars[1], pars[2], pars[3], q = 600)
```

Kappa3GF *Kappa3 distribution growth factors*

Description

Estimated growth factors as a function of return period, with inputs of Lcv & LSkew (linear coefficient of variation & linear skewness)

Usage

```
Kappa3GF(lcv, lskew, RP)
```

Arguments

lcv	linear coefficient of variation
lskew	linear skewness
RP	return period

Details

Growth factors are calculated by the method outlined in Kjeldsen, T (2019), 'The 3-parameter Kappa distribution as an alternative for use with FEH pooling groups.'Circulation - The Newsletter of the British Hydrological Society, no. 142

Value

Kappa3 distribution estimated growth factor

Author(s)

Anthony Hammond

Examples

```
# Get an ungauged pooled Lcv and LSkew for catchment 15006
pooled_res <- as.numeric(QuickResults(GetCDs(15006), plot = FALSE)[[2]])

# Calculate Kappa growth factor for the 100-year flood
Kappa3GF(pooled_res[1], pooled_res[2], RP = 100)
```

`Kappa3Pars` *Kappa3 distribution parameter estimates*

Description

Estimated parameters from a sample (using Lmoments) or from user supplied L1 (first L-moment), Lcv (linear coefficient of variation), and LSkew (linear skewness)

Usage

```
Kappa3Pars(x = NULL, L1 = NULL, LCV = NULL, LSKEW = NULL)
```

Arguments

`x`	numeric vector. The sample
`L1`	first Lmoment
`LCV`	linear coefficient of variation
`LSKEW`	linear skewness

Details

The L-moment estimated parameters are by the method detailed in 'Hosking J. and Wallis J. 1997 Regional Frequency Analysis: An Approach Based on L-moments. Cambridge University Press, New York'. This is the Kappa3 distribution as defined in Kjeldsen, T (2019), 'The 3-parameter Kappa distribution as an alternative for use with FEH pooling groups.' (Circulation - The Newsletter of the British Hydrological Society, no. 142).

This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

Parameter estimates (location, scale, shape)

Author(s)

Anthony Hammond

Examples

```
# Get an annual maximum sample and estimate the parameters
am_27090 <- GetAM(27090)
Kappa3Pars(am_27090$Flow)

# Calculate L-moments and estimate the parameters with L1, LCV, and LSKEW
l_pars <- as.numeric(LMoments(am_27090$Flow))[c(1, 5, 6)]
Kappa3Pars(L1 = l_pars[1], LCV = l_pars[2], LSKEW = l_pars[3])
```

Lcv *Linear coefficient of variation (Lcv)*

Description

Calculates the Lcv from a sample of data

Usage

```
Lcv(x)
```

Arguments

x	a numeric vector. The sample of interest

Details

Lcv calculated according to methods outlined by Hosking & Wallis (1997): Regional Frequency Analysis and approach based on LMoments. Also in the Flood Estimation Handbook (1999), volume 3.

Value

Numeric. The Lcv of a sample.

Author(s)

Anthony Hammond

Examples

```
# Get an AMAX sample and calculate the L-moments
am_27051 <- GetAM(27051)
Lcv(am_27051$Flow)
```

LcvUrb *Urban adjustment for the linear coefficient of variation (Lcv)*

Description

Urbanises or de-urbanises the Lcv using the methods outlined in the guidance by Wallingford HydroSolutions: 'WINFAP 4 Urban Adjustment Procedures'

Usage

```
LcvUrb(lcv, URBEXT2000, DeUrb = FALSE)
```

Arguments

`lcv`	the Lcv (numeric)
`URBEXT2000`	quantification of urban and suburbanisation for the subject catchment
`DeUrb`	logical argument with a default of FALSE. If set to TRUE, de-urbanisation adjustment is performed, if FALSE, urbanisation adjustment is performed

Details

The method for de-urbanisation isn't explicitly provided in 'WINFAP 4 Urban Adjustment Procedures', but the procedure is a re-arrangement of the urbanisation equation, solving for Lcv rather than Lcv-urban. The functionality assumes that the variance of an annual maximum flow sample is impacted by urbanisation and that this impact can be modelled as a function of URBEXT2000.

Value

The urban adjust Lcv or the de-urbanised Lcv

Author(s)

Anthony Hammond

Examples

```
# Choose an urban site (site 53006) from the NRFA data then apply a de-urban
# adjustment using the Lcv and URBEXT2000 displayed
NRFAData[which(rownames(NRFAData) == 53006), ]
LcvUrb(0.21, 0.1138, DeUrb = TRUE)

# Get the pooled L-moment ratios results for catchment 53006 and apply the
# urban adjustment using the pooled Lcv and the URBEXT2000 for site 53006
cds_53006 <- GetCDs(53006)
QuickResults(cds_53006)[[2]]
LcvUrb(0.196, 0.1138)
```

LKurt *Linear Kurtosis (LKurt)*

Description

Calculates the LKurtosis from a sample of data

Usage

```
LKurt(x)
```

Arguments

x	a numeric vector. The sample of interest

Details

LKurtosis calculated according to methods outlined by Hosking & Wallis (1997): Regional Frequency Analysis and approach based on LMoments. Also in the Flood Estimation Handbook (1999), volume 3.

Value

Numeric. The LSkew of a sample.

Author(s)

Anthony Hammond

Examples

```
# Get an AMAX sample and calculate the L-moments
am_27051 <- GetAM(27051)
LKurt(am_27051$Flow)
```

LMoments *Lmoments & Lmoment ratios*

Description

Calculates the Lmoments and Lmoment ratios from a sample of data

Usage

```
LMoments(x)
```

Arguments

x	a numeric vector. The sample of interest

Details

Lmoments calculated according to methods outlined by Hosking & Wallis (1997): Regional Frequency Analysis and approach based on LMoments. Also in the Flood Estimation Handbook (1999), volume 3.

Value

A data.frame with one row and column headings; L1, L2, L3, L4, Lcv, LSkew, and LKurt. The first four are the Lmoments and the next three are the Lmoment ratios.

Author(s)

Anthony Hammond

Examples

```
# Get an AMAX sample and calculate the L-moments
am_27051 <- GetAM(27051)
LMoments(am_27051$Flow)
```

LRatioChange — *Adjust L-Ratios in a pooling group*

Description

Adjusts the linear coefficient of variation (Lcv) and the linear skewness (LSkew) for a chosen site in a pooling group

Usage

```
LRatioChange(x, SiteID, lcv, lskew)
```

Arguments

x	pooling group derived with the Pool function
SiteID	the identification number of the site in the pooling group that is to be changed (character or integer)
lcv	The user supplied Lcv. numeric
lskew	The user supplied LSkew. numeric

Details

Pooling groups are formed from the NRFAData data.frame and all the Lcv and LSkew values are precalculated using the National River Flow Archive Peak flow dataset noted in the description file. The resulting pooled growth curve is calculated using the Lcv and Lskew in the pooled group. The user may have further data and be able to add further peak flows to the annual maximum samples within a pooling group. If that is the case a new Lcv and Lskew can be determined using the LMoments function. These new values can be added to the pooling group with this LRatioChange function. Also the non-flood years adjustment function may have been applied to a site, which provides a new Lcv and LSkew. In which case, the LRatioChange function can be applied. The function creates a new pooling group object and x will still exist in it's original state after the function is applied.

Value

A new pooling group, the same as x except for the user adjusted Lcv and Lskew for the user selected site.

Author(s)

Anthony Hammond

Examples

```
# Get some catchment descriptors and create a pooling group
cds_39001 <- GetCDs(39001)
pool_39001 <- Pool(cds_39001, iug = TRUE)

# Apply the function to create a new adjusted pooling group,
# changing the subject site's lcv and lskew to 0.187 and 0.164, respectively
pool_39001_adj <- LRatioChange(pool_39001, SiteID = 39001, lcv = 0.187, lskew = 0.164)
```

`LSkew` *Linear Skewness (LSkew)*

Description

Calculates the LSkew from a sample of data

Usage

```
LSkew(x)
```

Arguments

x	a numeric vector. The sample of interest

Details

LSkew calculated according to methods outlined by Hosking & Wallis (1997): Regional Frequency Analysis and approach based on LMoments. Also in the Flood Estimation Handbook (1999), volume 3.

Value

Numeric. The LSkew of a sample.

Author(s)

Anthony Hammond

Examples

```
# Get an AMAX sample and calculate the L-moments
am_27051 <- GetAM(27051)
LSkew(am_27051$Flow)
```

`LSkewUrb` *Urban adjustment for the linear skewness (LSkew)*

Description

Urbanises or de-urbanises the LSkew using the methods outlined in the guidance by Wallingford HydroSolutions: 'WINFAP 4 Urban Adjustment Procedures'

Usage

```
LSkewUrb(lskew, URBEXT2000, DeUrb = FALSE)
```

Arguments

`lskew`	the LSkew (numeric)
`URBEXT2000`	quantification of urban and suburbanisation for the subject site
`DeUrb`	logical argument with a default of FALSE. If set to TRUE, de-urbanisation adjustment is performed, if FALSE, urbanisation adjustment is performed

Details

The method for de-urbanisation isn't explicitly provided in 'WINFAP 4 Urban Adjustment Procedures', but the procedure is a re-arrangement of the urbanisation equation, solving for LSkew rather than LSkew-urban. The functionality assumes that the skewness of an annual maximum flow sample is impacted by urbanisation and that this impact can be modelled as a function of URBEXT2000.

Value

The urban adjust Lskew or the de-urbanised LSkew

Author(s)

Anthony Hammond

Examples

```
# Choose an urban site (site 53006) from the NRFA data then apply a de-urban
# adjustment using the Lcv and URBEXT2000 displayed
NRFAData[which(rownames(NRFAData) == 53006), ]
LSkewUrb(0.124, 0.1138, DeUrb = TRUE)

# Get the pooled L-moment ratios results for catchment 53006 and apply the
# urban adjustment using the pooled LSkew and the URBEXT2000 for site 53006
cds_53006 <- GetCDs(53006)
QuickResults(cds_53006)[[2]]
LSkewUrb(0.194, 0.1138)
```

MonthlyStats *Monthly Statistics*

Description

Derives monthly statistics from a data.frame with Dates or POSIXct in the first column and variable of interest in the second

Usage

```
MonthlyStats(
  x,
  Stat,
  AggStat = NULL,
  TS = FALSE,
  Plot = FALSE,
  ylab = "Magnitude",
  main = "Monthly Statistics",
  col = "grey"
)
```

Arguments

x	a data.frame with Dates or POSIXct in the first column and numeric vector in the second.
Stat	A user chosen function to calculate the statistic of interest; mean or sum for example. Could be a user developed function.

`AggStat` — the aggregating statistic. The default is mean. The function applied must have an na.rm argument (base R stat functions such as mean, max, and sum all have an na.rm argument.).

`TS` — A logical statement with a default of FALSE. If TRUE, instead of a dataframe of monthly statistics and average statistics, a monthly time series is returned.

`Plot` — logical argument with a default of FALSE. If TRUE the monthly statistics are plotted.

`ylab` — A label for the y axis of the plot. The default is "Magnitude"

`main` — A title for the plot. The default is "Monthly Statistics"

`col` — A choice of colour for the bar plot. A single colour or a vector (a colour for each bar).

Details

The statistic of interest for each month is calculated for each calendar year in the data.frame. An aggregated result is also calculated for each month using an aggregating statistic (the mean by default). The data.frame is first truncated at the first occurrence of January 1st and last occurrence of December 31st.

Value

A list with two elements. The first element is a data.frame with year in the first column and months in the next 12 (i.e. each row has the monthly stats for the year). The second element is a dataframe with month in the first column and the associated aggregated statistic in the second. i.e. the aggregated statistic (default is the mean) for each month is provided. However, of TS = TRUE, a monthly time series is returned - as a dataframe with date in the first column and monthly value in the second.

Author(s)

Anthony Hammond

Examples

```
# Get the mean flows for each month for the Thames at Kingston
qm_on_thames <- MonthlyStats(ThamesPQ[, c(1, 3)],
  Stat = mean,
  ylab = "Discharge (m^3/s)", main = "Thames at Kingston monthly mean flow", Plot = TRUE
)

# Get the monthly sums of rainfall for the Thames at Kingston
pm_on_thames <- MonthlyStats(ThamesPQ[, c(1, 2)],
  Stat = sum,
  ylab = "Rainfall (mm)", main = "Thames as Kingston monthly rainfall", Plot = TRUE
)
```

NGRDist *British national grid reference (NGR) distances*

Description

Calculates the Euclidean distance between two British national grid reference points using the Pythagorean/Euclidean method.

Usage

```
NGRDist(i, j)
```

Arguments

i	a numeric vector of length two. The first being the easting and the second being the northing of the first site
j	a numeric vector of length two. The first being the easting and the second being the northing of the second site

Details

Note, that the result is converted to km from m.

Value

A distance in kilometres (if British national grid easting and northing are applied)

Author(s)

Anthony Hammond

Examples

```
# Calculate the distance between the catchment centroid for the
# Kingston upon Thames river gauge and the catchment centroid for the
# gauge at Ardlethen on the River Ythan.
# First retrieve the catchment descriptors (CDs) to obtain eastings and northings
GetCDs(10001)
GetCDs(39001)

# Calculate the distance between two centroids (eastings and northings)
NGRDist(i = c(381355, 839183), j = c(462899, 187850))
```

NonFloodAdj *Non-flood adjustment*

Description

Adjusts the linear coefficient of variation (Lcv) and the linear skewness (LSkew) to account for non-flood years

Usage

```
NonFloodAdj(x)
```

Arguments

x	The annual maximum sample. Numeric vector

Details

The method is the "permeable adjustment method" detailed in chapter 19, volume three of the Flood Estimation Handbook, 1999. The method makes no difference for sites where there are no annual maximums (AM) in the sample that are < median(AM)/2. Once applied the results can be used with the LRatioChange function to update the associated member of a pooling group. There is also the NonFloodAdjPool() function which can be used for multiple sites in a pooling group. The non flood adjustment procedure makes the assumption that annual maxima below QMED/2 are not from the same distribution and will result in a biased estimate. In turn it assumes that the AMAX are from a stationary process. The process adds uncertainty to the usual fitting process for three main reasons. Firstly, the definition of non-flood year (QMED/2). Secondly, the reduced sample size. Thirdly, the calculation process is based, in part, on the proportion of non-flood years to flood years. This proportion has uncertainty as a function of the sample size and the proportion because the standard error of a proportion (p) = sqrt((p * (1 - p)) / n).

Value

A list is returned. The first element of the list is a dataframe with one row and two columns - the adjusted Lcv in the first column and Lskew in the second. The second element of the list is another dataframe with one row and three columns. Number of non-flood years in the first column, sample size in the second and the percent of non-flood year in the third.

Author(s)

Anthony Hammond

Examples

```
# Get an annual maximum sample with a BFIHOST above 0.65 and with some
# annual maxima lower than half the median of the AMAX series, then apply the function
NonFloodAdj(GetAM(44013)[, 2])
```

`NonFloodAdjPool` *Non-flood adjustment for pooling groups*

Description

Applies the NonFloodAdj function to adjust the LCV and LSKEW of one or more sites in a pooling group.

Usage

```
NonFloodAdjPool(x, Index = NULL, AutoP = NULL, ReturnStats = FALSE)
```

Arguments

`x`	A pooling group, derived from the Pool() or PoolSmall() functions.
`Index`	A vector of indices (row numbers) of sites to be adjusted. If Index = NULL (the default) the function is applied to all sites.
`AutoP`	A percentage (numeric) of non flood years. Any sites in the group exceeding this value will be adjusted. This is an automated approach so that the user doesn't need to specify Index. If no sites are above AutoP, the function is applied to all sites.
`ReturnStats`	Logical with a default of FALSE. If set to TRUE, a dataframe of non-flood year stats is returned (see 'Value' section below) instead of the adjusted Pooling group.

Details

For more details of the method for individual sites see the details section of the NonFloodAdj function. As a default this function applies NonFloodAdj to every member of the pooling group. Index can be supplied which is the row name/s of the members you wish to adjust. Or AutoP can be applied and is a percentage. Any member with a greater percentage of non-flood years than AutoP is then adjusted. The non-flood adjustment procedure makes the assumption that annual maxima below QMED/2 are not from the same distribution and will result in a biased estimate. In turn it assumes that the AMAX are from a stationary process. The process adds uncertainty to the usual fitting process for three main reasons. Firstly, the definition of non-flood year (QMED/2). Secondly, the reduced sample size. Thirdly, the calculation process is based, in part, on the proportion of non-flood years to flood years. This proportion has uncertainty as a function of the sample size and the proportion because the standard error of a proportion (p) = sqrt((p * (1 - p)) / n).

Value

By default the pooling group is returned with adjusted LCVs and LSKEWs for all sites indexed (or all sites when Index = NULL), or all sites with percentage of non-flood years above AutoP. No difference will be seen for sites with no AMAX < 0.5QMED. If ReturnStats is set to TRUE, a dataframe with Non-flood year stats is returned. The dataframe has a row for each site in the pooling group and three columns. The first is the number of non-flood years, the second is the number of years, and the third is the associated percentage.

Author(s)

Anthony Hammond

Examples

```
# Set up a pooling group for site 44013, then apply the function
pool_44013 <- Pool(GetCDs(44013))
pool_nf <- NonFloodAdjPool(pool_44013)

# Return the non-flood stats for the pooling group
NonFloodAdjPool(pool_44013, ReturnStats = TRUE)
```

NRFAData — *National River Flow Archive descriptors and calculated statistics for sites suitable for pooling*

Description

A data.frame of catchment descriptors, Lmoments, Lmoment ratios, sample size and median annual maximum flow (QMED). NRFA Peak Flow Dataset - Version 14.

Usage

```
NRFAData
```

Format

A data frame with 543 rows and 27 variables

Details

The functions for pooling group formation and estimation rely on this dataframe. However, the data frame is open for manipulation in case the user wishes to add sites that aren't included, or change parts where local knowledge has improved on the data. Although, usually, in the latter case, such changes will be more appropriately applied to the formed pooling group. If changes are made, they will only remain within the workspace. If a new workspace is opened and the UKFE package is loaded, the data frame will have returned to it's original state.

Source

https://nrfa.ceh.ac.uk/data/peak-flow-dataset

OptimPars *Optimise distribution parameters*

Description

Estimates the parameters of the Generalised extreme value, generalised logistic, Kappa3, or Gumbel distribution from known return period estimates

Usage

```
OptimPars(x, dist = "GenLog")
```

Arguments

x	a data.frame with RPs in the first column and associated estimates in the second column
dist	a choice of distribution for the estimates. The choices are "GenLog", "GEV", "Kappa3", or "Gumbel" - the generalised logistic, generalised extreme value, Kappa3, and Gumbel distribution, respectively. The default is "GenLog"

Details

Given a dataframe with return periods (RPs) in the first column and associated estimates in the second column, this function provides an estimate of the distribution parameters. Ideally the first RP should be 2. Extrapolation outside the RPs used for calibration comes with greater uncertainty.

Value

The parameters of one of four user chosen distributions; Generalised logistic, generalised extreme value, Gumbel, and Kappa3.

Author(s)

Anthony Hammond

Examples

```
# Get some catchment descriptors and some quick results
# Then estimate the GenLog parameters
results <- QuickResults(GetCDs(27051), plot = FALSE)[[1]]
OptimPars(results[, 1:2])
```

Pool *Create pooling group*

Description

Function to develop a pooling group based on catchment descriptors

Usage

```
Pool(
  CDs = NULL,
  AREA,
  SAAR,
  FARL,
  FPEXT,
  N = 500,
  exclude = NULL,
  iug = FALSE,
  UrbMax = 0.03,
  DeUrb = FALSE
)
```

Arguments

CDs	catchment descriptors derived from either GetCDs or CDsXML
AREA	catchment area in km2
SAAR	catchment standard average annual rainfall (1961-1990) in mm
FARL	catchment flood attenuation from reservoirs & lakes
FPEXT	catchment floodplain extent. The proportion of the catchment that is estimated to be inundated by a 100-year flood
N	minimum Number of total gauged record years for the pooling group
exclude	sites to exclude from the pooling group. Either a single site reference or a vector of site references (numeric)
iug	iug stands for 'include urban gauge' - which refers to a gauged subject site with URBEXT2000 >= UrbMax. It's a logical argument with default of FALSE. TRUE will over-ride the default and add the closest site in catchment descriptor space (should be the gauge of interest) to the pooling group if it has URBEXT2000 >= UrbMax
UrbMax	Maximum URBEXT2000 level with a default of 0.03. Any catchment with URBEXT2000 above this level will be excluded from the pooling group
DeUrb	logical argument with a default of FALSE. If true, the Lcv and LSkew of any site in the pooling group with URBEXT2000 > 0.03 will be de-urbanised

Details

A pooling group is created from a CDs object, derived from GetCDs or CDsXML, or specifically with the catchment descriptors (see arguments). To change the default pooling group, one or more sites can be excluded using the 'exclude' option, which requires either a site reference or multiple site references in a vector. If this is done, the site with the next lowest similarity distance measure is added to the group (until the total number of years is at least N). Sites with URBEXT2000 (urban extent) > 0.03 are excluded from the pooling group by default. This threshold can be adjusted with UrbMax. If a gauged assessment is required and URBEXT2000 at the site of interest is > UrbMax, the site should be included by setting iug = TRUE. The Lcv and Lskew (L-moment ratios) for sites in the pooling group with URBEXT2000 > 0.03 can be de-urbanised by setting DeUrb = TRUE. If the user has more data available for a particular site within the pooling group, the Lcv and Lskew for the site can be updated after the group has been finalised. An example of doing so is provided below.

The pooling method is outlined in Science Report: SC050050 - Improving the FEH statistical procedures for flood frequency estimation. The de-urbanisation functionality assumes that the growth curve associated with an annual maximum flow sample is impacted by urbanisation and that this impact can be modelled as a function of the catchment URBEXT2000. The method for pooling the catchments together is based on the similarity of AREA, SAAR, FARL, and FPEXT. These were seen to have the most significant impact on the LCV and LSKEW - and ultimately to provide the lowest 'Pooling Uncertainty Measure' (a statistic for assessing the similarity between pooled and single site gauged estimates).

Value

A data.frame of the pooling group with site reference row names and 24 columns, each providing catchment & gauge details for the sites in the pooling group.

Author(s)

Anthony Hammond

Examples

```
# Get some catchment descriptors
cds_73005 <- GetCDs(73005)

# Set up a pooling group object called pool_73005 excluding sites 79005 & 71011
# Then print the group to the console
pool_73005 <- Pool(cds_73005, exclude = c(79005, 71011))
pool_73005

# Form a pooling group, called pool_group, with the catchment descriptors specifically
pool_group <- Pool(AREA = 1000, SAAR = 800, FARL = 1, FPEXT = 0.01)

# Form a pooling group using an urban catchment which is intended for enhanced
# single site estimation - by including it in the group.
cds_39001 <- GetCDs(39001)
pool_39001 <- Pool(cds_39001, iug = TRUE, DeUrb = TRUE)

# Change the Lcv and LSkew of the top site in the pooling group to 0.19 & 0.18,
```

```
# respectively.
pool_update <- LRatioChange(pool_39001, SiteID = 39001, 0.19, 0.18)
```

PoolEst *Pooled flood estimates*

Description

Provides pooled results from a pooling group - gauged, ungauged and with urban adjustment if necessary.

Usage

```
PoolEst(
  x,
  gauged = FALSE,
  QMED,
  dist = "GenLog",
  RP = c(2, 5, 10, 20, 50, 75, 100, 200, 500, 1000),
  UrbAdj = FALSE,
  CDs = NULL,
  URBEXT = NULL,
  fseQMED = 1.46
)
```

Arguments

`x`	pooling group derived from the Pool function
`gauged`	logical argument with a default of FALSE. TRUE for gauged results and FALSE for ungauged
`QMED`	estimate of the median annual maximum flow
`dist`	a choice of distribution for the estimates. The choices are "GenLog", "GEV", "Kappa3", or "Gumbel"; the generalised logistic, generalised extreme value, Kappa3, and Gumbel distribution, respectively. The default is "GenLog"
`RP`	return period of interest. By default the following RPs are provided: 2, 5, 10, 20, 50, 75, 100, 200, 500, 1000
`UrbAdj`	logical argument with a default of FALSE. When TRUE, an urban adjustment is applied to the pooled Lcv and LSkew
`CDs`	catchment descriptors derived from either GetCDs or CDsXML
`URBEXT`	the catchment URBEXT2000, to be supplied if UrbAdj is TRUE and if CDs have not been
`fseQMED`	factorial standard error of the median annual maximum (QMED) estimate, used for quantifying ungauged uncertainty. Default is 1.46

Details

PoolEst is a function to provide results from a pooling group derived using the Pool function. QMED (median annual maximum flow) needs to be supplied and can be derived from the QMED function for ungauged estimates or the annual maximum sample for gauged estimates. The UrbAdj argument can be set to TRUE to provide urbanised results. If this is done, either URBEXT(urban & suburban extent) or CDs (the catchment descriptors derived from CDsXML or GetCDs) need to be provided. When UrbAdj = TRUE, urban adjustment is applied to the QMED estimate according to the method outlined in the guidance by Wallingford HydroSolutions: 'WINFAP 4 Urban Adjustment Procedures'. The gauged argument can be set to TRUE to form the growth curve with the gauged weighting procedure (often known as enhanced single site). Note that if Gauged = TRUE, the functionality assumes that the top site in the pooling group (i.e. the first row) is the subject "gauged" catchment. It is important to check that this is the case because if the site is urban it may not be included by default. The methods for estimating pooled growth curves are according to Science Report: SC050050 - Improving the FEH statistical procedures for flood frequency estimation. The methods for estimating the L-moments and growth factors are outlined in the Flood Estimation Handbook (1999), volume 3. The methods for quantifying uncertainty are detailed in Hammond, A. (2022). Easy methods for quantifying the uncertainty of FEH pooling analysis. Circulation - The Newsletter of the British Hydrological Society (152). The estimation procedure assumes that the pooled AMAX samples are from the same underlying distribution (aside from the QMED scaling factor), that the distribution is correctly specified, that the individual samples are all independent and identically distributed, and that the samples are independent of each other. The urban adjustment assumes that the growth curve associated with an annual maximum flow sample is impacted by urbanisation and that this impact can be modelled as a function of the catchment URBEXT2000. The confidence intervals are different between the gauged and ungauged pooling. This is because the intervals for the ungauged site are uncomfortably large, and therefore 68

Value

If RP is default then a list of length 4. Element one is a data frame with columns; return period (a range from 2 - 1000), peak flow estimates (Q), growth factor estimates (GF), lower and upper intervals of uncertainty (68 percent intervals for ungauged and 95 percent for gauged). The second element is the estimated Lcv and Lskew. The third provides distribution parameters for the growth curve. The fourth provides distribution parameters for the frequency curve. If RP is not the default only the first two elements are returned.

Author(s)

Anthony Hammond

Examples

```
# Get some catchment descriptors and form a pooling group. It's urban and
# therefore the site of interest is not included
cds_27083 <- GetCDs(27083)
pool_27083 <- Pool(cds_27083)

# Get results for the ungauged case, with urban adjustment
PoolEst(pool_27083, QMED = 12, UrbAdj = TRUE, CDs = cds_27083)
```

```
# Form the group again with the urban gauge included & undertake a gauged estimate
# with urban adjustment. QMED in this example is estimated as the median of the annual
# maximum series for site 27083.
pool_g_27083 <- Pool(cds_27083, iug = TRUE, DeUrb = TRUE)
PoolEst(pool_g_27083, QMED = 12.5, UrbAdj = TRUE, CDs = cds_27083)
```

PoolSmall *Create pooling group for small catchments*

Description

Function to develop a small catchments pooling group based on catchment descriptors

Usage

```
PoolSmall(
  CDs = NULL,
  AREA,
  SAAR,
  N = 500,
  exclude = NULL,
  iug = FALSE,
  UrbMax = 0.03,
  DeUrb = FALSE
)
```

Arguments

CDs	catchment descriptors derived from either GetCDs or CDsXML
AREA	catchment area in km2
SAAR	catchment standard average annual rainfall (1961-1990) in mm
N	minimum Number of total gauged record years for the pooling group
exclude	sites to exclude from the pooling group. Either a single site reference or a vector of site references (numeric)
iug	iug stands for 'include urban gauge' - which refers to a gauged subject site with URBEXT2000 >= UrbMax. It's a logical argument with default of FALSE. TRUE will over-ride the default and add the closest site in catchment descriptor space (should be the gauge of interest) to the pooling group if it has URBEXT2000 >= UrbMax
UrbMax	Maximum URBEXT2000 level with a default of 0.03. Any catchment with URBEXT2000 above this level will be excluded from the pooling group
DeUrb	logical argument with a default of FALSE. If true, the Lcv and LSkew of any site in the pooling group with URBEXT2000 > 0.03 will be de-urbanised

Details

A pooling group is created from a CDs object, derived from GetCDs or CDsXML, or specifically with the necessary catchment descriptors (see arguments). To change the default pooling group one or more sites can be excluded using the 'exclude' option, which requires either a site reference or multiple site references in a vector. If this is done, the site with the next lowest similarity distance measure is added to the group (until the total number of years is at least N). Sites with URBEXT2000 (urban extent) > 0.03 are excluded from the pooling group by default. This threshold can be adjusted with the UrbMax argument. If a gauged assessment is required and URBEXT2000 at the site of interest is > UrbMax, the site should be included by setting iug = TRUE. The Lcv and Lskew (L-moment ratios) for sites in the pooling group with URBEXT2000 > 0.03 can be deurbanised by setting DeUrb = TRUE. If the user has more data available for a particular site within the pooling group, the Lcv and Lskew for the site can be updated after the group has been finalised.

The de-urbanisation functionality assumes that the growth curve associated with an annual maximum flow sample is impacted by urbanisation and that this impact can be modelled as a function of the catchment URBEXT2000. The method for pooling the catchments together is based on the similarity of AREA and SAAR. These were seen to have the most significant impact on the LCV and LSKEW - and ultimately to provide the lowest 'Pooling Uncertainty Measure' (a statistic for assessing the similarity between pooled and single site gauged estimates).

Value

A data.frame of the pooling group with site reference row names and 24 columns, each providing catchment & gauge details for the sites in the pooling group.

Author(s)

Anthony Hammond

Examples

```
# Get some catchment descriptors
cds_21001 <- GetCDs(21001)

# Set up a pooling group object called pool_21001 excluding site 206006.
# Then print the group to the console
pool_21001 <- PoolSmall(cds_21001, exclude = 206006)
pool_21001

# Form a pooling group, called pool_group, with the catchment descriptors specifically
pool_group <- PoolSmall(AREA = 22, SAAR = 1702)
```

POTextract *Peaks over threshold (POT) data extraction*

Description

Extracts independent peaks over a threshold from a sample

Usage

```
POTextract(
  x,
  div = NULL,
  TimeDiv = NULL,
  thresh = 0.975,
  Plot = TRUE,
  ylab = "Magnitude",
  xlab = "Time",
  main = "Peaks over threshold"
)
```

Arguments

`x`	either a numeric vector or dataframe with date (or POSIXct) in the first column and hydrological variable in the second
`div`	numeric percentile (between 0 and thresh), either side of which two peaks over the threshold are considered independent. Default is the mean of the sample.
`TimeDiv`	Number of timesteps to define independence (supplements the div argument). As a default this is NULL and only 'div' defines independence. Currently this is only applicable for data.frames.
`thresh`	user chosen threshold. Default is 0.975
`Plot`	logical argument with a default of TRUE. When TRUE, the full hydrograph with the peaks over the threshold highlighted is plotted
`ylab`	Label for the plot yaxis. Default is "Magnitude"
`xlab`	Label (character) for the plot x axis. Default is "Time".
`main`	Title for the plot. Default is "Peaks over threshold"

Details

If the x argument is a numeric vector, the peaks will be extracted with no time information. x can instead be a data.frame with dates in the first column and the numeric vector in the second. In this latter case, the peaks will be time-stamped and a hydrograph, including POT, will be plotted by default. The method of extracting independent peaks assumes that there is a value either side of which events can be considered independent. For example, if two peaks above the chosen threshold are separated by the mean flow, they could be considered independent, but not if flow hasn't returned to the mean at any time between the peaks. Mean flow may not always be appropriate, in which case the 'div' argument can be applied (and is a percentile). The TimeDiv argument can also be applied to ensure the peaks are separated by a number of time-steps either side of the peaks. For extracting POT rainfall a div of zero could be used and TimeDiv can be used for further separation - which would be necessary for sub-daily time-series. In which case, with hourly data for example, TimeDiv could be set to 120 to ensure each peak is separated by five days either side as well as at least one hour with 0 rainfall. When plotted, the blue line is the threshold, and the green line is the independence line (div).

Value

Prints the number of peaks per year and returns a data.frame with columns; Date and peak, with the option of a plot. Or a numeric vector of peaks is returned if only a numeric vector of the hydrological variable is input.

Author(s)

Anthony Hammond

Examples

```
# Extract POT data from Thames mean daily flow 2000-10-01 to 2015-09-30 with
# div = mean (default) and threshold = 0.95, and display the first six rows
thames_q_pot <- POTextract(ThamesPQ[, c(1, 3)], thresh = 0.95)
head(thames_q_pot)

# Extract Thames POT from only the numeric vector of flows and display the
# first six rows
thames_q_pot <- POTextract(ThamesPQ[, 3], thresh = 0.9)
head(thames_q_pot)

# Extract the Thames POT precipitation with a div of 0, the default
# threshold, and five timesteps (days) either side of the peak, and display the first six rows
thames_p_pot <- POTextract(ThamesPQ[, c(1, 2)], div = 0, TimeDiv = 5)
head(thames_p_pot)
```

POTt — *Peaks over threshold (POT) data extraction (quick)*

Description

Extracts independent peaks over a threshold from a sample, using time as the independence criteria.

Usage

```
POTt(
  x,
  threshold = 0.975,
  div,
  Plot = TRUE,
  PlotType = "l",
  main = "Peaks over threhsold",
  ylab = "Magnitude",
  xlab = "Time"
)
```

Arguments

`x`	either a numeric vector or dataframe with date (or POSIXct) in the first column and hydrological variable in the second
`threshold`	user chosen threshold. Default is 0.975
`div`	number of time steps between peaks to ensure independence.
`Plot`	logical argument with a default of TRUE. When TRUE, the full hydrograph with the peaks over the threshold highlighted is plotted
`PlotType`	Type of plot with a default of "l" for line graph. For rainfall type "h" for bars could be used.
`main`	Title for the plot. Default is "Peaks over threshold"
`ylab`	Label (character) for the plot y axis. Default is "Magnitude"
`xlab`	Label (character) for the plot x axis. Default is "Time".

Details

This provides a quicker option than the POTextract function - useful for very long time series'. It only has the option of time division to ensure independence between peaks. If the x argument is a numeric vector, the peaks will be extracted with no time information. x can instead be a data.frame with dates in the first column and the numeric vector in the second. In this latter case, the peaks will be time-stamped and a hydrograph, including POT, will be plotted by default.

Value

A data.frame with columns; Date and peak, with the option of a plot. Or a numeric vector of peaks is returned if only a numeric vector of the variable is input as x.

Author(s)

Anthony Hammond

Examples

```
# Extract POT data from Thames catchment daily rainfall 2000-10-01 to 2015-09-30 with
# div = 14 (14 days) and threshold = 0.975, and display the first six rows
thames_p_pot <- POTt(ThamesPQ[, c(1, 2)], div = 14, threshold = 0.975)
head(thames_p_pot)

# Extract Thames rainfall POT from the numeric vector of rainfall, with threshold
# set to 0.95 and div set to 14, and display the first six rows
thames_p_pot <- POTt(ThamesPQ[, 2], threshold = 0.95, div = 14)
head(thames_p_pot)
```

QMED *QMED (median annual maximum flow) estimate from catchment descriptors*

Description

Estimated median annual maximum flow from catchment descriptors and donor sites

Usage

```
QMED(
  CDs = NULL,
  Don1 = NULL,
  Don2 = NULL,
  UrbAdj = FALSE,
  uef = FALSE,
  DonUrbAdj = FALSE,
  AREA,
  SAAR,
  FARL,
  BFIHOST,
  URBEXT2000 = NULL,
  Easting = NULL,
  Northing = NULL
)
```

Arguments

`CDs`	catchment descriptors derived from either GetCDs or CDsXML
`Don1`	numeric site reference for a single donor (for donor candidates see DonAdj function)
`Don2`	vector of two site references for two donors (for donor candidates see DonAdj function)
`UrbAdj`	logical argument with a default of FALSE. True applies an urban adjustment
`uef`	logical argument with a default of FALSE. If true an urban expansion factor is applied to the URBEXT2000 value - using the current year.
`DonUrbAdj`	logical argument with a default of FALSE. If TRUE, an urban adjustment is applied to the donor/s QMEDcds estimate.
`AREA`	catchment area in km2
`SAAR`	standard average annual rainfall (mm)
`FARL`	flood attenuation from reservoirs and lakes
`BFIHOST`	baseflow index calculated from the catchment hydrology of soil type classification
`URBEXT2000`	measure of catchment urbanisation
`Easting`	Easting. A six digit Easting (British national grid reference).
`Northing`	Northing. A six digit Northing (British national grid reference).

Details

QMED is estimated from catchment descriptors: QMED = 8.3062*AREA^0.8510 0.1536^(1000/SAAR) FARL^3.4451 0.0460^(BFIHOST^2) as derived in Science Report: SC050050 - Improving the FEH statistical procedures for flood frequency estimation. The single donor method is from the same paper. The method for two donors is outlined in 'Kjeldsen, T. (2019). Adjustment of QMED in ungauged catchments using two donor sites. Circulation - The Newsletter of the British Hydrological Society, 4'. This QMED model is a multiple linear regression with transformed predictor variables and is trained on log transformed observed QMED values. The following assumptions are therefore applied: the relationship between the transformed independent variables (predictors) and the logarithmically transformed dependent variable (QMED) is linear, observations (observed QMEDs used in calibration) are independent of each other, model and sampling errors are independent of each other, model and sampling errors are normally distributed and have a mean of zero, predictor variables are independent, the cross-correlation of model errors can be described by distance between catchment centroids and the form of the associated correlation matrix is known prior to and for the calibration process. When UrbAdj = TRUE, urban adjustment is applied to the QMED estimate according to the method outlined in the guidance by Wallingford HydroSolutions: 'WINFAP 4 Urban Adjustment Procedures'. The use of the urban adjustment factor (UrbAdj) assumes that QMED is impacted by urbanisation and this impact can be determined by the URBEXT2000 catchment descriptor. Urban donors should be avoided, but in the case that an urban donor is considered appropriate the QMEDcd estimate of the donor (or donors) should be urban adjusted by setting the DonUrbAdj argument to TRUE. Use of the uef option applies a nationally averaged urban expansion factor to the URBEXT2000 value, tending to overall underestimated urbanisation in more urban catchments and overestimated urbanisation in more rural catchments. Note that the distance-dependent moderation term (alpha) in the one-donor adjustment is not always appropriate, for example in some situations where the subject site is on the same watercourse as the donor. Similarly the two-donor distance-weighting method can give unsuitable results in some situations, for example where a subject site is in between the two donors on the same watercourse. Finally, for flexibility there is the option to input the relevant catchment descriptors directly rather than using a CDs object.

To derive an appropriate estimate when the donor catchment is urban ensure that DonUrbAdj is TRUE.

Value

An estimate of QMED from catchment descriptors. If two donors are used the associated weights are also returned

Author(s)

Anthony Hammond

Examples

```
# Get some catchment descriptors and calculate QMED as if it was ungauged, with
# no donors, with one donor, and with two donors
cds_55004 <- GetCDs(55004)
QMED(cds_55004)
QMED(cds_55004, Don1 = 55012)
QMED(cds_55004, Don2 = c(55012, 60007))
```

```
# Get CDs for urban gauge and calculate QMED with urban adjustment
cds_27083 <- GetCDs(27083)
QMED(cds_27083, UrbAdj = TRUE)
```

QMEDData — *National River Flow Archive descriptors and calculated statistics for sites suitable for QMED & pooling*

Description

A data.frame of catchment & data descriptors relating to the median annual maximum flow (QMED). NRFA Peak Flow Dataset - Version 14

Usage

```
QMEDData
```

Format

A data frame with 897 rows and 26 variables

Details

The functions for QMED estimation and retrieval of catchment descriptors rely on this dataframe. However, the data frame is open for manipulation in case the user wishes to add sites that aren't included, or change parts where local knowledge has improved on the data. If changes are made, they will only remain within the workspace. If a new workspace is opened and the UKFE package is loaded, the data frame will have returned to it's original state.

Source

https://nrfa.ceh.ac.uk/data/peak-flow-dataset

QMEDDonEq — *QMED donor adjustment*

Description

Applies a donor adjustment to the median annual maximum flow (QMED) estimate

Usage

```
QMEDDonEq(
  AREA,
  SAAR,
  FARL,
  BFIHOST,
  QMEDgObs,
  QMEDgCds,
  xSI,
  ySI,
  xDon,
  yDon,
  alpha = TRUE
)
```

Arguments

AREA	catchment area in km2 for the site of interest
SAAR	standardised average annual rainfall in mm for the site of interest
FARL	flood attenuation from reservoirs and lakes for the site of interest
BFIHOST	the baseflow index as a function of soil type for the site of interest
QMEDgObs	the observed QMED at the donor site
QMEDgCds	the QMED equation derived QMED at the donor site
xSI	the catchment centroid easting for the site of interest
ySI	the catchment centroid northing for the site of interest
xDon	the catchment centroid easting for the donor site
yDon	the catchment centroid northing for the donor site
alpha	a logical argument with a default of TRUE. When FALSE the exponent in the donor equation is set to one. Otherwise it is determined by the distance between the donor and the subject site

Details

Although a single donor adjustment can be applied with the DonAdj() function and the QMED() function, this additional function is provided for flexibility. The method is that of Science Report: SC050050 - Improving the FEH statistical procedures for flood frequency estimation (2008). The x and y grid reference inputs assume that the grid references for subject and donor site are using the same grid referencing system.

Author(s)

Anthony Hammond

Examples

```
# Get observed QMED for site 15006
q_ob <- median(GetAM(15006)[, 2])

# Get QMED equation estimated QMED for the donor site
q_cd <- QMED(CDs = GetCDs(15006))

# Display CDs for site 27051 & note the easting and northing
GetCDs(27051)

# Display CDs for site 15006 & note the easting and northing
GetCDs(15006)

# Apply the QMEDDonEq function with the information gained
QMEDDonEq(
  194, 1096, 0.955, 0.297, q_ob, q_cd,
  xSI = 289289, ySI = 947523,
  xDon = 280908, yDon = 953653
)
```

QMEDfseSS *QMED factorial standard error for gauged sites*

Description

Estimates the median annual maximum flow (QMED) factorial standard error (FSE) by bootstrapping the sample

Usage

```
QMEDfseSS(x)
```

Arguments

x	a numeric vector. The sample of interest

Details

The bootstrapping procedure resamples from the sample N*500 times with replacement. After splitting into 500 samples of size N, the median is calculated for each. Then the exponent of the standard deviation of the log transformed residuals is taken as the FSE. i.e. exp(sd(log(x)-mean(log(x)))), where x is the bootstrapped medians.

Value

The factorial standard error for the median of a sample.

Author(s)

Anthony Hammond

Examples

```
# Extract an AMAX sample and estimate the QMED factorial standard error
am_203018 <- GetAM(203018)
QMEDfseSS(am_203018$Flow)
```

QMEDLink *QMED Linking equation*

Description

Estimates the median annual maximum flow (QMED) from non-flood flows

Usage

```
QMEDLink(Q5dmf, Q10dmf, DPSBAR, BFI)
```

Arguments

Q5dmf	numeric. The daily mean flow that is exceeded 5 percent of the time
Q10dmf	numeric. The daily mean flow that is exceeded 10 percent of the time
DPSBAR	a catchment descriptor. The average drainage path slope of the catchment
BFI	the baseflow index of the gauged flow

Details

The QMED Linking equation estimates QMED as a function of the flow that is exceeded five percent of the time, the flow that is exceeded 10 percent of the time, the baseflow index, and the catchment descriptor drainage path slope (DPSBAR). All of these can be found for sites on the National River Flow Archive (NRFA) website. The method is provided in the guidance note 'WINFAP 4 QMED Linking equation' (2016) by Wallingford HydroSolutions.

Author(s)

Anthony Hammond

Examples

```
# Calculate the QMED for site 1001 (Wick at Tarroul)
QMEDLink(10.14, 7.352, 29.90, 0.39)
```

QMEDPOT *Empirical estimate of QMED from peaks over threshold (POT) data*

Description

Estimates the median annual maximum flow (QMED) from peaks over threshold data

Usage

```
QMEDPOT(x, ppy)
```

Arguments

x	numerical vector. POT data
ppy	number of peaks per year in the POT data

Details

If there are multiple peaks per year, the peaks per year (ppy) argument is used to convert to the annual scale to derive QMED. If ppy is one, then the median of the POT sample is returned (the median of x).

Author(s)

Anthony Hammond

Examples

```
# Extract some POT data and estimate QMED
thames_pot <- POTextract(ThamesPQ[, c(1, 3)], thresh = 0.90)
QMEDPOT(thames_pot$peak, ppy = 1.867263)
```

QuickResults *Quick pooled results*

Description

Provides pooled gauged, ungauged, or fake ungauged results, directly from the catchment descriptors

Usage

```
QuickResults(
  CDs,
  gauged = FALSE,
  dons = 2,
  Qmed = NULL,
  FUngauged = FALSE,
  plot = TRUE,
  dist = "GenLog"
)
```

Arguments

`CDs`	catchment descriptors derived from either GetCDs or CDsXML
`gauged`	logical argument with a default of FALSE. TRUE for gauged results and FALSE for ungauged
`dons`	number of donors required with a choice of 0, 1, or 2
`Qmed`	user supplied QMED which overrides the default QMED estimate
`FUngauged`	logical argument with a default of FALSE. TRUE provides a fake ungauged estimate (an ungauged estimate whilst excluding the gauged site (the site with the most similar CDs))
`plot`	logical argument with a default of TRUE. TRUE provides an extreme value plot. FALSE prevents the plot
`dist`	a choice of distribution for the estimates. The choices are "GenLog", "GEV", "Kappa3", or "Gumbel; the generalised logistic, generalised extreme value, Kappa3,and Gumbel distributions, respectively. The default is "GenLog"

Details

The quick results function provides results with a default pooling group. If gauged = FALSE the median annual maximum flood (QMED) is estimated from catchment descriptors using the QMED equation adjusted with the 2 closest non-urban gauged sites as donors (though 0 or 1 donors can instead be used if specified) and the growth curve is formed from the ungauged pooling group. If the ungauged site is urban (URBEXT > 0.03), an urban adjustment is made to the QMED and to the pooled growth curve. If gauged = TRUE QMED is the median of the gauged annual maxima and the growth curve is formed with the gauged weighting procedure (often known as enhanced single site). Note that if Gauged = TRUE, the functionality assumes that the top site in the pooling group (i.e. the first row) is the subject "gauged" catchment. If the gauged catchment is urban (URBEXT > 0.03), it's included in the pooling group and deurbanised before an urban adjustment is made to the final growth curve. However, the urban expansion (UEF) is not applied. Note that gauged = TRUE should only be applied where the site is suitable for pooling. If FUngauged = TRUE, the top site in the pooling group is excluded and the QMED and growth curve estimates are performed henceforth in the manner of gauged = FALSE. Note that FUngauged = TRUE should only be applied where the site is suitable for pooling and in conjunction with the argument gauged = FALSE.

This function applies the index flood procedure in a quick way and therefore encompasses all the associated assumptions. Note that it is recommended that the default results should only be used for broad-scale and/or initial assessments.

Value

A list of length two. Element one is a data frame with columns; return period (RP), peak flow estimates (Q) and growth factor estimates (GF). Two additional columns quantify the uncertainty. The second element is the estimated Lcv and Lskew (linear coefficient of variation and skewness). By default an extreme value plot is also returned

Author(s)

Anthony Hammond

Examples

```
# Get some catchment descriptors
cds_73005 <- GetCDs(73005)

# Get default ungauged results
QuickResults(cds_73005)

# Get gauged results with a GEV distribution
QuickResults(cds_73005, gauged = TRUE, dist = "GEV")

# Get fake ungauged results with one donor
QuickResults(cds_73005, FUngauged = TRUE, dons = 1)
```

Rating *Stage-Discharge equation optimisation*

Description

Optimises a power law rating equation from observed discharge and stage

Usage

```
Rating(x, a = NULL)
```

Arguments

x	a data.frame with discharge in the first column and stage in the second
a	a user defined stage correction

Details

The power law rating equation optimised here has the form q = c(h+a)^n; where 'q' is flow, 'h' is the stage, c' and 'n' are constants, and 'a' is the stage when flow is zero. The optimisation uses all the data provided in the dataframe (x). If separate rating limbs are necessary, x can be subset per limb. i.e. the rating function would be used multiple times, once for each subset of x. There is the option, with the 'a' argument, to hold the stage correction parameter (a), at a user defined level. If 'a' is NULL it will be calibrated with 'c' & 'n' as part of the optimisation procedure. Note that this is a purely statistical procedure and hydraulic considerations may prove useful for improving results (particularly where extrapolation is required).

Value

A list with three elements. The first is a vector of the three calibrated rating parameters. The second is the rating equation; discharge as a function of stage. The third is the rating equation; stage as a function of discharge. A rating plot is also returned.

Author(s)

Anthony Hammond

Examples

```
# Create some dummy data
flow <- c(177.685, 240.898, 221.954, 205.55, 383.051, 154.061, 216.582)
stage <- c(1.855, 2.109, 2.037, 1.972, 2.574, 1.748, 2.016)
observations <- data.frame(flow, stage)

# Apply the rating function
Rating(observations)

# Apply the rating function with the stage correction at zero
Rating(observations, a = 0)
```

ReFH *Revitalised Flood Hydrograph Model (ReFH)*

Description

Provides outputs of the ReFH model from catchment descriptors or user defined inputs

Usage

```
ReFH(
  CDs = NULL,
  Depth = NULL,
  duration = NULL,
  timestep = NULL,
  scaled = NULL,
  PlotTitle = NULL,
  RPa = NULL,
  alpha = TRUE,
  season = NULL,
  AREA = NULL,
```

```
  TP = NULL,
  BR = NULL,
  BL = NULL,
  Cmax = NULL,
  Cini = NULL,
  BFini = NULL,
  Rain = NULL
)
```

Arguments

`CDs`	catchment descriptors derived from either GetCDs or ImportCD
`Depth`	a numeric value. The depth of rainfall used as input in the estimation of a design hydrograph. The default, when Depth = NULL, is a two year rainfall.
`duration`	a numeric value. A duration (hrs) for the design rainfall
`timestep`	a numeric value. A user defined data interval. The default changes depending on the estimated time to peak to formulate a sensible looking result
`scaled`	a numeric value of peak flow in m3/s
`PlotTitle`	a character string. A user defined title for the ReFH plot
`RPa`	return period for alpha adjustment. This is only for the purposes of the alpha adjustment, it doesn't change the rainfall input
`alpha`	a logical argument with default TRUE. If TRUE the alpha adjustment is applied based on RPa. If FALSE, no alpha adjustment is made
`season`	a choice of "summer" or "winter". The default is "summer" in urban catchments (URBEXT2000 > 0.03) and "winter" in rural catchments
`AREA`	numeric. Catchment area in km2.
`TP`	numeric. Time to peak parameter (hours)
`BR`	numeric. Baseflow recharge parameter
`BL`	numeric. Baseflow lag parameter (hours)
`Cmax`	numeric. Maximum soil moisture capacity parameter (mm)
`Cini`	numeric. Initial soil moisture content (mm)
`BFini`	numeric. Initial baseflow (m3/s)
`Rain`	numeric. User input rainfall (hourly). A numeric vector

Details

The ReFH is described in the Flood Estimation Handbook Supplementary Report No.1 (2007). The method to derive design rainfall profiles is described in the Flood Estimation Handbook (1999), volume 2. Users can also input their own rainfall with the 'Rain' argument. As a default, when catchment descriptors (CDs) are provided the ReFH function uses catchment descriptors to estimate the parameters of the ReFH model and the two year rainfall for the critical duration. The latter is based on a quadratic interpolation of the catchment descriptors RMED1H, RMED1D, and RMED2D (then a seasonal correction factor is applied). Parameters and initial conditions can also be individually input by the user. If a parameter argument is used for one or more of the parameters, then these overwrite the CD derived parameters. If a value for the scaled argument is provided (m3/s), a scaled hydrograph is returned. The RPa argument doesn't change the rainfall input and is only needed for the alpha adjustment (see the FEH supplement report no.1). The scaling approach multiplies all the ordinates of the hydrograph by the peak flow as opposed to extending the rise of the hydrograph. i.e. there is an implicit assumption that the largest peak flow events are from initial flow that is also larger than for lower peak flow events. This ReFH model is not recommended for deterministic design flow estimation. Instead it can be used to analyse the plausible response to an input of rainfall.

Value

A list with two elements, and a plot. First element of the list is a data.frame of parameters, initial conditions and the catchment area. The second is a data.frame with columns Rain, NetRain, Runoff, Baseflow, and TotalFlow. If the scale argument is used a numeric vector containing the scaled hydrograph is returned instead of the results dataframe. The plot is of the ReFH output, with rainfall, net-rainfall, baseflow, runoff and total flow. If the scaled argument is used, a scaled hydrograph is plotted.

Author(s)

Anthony Hammond

Examples

```
# Get CDs and apply the ReFH function
cds_203018 <- GetCDs(203018)
ReFH(cds_203018)

# Apply the ReFH function, scale to a 100-year flow estimate and change the plot title accordingly
ReFH(cds_203018, scaled = 182, PlotTitle = "100-Year Design Hydrograph - Site 203018")

# Apply the ReFH function with a user defined initial baseflow
ReFH(cds_203018, BFini = 6)
```

SCF *Seasonal correction factor (SCF)*

Description

The results of applying the ratio of the seasonal annual maximum rainfall for a given duration to the annual maximum rainfall for the same duration

Usage

```
SCF(SAAR, duration)
```

Arguments

SAAR	standardised average annual rainfall. Numeric
duration	duration in hours. Numeric

Details

The SCF and its use is detailed in R&D Technical Report FD1913/TR - Revitalisation of the FSR/FEH rainfall runoff method (2005). The ReFH model has a design rainfall profile included for winter and summer but the depth duration frequency (DDF) model is calibrated on annual maximum peaks as opposed to seasonal peaks. The SCF is necessary to convert the DDF estimate to a seasonal one. Similarly, the DDF model is calibrated on point rainfall and the area reduction factor converts it to a catchment rainfall for use with a rainfall runoff model such as ReFH (see details of the ReFH function). The final depth, therefore, is; Depth = DDFdepth x ARF x SCF. Note that the SCF function (as detailed in FEH volume 2) was derived for durations of up to one day.

Value

A data.frame of one row and two columns: SCFSummer and SCFWinter.

Author(s)

Anthony Hammond

Examples

```
# Derive the SCF for a SAAR of 1981 and a duration of 6.5 hours
SCF(1981, 6.5)
```

SimData *Data simulator*

Description

Simulation of a random sample from the generalised extreme value, generalised logistic, Gumbel, Kappa3, or generalised Pareto distributions

Usage

```
SimData(n, pars = NULL, dist = "GenLog", GF = NULL)
```

Arguments

n	sample size to be simulated
pars	vector of parameters in the order of location, scale, shape (only location and shape for Gumbel)
dist	choice of distribution. Either "GEV", "GenLog", "Gumbel", "Kappa3", or "GenPareto"
GF	vector of GF inputs in the order of Lcv, LSkew, QMED (only Lcv and QMED if dist = "Gumbel")

Details

The simulated sample can be generated using the distribution parameters (pars) location, scale and shape, or the growth factor (GF) inputs linear coefficient of variation (Lcv), linear skewness (LSkew) & median annual maximum (QMED). This function applies a probability distribution model which assumes that the sample data is independent and identical, i.e. the assumption is that all observations in the sample would not impact or depend on any other. Furthermore, all observations are from the same underlying process which has not changed over the period of record (stationarity).

Value

A random sample of size n for the chosen distribution.

Author(s)

Anthony Hammond

Examples

```
# Simulate a sample of size 30 from a GenLog distribution with parameters 299, 51, -0.042
SimData(30, pars = c(299, 51, -0.042), dist = "GenLog")

# Now simulate using the Lcv, Lskew, and median (0.17, 0.04, 310)
SimData(30, GF = c(0.17, 0.04, 310), dist = "GenLog")
```

ThamesPQ *Kingston upon Thames daily flow and catchment precipitation 2000-10-01 to 2015-09-30*

Description

A data.frame of four columns; Date, Precipitation (P), & daily mean flow (Q)

Usage

```
ThamesPQ
```

Format

A data frame with 5478 rows and 4 columns:

Date Date

P Precipitation, in mm

Q Daily mean discharge, in m3/s

Source

https://nrfa.ceh.ac.uk/data/station/meanflow/39001

`TrendTest` *Trend hypothesis test*

Description

A hypothesis test for trend

Usage

```
TrendTest(x, method = "mk", alternative = "two.sided")
```

Arguments

`x`	a numeric vector or a data.frame with dates in the first column and chronologically ordered variable in the second.
`method`	a choice of test method. Choices are "mk" (Mann Kendall - the default), "pearson", and "spearman".
`alternative`	the alternative hypothesis. Options are "less", "greater", and "two.sided". The default is "two.sided".

Details

The test can be performed on a numeric vector, or a data.frame with dates in the first column and the associated variable of interest in the second. A choice can be made between Mann Kendall, Pearson, or Spearman tests. The Spearman and Mann Kendall are based on ranks and will therefore have the same results whether dates are included or not. The default is Mann Kendall. The default is to test for any trend (alternative = "two.sided"). For positive trend set alternative to "greater", and to test for negative trend set alternative to "less".

Interpretation: When testing for positive trend (alternative = "greater") the P_value is the probability of exceeding the observed statistic under the null hypothesis (that it is less than zero). The vice versa is true when testing for negative trend (alternative = "less"). For alternative = "two.sided" the P_value is the probability of exceeding the absolute value of the observed statistic under the null hypothesis (that it is different from zero). Low P values indicate that the null hypothesis is less likely.

Value

A data.frame with columns and associated values: P_value, statistic (Kendall's tau, Spearman's rho, or Pearson's correlation coefficient), and a standardised distribution value. The latter is either the z score (for MK test) or students 't' of the observed statistic under the null hypothesis.

Author(s)

Anthony Hammond

Examples

```
# Get an AMAX sample and apply a trend test with the default Mann-Kendall test
am_27083 <- GetAM(27083)
TrendTest(am_27083)

# Apply the test with the Pearson correlation method with dates
# included (full object) and not (flow values only)
TrendTest(am_27083, method = "pearson")
TrendTest(am_27083$Flow, method = "pearson")

# Apply the default Mann-Kendall test for positive trend
TrendTest(am_27083$Flow, alternative = "greater")
```

UAF — *Urban adjustment factor (UAF) and percentage runoff urban adjustment factor (PRUAF)*

Description

UAF and PRUAF from catchment descriptors for QMED estimation in ungauged urban catchments

Usage

```
UAF(CDs = NULL, URBEXT2000, BFIHOST)
```

Arguments

CDs	catchment descriptors derived from either GetCDs or CDsXML
URBEXT2000	quantification of catchment urbanisation (used when CDs is not)
BFIHOST	baseflow index as a function of hydrological soil type of the catchment (used when CDs is not)

Details

The urban adjustment factor is to adjust the rural QMED estimates (as estimated using the QMED function) to urban estimates. This is necessary because the QMED equation is calibrated on rural catchments. The assumption is that the magnitude of QMED is impacted by urbanisation and that this impact can be modelled as a function of the catchment descriptors URBEXT and BFIHOST. This UAF function is based on URBEXT2000 and BFIHOST19.

Value

a data.frame with columns UAF and PRUAF

Author(s)

Anthony Hammond

Examples

```
# Get some catchment descriptors for an urban catchment and calculate the UAF & PRUAF
cds_53006 <- GetCDs(53006)
UAF(cds_53006)

# Calculate UAF and PRUAF using a user input URBEXT2000 and BFIHOST
UAF(URBEXT2000 = 0.1138, BFIHOST = 0.3620)
```

UEF *Urban expansion factor*

Description

This function provides a coefficient to multiply by URBEXT2000 to adjust it to a given year

Usage

```
UEF(Year)
```

Arguments

Year	The year for consideration. Numeric

Details

The urban expansion factor is detailed in Bayliss, A. Black, K. Fava-Verde, A. Kjeldsen, T. (2006). URBEXT2000 - A new FEH catchment descriptor: Calculation, dissemination and application. R&D Technical Report FD1919/TR, DEFRA, CEH Wallingford. The urban expansion model assumes a national average expansion as a function of year. This means that on some catchments the value will be overestimated (primarily on rural ones) and on others the value will be underestimated (primarily on urban ones).

Value

A numeric urban expansion factor.

Author(s)

Anthony Hammond

Examples

```
# Get an expansion factor for the year 2023
UEF(2023)
```

`UKOutline` *UK outline*

Description

Easting and northing national grid reference points around the coast of the UK

Usage

```
UKOutline
```

Format

A data frame with 3867 rows and 2 variables

X_BNG Easting, British national grid reference

Y_BNG Northing, British national grid reference

Source

https://environment.data.gov.uk/

Uncertainty *Uncertainty quantification for gauged and ungauged pooled estimates*

Description

Quantification of aleatoric uncertainty for pooling results for the gauged and ungauged case

Usage

```
Uncertainty(
  x,
  Gauged = FALSE,
  qmed = NULL,
  Dist = "GenLog",
  Conf = 0.95,
  fseQMED = 1.55,
  UrbAdj = FALSE,
  URBEXT = NULL,
  Plot = TRUE,
  IncAMest = TRUE,
  Parametric = TRUE
)
```

Arguments

x	the pooled group derived from the Pool() or PoolSmall() function.
Gauged	a logical argument with a default of FALSE. If FALSE the uncertainty intervals are calculated for the ungauged case. If TRUE they are calculated for the gauged case.
qmed	the QMED estimate for the ungauged case. It is derived from the observed AMAX if Gauged equals TRUE.
Dist	a choice of distribution to use for the estimates. Choices are "GEV", "GenLog", "Gumbel", or "Kappa3". The default is "GenLog".
Conf	the confidence level of the uncertainty intervals. Default is 0.95. Must be between 0 and 1.
fseQMED	the factorial standard error of the QMED estimate for an ungauged assessment. The default is 1.55.
UrbAdj	applies an urban adjustment to the growth curves.
URBEXT	URBEXT value for the site of interest. This is necessary if UrbAdj equals TRUE.
Plot	logical argument with a default of TRUE. If TRUE a return level plot with results and margin of error is plotted. If FALSE, it is not.

IncAMest
: logical argument with a default of TRUE. Sometimes when doing gauged (enhanced single site analysis), the central estimate of the single site estimate is outside the intervals of the ESS estimate. When this argument is true the confidence interval is expanded to include the central estimate for the single site. If FALSE, it is not.

Parametric
: logical argument with a default of TRUE. If TRUE, the bootstrapping is done by simulation with the distribution of choice. If FALSE the bootstrapping is done by resampling with replacement.

Details

Uncertainty for both the gauged (enhanced single site) and ungauged case are quantified according to the bootstrapping procedures, which account for weights in the pooling group, detailed in Hammond, A. (2021). Sampling uncertainty of UK design flood estimation. Hydrology Research. 1357-1371. 52 (6). Note that this function only quantifies sampling (aleatoric) uncertainty. It does not quantify uncertainty associated with models, model choices applied or hydrometric data. Lastly, the method assumes that AMAX samples within the pooling group are independent of each other and serially independent and identically distributed.

Value

A dataframe with 10 rows and four columns. Return period in the first column, central estimate in the second, lower in the third, and upper in the fourth. If Plot = TRUE, a return level plot is also returned.

Author(s)

Anthony Hammond

Examples

```
# Get an ungauged pooling group
pool_203018 <- Pool(GetCDs(203018), exclude = 203018)

# Quantify the central estimate and uncertainty
Uncertainty(pool_203018, qmed = QMED(GetCDs(203018)))

# Get a pooling group with subject site included
pool_203018 <- Pool(GetCDs(203018))

# Quantify the central estimate and uncertainty
Uncertainty(pool_203018, Gauged = TRUE)
```

WeightsGLcv *Site gauged linear coefficient of variation (Lcv) weightings*

Description

Provides the gauged Lcv weights for each site in a pooling group

Usage

```
WeightsGLcv(x)
```

Arguments

x	pooling group derived with the Pool() function

Details

Weighting method as according to Science Report: SC050050 - Improving the FEH statistical procedures for flood frequency estimation

Value

A data.frame with site references in the first column and associated weights in the second

Author(s)

Anthony Hammond

Examples

```
# Get some CDs, form a gauged pooling group, and estimate gauged Lcv
cds_27051 <- GetCDs(27051)
pool_27051 <- Pool(cds_27051)
WeightsGLcv(pool_27051)
```

WeightsGLSkew *Site gauged linear skewness (LSkew) weightings*

Description

Provides the gauged LSkew weights for each site in a pooling group

Usage

```
WeightsGLSkew(x)
```

Arguments

x pooling group derived with the Pool() function

Details

Weighting method as according to Science Report: SC050050 - Improving the FEH statistical procedures for flood frequency estimation

Value

A data.frame with site references in the first column and associated weights in the second

Author(s)

Anthony Hammond

Examples

```
# Get some CDs, form a gauged pooling group, and estimate gauged LSkew
cds_27051 <- GetCDs(27051)
pool_27051 <- Pool(cds_27051)
WeightsGLSkew(pool_27051)
```

WeightsUnLcv *Site ungauged linear coefficient of variation (Lcv) weightings*

Description

Provides the ungauged Lcv weights for each site in a pooling group

Usage

```
WeightsUnLcv(x)
```

Arguments

x pooling group derived with the Pool() function

Details

Weighting method as according to Science Report: SC050050 - Improving the FEH statistical procedures for flood frequency estimation

Value

A data.frame with site references in the first column and associated weights in the second

Author(s)

Anthony Hammond

Examples

```
# Get some CDs, form an ungauged pooling group, and estimate ungauged Lcv
cds_27051 <- GetCDs(27051)
pool_27051 <- Pool(cds_27051, exclude = 27051)
WeightsUnLcv(pool_27051)
```

WeightsUnLSkew *Site ungauged linear skewness (LSkew) weightings*

Description

Provides the ungauged LSkew weights for each site in a pooling group

Usage

```
WeightsUnLSkew(x)
```

Arguments

x	pooling group derived with the Pool() function

Details

Weighting method as according to Science Report: SC050050 - Improving the FEH statistical procedures for flood frequency estimation

Value

A data.frame with site references in the first column and associated weights in the second

Author(s)

Anthony Hammond

Examples

```
# Get some CDs, form an ungauged pooling group, and estimate ungauged LSkew
cds_27051 <- GetCDs(27051)
pool_27051 <- Pool(cds_27051, exclude = 27051)
WeightsUnLSkew(pool_27051)
```

`WGaugLcv` *Gauged pool weighted linear coefficient of variation (Lcv)*

Description

Calculates the gauged weighted Lcv from a pooling group (enhanced single site)

Usage

```
WGaugLcv(x)
```

Arguments

x	pooling group derived with the Pool() function

Details

Weighting method as according to Science Report: SC050050 - Improving the FEH statistical procedures for flood frequency estimation

Value

the gauged weighted Lcv from a pooling group

Author(s)

Anthony Hammond

Examples

```
# Get some CDs, form a gauged pooling group, and estimate gauged Lcv
cds_27051 <- GetCDs(27051)
pool_27051 <- Pool(cds_27051)
WGaugLcv(pool_27051)
```

`WGaugLSkew` *Gauged pool weighted linear skewness (LSkew)*

Description

Calculates the gauged weighted LSkew from a pooling group (enhanced single site)

Usage

```
WGaugLSkew(x)
```

Arguments

x pooling group derived with the Pool() function

Details

Weighting method as according to Science Report: SC050050 - Improving the FEH statistical procedures for flood frequency estimation

Value

the gauged weighted LSkew from a pooling group

Author(s)

Anthony Hammond

Examples

```
# Get some CDs, form a gauged pooling group, and estimate gauged LSkew
cds_27051 <- GetCDs(27051)
pool_27051 <- Pool(cds_27051)
WGaugLSkew(pool_27051)
```

WungLcv *Ungauged pool weighted linear coefficient of variation (Lcv)*

Description

Calculates the ungauged weighted Lcv from a pooling group

Usage

```
WungLcv(x)
```

Arguments

x pooling group derived with the Pool() function

Details

Weighting method as according to Science Report: SC050050 - Improving the FEH statistical procedures for flood frequency estimation

Value

the ungauged weighted Lcv from a pooling group

Author(s)

Anthony Hammond

Examples

```
# Get some CDs, form an ungauged pooling group, and estimate ungauged Lcv
cds_27051 <- GetCDs(27051)
pool_27051 <- Pool(cds_27051, exclude = 27051)
WungLcv(pool_27051)
```

WungLSkew — *Ungauged pool weighted linear skewness (LSkew)*

Description

Calculates the ungauged weighted LSkew from a pooling group

Usage

```
WungLSkew(x)
```

Arguments

x	pooling group derived with the Pool() function

Details

Weighting method as according to Science Report: SC050050 - Improving the FEH statistical procedures for flood frequency estimation

Value

the ungauged weighted LSkew from a pooling group

Author(s)

Anthony Hammond

Examples

```
# Get some CDs, form an ungauged pooling group, and estimate ungauged LSkew
cds_27051 <- GetCDs(27051)
pool_27051 <- Pool(cds_27051, exclude = 27051)
WungLSkew(pool_27051)
```

Zdists — *Zdist Goodness of fit measure for pooling groups*

Description

Calculates the goodness of fit score for pooling groups.

Usage

```
Zdists(x)
```

Arguments

`x`	pooling group derived from the Pool() function

Details

The goodness of fit measure provides a Z-Score which quantifies the number of standard deviations from the mean of a normal distribution. To determine goodness of fit for a given distribution (assume GEV for this example), 500 pooling groups are formed which match the number of sites and samples sizes of the pooling group of interest. These are formed by simulation with the GEV distribution having LCV and LSKEW which are the weighted mean LCV and LSKEW of the pooling group (weighted by sample size) and a median of 1. The weighted mean L-Kurtosis of the observed pooling group (tR4) is compared to the mean and standard deviation (sd) of L-Kurtosis from the simulated pooling groups (tR4_Dist) by calculating the associated Z-score: (tR4 - mean(tR4_Dist)) / sd(tR4_Dist). The fit of the distribution can be considered acceptable if the absolute Z-Score is less than 1.645 (essentially a hypothesis test with alpha level equal to 0.1). This is done for all candidate distributions and the lowest absolute score is considered the best fit.

NOTE: This is slightly different from the zdist function described in the science report 'Improving the FEH statistical procedures for flood frequency estimation, Environment Agency (2008)'. That function assumes a theoretical LKurtosis as a function of the pooled LSKEW to compare with a distribution of LKurtosis from simulated pooling groups. This means that the Gumbel distribution cannot be compared (hence the change which is a recommendation in 'Regional Frequency Analysis' by Hosking & Wallis (1997)), i.e. the Gumbel distribution is now included whereas it previously could not be.

Value

A list with the first element a data.frame of four Z-Scores related to the columns; "GEV", "GenLog", "Gumbel", and "Kappa3". The second element is a character stating which has the best fit.

Author(s)

Anthony Hammond

Examples

```
# Get CDs, form a pooling group, and calculate the Z-dists
cds_203018 <- GetCDs(203018)
pool_203018 <- Pool(cds_203018)
Zdists(pool_203018)
```

Index