

# Package ‘hydrostats’

August 12, 2021

**Type** Package

**Title** Hydrologic Indices for Daily Time Series Data

**Version** 0.2.8

**Date** 2021-08-05

**Author** Nick Bond

**Maintainer** Nick Bond <n.bond@latrobe.edu.au>

**Description** Calculates a suite of hydrologic indices for daily time series data that are widely used in hydrology and stream ecology.

**License** GPL (>= 2)

**URL** <https://github.com/nickbond/hydrostats>

**Suggests** dplyr, plyr

**Imports** stats, utils, graphics

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-08-12 04:30:02 UTC

## R topics documented:

Acheron . . . . .	2
ann.cv . . . . .	3
baseflows . . . . .	3
Colwells . . . . .	5
Cooper . . . . .	7
CTF . . . . .	8
daily.cv . . . . .	9
day.dist . . . . .	10
flood.length.max . . . . .	11
four.digit.year . . . . .	12
get.days . . . . .	13
high.spell.lengths . . . . .	14

high.spells . . . . .	15
hydro.year . . . . .	18
hydrostats . . . . .	19
low.spell.lengths . . . . .	20
low.spells . . . . .	21
monthly.cv . . . . .	24
partial.series . . . . .	24
recode . . . . .	26
seasonality . . . . .	27
ts.format . . . . .	28

<b>Index</b>	<b>29</b>
--------------	-----------

---

Acheron	<i>Acheron River flow data</i>
---------	--------------------------------

---

## Description

Daily discharge time series in megalitres per day (ML/day) for the Acheron River @ Taggerty (Gauge No. 405209), Victoria, Australia, from 1971-2000.

## Usage

```
data(Acheron)
```

## Format

A data frame with 10944 observations (from 1971-2000) on 2 variables.

```
[,'Date'] date (format dd/mm/yy) [,'Q'] discharge (ML/day)
```

## Source

Data provided by the State of Victoria, Department of Environment and Primary Industries, under Creative Commons Licence 3.0.

## Examples

```
data(Acheron)
Acheron<-ts.format(Acheron)
plot(Acheron[, "Date"], Acheron[, "Q"], type="l", xlab="Date", ylab="Discharge (ML/day)")
```

---

ann.cv                      *Calculate interannual coefficient of variation*

---

**Description**

This function takes a daily time series and returns the coefficient of variation of mean annual flow expressed as a percentage.

i.e.  $(sd/mean)*100$

Missing values are ignored.

**Usage**

```
ann.cv(flow.ts)
```

**Arguments**

flow.ts                      Dataframe with date and discharge data in columns named "Date" and "Q" respectively. Date must be in POSIX format (see ts.format). Missing values are ignored.

**Value**

A dataframe with one column (ann.cv).

**Author(s)**

Nick Bond <n.bond@latrobe.edu.au>

**Examples**

```
data(Cooper)
cooper<-ts.format(Cooper)
ann.cv(Cooper)
```

---

baseflows                      *Measures of central tendency and baseflow via the Lynne-Hollick (LH) baseflow filter*

---

**Description**

Calculate measure of central tendency and baseflow indices using the Lynne-Hollick filter

**Usage**

```
baseflows(flow.ts, a, n.reflected = 30, ts = "mean")
```

**Arguments**

<code>flow.ts</code>	Dataframe with date and discharge data in columns named "Date" and "Q" respectively. Date must be in POSIX format (see <code>ts.format</code> ). Missing values are ignored.
<code>a</code>	The alpha value used in the Lynne-Hollick filter for digital baseflow separation. Default value is 0.975
<code>n.reflected</code>	The number of days that are reflected at the start and end of the series to provide a burn in for the digital filter. Default value is 30. (See Ladson et al. 2013).
<code>ts</code>	<code>ts="mean"</code> returns means for the entire time series <code>ts="annual"</code> returns annual averages. Note this function does not currently use hydrologic years even when defined <code>ts="daily"</code> returns complete series

**Details**

Technically the LH filter cannot be calculated where there are missing data. Here the function removes missing values and is applied to a concatenated version of the time series. Missing dates are reinserted after the filter has been applied for the purpose of returning annual or daily series. The function further reports the number of missing values leaving the user to decide on the reliability of the baseflow estimates.

**Value**

A dataframe. See below for details.

`ts="mean"`

<code>n.years</code>	The number of years of record in the series
<code>prop.obs</code>	proportion of non-missing observations
<code>MDF</code>	mean daily flow
<code>Q50</code>	median daily flow
<code>mean.bf</code>	mean baseflow volume
<code>mean.bfi</code>	mean baseflow index

`ts="annual"`

A dataframe containing columns "year", "no.obs", "Q", "bf", "bfi", where `no.obs` is the number of observations, and `bf` and `bfi` are the mean baseflow and baseflow index, respectively, for each year in the record.

`ts="daily"`

The original dataframe with appended columns "bf" and "bfi". See `ts="annual"` for details.

**Author(s)**

Nick Bond <n.bond@latrobe.edu.au>

## References

- Ladson, A. R., R. Brown, B. Neal and R. Nathan (2013) A standard approach to baseflow separation using the Lyne and Hollick filter. *Australian Journal of Water Resources* 17(1): 173-18
- Lynne, V., Hollick, M. (1979) Stochastic time-variable rainfall-runoff modelling. In: pp. 89-93 Institute of Engineers Australia National Conference. Perth.

## Examples

```
data(Acheron)
Acheron<-ts.format(Acheron)

baseflows(Acheron,a=0.975, ts="mean")
baseflows(Acheron,a=0.975, ts="annual")
head(baseflows(Acheron,a=0.975, ts="daily"))
```

---

Colwells

*Colwells indices*

---

## Description

Calculates Colwell's (1974), which provide a measure of the seasonal predictability of environmental phenomena. Defined in terms of Predictability (P), Constancy (C) and Contingency (M). For detailed information on the calculation and description of Colwell's indices refer to (Colwell, 1974).

## Usage

```
Colwells(flow.ts, fn = "mean",
         boundaries = "transform", s = 11,
         base.binning = 2, from = 0.5, by = 0.25, base.entropy=2, indices.only=FALSE)
```

## Arguments

- |            |   |
|------------|---|
| flow.ts    | Dataframe with date and discharge data in columns named "Date" and "Q" respectively. Date must be in POSIX format (see ts.format). Missing values are ignored.  |
| fn         | The function used to summarise daily data (default mean) and for scaling break points when binning data. Can also use median, min, max.   |
| boundaries | The method used to define break points when binning data.<br>boundaries="equal" splits the data into s equal sized bins<br>boundaries="transform" (default) first applies a $\log_{10}(x+1)$ transformation and then splits the data into s equal size bins<br>boundaries="log_class_size" generates breaks based on logarithmic scale (default base 2) with a roughly equal number of bins above and below 1 |

	boundaries="weighted_log_class_size" generates breaks based on logarithmic scale (default base 2) * mean (or other summary statistic) of the variable. A roughly equal number of bins occur above and below the mean (or other summary statistic)
	boundaries="Gan" creates bins that match those of Gan et al. (1991). Requires: from (default 0.25), by (default 0.25) and s (number of bins)
s	The number of classes the flow data is broken into (default 11)
base.binning	The base integer for defining classes when using the "log_class_size" or "weighted_log_class_size" boundaries
from	The lowest break point for defining classes when using the "Gan" boundaries (default 0.25)
by	The bin width when using the "Gan" boundaries (default 0.25)
base.entropy	The base integer used for the entropy calculations (default=2)
indices.only	Logical. If FALSE (default), the function returns a list of length 3, including the breaks, a table of frequencies, and Colwell's indices. If TRUE, the function returns just a dataframe of indices, useful for combining output with that from other functions.

### Details

Predictability measures how tightly an event is linked to a season; Constancy measures how uniformly the event occurs through all seasons, and Contingency measures the repeatability of season patterns. Predictability is the sum of Constancy and Contingency, and reflects the likelihood of being able to predict a flow occurrence. It is maximized when the flow is constant throughout the year (Constancy Maximised), or if the pattern of high or low flow occurrence is repeated across all years (Contingency maximized).

### Value

A list or dataframe (see above).

breaks	shows the break points used between classes. Not returned for all boundary options. The upper and lower classes are always open even if -Inf/Inf are not shown.
flow.table	Table showing the number of times the monthly flows fall into each flow class in each month. Useful for examining the results of different binning techniques.
P	Predictability
C	Constancy
M	Contingency
CP	C/P
MP	M/P

### Author(s)

Nick Bond <n.bond@latrobe.edu.au>

## References

Colwell, R.K. 1974. Predictability, constancy, and contingency of periodic phenomena. *Ecology* 55(5): 1148-53.

Gan, K.C., McMahon, T.A., and Finlayson, B.L. 1991. Analysis of periodicity in streamflow and rainfall data by Colwell's indices. *Journal of Hydrology* 123(1-2): 105-18.

## Examples

```
data(Cooper)
Cooper<-ts.format(Cooper)

Colwells(Cooper, s=5)
Colwells(Cooper, boundaries="equal", s=11)
Colwells(Cooper, boundaries="log_class_size", s=11)
Colwells(Cooper, boundaries="weighted_log_class_size", s=11)

Colwells(Cooper, boundaries="Gan", from=1,by=1, s=4)
Colwells(Cooper, boundaries="Gan", from=0.25,by=0.25, s=9)

Colwells(Cooper, boundaries="Gan", from=0.25,by=0.25, s=9, indices.only=TRUE)

require(plyr)
data(Acheron)
Acheron<-ts.format(Acheron)
flow.ts<-rbind(data.frame(River="Acheron", Acheron), data.frame(River="Cooper", Cooper))

ddply(flow.ts, .(River), function(x)
Colwells(x, boundaries="weighted_log_class_size", s=11, indices.only=TRUE))
```

---

Cooper

*Cooper Creek flow data*

---

## Description

Daily discharge time series in megalitres per day (ML/day) for Coopers Creek @ Currareva (Gauge No. 003101), Qld, Australia, from 1967-1987.

## Usage

```
data(Cooper)
```

## Format

A data frame with 7670 observations (from 1967-1987) on 2 variables.

[,'Date'] date (format dd/mm/yy) [,'Q'] discharge (ML/day)

**Source**

Data provided by the State of Queensland, Department of Natural Resources and Mines, under creative commons licence agreement. Details available at <http://watermonitoring.dnrm.qld.gov.au/wini/copyright.htm>

**Examples**

```
data(Cooper)
Cooper<-ts.format(Cooper)
plot(Cooper[, "Date"],Cooper[, "Q"],type="l", xlab="Date",ylab="Discharge (ML/day)")
```

---

CTF

*Cease-to-flow (CTF) spell statistics*


---

**Description**

Calculates summary statistics describing cease-to-flow spell characteristics.

**Usage**

```
CTF(flow.ts, threshold = 0.1)
```

**Arguments**

flow.ts	Dataframe with date and discharge data in columns named "Date" and "Q" respectively. Date must be in POSIX format (see ts.format).
threshold	values below this threshold (default 0.1) are treated as zero for the purpose of defining cease to flow spells to account for the fact that cease to flow levels are poorly defined for many gauging sites.

**Value**

A dataframe with 5 columns (see below).

p.CTF	Proportion time cease to flows occur
avg.CTF	Average cease-to-flow spell duration
med.CTF	Median cease-to-flow spell duration
min.CTF	Minimum cease-to-flow spell duration
max.CTF	Maximum cease-to-flow spell duration

**Author(s)**

Nick Bond <n.bond@latrobe.edu.au>



**Examples**

```
data(Cooper)
Cooper<-ts.format(Cooper)

CTF(Cooper)
CTF(Cooper, threshold=0)
```

---

`daily.cv`*Calculate daily coefficient of variation*

---

**Description**

This function takes a daily time series and returns the coefficient of variation of daily flows expressed as a percentage.

i.e.  $(sd/mean)*100$

Missing values are ignored.

**Usage**

```
daily.cv(flow.ts)
```

**Arguments**

`flow.ts` Dataframe with date and discharge data in columns named "Date" and "Q" respectively. Date must be in POSIX format (see `ts.format`)

**Value**

A dataframe with one column (`daily.cv`).

**Author(s)**

Nick Bond <n.bond@latrobe.edu.au>

**Examples**

```
data(Cooper)
Cooper<-ts.format(Cooper)

daily.cv(Cooper)
```

---

day.dist	<i>Calculate average timing of events (doy)</i>
----------	---

---

### Description

Calculates the mean and standard deviation of the timing of annual events. Given a dataframe consisting of years in column one and a day of year (0-365 [366 for leap years]) in column two, day.dist returns the mean day of the year (doy) and standard deviation of days around the mean.

Circular statistics are used to account for the proximity of days close to the start and the end of the year (i.e. numbers close to 0 and 365), which would notionally have a mean of approximately 182 (see Bayliss and Jones (1993)). The mean that is returned can be interpreted as a calendar day, and data that are strongly directional will have a standard deviation close to zero.

### Usage

```
day.dist(Dates, days, years)
```

### Arguments

Dates	A vector of POSIX dates from which days and years are extracted. If Dates are not provided, days and years must be
days	A vector of days in numeric format. Not required if POSIXct dates are provided
years	A vector of years in numeric format. Not required if POSIXct dates are provided

### Value

A dataframe with two columns.

mean.doy	mean day of year events occur on
sd.doy	standard deviation indicating the spread of event timing

### Author(s)

Nick Bond <n.bond@latrobe.edu.au>

### References

Bayliss, A. C., Jones, R. C. (1993) Peaks-over-threshold flood database: Summary statistics and seasonality. Institute of Hydrology. Wallingford, UK.

### Examples

```
days<-c(366,1,365,1,366)
years<-c("1968","1975","1983","1990","2004")
day.dist(days=days, years=years)

days<-c(170,180,1,365,170)
```

```
day.dist(days=days, years=years)

dates<-c("1968-06-18", "1975-06-29", "1983-01-01", "1990-12-31", "2004-06-18")
dates<-as.POSIXct(dates, format = "%Y-%m-%d", tz="")
day.dist(Dates=dates)
```

---

flood.length.max	<i>Calculates the maximum flood length in a time series.</i>
------------------	--

---

### Description

Calculates the maximum flood length above a user defined threshold in a time series. Used with `ddply` (from the `plyr` package) it can be used to return a vector of maximum flood lengths for multiple gauges or for multiple years (see examples). Alternatively, the function `high.spell.lengths` can be used to return the length of all events above a threshold.

### Usage

```
flood.length.max(flow.ts, threshold, ind.days = 5)
```

### Arguments

<code>flow.ts</code>	Dataframe with date and discharge data in columns named "Date" and "Q" respectively. Date must be in POSIX format (see <code>ts.format</code> )
<code>threshold</code>	A user supplied threshold for defining spells. This would typically be derived from hydraulic models or similar knowledge pertaining to a gauge site
<code>ind.days</code>	Periods between spells of less than <code>ind.days</code> (default 5) are considered to be 'in spell' for the purpose of further calculations. A value of 0 means spells 1 day apart are considered independent

### Value

A dataframe with one column (`flood.length.max`).

### Author(s)

Nick Bond <n.bond@latrobe.edu.au>

### Examples

```
data(Cooper)
Cooper<-ts.format(Cooper)

flood.length.max(Cooper, threshold = 50000, ind.days = 5)

# Return annual maximum flood length based on calendar year using ddply (from plyr package)
```

```

require(plyr)
Cooper$Year=format(Cooper$Date, format="%Y")
ddply(Cooper, .(Year), flood.length.max, threshold = 50000)

require(dplyr)
Cooper %>%
dplyr::group_by(Year) %>%
dplyr::do(flood.length.max(., threshold = 50000))

# Based on hydrologic year.
Cooper<-hydro.year(Cooper)
plyr::ddply(Cooper, .(Year), flood.length.max, threshold = 50000)

```

---

four.digit.year

---

*Correct conversion of years to four digit format*


---

### Description

Converts from two to four digit representation of years correcting the century for years earlier than that specified. Addresses the fact that under POSIX specifications, values 00 to 68 are prefixed by 20 and 69 to 99 by 19 when converting from two digit years, which can affect longer time series and older data sets.

### Usage

```
four.digit.year(x, year=1968)
```

### Arguments

x	A vector of POSIXct dates, presumably with some years (often those earlier than 1969) assigned to the wrong century.
year	The year (in four digit format) indicating the cutoff for setting the century to 1900's or 2000's.

### Value

A vector of same length as input with years in four digit format.

### Author(s)

Nick Bond <n.bond@latrobe.edu.au>

### See Also

[strptime](#).

**Examples**

```
x <- as.POSIXct(c("01/01/43", "01/01/68", "01/01/69", "01/01/99", "01/01/04"), format="%d/%m/%y")
x
four.digit.year(x, year=1968)
four.digit.year(x, year=1942)
```

---

`get.days`*Function to determine the number of days 365 or 366 in a given year.*

---

**Description**

A helper function for circular statistic functions. Determines the number of days in any given year (i.e. 365 or 366)

**Usage**

```
get.days(year)
```

**Arguments**

year            A vector of years in numeric format

**Value**

A vector containing the number of days in each year in the input vector

**Author(s)**

Nick Bond <n.bond@latrobe.edu.au>

**Examples**

```
years<-c("1968", "1975", "1983", "1990", "2004")
get.days(years)
```

---

high.spell.lengths      *Calculate the length of all high flow spells*

---

### Description

Returns the length (and start date) of all flow spells above (or below) a given percentile or user defined threshold.

Independence criteria allow short periods below the spell threshold to be ignored and flows below a threshold (e.g. zero flows) can be ignored when calculating percentile flows (useful in ephemeral rivers).

### Usage

```
high.spell.lengths(flow.ts, quant = 0.9, threshold,
ind.days = 5, ignore.zeros = T, ctf.threshold = 0.1, inter.flood=FALSE)
```

### Arguments

flow.ts	Dataframe with date and discharge data in columns named "Date" and "Q" respectively. Date must be in POSIX format (see ts.format).
quant	Percentile/quantile to use for defining event magnitude (default 0.9). A value of 0.9 is the upper 90th percentile (i.e. a volume exceeded 10% of the time).
threshold	A user supplied threshold for defining spells. This would typically be derived from hydraulic models or similar knowledge pertaining to a gauge site.
ind.days	Periods between spells of less than ind.days (default 5) are considered to be 'in spell' for the purpose of further calculations. A value of 0 means spells 1 day apart are considered independent.
ignore.zeros	logical. If TRUE, days below a user defined cease-to-flow threshold (default 0.1) will be excluded when estimating the spell threshold for a given percentile. This is primarily of interest in highly ephemeral rivers, where flow may only occur for a small fraction of the time. In such cases, the inclusion of zeros will skew estimates of high flow events downwards, which may be undesirable.
ctf.threshold	values below this threshold are treated as zero for the purpose of percentile based calculations (see ignore zeros).
inter.flood	logical. If TRUE, the function returns the spell lengths and start dates for periods below (rather than above) the defined threshold.

### Value

Returns a dataframe of spell lengths and their associated starting dates.

Note that spells will always end at NAs.

### Author(s)

Nick Bond <n.bond@latrobe.edu.au>

**Examples**

```
data(Cooper)
Cooper<-ts.format(Cooper)

high.spell.lengths(Cooper, threshold=50000)
```

---

high.spells	<i>Calculates high flow spell statistics</i>
-------------	--

---

**Description**

Calculates a suite of statistics describing flood characteristics, such as the timing, frequency and duration of events. The event threshold can be defined as a flow quantile (e.g. upper 90th percentile [default]) or a specific threshold volume (e.g. ML/day).

For the purpose of deriving annual flood statistics, the function can also be applied based on the hydrologic year. This is advisable where the high flow season spans years, such that prolonged spells may span years. Setting the parameter `hydro.year=TRUE` uses the `hydro.year` function to determine the appropriate hydrologic year for each record, which is then used for deriving annual spell characteristics.

It is possible for there to be multiple days with the same annual maximum flow value (although less likely than for low flows). In estimating the average timing (and sd of timing) of minimum flows, the function calculates the average day of year (DOY) of minimum flows in each year first, before calculating the average across years. Circular functions are used to address the proximity between days toward the beginning and end of the year.

Missing values are allowed for convenience (NA's are removed and the time-series is concatenated before functions are applied), but may lead to biased results. For the purpose of the annual statistics years with fewer than 350 days of available record are ignored.

When used with `ddply` to compute outputs for multiple gauges or time periods simultaneously, results, including graphs are produced for each factor level, including graphs. Note the function will return warnings if annual stats are calculated when year is used as a factor.

**Usage**

```
high.spells(flow.ts, quant = 0.9, threshold = NULL,
ind.days = 5, duration = TRUE, volume = TRUE, plot = TRUE, ignore.zeros = FALSE,
ctf.threshold = 0.1, ann.stats = TRUE, ann.stats.only = FALSE, inter.flood = FALSE,
hydro.year=FALSE)
```

**Arguments**

<code>flow.ts</code>	Dataframe with date and discharge data in columns named "Date" and "Q" respectively. Date must be in POSIX format (see <code>ts.format</code> ). If a third column exists then this is assumed to provide a vector of years for the purpose of calculating annual spell statistics based on a predetermined hydrologic year.
----------------------	---

quant	Percentile/quantile to use for defining event magnitude (default 0.9). A value of 0.9 is the upper 90th percentile (i.e. a volume exceeded 10% of the time), and corresponds to Q90.
threshold	A user supplied threshold for defining spells. This would typically be derived from hydraulic models or similar knowledge pertaining to a gauge site.
ind.days	Periods between spells of less than ind.days (default 5) are considered to be 'in spell' for the purpose of further calculations. A value of 0 means spells 1 day apart are considered independent.
duration	logical. Should statistics describing spell duration be returned?
volume	logical. Should statistics describing spell volumes be returned? Note that for days considered 'in-spell', the returned values have the threshold volume subtracted first, and hence reflect the amount of water that was flowing past the gauge above threshold. This is most useful in water planning scenarios.
plot	logical. Should the time-series be plotted? Data points considered 'within spell' are identified using red circles and the threshold is identified with a horizontal line.
ignore.zeros	logical. If TRUE, days below a user defined cease-to-flow threshold (default 0.1) will be excluded when estimating the spell threshold for a given percentile. This is primarily of interest in highly ephemeral rivers, where flow may only occur for a small fraction of the time. In such cases, the inclusion of zeros will skew estimates of high flow events downwards, which may be undesirable.
ctf.threshold	values below this threshold are treated as zero for the purpose of percentile based calculations (see ignore zeros).
ann.stats	logical. If TRUE, the function returns results describing the annual maximum series (i.e. that describing the characteristics of the largest flood event in each year of the time-series). The duration of each annual high.spell is defined as the number of days above the smallest annual maximum for the largest (and longest) high.spell event in each year.
ann.stats.only	logical. If TRUE, statistics describing the annual series only are returned.
inter.flood	logical. If TRUE, statistics describing inter-flood spell characteristics are reported.
hydro.year	logical. If TRUE, each record is first assigned to a hydrologic year based on the timing of minimum flows. See <a href="#">hydro.year</a> for further details.

### Value

A dataframe with the following columns.

flood indices

high.spell.threshold

The high spell threshold applied in the analysis)

n.events

The number of events in the series greater than or equal to the high.spell.threshold

spell.freq

The frequency of spell events (no. per year)

ari

Average Recurrence Interval of events in years (1/spell.freq)



<code>min.high.spell.dur</code>	Minimum duration of spell events
<code>avg.high.spell.dur</code>	Average duration of spell events
<code>med.high.spell.dur</code>	Median duration of spell events
<code>max.high.spell.dur</code>	Maximum duration of spell events
<code>avg.spell.volume</code>	Average spell volume (volumes above the threshold only)
<code>avg.spell.peak</code>	Average spell peak
<code>sd.spell.peak</code>	Standard deviation of spell speaks
<code>avg.rise</code>	Average absolute rate of daily rise during spell events
<code>avg.fall</code>	Average absolute rate of daily fall during spell events
interflood indices	
<code>average.interval</code>	The average time between spells (years)
<code>min.interval</code>	The minimum time between spells (years)
<code>max.interval</code>	The maximum time between spells (years)
Annual flood statistics	
<code>avg.max.ann</code>	The average annual maximum flow
<code>cv.max.ann</code>	The coefficient of variation of annual maximum flows
<code>flood.skewness</code>	The average annual maximum / mean daily flow
<code>ann.max.timing</code>	The average day of the year (0-366) on which maximum flows occur
<code>ann.max.timing.sd</code>	circular standard deviation of the average timing of annual maximum flows
<code>ann.max.min.dur</code>	Minimum duration of the annual maximum spells (always equal to 1)
<code>ann.max.avg.dur</code>	Average duration of the annual maximum spells
<code>ann.max.max.dur</code>	Maximum duration of the annual maximum spells
<code>ann.max.cv.dur</code>	The coefficient of variation of the duration of annual maximum spells

**Author(s)**

Nick Bond <n.bond@latrobe.edu.au>

## Examples

```

data(Cooper)
Cooper<-ts.format(Cooper)

high.spells(Cooper, quant=0.9)

high.spells(Cooper, quant=0.9, ann.stats=FALSE, plot=FALSE)

high.spells(Cooper, quant=0.9, ann.stats=FALSE, ignore.zeros=TRUE)

high.spells(Cooper, quant=0.9, ann.stats=FALSE, ignore.zeros=TRUE, hydro.year=TRUE)

require(plyr)
Cooper$year<-strftime(Cooper$Date, format="%Y")
ddply(Cooper, .(year), function(x) high.spells(x, ann.stats=FALSE))

Cooper$time.period <- ifelse(Cooper$year<1980,"pre_1980","post_1980")
ddply(Cooper, .(time.period), function(x) high.spells(x, ann.stats=FALSE))

```

---

hydro.year	<i>Determine hydrologic year for the purposes of deriving spell characteristics</i>
------------	---

---

## Description

Defines a hydrologic year to minimise the risk that defined spells are interrupted by transitions between calendar years. The function can be called by several other functions in the hydrostats package (e.g. [high.spells](#), [low.spells](#), [high.spell.lengths](#)).

## Usage

```
hydro.year(flow.ts, hydro.year = "hydro", year.only=FALSE)
```

## Arguments

flow.ts	Dataframe with date and discharge data in columns named "Date" and "Q" respectively. Date must be in POSIX format (see <a href="#">ts.format</a> ).
hydro.year	hydro.year="hydro" calculates the hydrologic year and returns a dataframe with an additional column indicating the hydrologic year to which each observation belongs. The hydrologic year is defined as starting in the first month of the average driest 6 month period across all years. This maximises the likelihood that low-flow and high-flow spells will be contained within a rolling 12 month period. Other options may be added in the future.

`year.only` logical. If FALSE (default), a column indicating the hydrologic year of each record is added to the original data.frame. If TRUE, a vector indicating the hydrologic year of each record is returned.

### Value

If `year.only=FALSE` (default), the function returns the original dataframe with an added column "hydro.year" indicating the hydrologic year to which each case belongs. Otherwise, if `year.only=TRUE`, a vector of hydrologic years is returned.

### Author(s)

Nick Bond <n.bond@latrobe.edu.au>

### See Also

[high.spells](#), [low.spells](#).

### Examples

```
data(Cooper)
Cooper<-ts.format(Cooper)

head(hydro.year(Cooper))
head(hydro.year(Cooper, year.only=TRUE))
```

---

hydrostats                      *calculate hydrologic statistics*

---

### Description

Calculate a range of hydrologic statistics based on daily time series data and which are widely used in hydrology and ecological applications.

### Details

Package: hydrostats  
Type: Package  
Version: 0.2.4  
Date: 2015-10-16  
License: GPL (>= 2)

Data must be provided as a dataframe in which the date is in POSIXct format. The function [ts.format](#) can be used to specify the Date and discharge columns (named Date and Q respectively) in a dataframe, and convert dates to POSIXct format. The date and discharge data must be in columns labelled "Date" and "Q" for the functions to work.

Includes several sample datasets.

data(Cooper) - Flow data for Coopers Creek, Australia. Gauge 003101@Currareva

data(Acheron) - Flow data for Acheron River, Australia, Gauge 405209@Taggerty

### Author(s)

Nick Bond Maintainer: Nick Bond <n.bond@latrobe.edu.au>

### Examples

```
data(Acheron)
Acheron<-ts.format(Acheron)
with(Acheron, plot(Q~Date))

high.spell.lengths(Acheron, threshold=50000)
```

---

low.spell.lengths	<i>Calculate the length of all low flow spells</i>
-------------------	--

---

### Description

Returns the length (and start date) of all flow spells above (or below) a given percentile or user defined threshold.

Independence criteria allow short periods below the spell threshold to be ignored and flows below a threshold (e.g. zero flows) can be ignored when calculating percentile flows (useful in ephemeral rivers).

### Usage

```
low.spell.lengths(flow.ts, quant = 0.1, threshold,
ind.days = 5, ignore.zeros = F, ctf.threshold = 0.1, inter.spell=FALSE)
```

### Arguments

flow.ts	Dataframe with date and discharge data in columns named "Date" and "Q" respectively. Date must be in POSIX format (see ts.format).
quant	Percentile/quantile to use for defining event magnitude (default 0.9). A value of 0.9 is the upper 90th percentile (i.e. a volume exceeded 10% of the time).
threshold	A user supplied threshold for defining spells. This would typically be derived from hydraulic models or similar knowledge pertaining to a gauge site.
ind.days	Periods between spells of less than ind.days (default 5) are considered to be 'in spell' for the purpose of further calculations. A value of 0 means spells 1 day apart are considered independent.

<code>ignore.zeros</code>	logical. If TRUE, days below a user defined cease-to-flow threshold (default 0.1) will be excluded when estimating the spell threshold for a given percentile. This is primarily of interest in highly ephemeral rivers, where flow may only occur for a small fraction of the time. In such cases, the inclusion of zeros will skew estimates of low flow events downwards, which may be undesirable.
<code>ctf.threshold</code>	values below this threshold are treated as zero for the purpose of percentile based calculations (see <code>ignore.zeros</code> ).
<code>inter.spell</code>	logical. If TRUE, the function returns the spell lengths and start dates for periods above (rather than below) the defined threshold.

**Value**

Returns a dataframe of spell lengths and their associated starting dates.

Note that spells will always end at NAs.

**Author(s)**

Nick Bond <n.bond@latrobe.edu.au>

**Examples**

```
data(Cooper)
Cooper<-ts.format(Cooper)

low.spell.lengths(Cooper, threshold=50000)
```

---

low.spells

*Calculate low flow spell statistics*

---

**Description**

Calculates a suite of statistics describing low-flow spell characteristics, such as the timing, frequency and duration of events below a threshold. The event threshold can be defined as a flow quantile (e.g. the 10th percentile, which is flows exceeded 90% of the time, corresponding to Q10) or a specific threshold volume (e.g. ML/day).

For the purpose of deriving annual low-flow spell statistics, the function can also be applied based on the hydrologic year. This is advisable where the low flow season spans calendar years, such that prolonged spells may be split at the transition from one calendar year to the next. This first requires the time series be processed using the `hydro.year` function. This adds an additional column indicating the hydrologic year to which each row belongs, which is used for deriving annual spell characteristics.

It is possible for there to be multiple days with the same low flow value (especially zero flows). In estimating the average timing (and sd of timing) of minimum flows, the function calculates the average DOY (day of year) of minimum flows in each year first, before calculating the average across

years. Circular functions are used to address the proximity between days toward the beginning and end of the year.

Missing values are allowed for convenience (NA's are removed and the time-series is concatenated before functions are applied), but of course may lead to biased results. For the purpose of the annual statistics years with fewer than 350 days of available record are ignored.

When used with `ddply` to compute outputs for multiple gauges or time periods simultaneously, results, including graphs are produced for each factor level, including graphs. Note however that the function will return warnings if annual stats are calculated when year is used as a factor.

## Usage

```
low.spells(flow.ts, quant = 0.1, threshold=NULL,
           duration = T, volume = T, plot = T, ann.stats = T, ann.stats.only = F,
           hydro.year=FALSE)
```

## Arguments

<code>flow.ts</code>	Dataframe with date and discharge data in columns named "Date" and "Q" respectively. Date must be in POSIX format (see <code>ts.format</code> ). If a third column exists then this is assumed to provide a vector of years for the purpose of calculating annual spell statistics based on a predetermined hydrologic year.
<code>quant</code>	Percentile/quantile to use for defining event magnitude (default 0.1). A value of 0.1 is the lower 10th percentile (i.e. a volume exceeded 90% of the time).
<code>threshold</code>	A user supplied threshold for defining spells. This would typically be derived from hydraulic models or similar knowledge pertaining to a gauge site.
<code>duration</code>	logical. Should statistics describing spell duration be returned?
<code>volume</code>	logical. Should statistics describing spell volumes be returned?
<code>plot</code>	logical. Should the time-series be plotted? Data points considered 'within spell' are identified using red circles and the threshold is identified with a horizontal line.
<code>ann.stats</code>	logical. If TRUE, the function returns results describing the annual series (i.e. the characteristics of the spells associated with the lowest annual daily flow). The duration of each annual low spell is defined as the number of days below the smallest annual minimum for the lowest (and longest) low flow event in each year.
<code>ann.stats.only</code>	logical. If TRUE, statistics describing the annual series only are returned.
<code>hydro.year</code>	logical. If TRUE, each record is first assigned to a hydrologic year based on the timing of minimum flows. See <a href="#">hydro.year</a> for further details.

## Value

A dataframe with the following columns.

low flow indices

`low.spell.threshold`

The low spell threshold applied in the analysis

```

min.low.spell.duration      Minimum duration of spell events
avg.low.spell.duration      Average duration of spell events
med.low.spell.duration      Median duration of spell events
max.low.duration            Maximum duration of spell events
low.spell.freq              The frequency of spell events (no. per year)

Annual low flow statistics

avg.min.ann                 The average annual minimum flow
cv.min.ann                  The coefficient of variation of annual minimum flows
ann.min.timing              The average day of the year (0-366) on which the minimum flow(s) occur.
ann.min.timing.sd           circular standard deviation of the average timing of annual minimum flows
ann.min.min.dur             Minimum duration of the annual maximum spells (always equal to 1)
ann.min.avg.dur             Average duration of the annual maximum spells
ann.min.max.dur             Maximum duration of the annual maximum spells

```

**Author(s)**

Nick Bond <n.bond@latrobe.edu.au>

**Examples**

```

data(Cooper)
Cooper<-ts.format(Cooper)

low.spells(Cooper, quant=0.1)

low.spells(Cooper, quant=0.1, hydro.year=TRUE)

#generate results for each year
Cooper$year<-strftime(Cooper$Date, format="%Y")

require(plyr)
ddply(Cooper, .(year), function(x) low.spells(x, threshold=20, ann.stats=FALSE))

#generate separate results prior to 1980.
Cooper$time.period<-ifelse(Cooper$year<1980, "pre_1980", "post_1980")
ddply(Cooper, .(time.period), function(x) low.spells(x, threshold=20, ann.stats=FALSE))

```

---

monthly.cv	<i>Calculate monthly coefficient of variation</i>
------------	---

---

**Description**

This function takes a daily time series and returns the coefficient of variation of mean monthly flow expressed as a percentage.

**Usage**

```
monthly.cv(flow.ts)
```

**Arguments**

flow.ts	Dataframe with date and discharge data in columns named "Date" and "Q" respectively. Date must be in POSIX format (see ts.format). Missing values are ignored.
---------	--

**Value**

a dataframe with 1 column (monthly.cv)

**Author(s)**

Nick Bond <n.bond@latrobe.edu.au>

**Examples**

```
data(Cooper)
Cooper<-ts.format(Cooper)
ann.cv(Cooper)
```

---

partial.series	<i>Partial and annual exceedence series.</i>
----------------	--

---

**Description**

Returns a partial or annual exceedence series (ari=1) based on a user defined recurrence interval (ari).

For analyses based on a defined threshold (rather than recurrence interval) use [high.spells](#) instead.

**Usage**

```
partial.series(flow.ts, ari = 2, ind.days = 7, duration = T, plot = F, volume = T,
series = FALSE)
```



**Arguments**

flow.ts	Dataframe with date and discharge data in columns named "Date" and "Q" respectively. Date must be in POSIX format (see ts.format)
ari	The desired average return interval. As a partial series, an ari of 1 will return statistics for the n largest floods in n years of record (also referred to as the annual exceedence series). The annual maximum series can be derived from high.spells with annual.stats=T
ind.days	Spells of less than ind.days (default 7) are considered to be non-independent, and only the larger of the two spells is included in the results. This behaviour differs from high.spells, where periods below the determined spell threshold of less than the independence period are infilled for the purposes of determining spell duration. This behaviour may change in the future
duration	logical. If TRUE (default), statistics describing the duration of events are returned
plot	logical. If TRUE a plot is returned showing the events included in the partial series
volume	logical. If TRUE, statistics are returned describing the volume of spells (above the spell threshold)
series	logical. If TRUE, the partial series is returned. If FALSE (default), only the indices describing the partial series are returned

**Value**

A list or dataframe depending on whether series = TRUE. If TRUE, a list is returned (see below). If FALSE a dataframe is returned with all indices but without the actual partial series (p.series).

**Indices**

p.series	A dataframe containing an ordered partial series showing the timing and peak magnitude of events
.	
n.years	Number of (almost) complete years in the series. Years with fewer than 350 non-missing values are ignored
.	
n.events	Number of events in the partial series
flow.threshold	the peak volume of the smallest event include in the series
avg.duration	the average duration of events in the series
max.duration	the maximum duration of events in the series
med.spell.volume	the median volume (above the threshold) of events in the series

**Author(s)**

Nick Bond <n.bond@latrobe.edu.au>

**Examples**

```
data(Cooper)
Cooper<-ts.format(Cooper)

partial.series(Cooper,ari=2)
partial.series(Cooper, ari=5, plot=TRUE, ind.days=2)
partial.series(Cooper, ari=5, plot=TRUE, ind.days=10)
```

---

recode

*recode*

---

**Description**

Recodes values in a vector based on original and new values provided as two vectors.

**Usage**

```
recode(x, oldvalue, newvalue)
```

**Arguments**

x	A vector with values to be replaced.
oldvalue	A vector of original values to be recoded.
newvalue	A vector of replacement values of the same length as oldvalue.

**Value**

A vector of same length as input.

**Author(s)**

Nick Bond <n.bond@latrobe.edu.au>

**Examples**

```
x<-seq(1:10)
recode(x, c(1,5,10), c(-1,-5,-10))
```

---

seasonality	<i>Calculate seasonality of discharge</i>
-------------	---

---

**Description**

Returns statistics describing seasonal variation in runoff.

**Usage**

```
seasonality(flow.ts, monthly.range = FALSE)
```

**Arguments**

flow.ts	Dataframe with date and discharge data in columns named "Date" and "Q" respectively. Date must be in POSIX format (see ts.format). Missing values are ignored.
monthly.range	logical. If FALSE (default), the function returns the percentage of runoff occurring during the average driest 6 month period (as defined across all years). If true, additional statistics describing cumulative average monthly flows, the range between the runoff in the wettest and driest months, and the average number of months between the wettest and driest periods of runoff.

**Value**

If monthly.range=FALSE (default) the function returns a dataframe with one column with the percentage of annual runoff delivered during the average driest 6 month period.

If monthly.range=TRUE, the function returns a list with the following elements:

seasonality	The percentage of annual runoff delivered during the driest 6 months
monthly.means	Average flow in each month of the year
avg.ann.month.range	Average difference between the monthly minimum and maximum
max.min.time.dif	Average number of months between the highest and lowest monthly runoff

**Author(s)**

Nick Bond <n.bond@latrobe.edu.au>

**Examples**

```
data(Cooper)
Cooper<-ts.format(Cooper)

seasonality(Cooper, monthly.range=TRUE)
```

---

`ts.format`*Format dates as POSIXct*

---

**Description**

Converts dates from class character (format dd/mm/yyyy or other as specified) into class POSIXct and renames columns for use with other functions in the hydrostats package.

**Usage**

```
ts.format(x, format="%d/%m/%Y", cols=c(1,2))
```

**Arguments**

<code>x</code>	Dataframe including date and discharge data. Dates are assumed to be of class character (see format). The columns containing date and discharge data are required (defaults to renaming columns 1 and 2 to Date and Q respectively if no other columns are specified (see cols)).
<code>format</code>	Format of dates in existing data frame.
<code>cols</code>	A vector of column indices for the date and discharge data. Used to rename columns.

**Details**

Default assumes the date is of class character and in the first column, with discharge in the second column of the data frame. These columns can be specified if the defaults are not appropriate. The date and discharge columns are renamed to 'Date' and 'Q' respectively. For more flexibility in formatting dates/times see the lubridate package.

**Value**

A dataframe with the dates formatted as POSIXct and named columns Date and Q.

**Author(s)**

Nick Bond <n.bond@latrobe.edu.au>

**Examples**

```
data(Cooper)
Cooper<-ts.format(Cooper)
```

# Index

Acheron, [2](#)  
ann.cv, [3](#)

baseflows, [3](#)

Colwells, [5](#)  
Cooper, [7](#)  
CTF, [8](#)

daily.cv, [9](#)  
day.dist, [10](#)

flood.length.max, [11](#)  
four.digit.year, [12](#)

get.days, [13](#)

high.spell.lengths, [11](#), [14](#), [18](#)  
high.spells, [15](#), [18](#), [19](#), [24](#)  
hydro.year, [16](#), [18](#), [22](#)  
hydrostats, [19](#)

low.spell.lengths, [20](#)  
low.spells, [18](#), [19](#), [21](#)

monthly.cv, [24](#)

partial.series, [24](#)

recode, [26](#)

seasonality, [27](#)  
strptime, [12](#)

ts.format, [19](#), [28](#)