Package ‘FlowScreen’

May 7, 2026

Title Daily Streamflow Trend and Change Point Screening

Version 2.1

Description Screens daily streamflow time series for temporal trends and change-points. This package has been primarily developed for assessing the quality of daily streamflow time series. It also contains tools for plotting and calculating many different streamflow metrics. The package can be used to produce summary screening plots showing change-points and significant temporal trends for high flow, low flow, and/or baseflow statistics, or it can be used to perform more detailed hydrological time series analyses. The package was designed for screening daily streamflow time series from Water Survey Canada and the United States Geological Survey but will also work with streamflow time series from many other agencies.
Package update to version 2.0 made updates to read.flows function to allow loading of GRDC and ROBIN streamflow record formats.
This package uses the \`changepoint\` package for change point detection.
For more information on change point methods, see the changepoint package at <https://cran.r-project.org/package=changepoint>.

Depends R (>= 3.5.0)

Imports zyp, changepoint, evir, graphics, grDevices, stats, utils, tools

License GPL (>= 2)

LazyData true

RoxygenNote 7.2.2

Encoding UTF-8

Maintainer Jennifer Dierauer <jen.r.brand@gmail.com>

NeedsCompilation no

Author Jennifer Dierauer [aut, cre],
Paul Whitfield [aut]

Repository CRAN

Date/Publication 2025-07-05 18:20:02 UTC

Contents

add.station.metadata . . . 3
axis_doy.internal . . . 4
bf.seas . . . 5
bf.stats . . . 6
bf_boughton . . . 7
bf_eckhardt . . . 8
bf_oneparam . . . 9
cania.sub.ts . . . 9
caniapiscau . . . 11
caniapiscau.res . . . 12
check_completeness . . . 12
create.ts . . . 14
dr.events . . . 15
dr.pds . . . 16
dr.seas . . . 18
drop.years . . . 19
FDC . . . 20
FlowScreen . . . 21
get.station.internal . . . 23
get.titles.internal . . . 23
hyear.internal . . . 24
MAMn . . . 24
metrics.all . . . 25
missingness . . . 28
mqt . . . 30
NA.count.runs . . . 31
NA.runs . . . 32
pk.bf.stats . . . 33
pk.cov . . . 34
pk.max . . . 35
pk.max.doy . . . 36
pks . . . 37
pks.dur . . . 38
Qn . . . 38
read.flows . . . 39
regime . . . 41
remove.station.metadata . . . 42
screen.cpts . . . 43
screen.frames . . . 44
screen.frames.internal . . . 47
screen.metric . . . 48
screen.series . . . 49
screen.summary . . . 50
screen.summary.internal . . . 51
set.plot.titles . . . 52
station.info . . . 53

YMD.internal 55

Index **56**

`add.station.metadata` *Add MetaData to Database*

Description

Adds user-supplied station metadata to package database. Can also be used to update metadata for a station that is already present in the metadata.

Usage

```
add.station.metadata(
  Agency,
  StationID,
  StnName,
  StateProv = NA,
  Country,
  Lat,
  Lon,
  CatchmentArea_km2 = NA,
  RHN = FALSE,
  StationID_Alternate = NA,
  Overwrite = FALSE
)
```

Arguments

`Agency`	string indicating the source of the streamflow data, e.g. USGS, WSC, etc. Cannot be NA, but can be any user-specified string, e.g. "Agency A".
`StationID`	string, cannot be NA.
`StnName`	string, cannot be NA.
`StateProv`	string, State, Province, or Territory. Can be NA.
`Country`	string, can be an abbreviation, e.g. USA, CA, or full name.
`Lat`	numeric indicating latitude (in decimal degrees) for the gauge location. Can be NA.
`Lon`	numeric indicating longitude (in decimal degrees) for the gauge location. Can be NA.
`CatchmentArea_km2`	numeric, the total drainage area in square kilometers.
`RHN`	TRUE or FALSE indication of whether the station is part of a reference hydrologic network, representing a catchment that has minimal human impacts. Default is FALSE.

StationID_Alternate
: Optional alternate station ID, default is NA.

Overwrite
: Indication of whether a record in the metadata should be overwritten if a match is found. Match is based on the Agency AND StationID. Default is FALSE.

Author(s)

Jennifer Dierauer

Examples

```
## Not run:
met_added <- add.station.metadata(
  Agency = "Foo Bar",
  StationID = "01234",
  StnName = "Example Station",
  Country = NA,
  Lat = 40.0,
  Lon = -89.0
)

## End(Not run)
```

---

axis_doy.internal *Create custom axis starting on hydrologic year start month*

Description

Create custom axis starting on hydrologic year start month

Usage

```
axis_doy.internal(hyrstart = 10)
```

Arguments

hyrstart	numeric indicating month for start of the hydrologic year (water year).

Author(s)

Paul Whitfield

bf.seas *Seasonal baseflow percentage*

Description

This function estimates the percentage of baseflow in a given period relative to the total annual baseflow.

Usage

```
bf.seas(TS, seas = c(6:8))
```

Arguments

TS	data.frame of streamflow time series loaded with read.flows.
seas	Integers representing months of the year. Default is c(6:8), i.e. June-August.

Details

This function calls bf_eckhardt to complete the baseflow separation.

Value

Returns a vector containing the calculated percentage for each year in the input time series. The "times" attribute provides the corresponding year for each calculated value.

Author(s)

Jennifer Dierauer

See Also

See bf.stats to calculate additional baseflow metrics.

Examples

```
data(cania.sub.ts)
cania.sub.ts <- set.plot.titles(cania.sub.ts, title.elements = c("StationID", "Country"))
res <- bf.seas(cania.sub.ts)
res2 <- screen.metric(res, "Percent Annual Baseflow in Jun-Aug", title = TRUE)
```

`bf.stats` *Baseflow statistics*

Description

This function estimates the baseflow and calculates the mean, max, and min baseflow and baseflow index for a user defined time period.

Usage

```
bf.stats(TS, by = "hyear")
```

Arguments

`TS`	data.frame of streamflow time series loaded with `read.flows`.
`by`	summary period. Options are "year", "hyear", "month", or "doy". Default is "hyear".

Details

This function calls `bf_eckhardt` to complete the baseflow separation.

Value

Returns a data.frame with the following columns:

- By - Unique values representing the summary periods, e.g. a list of unique years, months, or days of year
- MeanQ - Mean daily streamflow for the summary period, in m3/s
- MeanBF - Mean daily baseflow for the summary period, in m3/s
- MaxBF - Maximum daily baseflow for the summary period, in m3/s
- MinBF - Minimum daily baseflow for the summary period, in m3/s
- BFVol - Baseflow volume for the summary period, in km3
- MeanBFI - Mean daily baseflow index for the summary period, dimensionless
- MaxBFI - Maximum daily baseflow index for the summary period, dimensionless
- MinBFI - Minimum daily baseflow index for the summary period, dimensionless

Author(s)

Jennifer Dierauer

Examples

```
data(cania.sub.ts)

cania.sub.ts <- set.plot.titles(cania.sub.ts, title.elements = c("StationID", "Country"))
res <- bf.stats(cania.sub.ts)
res2 <- screen.metric(res[,2], "m3/s", title = TRUE)
```

`bf_boughton` *Boughton recursive digital filter*

Description

This function estimates baseflow

Usage

```
bf_boughton(discharge, k, C)
```

Arguments

discharge	Nnumeric vector of daily flow data
k	Numeric value of the recession constant (dimensionless).
C	Numeric value of the partitioning factor (dimensionless).

Value

Returns a numeric vector of the estimated baseflow.

Author(s)

Paul H. Whitfield

References

Boughton, WC. 1993. A hydrograph-based model for estimating the water yield of ungauged catchments.In Hydrology and Water Resources Symposium, Institution of Engineers Australia, Newcastle, NSW; 317-324.

Examples

```
data(cania.sub.ts)
res <- bf_boughton(cania.sub.ts$Flow, k=0.9, C=0.1)
plot(cania.sub.ts$Date, cania.sub.ts$Flow, xlab="", ylab="Q (m3/s)", type="l")
points(cania.sub.ts$Date, res, type="l", col="blue")
```

bf_eckhardt *Eckhardt two parameter recursive digital filter*

Description

This function takes vector of discharge data and estimates the baseflow

Usage

```
bf_eckhardt(discharge, a, BFI)
```

Arguments

discharge	vector of daily discharge observations
a	Numeric value.
BFI	Numeric value.

Value

Returns

Author(s)

Paul Whitfield

References

Eckhardt, K. 2012. Technical note: Analytical sensitivity analysis of two parameter recursive digital baseflow separation filter. Hydrology and Earth System Sciences 16: 451-455.

Examples

```
data(cania.sub.ts)
bf <- bf_eckhardt(cania.sub.ts$Flow, 0.97, 0.8)
plot(cania.sub.ts$Date, cania.sub.ts$Flow, type="l")
points(cania.sub.ts$Date, bf, type="l", col="blue")
```

bf_oneparam — *One parameter recursive digital filter*

Description

This function estimates baseflow.

Usage

```
bf_oneparam(discharge, k)
```

Arguments

discharge	Numeric vector of daily flow data
k	Numeric value for the recession constant (dimensionless).

Value

Returns a numeric vector of the estimated baseflow.

Author(s)

Paul H. Whitfield

References

Eckhardt, K. 2005. How to construct recursive digital filters for baseflow separation methods. Journal of Hydrology 352: 168-173.

Examples

```
data(cania.sub.ts)
res <- bf_oneparam(cania.sub.ts$Flow, k=0.9)
plot(cania.sub.ts$Date, cania.sub.ts$Flow, xlab="", ylab="Q (m3/s)", type="l")
points(cania.sub.ts$Date, res, type="l", col="blue")
```

cania.sub.ts — *Subset of the Caniapiscau River Daily Flows*

Description

This data set includes a subset of the mean daily streamflow for the Caniapiscau Rivers. It includes observations from 1970-1995 (hydrologic years). The code used to subset and modify the original data is shown below.

Usage

```
data(caniapiscau)
```

Format

Formatted as a data.frame with the following columns:

- ID - Water Survey Canada Station ID
- Date - Date of observation, formatted as YYYY-mm-dd
- Flow - Mean daily streamflow, measured in m3/s
- Code - Data Quality Code
- Agency - Source Agency (Water Survey Canada)
- FlowUnits - Unit of streamflow
- year - Calendar year
- month - Calendar month
- doy - Calendar day of year
- hyear - Hydrologic year
- hmonth - Hydrologic month
- hdoy - Hydrologic day of year

Source

Environment Canada. 2010. EC Data Explorer V1.2.30.
Water Survey of Canada V1.2.30 https://www.ec.gc.ca/rhc-wsc/

Examples

```
# Code used to subset and modify original Caniapiscau series:
## Not run:
data(caniapiscau)
cania.ts <- create.ts(caniapiscau, hyrstart=3)
cania.sub.ts <- subset(cania.ts, cania.ts$hyear %in% c(1963:1976))

## End(Not run)
# example use of example subset flow series
data(cania.sub.ts)
head(cania.sub.ts)
str(cania.sub.ts)
```

`caniapiscau` *Caniapiscau River Daily Flows*

Description

This data set includes the mean daily streamflow for the Caniapiscau River. The file has been read from the original .csv format using `read.flows`. The Caniapiscau River is located in Nunavik, Quebec, Canada, and flows northward. The headwaters (representing 45 percent of the total flow) were dammed to create the Caniapiscau Reservoir, which started filling in 1981. In 1985, the reservoir was diverted to the west into the La Grande hydroelectric complex. This flow time series is used as an example of a river with a known change point to demonstrate the package's screening capabilities.

Usage

```
data(caniapiscau)
```

Format

Formatted as a data.frame with the following columns:

- ID - Water Survey Canada Station ID
- PARAM - Parameter ID (1 indicates flow)
- Date - Date of observation, formatted as YYYY-mm-dd
- Flow - Mean daily streamflow, measured in m3/s
- Agency - Source Agency (Water Survey Canada)

Source

Environment Canada. 2010. EC Data Explorer V1.2.30.
Water Survey of Canada V1.2.30 https://www.ec.gc.ca/rhc-wsc/

Examples

```
data(caniapiscau)
head(caniapiscau)
str(caniapiscau)
```

caniapiscau.res *Screening results for the Caniapiscau River*

Description

Contains the results from metrics.all for the full Caniapiscau River daily flow series. Data set created as indicated below. This data set is used in the example documentation for the screen.frames, screen.summary, and screen.cpts functions in order to reduce example run times.

Usage

```
data(caniapiscau)
```

Format

Formatted as indicated in the documentation for metrics.all

Source

Original flow series from Environment Canada. 2010. EC Data Explorer V1.2.30.
Water Survey of Canada V1.2.30 https://www.ec.gc.ca/rhc-wsc/

Examples

```
# Code used produce this data set:
## Not run:
data(caniapiscau)
caniapiscau.ts <- create.ts(caniapiscau, hyrstart=3)
caniapiscau.ts <- subset(caniapiscau.ts, caniapiscau.ts$hyear > 1962)
caniapiscau.res <- metrics.all(caniapiscau.ts)

## End(Not run)
# example use of example subset flow series
data(caniapiscau.res)
```

check_completeness *Check Completeness*

Description

Determine if the desired completeness criteria are being met. This considers if the date range of interest is complete, and whether internal gaps are longer than the criteria. There is an option for allowing the data set to be padded with an optional number of years and then tested using the same criteria. Function returns TRUE if the criteria are met and FALSE if not, and a numeric code that indicates the mode of failure or success.

Usage

```
check_completeness(
  data,
  st_yr,
  nd_yr,
  allowed = 3,
  period = 10,
  pad = NULL,
  my = NULL
)
```

Arguments

data	Result from missingness.
st_yr	Starting year of the desired period.
nd_yr	Ending year of the desired period.
allowed	Maximum number of years allowed to be missing in the period.
period	Period of years that cannot exceed an allowed gap.
pad	Optional number of years to pad the data set.
my	Optional maximum number of years allowed to be missing with padding.

Value

A list containing:

- TRUE or FALSE indicating conditions were met or not
- code: 0 "met", 1 "not long enough period", 2 "gaps longer than allowed", and 3 "more gaps in period than allowed", 10 "met with pad", 11 "not long enough period with pad", 12 "gaps longer than allowed with pad", and 13 "more gaps in period than allowed with pad"

Messages indicating the reason for failure

Author(s)

Paul Whitfield

Examples

```
robin_path <- system.file("extdata", "ROBIN_example.csv", package = "FlowScreen")

TS <- read.flows(robin_path)
res <- missingness(TS, title = TRUE)
check_completeness(res, st_yr = 1985, nd_yr = 2014, allowed = 3, period = 10)
```

create.ts *Create a Time Series of daily streamflow observations*

Description

This function creates a daily time series formatted for use with the functions in this package. This function is executed within the `read.flows` function. To use separately, the 'Flows' input data.frame must have include the columns: ID, PARAM, date, Flow, SYM, Agency, and FlowUnits. This function would be used in the case the user has data files containing dates and flows and this function would convert the original data into the the form used by the FlowScreen functions.

Usage

```
create.ts(Flows, hyrstart = 10)
```

Arguments

`Flows`	Data.frame containing daily streamflow time series loaded with the `read.flows` function.
`hyrstart`	define start month of hydrologic year. Defaults to 10 (October).

Value

Returns a data.frame with year, month, doy, and hyear columns appended to the original input data.frame.

Author(s)

Jennifer Dierauer

Examples

```
data(caniapiscau)
# subset flow series for shorter example run time
# first, drop the rows with missing streamflow
caniapiscau <- caniapiscau[!is.na(caniapiscau$Flow),]
caniapiscau.sub <- caniapiscau[300:1800,]
caniapiscau.sub.ts <- create.ts(caniapiscau.sub, hyrstart = 3)
```

`dr.events` *Partial Duration Series and Event Statistics for streamflow droughts*

Description

This function extracts the partial duration series for all streamflow droughts based on a moving window quantile threshold. Also returns summary information (start date, end date, duration, deficit volume) for each drought event.

Usage

```
dr.events(TS, Qdr = 0.2, WinSize = 30, IntEventDur = 10, EventDur = 15)
```

Arguments

`TS`	data.frame of streamflow time series loaded with `read.flows`.
`Qdr`	Numeric value of the drought threshold quantile. Default is 0.2.
`WinSize`	Numeric value specifying the size of the moving window in days. Default is 30.
`IntEventDur`	Numeric value for the minimum inter-event duration in days. Drought events with less than the specified number of days between will be pooled and considered as one event.
`EventDur`	Numeric value for the minimum drought duration in days. Default is 15.

Value

Returns a list with the following elements:

DroughtEvents: A data.frame with the following columns:

- Event - Integer indicating the original event number assigned before minor drought events were removed.
- Start - Date of the start of the drought event.
- End - Date of the end of the drought event
- maxDef - Numeric value of the maximum streamflow deficit.
- Severity - Numeric value indicating the drought severity, calculated as the cumulative daily streamflow deficit in m3/s.
- Duration - Numeric value of the drought duration in days.
- Magnitude - Numeric value indicating the drought magnitude, which is calculated as the mean daily streamflow deficit in m3/s.
- stdtotDef - Numeric value indicating the standardized cumulative streamflow deficit, calculated as the drought severity divided by the mean annual daily streamflow.

DroughtPDS: A data.frame of the original input TS that has been subset to include only the days on which the streamflow was below the drought threshold. The data.frame also has the following columns appended:

- Thresh - Numeric value indicating the streamflow drought threshold, as calculated by mqt
- BelowThresh - Logical indicating whether the observed streamflow was below the streamflow drought threshold.
- Def - Numeric value of the streamflow defict, calculated as the streamflow drought threshold (m3/s) minus the observed streamflow (m3/s).

Author(s)

Jennifer Dierauer

See Also

See dr.seas to calculate metrics for droughts occurring in a user-defined season.

This function calls dr.pds which calls mqt.

Examples

```
data(cania.sub.ts)
res1 <- dr.events(cania.sub.ts)
events <- res1$DroughtEvents

opar <- graphics::par(no.readonly = TRUE)
par(mar=c(5,6,2,2))
plot(events$Start, events$Duration, pch=19, ylab="Drought Duration (days)", xlab="")
graphics::par(opar)
```

---

dr.pds — *Get the partial duration series for streamflow droughts*

Description

This function returns the partial duration series for streamflow droughts based on a moving window quantile threshold.

Usage

```
dr.pds(TS, Qdr = 0.2, WinSize = 30)
```

Arguments

TS	data.frame of streamflow time series loaded with read.flows.
Qdr	Numeric value of the drought threshold quantile. Default is 0.2.
WinSize	Numeric value specifying the size of the moving window in days. Default is 30.

Details

This function defines a daily streamflow threshold and finds the partial duration series of streamflow droughts. Drought events are identified in the daily streamflow time series with the threshold level approach. In this function, the threshold is defined by a moving quantile, where daily threshold values are based on the 80th percentile of the flow duration curve from a 30-day moving window (Beyene et al. 2014). With this method, every day of the year has a different threshold based on the streamflow measured on the day, the 15 days before the day, and the 15 days after the day. The size of the moving window can be modified with the WinSize argument, and the percentile can be modified with the Qdr argument.

Value

Returns the input TS data.frame with "Thresh" and "BelowThresh" columns appended. The Thresh column contains the daily flow threshold, and the BelowThresh column is a binary indicating whether the flow on each day was below the drought threshold.

Author(s)

Jennifer Dierauer

References

Beyene, B.S., Van Loon, A.F., Van Lanen, H.A.J., Torfs, P.J.J.F., 2014. Investigation of variable threshold level approaches for hydrological drought identification. Hydrol. Earth Syst. Sci. Discuss. 11, 12765-12797. http://dx.doi.org/10.5194/hessd-11-12765-2014.

See Also

See `create.ts` to format the input flow series.

See `mqt` to return only the daily moving quantile threshold.

See `dr.events` to pool drought events, remove minor events, and calculate metrics.

See `dr.seas` to calculate metrics for streamflow droughts that start in a specific month or months.

Examples

```
data(cania.sub.ts)
pds <- dr.pds(cania.sub.ts)
pds <- subset(pds, pds$BelowThresh==TRUE)

# plot the flow time series with black and the drought events in red
plot(cania.sub.ts$Date, cania.sub.ts$Flow, ylab="m3/s", xlab="", type="l")
points(pds$Date, pds$Flow, pch=19, cex=0.7, col="red")
```

dr.seas *Find the start, middle, end, and duration of seasonal droughts*

Description

This function returns the day of year for the start, middle, and end of seasonal droughts. It also returns the duration and severity of each drought event. The function allows for seasonal analysis by defining a season argument which lists months during which droughts of interest may start.

Usage

```
dr.seas(
  TS,
  Qdr = 0.2,
  WinSize = 30,
  IntEventDur = 10,
  EventDur = 15,
  Season = c(4:9)
)
```

Arguments

`TS`	data.frame of streamflow time series loaded with `read.flows`.
`Qdr`	Numeric value for drought quantile. Default is 0.2.
`WinSize`	Numeric value for moving window size in days. Default is 30.
`IntEventDur`	Numeric value for the minimum inter-event duration in days. Drought events with less than the specified number of days between will be pooled and considered as one event. Default is 10.
`EventDur`	Numeric value for the minimum drought duration in days. Default is 15.
`Season`	Numeric vector of months during which droughts start. Default is c(4:9) for non-frost season droughts.

Details

This function calls `dr.events` which calls `dr.pds` and `mqt`

Value

Returns a data.frame of drought event metrics; the columns are:

- StartDay - day of year that the drought event started on
- MidDay - day of year for the middle of the drought event, which is defined as the day when the cumulative drought deficit reached 50 total cumulative daily streamflow deficit. Total cumulative streamflow deficit is also referred to as drought severity in this package.
- EndDay - day of year that the drought ended on

- Duration - length of the drought event, in days
- Severity - severity of the drought event, calculated as the total cumulative daily streamflow deficit

The "times" attribute provides the start date to preserve year information and aid in plotting the time series.

Author(s)

Jennifer Dierauer

See Also

See `create.ts` to format the input flow series.
See `dr.events` and `mqt` for details on how drought events are defined.

Examples

```
data(cania.sub.ts)
res <- dr.seas(cania.sub.ts)
res2 <- screen.metric(res[,1], "Day of Year")
```

`drop.years` *Drop hydrologic years*

Description

Removesthose hydrologic years where the fraction of missing data is above the defined threshold.

Usage

```
drop.years(TS, NAthresh = 0.8)
```

Arguments

`TS`	data.frame of streamflow time series loaded with `read.flows`.
`NAthresh`	Numeric value indicating the threshold for missing data points in any one year. Default is 0.80, indicating that years with more than 80 percent missing data will be omitted from the metric calculations. This value should always be set to greater than 0.1, as years with fewer observations than approximately 1 month will cause errors.

Value

Returns TS data.frame with hydrologic years with above the user-defined threshold dropped.

Author(s)

Jennifer Dierauer

Examples

```
data(caniapiscau)
cania.ts <- create.ts(caniapiscau, hyrstart = 4)
cania.ts <- drop.years(cania.ts, NAthresh = 0.75)
```

FDC *Flow Duration Curve*

Description

Produces a flow duration curve plot with optional Gustard type-curves that can be used to estimate catchment permeability.

Usage

```
FDC(
  TS,
  normalize.flow = TRUE,
  ylog = TRUE,
  title = FALSE,
  normal = FALSE,
  gust = TRUE,
  ylimits = NULL
)
```

Arguments

`TS`	A data frame of streamflow time series loaded with `read.flows`.
`normalize.flow`	Boolean to indicate whether or not the streamflow should be normalized by dividing by the mean. Default is TRUE. Gustard's Type Curves can only be included when this is TRUE.
`ylog`	Boolean indicating whether or not to plot the y-axis as a logarithmic scale. Default is TRUE.
`title`	optional plot title. Default is FALSE indicating no plot title is wanted. Set to TRUE to use the the default plot title, which will look for 'plot title' attribute of the data.frame set by `set.plot.titles`. All values other values will be used as a custom plot title.
`normal`	A logical indicating whether to use normal probability x-axis (normal=TRUE) or a linear probability x-axis (default, normal=FALSE).
`gust`	A logical indicating whether to include Gustard's Type Curves (default is TRUE). Type curves can only be plotted when normalize.flow is TRUE.
`ylimits`	A numeric vector of length 2 to set y-axis limits (default is NULL).

Author(s)

Paul Whitfield

References

Gustard, A., Bullock, A., and Dixon, J.M. (1992). Report No. 108: Low flow estimation in the United Kingdom. Oxfordshire, United Kingdom: Institute of Hydrology.

Examples

```
data(caniapiscau)
FDC(caniapiscau, title="Caniapiscau River")
```

`FlowScreen` *Screen Daily Discharge Time Series for Temporal Trends and Change Points*

Description

This package can be used to calculate more than 30 different streamflow metrics and identify temporal trends and changepoints. It is intended for use as a data quality screening tool aimed at identifying streamflow records that may have anthropogenic impacts or data inhomogeneity.

Details

Package:	FlowScreen
Type:	Package
Version:	1.2.6
Date:	2019-04-05
License:	GPL (>= 2)

Daily streamflow time series downloaded with the Environment Canada Data Explorer can be loaded with `read.flows`. The `read.flows` function can also be used to load daily streamflow time series from the USGS. The streamflow regime can be visualized with `regime`. A list of 30 streamflow metrics that describe high flows, low flows, and baseflows can be calculated using `metrics.all`. The temporal occurrence of changepoints for all metrics or for only the high flow, baseflow, or low flow metrics can be analyzed using `screen.cpts`. If the streamflow time series has multiple metrics exhibiting changepoints within the same year (or few years), the time series can be further analyzed using `screen.summary` which creates a summary plot showing the significant temporal trends and changepoints for the high flow, low flow, or baseflow metrics. The `screen.metric` can be used to create a time series plot for one metric at a time. The `screen.metric` function works with individual metrics output from the following functions: `pk.max`, `pk.max.doy`, `Qn`, `pk.bf.stats`, `dr.seas`, `MAMn`, `bf.stats`, `pk.cov`, and `bf.seas`.The `screen.frames` function creates individual plots from the `screen.summary` function. The `screen.frames` function can also be used to create custom summary plots, see the example code in the function documentation.

Author(s)

Jennifer Dierauer, Paul H. Whitfield

Maintainer: Jennifer Dierauer <jen.r.brand@gmail.com>

References

Bard, A., Renard, B., Lang, M. 2011. The AdaptAlp Dataset: Description, guidance, and analyses. In AdaptAlp WP 4 Report, 15. Lyon, France: Cemagraf.

Bard, A., Renard, B., Lang, M., Giuntoli, I., Korck, J., Koboltschnig, G., Janza, M., d'Amico, M., Volken, D. 2015. Trends in the hydrologic regime of Alpine rivers. Journal of Hydrology online.

Svensson, C., Kundzewicz, Z.W., Maurer, T. 2005. Trend detection in river flow series: 2. Flood and low-flow index series. Hydrological Sciences Journal 50:811-824.

Whitfield, P.H. 2012. Why the provenance of data matters: Assessing "Fitness for Purpose" for environmental data. Canadian Water Resources Journal 37:23-36.

Whitfield, P.H. 2013. Is 'Center of Volume' a robust indicator of changes in snowmelt timing? Hydrological Processes 27:2691-2698.

See Also

pot, decluster, cpt.meanvar, zyp.trend.vector, Kendall

Examples

```
## Not run:
# load daily streamflow time series for the Caniapiscau River
data(caniapiscau)

# summary plot of the annual flow regime
caniapiscau.ts <- create.ts(caniapiscau)
regime(caniapiscau.ts)

# calculate high flow, low flow, and baseflow metrics
res <- metrics.all(caniapiscau.ts)

# plot histogram of changepoints for high flow, low flow, and baseflow metrics
screen.cpts(res, type="h")
screen.cpts(res, type="l")
screen.cpts(res, type="b")

# or plot all changepoints together
cpts <- screen.cpts(res)

# create screening plots for high, low, and baseflow metrics
screen.summary(res, type="h")
screen.summary(res, type="l")
screen.summary(res, type="b")

## End(Not run)
```

get.station.internal — *Get station information for hydrometric stations*

Description

Get station information for ROBIN, USGS, or WSC hydrometric stations.

Usage

```
get.station.internal(Agency, StationID)
```

Arguments

`Agency`	Character string of Agency.
`StationID`	Character string of station ID.

Value

Returns a data.frame of station information

Author(s)

Jennifer Dierauer

get.titles.internal — *Returns plot titles and labels based on plot type and language preference*

Description

Returns plot titles and labels based on plot type and language preference

Usage

```
get.titles.internal(type, flow.units, language = "English", Qmax)
```

Arguments

`type`	character indicating the type of summary plot
`flow.units`	Character string indicating the units for streamflow values, one of either 'ft3/s' or 'm3/s', taken from the flow time series data.frame created with `read.flows`
`language`	"English" or "French"
`Qmax`	the flow quantile used to define peaks of threshold, e.g. 0.95

Author(s)

Jennifer Dierauer

`hyear.internal` *Add hydrologic Year, month, and doy columns to a daily time series*

Description

Add hydrologic Year, month, and doy columns to a daily time series

Usage

```
hyear.internal(TS, hyrstart = 10)
```

Arguments

`TS`	data.frame of streamflow time series loaded with `read.flows`.
`hyrstart`	define start month of hydrologic year. Defaults to 10 (October).

Value

Returns a data.frame with hyear, hmonth, and hdoy columns appended to the original input data.frame.

Author(s)

Jennifer Dierauer

`MAMn` *Calculate mean annual minimum n-day flows*

Description

This function calculates the calculates the mean annual minimum n-day flow by calendar year or by hydrologic year. This function can also be used to find the annual minimum series by setting n=1.

Usage

```
MAMn(TS, n = 7, by = "hyear", threshold.missing = 0.5)
```

Arguments

`TS`	data.frame of streamflow time series loaded with `read.flows`.
`n`	Numeric value for the number of days in the n-day flow period. Default is 7.
`by`	Character string indicating whether to use hydrologic years or calendar years. Default is "hyear". Other option is "year".
`threshold.missing`	Numeric value indicating the fraction of data that can be missing in a single year. Years with a missing data above this threshold will have NA values returned. Default is 0.5 (max of 50% missing data allowed).

Value

Returns a numeric vector containing the calculated MAM n-day flow for each year in the input time series. The "times" attribute provides the corresponding year for each calculated value. Note: a partial start year or end year in the time series that exceeds the threshold set by 'threshold.missing' will be automatically truncated from the output.

Author(s)

Jennifer Dierauer

See Also

screen.metric

Examples

```
data(caniapiscau)
cania.ts <- create.ts(caniapiscau, hyrstart = 4)
cania.ts <- drop.years(cania.ts)
cania.ts <- set.plot.titles(cania.ts,
title.elements = c("StationID", "StnName", "StateProv"))

# find the annual minimum series and plot
res <- MAMn(cania.ts, n=1)
res2 <- screen.metric(res, ylabel = "Q (m3/s)", title = TRUE)

# do the same with MAM 7-day flow instead of annual minimum
res <- MAMn(cania.ts, n=7)
res2 <- screen.metric(res, ylabel = "Q (m3/s)", title = TRUE)
```

---

metrics.all *Streamflow metrics*

Description

Calculates 30 different flow metrics, 10 each for high flows, low flows, and baseflow.

Usage

```
metrics.all(
  TS,
  Qmax = 0.95,
  Dur = 5,
  Qdr = 0.2,
  WinSize = 30,
  Season = c(4:9),
  NAthresh = 0.5,
  language = "English"
)
```

Arguments

`TS`	data.frame of streamflow time series loaded with `read.flows`.
`Qmax`	Numeric value for peaks over threshold quantile. Default is 0.95.
`Dur`	Numeric value for minimum number of days between flood peaks. Default is 5.
`Qdr`	Numeric value for drought quantile. Default is 0.2, i.e. the 80th percentile of the flow duration curve.
`WinSize`	Numeric value for moving window size (in days) for the moving window quantile drought threshold. See `mqt`. Default is 30.
`Season`	Numeric vector of months during which droughts start. Default is c(4:9) for non-frost season droughts.
`NAthresh`	Numeric value indicating the threshold for missing data points in any one year. Default is 0.50, indicating that years with more than 50 percent missing data will be omitted from the metric calculations. This value should always be set to greater than 0.1, as years with fewer observations than approximately 1 month will cause errors.
`language`	Character string indicating the language to be used for naming the different plot metrics. These names are used in `screen.summary` to label individual plots. Options are "English" or "French". Default is "English".

Details

This function calculates streamflow metrics and calculates the prewhitened trend using `zyp.trend.vector` and looks for changepoints in mean and variance using `cpt.meanvar` This function is intended for use as a data quality screening tool aimed at identifying streamflow records with anthropogenic impacts and should not be used to complete a temporal trend analysis, as the calculated metrics may not be appropriate for all catchments. See the functions linked in the following section for details on how each metric is calculated.

Value

Returns a list with the following elements:

metricTS: a list containing a vector of each metric calculated. Each vector has a times attribute providing either the year for metrics with one observation per year or a date for metrics that may have more than one observation per year (e.g., Peaks Over Threshold). This list has the following elements:

- Annual Maximum Series - calculated with `pk.max`
- Day of Annual Maximum - calculated with `pk.max.doy`
- Peaks Over Threshold (Qmax) - calculated with `pks`
- Inter-Event Duration - calculated with `pks.dur`
- Q80 - calculated with `Qn`
- Q90 - calculated with `Qn`
- Day of Year 25 percent Annual Volume - calculated with `pk.cov`
- Center of Volume - calculated with `pk.cov`

- Day of Year 75 percent Annual Volume - calculated with pk.cov
- Duration between 25 percent and 75 percent Annual Volume - calculated with pk.cov
- Q10 - calculated with Qn
- Q25 - calculated with Qn
- Drought Start - calculated with dr.seas
- Drought Center - calculated with dr.seas
- Drought End - calculated with dr.seas
- Drought Duration - calculated with dr.seas
- Drought Severity - calculated with dr.seas
- Annual Minimum Flow - calculated with MAMn
- Mean Annual Minimum 7-day Flow - calculated with MAMn
- Mean Annual Minimum 10-day Flow - calculated with MAMn
- Mean Daily Discharge - calculated with bf.stats
- Annual Baseflow Volume - calculated with bf.stats
- Annual Mean Baseflow - calculated with bf.stats
- Annual Maximum Baseflow - calculated with bf.stats
- Annual Minimum Baseflow - calculated with bf.stats
- Mean Annual Baseflow Index - calculated with bf.stats
- Day of Year 25 percent Baseflow Volume - calculated with pk.bf.stats
- Center of Volume Baseflow - calculated with pk.bf.stats
- Day of Year 75 percent Baseflow Volume - calculated with pk.bf.stats
- Duration between 25 percent and 75 percent Baseflow Volume - calculated with pk.bf.stats

tcpRes: this list contains the results of the trend and changepoint analysis for each of the metrics in the metricTS list described above. Each list element is a list containing the following elements:

- MetricID - integer used to identify the metric
- MetricName - Name of the metric.
- Slope - numeric vector containing the intercept and slope of the prewhitened linear trend calculated using the Yue Pilon method. See zyp.trend.vector
- ci1 - upper bound of the trend's 95 percent confidence interval
- ci2 - lower bound of the trend's 95 percent conficence interval
- pval - Kendall's P-value computed for the detrended time series
- cpts - Most probable location of a changepoint, if one is detected.
- means - Mean before and after the changepoint
- NumObs - The number of data points for the metric

inData: A data.frame of the original input daily streamflow time series.

OmitYrs: A data.frame containing the years and the number of observations for any years omitted from the analysis due to insufficient data. If no years were omitted, NA is returned.

Author(s)

Jennifer Dierauer

See Also

See the documentation for individual functions linked in the output description for details on methods.

See `screen.metric` to create individual plots for each metric.

Examples

```
# load subset of daily streamflow time series for the Caniapiscau River
data(cania.sub.ts)

## Not run:
# calculate low flow, high flow, and baseflow metrics
res <- metrics.all(cania.sub.ts)

## End(Not run)
```

`missingness` *missingness test*

Description

Determine the annual amount of missing data and generate an optional missingness plot.

Usage

```
missingness(
  TS,
  title = FALSE,
  plot = TRUE,
  increasing = TRUE,
  cols = c("white", "blue"),
  omar = c(2, 2, 2, 2),
  mar = c(3, 5, 3, 2)
)
```

Arguments

`TS`	A data frame of streamflow time series loaded with `read.flows`.
`title`	optional plot title. Default is FALSE indicating no plot title is wanted. Set to TRUE to use the the default plot title, which will look for 'plot title' attribute of the data.frame set by `set.plot.titles`. All values other values will be used as a custom plot title.
`plot`	Logical, default is TRUE. If FALSE, does not produce a plot.

increasing	Logical, default is TRUE. If FALSE, years are ordered from top to bottom.
cols	Plot colors, default is white and blue. White always corresponds to NA. Only observed color can be changed.
omar	Vector of length 4, outer margins for the plot.
mar	Vector of length 4, margins for the plot.

Details

Determine the Annual Amount of Missing Data and Generate a Missingness Plot

Value

A list containing:

- `years_total`: Number of years from start to ending year.
- `years_with_obs`: Number of years with observations.
- `years_no_missing_obs`: Number of years with no missing days.
- `complete_years`: Individual years with no missing data.
- `partial_years`: Individual years with some observations.
- `longest_common_period_years`: Number of sequential years with complete data.
- `lcperiod_st`: Starting year of the sequence of years with complete data.
- `lcperiod_nd`: Ending year of the sequence of years with complete data.
- `table`: A dataframe with years, and counts and fractions of missing data.
- `missingness plot`: A plot showing the missingness data (if `plot = TRUE`).

Author(s)

Paul Whitfield

Examples

```
robin_path <- system.file("extdata", "ROBIN_example.csv", package = "FlowScreen")

TS <- read.flows(robin_path)
res <- missingness(TS, cols = c("white", "red"), increasing = FALSE)
```

mqt — *Moving quantile threshold*

Description

This function calculates the daily moving window quantile threshold for use in identifying the partial duration series of streamflow droughts.

Usage

```
mqt(TS, Qdr = 0.2, WinSize = 30)
```

Arguments

TS	data.frame of streamflow time series loaded with `read.flows`.
Qdr	Numeric value of the drought threshold quantile. Default is 0.2.
WinSize	Numeric value specifying the size of the moving window in days. Default is 30.

Details

The threshold is defined by a moving quantile, where daily threshold values are based on the 80th percentile of the flow duration curve (i.e. 0.2 quantile) from a 30-day moving window (Beyene et al. 2014). With this method, every day of the year has a different threshold based on the streamflow measured on the day, the 15 days before the day, and the 15 days after the day.The size of the moving window can be modified with the WinSize argument, and the percentile can be modified with the Qdr argument.

Value

Returns a numeric vector containing the streamflow drought threshold in m3/s for each day of the year.

Author(s)

Jennifer Dierauer

References

Beyene, B.S., Van Loon, A.F., Van Lanen, H.A.J., Torfs, P.J.J.F., 2014. Investigation of variable threshold level approaches for hydrological drought identification. Hydrol. Earth Syst. Sci. Discuss. 11, 12765-12797. http://dx.doi.org/10.5194/hessd-11-12765-2014.

See Also

See `create.ts` to format the input flow series.

The following functions use this function: `dr.pds`, `dr.events`, `dr.seas`

Examples

```
data(cania.sub.ts)
res <- mqt(cania.sub.ts)

# subset one year of the flow series
flow.sub <- cania.sub.ts[cania.sub.ts$year == 1972,]

# plot the 1972 observed flows in dark blue and the daily drought threshold in red
plot(flow.sub$doy, flow.sub$Flow, ylab="Q (m3/s)", xlab="Day of Year",
 pch=19, col="darkblue", type="b")
points(res, pch=19, cex=0.7, col="red")
```

NA.count.runs *Sum missing data points from a daily time series*

Description

Counts the number of missing data points by calendar year, hydrologic year, or month

Usage

```
NA.count.runs(input, by = "hyear", hyrstart = 1)
```

Arguments

input	output from NA.runs
by	character string identifying the time period to summarize by. Defaults is hydrologic year ("hyear"), other choices are "year" and "month". The "month" option will return the number of missing data points for each month in the time series.
hyrstart	optional argument, define start month of hydrologic year

Value

Returns a numeric vector of the number of missing observations per summary period. The "times" attribute of the returned vector provides the corresponding year, hyear, or month.

Author(s)

Jennifer Dierauer

See Also

NA.runs

Examples

```
data(caniapiscau)
cania.ts <- create.ts(caniapiscau)
res <- NA.runs(cania.ts)
print(res)
res2 <- NA.count.runs(res)
print(res2)
```

NA.runs *Missing data runs for daily time series.*

Description

This function takes a data.frame from create.ts and returns a data.frame of missing data runs.

Usage

```
NA.runs(TS, quiet = FALSE)
```

Arguments

TS	data.frame of streamflow time series loaded with read.flows.
quiet	A boolean to indicate message printing.

Value

Returns a data.frame with the following columns:

- Start - Date of the start of the missing data period
- End - Date of the end of the missing data period
- Duration - number of days in the missing data period

Author(s)

Jennifer Dierauer

See Also

create.ts to create input, NA.count.runs to sum the the missing data occurrences by year or month.

Examples

```
data(caniapiscau)
cania.sub <- caniapiscau[300:1800,]
cania.ts <- create.ts(cania.sub)
res <- NA.runs(cania.ts)
print(res)
```

pk.bf.stats *Calculate baseflow peak statistics*

Description

This function finds the start, middle, end, and duration of the baseflow peak based on percent of the total annual baseflow volume. A value of 0 is returned for years with no flow. Hydrologic years with fewer than normal observations (outliers) are excluded from the analysis, and for stations with seasonal flow records, additional seasonal subsetting is done to include only days with observations in all years.

Usage

```
pk.bf.stats(TS, bfpct = c(25, 50, 75))
```

Arguments

`TS`	data.frame of streamflow time series loaded with `read.flows`.
`bfpct`	numeric vector of percentages used to define the start, middle, and end of the baseflow peak. Default is c(25, 50, 75)

Details

This function calculates metrics intended to focus on snowmelt-related streamflow occuring in spring and summer. For catchments in cold climates, the baseflow peak can be interpreted as snowmelt-induced. Baseflow is estimated with `bf_eckhardt`. If total annual flow is equal to 0, returns NA for that year.

Value

Returns a data.frame with the following columns:

- Start - day of year defining the start of the baseflow peak
- Mid - day of year defining the middle of the baseflow peak
- End - day of year defining the end of the baseflow peak
- Dur - duration of the baseflow peak, in days

Author(s)

Jennifer Dierauer

Examples

```
data(caniapiscau)
cania.ts <- create.ts(caniapiscau, hyrstart = 4)
cania.ts <- drop.years(cania.ts)
cania.ts <- set.plot.titles(cania.ts,
title.elements = c("StationID", "StnName", "StateProv"))
res1 <- pk.bf.stats(cania.ts)

# trend and changepoint plot for baseflow peak start doy
res2 <- screen.metric(res1[,1], ylabel = "Day of Year")
```

---

pk.cov *Center of Volume*

Description

This function calculates center of volume metrics, including the day of the hydrologic year that 25 percent, 50 percent, and 75 percent of the total annual streamflow is reached. A value of 0 is returned for years with no flow. Hydrologic years with fewer than normal observations (outliers) are excluded from the analysis, and for stations with seasonal flow records, additional seasonal subsetting is done to include only days with observations in all years.

Usage

```
pk.cov(TS, threshold.missing = 0.5)
```

Arguments

`TS`	data.frame of streamflow time series loaded with `read.flows`.
`threshold.missing`	Numeric value indicating the fraction of data that can be missing in a single year. Years with a missing data above this threshold will have NA values returned. Default is 0.5 (max of 50% missing data allowed).

Value

Returns a data.frame with the following columns:

- hYear - Hydrologic Years
- Q25 - day of hydrologic year for 25 percent of the total annual streamflow
- Q50 - day of hydrologic year for 50 percent of the total annual streamflow, i.e. Center of Volume
- Q75 - day of hydrologic year for 75 percent of the total annual streamflow
- Dur - duration of between the 25 percent and 75 percent day of year, in days

Author(s)

Jennifer Dierauer

Examples

```
data(cania.sub.ts)
cania.sub.ts <- set.plot.titles(cania.sub.ts,
title.elements = c("StationID", "StnName", "StateProv"))
res1 <- pk.cov(cania.sub.ts)

# trend and changepoint plot for baseflow peak start doy
res2 <- screen.metric(res1[,2], "Day of Year")
res2 <- screen.metric(res1[,2], "Day of Year", title = TRUE)
```

---

pk.max *Annual maximum series*

Description

This function returns the annual maximum series from a daily streamflow time series.

Usage

```
pk.max(TS)
```

Arguments

TS	data.frame of streamflow time series loaded with `read.flows`.

Value

Returns a numeric vector containing the annual maximum flow (m3/s) series, by hydrologic year. The "times" attribute contains the hydrologic year for each element in the vector.

Author(s)

Jennifer Dierauer

See Also

See `create.ts` to format the input flow series.

See `pk.max.doy` to find the day of year for each annual maximum flow event.

Examples

```
data(caniapiscau)
cania.ts <- create.ts(caniapiscau, hyrstart = 4)
cania.ts <- drop.years(cania.ts)
cania.ts <- set.plot.titles(cania.ts,
title.elements = c("StationID", "StnName", "StateProv"))

res <- pk.max(cania.ts)
res2 <- screen.metric(res, ylabel = "Q (m3/s)", title = TRUE)
```

pk.max.doy *Day of year for annual maximum series*

Description

This function returns the day of the hydrologic year for each annual maximum flow.

Usage

```
pk.max.doy(TS)
```

Arguments

TS	data.frame of streamflow time series loaded with `read.flows`.

Value

Returns a numeric vector containing the day of the (hydrologic) year for each annual maximum flow. The "times" attribute contains the hydrologic year for each element in the vector.

Author(s)

Jennifer Dierauer

See Also

See `create.ts` to format the input flow series.

See `pk.max` for the annual maximum flow series.

Examples

```
data(caniapiscau)
cania.ts <- create.ts(caniapiscau, hyrstart = 4)
cania.ts <- drop.years(cania.ts)
cania.ts <- set.plot.titles(cania.ts,
title.elements = c("StationID", "StnName", "StateProv"))

res <- pk.max.doy(cania.ts)
res2 <- screen.metric(res, ylabel = "Day of Year", title = TRUE)
```

pks *Get the flow peaks over a threshold*

Description

This function finds the flow peaks over a user defined threshold and declusters to remove dependent peaks.

Usage

```
pks(TS, Dur = 5, Qmax = 0.95)
```

Arguments

TS	data.frame of streamflow time series loaded with `read.flows`.
Dur	numeric value of the minimum number of days between peaks
Qmax	numeric value for peaks over threshold quantile. Default is 0.95.

Details

Peaks Over Threshold values are calcuated as mean daily streamflow (m3/s) minus the threshold streamflow value (m3/s) defined by the input quantile (Qmax). Peaks are identified with `pot` and the minimum inter-event duration (Dur) is applied by `decluster`.

Value

Returns a numeric vector of peaks of threshold values in m3/s. The "times" attribute contains the date by calendar year, and the "names" attribute contains the hydrologic year and hydrologic day of year, e.g., 2012 55.

Author(s)

Jennifer Dierauer

Examples

```
data(caniapiscau)
cania.ts <- create.ts(caniapiscau, hyrstart = 4)
cania.ts <- drop.years(cania.ts)
cania.ts <- set.plot.titles(cania.ts,
title.elements = c("StationID", "StnName", "StateProv"))

res <- pks(cania.ts)
res2 <- screen.metric(res, ylabel = "Peak Over Threshold (m3/s)", title = TRUE)
```

pks.dur *Calculate the inter-event duration*

Description

This function calculates duration (in days) between flow peaks.

Usage

```
pks.dur(Peaks)
```

Arguments

Peaks Output from pks.

Value

Returns a numeric vector containing the duration (in days) between peaks over threshold from pks. The "times" attribute contains the calendar year date of the earlier peak. The "names" attribute contains the hydrologic year and the day (1-366) of the hydrologic year.

Author(s)

Jennifer Dierauer

Examples

```
data(caniapiscau)
cania.ts <- create.ts(caniapiscau, hyrstart = 4)
cania.ts <- drop.years(cania.ts)
cania.ts <- set.plot.titles(cania.ts,
title.elements = c("StationID", "StnName", "StateProv"))

res1 <- pks(cania.ts)
res2 <- pks.dur(res1)
res3 <- screen.metric(res2, ylabel = "Inter-Event Duration (days)")
```

Qn *Calculate flow quantiles*

Description

This function calculates flow quantiles by hydrologic year, calendar year, month, or doy.

Usage

```
Qn(TS, n = 0.1, by = "hyear")
```

Arguments

TS	data.frame of streamflow time series loaded with `read.flows`.
n	Numeric value of the quantile. Default is 0.1.
by	Character string indicating time unit to summarize by. Default is "hyear" for hydrologic year, see `create.ts`. Other options are "year" for calendar year, "month", or "doy" for day of year.

Value

Returns a numeric vector of the calculated flow quantile for the time periods indicated with the "by" argument. The "times" attribute contains the hydrologic year, calendar year, month, or day of year for each data point.

Author(s)

Jennifer Dierauer

Examples

```
data(caniapiscau)
cania.ts <- create.ts(caniapiscau, hyrstart = 4)
cania.ts <- drop.years(cania.ts)
cania.ts <- set.plot.titles(cania.ts,
title.elements = c("StationID", "StnName", "StateProv"))

# 50% quantile, i.e. mean, by calendar year
res <- Qn(cania.ts, n=0.5, by="year")
res2 <- screen.metric(res, ylabel = "Q (m3/s)")

# Default 10% quantile, by hydrologic year
res <- Qn(cania.ts)
res2 <- screen.metric(res, ylabel = "Q (m3/s)")
```

---

read.flows *Read file of streamflows*

Description

Reads .csv, .Rdata, or .rds files of daily streamflow time series. Recognizes several formats, including those used by Water Survey Canada (WSC), United States Geological Survey (USGS), and ROBIN. Reads fixed width .txt files in GRDC format only. Uses read.csv(), load(), readRDS(), read.fwf() functions from base package and returns data frame with ID, Date, Flow, Agency, and, if available, associated quality codes and source agency. Replaces negative values that are sometimes used to denote missing data with NAs.

Usage

```
read.flows(filename, flow.units = "m3/s", convert.to = NULL, hyrstart = 10)
```

Arguments

- `filename` name of .csv, .txt, .rds, or .rdata file to be read. Filename should contain the file type extension, e.g. "station1.csv"
- `flow.units` Character string indicating the units for streamflow values, one of either 'ft3/s' or 'm3/s'. If the streamflow is in different units, it must be converted prior to use of the package functions, or the units will be labeled as 'Unknown'. Default is 'm3/s'.
- `convert.to` Character string indicating desired flow units (if different from original flow units). Options are 'm3/s' or 'ft3/s'. Default is `NULL`, indicating no unit conversion. If input matches the `flow.units` parameter, nothing will be done.
- `hyrstart` integer used to define start month of hydrologic year. Defaults to 10 (October).

Details

Streamflow records in .csv, .Rdata, or .rds format that are not from the USGS, WSC, ROBIN, or GRDC can be read by read.flows() if they contain the following required columns. Date format is auto-detected as long as it is some version of YYYY/mm/dd, mm/dd/YYYY, mm-dd-yy, etc. The file-to-be-read must contain, at a minimum, columns containing a partial matches to the following (not case sensitive):

- flow | val | value for the daily streamflow discharge (Flow column)
- id | site for the ID column
- date for the Date column

Optional columns names for partial matching include:

- sym | code | flag for the SYM column (quality codes)
- agency for the Agency column

Author(s)

Jennifer Dierauer

Examples

```
# example code to read a file, not run
# my_file_path <- "/Project/file1.csv"
# dat1 <- read.flows(my_file_path)

# Example code using external files included with the package
wsc_path <- system.file("extdata", "WSC_example.csv", package = "FlowScreen")
wsc_dat <- read.flows(wsc_path)

usgs_path <- system.file("extdata", "USGS_example.csv", package = "FlowScreen")
usgs_dat <- read.flows(usgs_path, flow.units = 'ft3/s', convert.to = 'm3/s')

robin_path <- system.file("extdata", "ROBIN_example.csv", package = "FlowScreen")
robin_dat <- read.flows(robin_path)
```

```
## Not run:
grdc_path <- system.file("extdata", "GRDC_example.txt", package = "FlowScreen")
grdc_dat <- read.flows(grdc_path)

## End(Not run)
```

---

regime *Plot flow regime*

Description

This function plots the min, max, mean, and two user-defined quantiles of daily streamflow to provide visual summary of the flow regime. Flow record must have at least 10 years of data to produce a plot. A visual summary is not shown for any days of the year that are missing >80 Area between the upper and lower quantile is shaded grey, the dark blue line represents the mean daily discharge, gray line represents the median daily discharge, and the period of record daily maximum and minimum are shown with the blue points.

Usage

```
regime(
  TS,
  q = c(0.9, 0.1),
  title = FALSE,
  hyear.start = 10,
  y.lims = NA,
  legend = TRUE,
  change.margins = TRUE
)
```

Arguments

TS	data.frame of streamflow time series loaded with read.flows.
q	Numeric vector of the upper and lower quantile values. Default is c(0.9, 0.1).
title	optional plot title. Default is FALSE indicating no plot title is wanted. Set to TRUE to use the the default plot title, which will look for 'plot title' attribute of the data.frame set by set.plot.titles. All values other values will be used as a custom plot title.
hyear.start	Integer indicating the start month for the regime plot. Default is 10 (October).
y.lims	optional user-defined y-axis minimum and maximum. e.g. c(0, 500)
legend	TRUE or FALSE to indicate whether a legend should be included. Default is TRUE.
change.margins	TRUE or FALSE to indicate whether the user's current margin settings should be used, or if the margins should be set within the function. Default is TRUE, to set margins to the minimal amount.

Author(s)

Jennifer Dierauer

Examples

```
# Load example ROBIN streamflow data
robin_path <- system.file("extdata", "ROBIN_example.csv", package = "FlowScreen")

TS <- read.flows(robin_path)
TS <- set.plot.titles(TS)
regime(TS, title = TRUE)
```

---

remove.station.metadata

Remove MetaData for one station from database

Description

Removes a station's metadata from the package database based on the Agency and StationID. If the agency is "USGS" and the station is not found, it will also check by adding a "0" to the beginning of the StationID. Used to remove a case added in error.

Usage

```
remove.station.metadata(Agency, StationID)
```

Arguments

`Agency`	string indicating the source of the streamflow data, e.g. USGS, WSC, etc. Cannot be NA.
`StationID`	string, cannot be NA.

Value

The metadata of the removed station if found and removed, or NULL if not found.

Examples

```
## Not run:
# Add station metadata
met_added <- add.station.metadata(
  Agency = "Foo Bar",
  StationID = "01234",
  StnName = "Example Station",
  StateProv = "Example State",
  Country = "Example Country",
  Lat = 40.0,
  Lon = -89.0,
```

```
    CatchmentArea_km2 = 500,
    RHN = TRUE,
    StationID_Alternate = "01234A",
    Overwrite = FALSE
  )


  # Remove the added station metadata
  met_removed <- remove.station.metadata(
    Agency = "Foo Bar",
    StationID = "01234"
  )

  ## End(Not run)
```

`screen.cpts` *Change point time series plot*

Description

Compiles change point information for all metrics and outputs a daily flow time series plot overlain with a bar plot of changepoint counts by year.

Usage

```
screen.cpts(metrics, type = "a", title = FALSE, change.margins = TRUE)
```

Arguments

`metrics`	output from `metrics.all`
`type`	character indicating which type of metric to compile change points for. Options are "h" for high flow metrics, "l" for low flow metrics, "b" for baseflow metrics, or "a" for all 30 metrics (10 high, 10 low, 10 baseflow).
`title`	optional plot title. Default is FALSE indicating no plot title is wanted. Set to TRUE to use the the default plot title, which will look for 'plot title' attribute of the data.frame set by `set.plot.titles`. All values other values will be used as a custom plot title.
`change.margins`	TRUE or FALSE to indicate whether the user's current margin settings should be used, or if the margins should be set within the function. Default is TRUE, to set margins to the minimal amount.

Value

When type="a", returns a data.frame of changepoint counts by metric type and year.

Author(s)

Jennifer Dierauer

See Also

metrics.all

Examples

```
# load results from metrics.all function for the Caniapiscau River
data(caniapiscau.res)

# plot changepoints for all groups of metrics
screen.cpts(caniapiscau.res, type="l")
screen.cpts(caniapiscau.res, type="h")
screen.cpts(caniapiscau.res, type="b")
```

---

screen.frames — *Plot one or more frames from the summary screening plot*

Description

This function plots one or more frames (i.e. time series plot) from any of the three plot.screening summary plots at a time. It can be used to create custom summary plots - see the example code.

Usage

```
screen.frames(
  metrics,
  type = "h",
  element = NULL,
  language = "English",
  mmar = c(3, 4, 0.5, 0.5),
  title = FALSE,
  multi = F,
  xaxis = T
)
```

Arguments

metrics	output from metrics.all
type	Character string indicating the set of metrics to plot. Options are "h" for high flow metrics, "l" for low flow metrics, or "b" for baseflow metrics.
element	Numeric index(es) (1-10) of the frame(s) to plot, see details of this function for the list of metrics for each category (high, low, baseflow). Each category has ten different metrics that can be plotted individually. Default is NULL, which creates individual plots for all ten metrics. A list of elements c(1, 5, 10) can be specified or a range c(1:5).
language	Language for plot labels. Choice of either "English" or "French". Default is "English".

`mmar`	Numeric vector specifying plot margins. Default is c(3,4,0.5,0.5)
`title`	optional plot title. Default is FALSE indicating no plot title is wanted. Set to TRUE to use the the default plot title, which will look for 'plot title' attribute of the data.frame set by `set.plot.titles`. All values other values will be used as a custom plot title. Set to FALSE to use this function in a multi-plot layout.
`multi`	Boolean indicating whether the function is being used to create one plot in a multi-plot layout. Default is F. If T, suppresses the reset of plot parameter settings. This plot function will only work for a multi-plot layout if text=F
`xaxis`	Boolean indicating whether to plot an x-axis. Default = T.

Details

High flow metrics include:

1. Annual Maximum Series
2. Annual Maximum Day of Year
3. Peaks Over Threshold (Qmax)
4. Inter-Event Duration
5. Q80
6. Q90
7. Day of Year 25 percent Annual Flow
8. Center of Volume
9. Day of Year 75 percent Annual Flow
10. Duration between 25 percent and 75 percent Annual Flow

Low flow metrics include:

1. Q10
2. Q25
3. Drought Start
4. Drought Center
5. Drought End
6. Drought Duration
7. Drought Severity
8. Annual Minimum Flow
9. Mean Annual Minimum 7-day Flow
10. Mean Annual Minimum 10-day Flow

Baseflow metrics include:

1. Mean Daily Discharge
2. Annual Baseflow Volume
3. Annual Mean Baseflow

4. Annual Maximum Baseflow
5. Annual Minimum Baseflow
6. Mean Annual Baseflow Index
7. Day of Year 25 percent Baseflow Volume
8. Center of Volume Baseflow
9. Day of Year 75 percent Baseflow Volume
10. Duration between 25 percent and 75 percent Baseflow Volume

Author(s)

Jennifer Dierauer and Paul Whitfield

Examples

```
# load results from metrics.all function for the Caniapiscau River
data(caniapiscau.res)
caniapiscau.ts <- caniapiscau.res$indata

# plot one frame from the baseflow screening plot
screen.frames(caniapiscau.res, type="b", element=1)

# plot three frames from the low flow screening plot
screen.frames(caniapiscau.res, type="l", element=c(1:3))

# create a custom summary plot
opar <- par(no.readonly = TRUE)
layout(matrix(c(1,2,3,4), 2, 2, byrow=TRUE))
par(oma=c(0,0,3,0))
stninfo <- station.info(caniapiscau.ts$Agency[1], caniapiscau.ts$ID[1], Plot=TRUE)
screen.frames(caniapiscau.res, type="h", element=1, multi=TRUE)
screen.frames(caniapiscau.res, type="l", element=1, multi=TRUE)
screen.frames(caniapiscau.res, type="b", element=1, multi=TRUE)

par <- opar
layout(1,1,1)

# or plot everything!
opar <- par(no.readonly = TRUE)
layout(matrix(c(1:30), 5, 6, byrow=TRUE))
screen.frames(caniapiscau.res, type="h", multi=TRUE)
screen.frames(caniapiscau.res, type="l", multi=TRUE)
screen.frames(caniapiscau.res, type="b", multi=TRUE)
par <- opar
layout(1,1,1)
```

`screen.frames.internal` *Internal wrapper for creating trend and change-point plots*

Description

Internal wrapper for creating trend and change-point plots

Usage

```
screen.frames.internal(
  input,
  mparam,
  mylab,
  DataType,
  maf,
  mmar,
  text,
  xaxis,
  Year1,
  YearEnd,
  hyrstart
)
```

Arguments

`input`	metric time series
`mparam`	trend and change point info
`mylab`	y axis label
`DataType`	numeric indicating data type
`maf`	mean annual flow series
`mmar`	plot margins
`text`	title passed from screen.frames (text or NULL)
`xaxis`	boolean indicating whether to plot the x axis
`Year1`	start year of original time series
`YearEnd`	end year of original time series
`hyrstart`	numeric indicating month for start of the hydrologic year

Author(s)

Jennifer Dierauer

`screen.metric` *Plot a metric with trend and change points*

Description

This function plots a time series of a streamflow metric with the prewhitened linear trend and any detected changepoints in mean and variance.

Usage

```
screen.metric(y, ylabel = "", title = FALSE, change.margins = TRUE)
```

Arguments

`y`	Numeric vector with "times" attribute, and, optionally, a 'StationID' and a 'Agency' attribute if you want the function to auto-generate a default plot title.
`ylabel`	Character string for the y-axis label
`title`	optional plot title. Default is FALSE indicating no plot title is wanted. Set to TRUE to use the the default plot title, which will look for 'plot title' attribute of the data.frame set by `set.plot.titles`. All values other values will be used as a custom plot title. Set to FALSE to use this function in a multi-plot layout.
`change.margins`	TRUE or FALSE to indicate whether the user's current margin settings should be used, or if the margins should be set within the function. Default is TRUE, to set margins to the minimal amount.

Details

This function plots detected changepoints as a vertical dashed line. The means on either side of a changepoint are plotted as solid black lines. If the temporal trend is significant (p-value < 0.1), the trend is plotted as a blue or red line for an increasing or decreasing trend, respectively. The upper and lower 95 dotted red or blue lines. If a trend is not significant, it is not plotted.

Value

Returns a list containing results from the trend and changepoint analysis. This list has the following elements:

- slope - Numeric vector containing the intercept and slope of the prewhitened linear trend computed with `zyp.trend.vector` using Yue Pilon's method
- ci1 - numeric vector containing the intercept and slope of the upper confidence bound. See `confint.zyp`
- ci2 - numeric vector of length 2 containing the intercept and slope of the lower confidence bound. See `confint.zyp`
- pval - numeric value indicatng the significance value of the detected trend, Kendall test computed within `zyp.trend.vector`

- cpts - numeric vector of changepoints if any are found, computed with `cpt.meanvar`. Will be NULL if changepoint analysis was not run due to insufficient data.
- means - numeric vector of means computed with `cpt.meanvar`. Will be NULL if changepoint analysis was not run due to insufficient data.

Author(s)

Jennifer Dierauer

See Also

See `screen.summary` to create a summary screening plot of high flow, low flow, or baseflow metrics.

See `metrics.all` to calculate 30 different streamflow metrics at once. The `screen.metric` function could then be used to loop through the metrics and create an individual plot for each.

Examples

```
data(cania.sub.ts)

# calculate and plot the annual maximum series
cania.sub.ts <- set.plot.titles(cania.sub.ts)
res <- pk.max(cania.sub.ts)
res1 <- screen.metric(res, ylabel=cania.sub.ts$FlowUnit[1],
title = TRUE)

# calculate and plot the annual minimum series
res <- MAMn(cania.sub.ts, n=1)
res1 <- screen.metric(res, ylabel="Discharge (m3/s)",
title = TRUE)
```

`screen.series` *Create a plot of the daily streamflow time series*

Description

Plots the daily streamflow time series and color codes points by data quality codes if the data are from Water Survey Canada. Also highlights date ranges with missing observations.

Usage

```
screen.series(TS, title = FALSE, change.margins = TRUE)
```

Arguments

TS	data.frame of streamflow time series loaded with read.flows.
title	optional plot title. Default is FALSE indicating no plot title is wanted. Set to TRUE to use the the default plot title, which will look for 'plot title' attribute of the data.frame set by set.plot.titles. All values other values will be used as a custom plot title.
change.margins	TRUE or FALSE to indicate whether the user's current margin settings should be used, or if the margins should be set within the function. Default is TRUE, to set margins to the minimal amount.

Author(s)

Jennifer Dierauer and Paul Whitfield

Examples

```
# load flow time series for the Caniapiscau River
data(cania.sub.ts)

# plot daily time series with default margin text
screen.series(cania.sub.ts)
```

---

screen.summary *Create a summary screening plot*

Description

Produces summary screening plots of high flow, low flow, or baseflow metrics. Each plot shows significant temporal trends and step changes. Intended for use as a data quality screening tool aimed at identifying streamflow records with anthropogenic impacts or data inhomogeneities.

Usage

```
screen.summary(metrics, type = "h", language = "English")
```

Arguments

metrics	output from metrics.all
type	Character indicating the set of metrics to plot. Options are "h" for high flow metrics, "l" for low flow metrics, or "b" for baseflow metrics.
language	Language for plot labels. Choice of either "English" or "French". Default is "English".

Details

For the center of volume (COV) plots on the high flow and baseflow screening plots, the correlation coefficients for COV and years and for mean annual flow (MAF) and years are added to the plot. The ratio of the correlation coefficients (r COV-years / r COV-MAF) is included as a rudimentary indication of whether or not the temporal trend in COV is meaningful. See Whitfield (2013) for a discussion of COV.

Drought metrics for the low flow plot may not be applicable to intermittent streams, and plots will be empty in this case.

Important note: If "French" is the language wanted for the plot labels, the language option must also be specified in `metrics.all`, as this plotting function pulls the metric names from the output metrics.all output.

Author(s)

Jennifer Dierauer

References

Whitfield, P.H. 2013. Is 'Center of Volume' a robust indicator of changes in snowmelt timing? Hydrological Processes 27:2691-8.

Examples

```
# load results from metrics.all function for the Caniapiscau River
data(caniapiscau.res)

# create a summary flow screening plot of the high flow metrics
screen.summary(caniapiscau.res, type="h")
# screen.summary(caniapiscau.res, type = "l")
# screen.summary(caniapiscau.res, type = "b)
```

`screen.summary.internal` *Internal wrapper for creating trend and change-point summary plots*

Description

Internal wrapper for creating trend and change-point summary plots

Usage

```
screen.summary.internal(
  input,
  mparam,
  type,
  ylabs,
  i,
```

```
  DataType,
  maf,
  Year1,
  YearEnd,
  hyrstart
)
```

Arguments

`input`	metric time series
`mparam`	trend and change point info
`type`	character indicating type of summary plot
`ylabs`	y axis labels
`i`	plot position
`DataType`	numeric indicating data type
`maf`	mean annual flow series
`Year1`	start year of original time series
`YearEnd`	end year of original time series
`hyrstart`	numeric indicating month for start of the hydrologic year

Author(s)

Jennifer Dierauer

`set.plot.titles` *Set plot titles*

Description

Sets the title to be used for all plots.

Usage

```
set.plot.titles(
  TS,
  title.elements = c("StationID", "StnName", "Country"),
  delimeter = " - ",
  custom.title = NULL,
  title.size = 1
)
```

Arguments

TS	data.frame of streamflow time series loaded with `read.flows`.
title.elements	A character vector with the title elements you want to include in the plot title, in the desired order. Possible values are: Agency, StationID, StnName, StateProv, Country, Lat, Lon, CatchmentArea_km2, MetadataSource. Default is `c("StationID", "StnName", "Country")`. Additional examples: `c("StnName", "StateProv")`, `c("StnName", "StationID")`, etc.
delimeter	separator for title elements, default is " - "
custom.title	String of a custom plot title. Default is NULL. Will supersede title.format if not NULL.
title.size	parameter cex for the base::plot function. Number indicating the amount by which plotting text and symbols should be scaled relative to the default. 1=default, 1.5 is 50 percent larger, 0.5 is 50 percent smaller, etc.

Value

Returns the input TS data.frame with a 'plot title' attribute added. This attribute will be the default option used for all plot titles unless an alternative title is passed to the plotting function, e.g. with `regime`.

Author(s)

Jennifer Dierauer

Examples

```
# Load example ROBIN streamflow data
robin_path <- system.file("extdata", "ROBIN_example.csv", package = "FlowScreen")
TS <- read.flows(robin_path)
TS <- set.plot.titles(TS, title.elements = c("StationID", "StnName"))
regime(TS, title = TRUE)

TS <- set.plot.titles(TS, custom.title = "My Custom Plot Title")
regime(TS, title = TRUE)
```

---

station.info *Retrieve Station Info*

Description

Returns station information from metadata included in the package data files. If there is no metadata match for the StationID AND Agency, returns NA values for all other columns.

Usage

```
station.info(Agency, StationID, Plot = FALSE, Language = "English")
```

Arguments

Agency	String indicating Agency where streamflow data.
StationID	String of the Station ID
Plot	Boolean indicating whether a plot of station information should be created. Default is FALSE. Plot is intended for use as the upper-left panel of the plot produced by screen.summary.
Language	Language for plotting when Plot = T. Choice of either "English" or "French". Default is "English".

Value

Returns a data.frame of the following station information:

- $Agency - Name of Agency from which the record came
- $StationID
- $StnName
- $StateProv - State, Province, or Territory where the station is located
- $Country - Country in which the station is located
- $Lat - Latitude of the station, numeric
- $Lon - Longitude of the station, numeric
- $CatchmentArea_km2 - total drainage area in square kilometers
- $RHN - Indication whether the station is part of a reference hydrologic network, TRUE/FALSE
- $MetadataSource - Indication of where the metadata came from, e.g. WSC Hydat, USGS, user-supplied.
- $StationID_Alternate - Alternate station ID, e.g. original station ID versus ROBIN database

Author(s)

Jennifer Dierauer

Examples

```
data(cania.sub.ts)
stn_metdat <- station.info(cania.sub.ts$Agency[1], cania.sub.ts$ID[1])
print(stn_metdat)
```

`YMD.internal` *Add calendar year, month, and day of year columns*

Description

Add calendar year, month, and day of year columns

Usage

```
YMD.internal(TS)
```

Arguments

`TS`	data.frame of streamflow time series loaded with `read.flows`.

Value

Returns a data.frame with year, month, and doy columns appended.

Author(s)

Jennifer Dierauer

Index

∗ **datasets**
  cania.sub.ts, 9
  caniapiscau, 11
  caniapiscau.res, 12
∗ **package**
  FlowScreen, 21

add.station.metadata, 3
axis_doy.internal, 4

bf.seas, 5, *21*
bf.stats, *5*, 6, *21*, *27*
bf_boughton, 7
bf_eckhardt, *5, 6*, 8, *33*
bf_oneparam, 9

cania.sub.ts, 9
caniapiscau, 11
caniapiscau.res, 12
check_completeness, 12
confint.zyp, *48*
cpt.meanvar, *22*, *26*, *49*
create.ts, 14, *17*, *19*, *30*, *32*, *35, 36*, *39*

decluster, *22*, *37*
dr.events, 15, *17–19*, *30*
dr.pds, *16*, 16, *18*, *30*
dr.seas, *16, 17*, 18, *21*, *27*, *30*
drop.years, 19

FDC, 20
FlowScreen, 21

get.station.internal, 23
get.titles.internal, 23

hyear.internal, 24

Kendall, *22*

MAMn, *21*, 24, *27*

metrics.all, *12*, *21*, 25, *43, 44*, *49–51*
missingness, *13*, 28
mqt, *16–19*, *26*, 30

NA.count.runs, 31, *32*
NA.runs, *31*, 32

pk.bf.stats, *21*, *27*, 33
pk.cov, *21*, *26, 27*, 34
pk.max, *21*, *26*, 35, *36*
pk.max.doy, *21*, *26*, *35*, 36
pks, *26*, 37, *38*
pks.dur, *26*, 38
pot, *22*, *37*

Qn, *21*, *26, 27*, 38

read.flows, *5, 6*, *11*, *14–16*, *18–21*, *24*, *26*, *28*, *30*, *32–37*, *39*, 39, *41*, *50*, *53*, *55*
regime, *21*, 41, *53*
remove.station.metadata, 42

screen.cpts, *12*, *21*, 43
screen.frames, *12*, *21*, 44
screen.frames.internal, 47
screen.metric, *21*, *25*, *28*, 48, *49*
screen.series, 49
screen.summary, *12*, *21*, *26*, *49*, 50, *54*
screen.summary.internal, 51
set.plot.titles, *20*, *28*, *41*, *43*, *45*, *48*, *50*, 52
station.info, 53

YMD.internal, 55

zyp.trend.vector, *22*, *26, 27*, *48*