Package 'CSHShydRology'

March 12, 2019

Type Package

Title Canadian hydrological analyses

Version 1.0.0

Date 2018-07-31

Description A collection of user submitted functions to aid in the analysis of hydrological data.

License AGPL-3 | file LICENSE

URL https://github.com/CSHS-hydRology/CSHShydRology

Depends R (>= 3.1)

Imports fields,
Kendall,
lubridate (>= 1.3),
plotrix,
timeDate

Suggests knitr,
testthat

VignetteBuilder knitr

LazyData true

RoxygenNote 6.1.1

R topics documented:

`CSHShydRology-package` *Functions for Canadian hydrological analyses*

Description

CSHShydRology is intended for the use of hydrologists, particularly those in Canada. It will contain functions which focus on the use of Canadian data sets, such as those from Environment Canada. The package will also contain functions which are suited to Canadian hydrology, such as the important cold-region hydrological processes. **CSHShydRology** will also contain functions which work with Canadian hydrological models, such as Raven, CRHM, Watflood, and MESH.

This packages has been developed with the assistance of the Canadian Society for Hydrological Sciences (CSHS) https://cwra.org/en/branches/affiliates/cshs-a which is an affiliated society of the Canadian Water Resources Association (CWRA) cwra.org.

The **CSHShydRology** will contain functions grouped into several themes, including:

Statistical hydrology trend detection, data screening, frequency analysis, regionalization

Basic data manipulations input/conversion/adapter functions, missing data infilling

Visualization data visualization, standardized plotting functions

Spatial hydrology basin delineation, landscape data analysis, working with GIS

Streamflow measurement analysis rating curve analysis, velocity profiles, naturalization

Network design/analysis homogeneity assessment

Ecohydrology fisheries and ecological analysis

Wrappers/unwrappers between other packages and **CSHShydRology**

References

To cite **CSHShydRology** in publications, use the command `citation("CSHShydRology")` to get the current version of the citation.

`axis_doy` *Generates the x axis for day of year*

Description

Used by `regime_plot`. This code deals only with the axis adjustments. Day of water year needs to be done separately

Usage

```
axis_doy(wyear = 1)
```

Arguments

`wyear`	Month to begin water year. Use `wyear = 1` for calendar year, `wyear = 10` for October 1.

Author(s)

Paul Whitfield

`Basic_data_manipulations-methods`
Basic data manipulation functions

Description

These functions read in or convert values among formats

read_ECDE_flows Reads a file of WSC daily flows from ECDataExplorer

get_wscstation Reads station information from a data file produced by ECDE

`binned_MannWhitney` *Compares two time periods of data using Mann-Whitney test*

Description

It bins data based upon a bin size, extracting data for two time periods and tests for change between two such periods. Result can be passed to `polar_plot` for visualization

Usage

```
binned_MannWhitney(mdata, step, range1, range2, ptest = 0.05,
  station_ID = "", station_name = "", variable = "discharge")
```

Arguments

`mdata`	A data frame of hydrometric data. Must contain the variables `Date` and `Flow`.
`step`	An integer indicating the degree of smoothing eg. 1, 5, 11.
`range1`	The first and last year of first period, as `c(first,last)`
`range2`	The first and last year of second period as codec(first,last)
`ptest`	The significance level. The default is `0.05`.
`station_ID`	Optional ID of station.
`station_name`	Optional name of station.
`variable`	Name of variable. Default is 'discharge'

Value

Returns a list containing:

StationID ID of station

Station_lname Name of station

bin_width Smoothing time step

range1 range1 years

range2 range2 years

p_used p value used

fail TRUE if test failed due to missing values

bin_method method used for binning

test_method Mann-Whitney U

series a data frame containing:

period period numbers i.e. 1:365/step

period1 median values for each bin in period 1

period2 median values for each bin in period 2

mwu Mann Whitney U-statistic for each bin between the two periods

prob probability of U for each period

code significance codes for each bin

Author(s)

Paul Whitfield <paul.h.whitfield@gmail.com>

References

Whitfield, P.H., Cannon, A.J., 2000. Recent variations in climate and hydrology in Canada. Canadian Water Resources Journal 25: 19-65.

See Also

polar_plot polar_plot_prep

Examples

```
## Not run:
# fails due to missing data in both periods
range1 <- c(1960,1969)
range2 <- c(1990,1999)
b_MW <- binned_MannWhitney(W05AA008, step=5, range1, range2, ptest=0.05)
## End(Not run)

range1 <- c(1970,1979)
range2 <- c(1990,1999)
b_MW <- binned_MannWhitney(W05AA008, step = 5, range1, range2,
ptest = 0.05, station_ID = "05AA008", station_name= "Crowsnest River at Frank")
```

---

booth_plot *Create a Booth plot of peaks over a threshold*

Description

A Booth plot is a plot of peaks over threshold flood events with duration on the horizontal and either magnitude (default) or volume on the vertical axis

Usage

```
booth_plot(events, threshold, title, type = "mag", colour1 = 1,
  colour2 = 1)
```

Arguments

events	A data frame of POT events from the function get_peaks
threshold	The threshold used by get_peaks
title	Plot title
type	The plot type, either 'mag' (magnitude, the default) or 'vol' (volume)

colour1	A vector of length 12 with line colours of rings or symbols. Defaults to those used by Booth.
colour2	A vector of length 12 with fill colours of rings or symbols. Defaults to those used by Booth.

References

Booth, E.G., Mount, J.F., Viers, J.H. 2006. Hydrologic Variability of the Cosumnes River Floodplain. San Francisco Estuary & Watershed Science 4:21.

Whitfield, P.H., and J.W. Pomeroy. 2016. Changes to flood peaks of a mountain river: implications for analysis of the 2013 flood in the Upper Bow River, Canada. Hydrological Processes 30:4657-73. doi: 10.1002/hyp.10957.

See Also

get_peaks

Examples

```
threshold <- 0.1 * max(W05AA008$Flow)  # arbitrary threshold
peaks <- get_peaks(W05AA008, threshold)
events <- peaks$POTevents
booth_plot(events, threshold, title = "05AA008", type='mag')
booth_plot(events, threshold, title = "05AA008", type='vol')
```

---

cut_block *Cuts a block in time from a time series*

Description

Allows the user to select a time period from a longer record. Could be used to get the same period of time from several stations for comparison.

Usage

```
cut_block(dataframe, st_date, end_date)
```

Arguments

dataframe	A time series dataframe with a Date variable
st_date	Starting date as a string with the format 'Y/m/d'
end_date	Ending date as a string with the format 'Y/m/d'

Value

Returns a data frame with the same columns as the original data frame

Author(s)

Paul Whitfield

Examples

```
subset <- cut_block(W05AA008, "2000/01/01", "2010/12/31")
```

`date_subset` *Subset date by String This function subsets a data frame by an specified date range, provided as a string by the* `prd` *argument. This function is meant to emulate the subsetting capability of the* **xts** *package.*

Description

Subset date by String This function subsets a data frame by an specified date range, provided as a string by the `prd` argument. This function is meant to emulate the subsetting capability of the **xts** package.

Usage

```
date_subset(df, prd)
```

Arguments

`df`	data frame of time series data; includes a variable called `Date`
`prd`	date range as string formatted as 'YYYY-MM-DD/YYYY-MM-DD'

Value

`df`	subsetted data frame

Author(s)

Robert Chlumsky <rchlumsk@gmail.com>

Examples

```
{
dd <- seq.Date(as.Date("2010-10-01"), as.Date("2013-09-30"), by = 1)
x <- rnorm(length(dd))
y <- abs(rnorm(length(dd)))*2
df <- data.frame("Date" = dd,x,y)
prd <- "2011-10-01/2012-09-30"
summary(date_subset(df,prd))}
```

doys *Days of year and water year*

Description

Converts a vector of dates into a dataframe with date, doy, dowy, year

Usage

```
doys(Date, mon = 10)
```

Arguments

Date	A vector of R dates.
mon	The month starting the water year, default is 10 (October).

Value

Returns a dataframe with differently-formatted dates

Date original Date

year numeric calendar year

month numeric calendar month

doy numeric day of year

wyear numeric water year

dwy numeric day of water year

Author(s)

Paul Whitfield <paul.h.whitfield@gmail.com>

Examples

```
dd <- seq.Date(as.Date("2010-01-01"), as.Date("2018-01-01"), by = 1)
output <- doys(dd)
head(output)
```

`fdcurve` *Plot Flow Duration Curve*

Description

A flow duration curve is a plot of flow magnitude against exceedance probability. The plot may contain the Gustard Curves or they can be omitted. The default is for curves to be plotted against probability, but an option is to plot against the normalized exceedance probability. In that case, the x axis represents a normal distribution.

Usage

```
fdcurve(flow, title = "", normal = FALSE, gust = TRUE)
```

Arguments

`flow`	Vector containing daily flows
`title`	The plot title
`normal`	If `normal = TRUE` then exceedance probability is normalized. Default is `FALSE`
`gust`	If `TRUE` (the default), adds the curves from Gustard et al 1992

Value

Plots the flow durations and returns a data frame containing the exceedance probabilty and flow

Author(s)

Paul Whitfield <paul.h.whitfield@gmail.com>

References

Gustard, A., A. Bullock, and J.M. Dixon. 1992. Low flow estimation in the United Kingdom. Institute of Hydrology, 292. Wallingford: Institute of Hydrology.

Vogel, R.M., and N.M. Fennessy. 1994. Flow-duration curves. I: New Interpretation and confidence intervals. Journal of Water Resources Planning and Management ASCE 120:485-504.

Examples

```
flow <- W05AA008$Flow
# plot with Gustard 1992 curves
test <- fdcurve(flow, title="Station", normal=FALSE, gust=TRUE)

# plot with normalized exceedance probability
test <- fdcurve(flow, title="Station", normal=TRUE, gust=FALSE)
```

`flow_raster` *Raster plot of streamflows*

Description

Produces a raster plot: years x day of year, showing magnitude of flows. This produces a plot showing the flow data in colours, showing different context than in a hydrograph. High flows are in warm colours

Usage

```
flow_raster(dframe, title = "", rastercolours = c("lightblue", "cyan",
  "blue", "slateblue", "orange", "red"))
```

Arguments

`dframe`	A data frame of hydrometric data. Must contain the variables `Date` and `Flow`.
`title`	The (optional) title for the plot
`rastercolours`	A vector of colours used for the raster plot. The default is `c("lightblue", "cyan", "blue", "slateblu`

Value

No value is returned; a standard R graphic is created.

Author(s)

Paul Whitfield <paul.h.whitfield@gmail.com>

See Also

`flow_raster_trend` `flow_raster_qa`

Examples

```
flow_raster(W05AA008)
```

flow_raster_qa *Raster plot of streamflows with WSC quality flags*

Description

Produces a raster plot of years x day of year showing the flow data in grayscale overlain by the Water Survey of Canada quality flags. Colours are consistent with ECDataExplorer. Raster layout lets the use see the flags in a different context than in a hydrograph. The data flags are:

A (Partial) green

B (Below Ice) blue

D (Dry) yellow

E (Estimated) red

Usage

```
flow_raster_qa(dframe, title = "")
```

Arguments

dframe	A data frame of WSC hydrometric data. Must contain the variables Date, Flow and SYM which is the WSC data flag symbol.
title	The (optional) title for the plot

Value

No value is returned; a standard R graphic is created.

Author(s)

Paul Whitfield <paul.h.whitfield@gmail.com>

See Also

flow_raster_trend flow_raster

Examples

```
flow_raster_qa(W05AA008, "Station W05AA008")
```

`flow_raster_trend` *Raster plot and simple trends of observed streamflows*

Description

Creates a raster plot plus trend plots for day of year, and over time which may be binned by a number of days

Usage

```
flow_raster_trend(date, flow, step = 5, stationID = "", title = "",
  missing = FALSE, colours = c("lightblue", "cyan", "blue",
  "slateblue", "darkblue", "red"))
```

Arguments

`date`	A numeric vector of the date as an R date. Must be same length as the `flow`
`flow`	A numeric vector of daily streamflows
`step`	An integer indicating the degree of smoothing eg. 1, 5, 11.
`stationID`	Station ID number, e.g. "05BB001". This value is optional, but is included in the output to help you identify the results.
`title`	Title of the plot
`missing`	If `FALSE` years with missing data are excluded. If `TRUE` partial years are included.
`colours`	A vector of colours used for the raster plot. The default is `c("lightblue","cyan", "blue", "slateblu`

Details

The plot contains four panels based upon binned data:

1. The annual maximum, minimum, and median dlow with a trend test number for each period: red arrows indicate decreases, blue arrows indicate increases.
2. The scale bar for the colours used in the raster plot,
3. The rasterplot with a colour for each period and each year where data exist, and
4. A time series plot of the minimum, median, and maximum annual bin values. If there is no trend ($p > 0.05$) the points are black. Decreasing trend are in red, increasing trends are in blue..

Value

a list containing:

stationID Station ID eg 05BB001

missing How missing values were used FALSE=used, TRUE=removed

step number of days in a bin

periods number of periods in a year

period period numbers i.e. 1:365/step

bins values for each period in each year

med_period median for each period

max_period maximum for each period

min_period minimum for each period

tau_period Kendall's Tau for each period

prob_period probability of Tau for each period

year years spanning the data

median_year median bin for each year

max_year maximum bin for each year

min_year minimum bin for each year

tau_median_year value of tau and probability for median_year

tau_maximum_year value of tau and probability for max_year

tau_minimum_year value of tau and probability for min_year

Author(s)

Paul Whitfield <paul.h.whitfield@gmail.com>

See Also

flow_raster

Examples

```
mdoy <- doys(W05AA008$Date)
mplot <- flow_raster_trend(W05AA008$Date, W05AA008$Flow, step=5, station="05AA008")
```

---

get_peaks *Extracts peak flows over a threshold*

Description

This function is development code being shared as is. It is expected that the user will be interested in the dataframe returned for POT analysis and for plotting (i.e. Booth_plots).

Usage

```
get_peaks(dataframe, threshold)
```

Arguments

dataframe	a data frame of streamflow data containing columns named 'Date' and 'Flow'
threshold	a value for the threshold. Values above the threshold are tested for peaks.

Details

This function retrieves peaks greater than the prescribed threshold. It returns a dataframe of peak characteristics suitable for subsequent analysis.

The portion under development is the It also returns a list of the flows during an event with the values of the three preceeding dates and three subsequnt dates.

Value

a list containing:

POTevents a dataframe contining details of the events

events a vector with the value 0 when the flow is below the threshold and 1 when above.

event_num a vector with the value 0 when the flow is below a threshold or the index of the events when the threshold was exceeded. i.e. 1,2,3, etc

st_date start date of events

case a list of the flows in each individual event (see details for more information)

The POTevents dataframe contains five columns: st_date (starting date), max_date (date of maximum in the event), max (maximum discharge), volume (volume of the event), and duration (in days).

The case list contains the flows during an event and also for three preceeding and subsequent days. The lists range from seven to n days in length.

References

Burn, D.H., Whitfield, P.H., Sharif, M., 2016. Identification of changes in floods and flood regimes in Canada using a peaks over threshold approach. Hydrological Processes, 39: 3303-3314. DOI:10.1002/hyp.10861

Whitfield, P.H., and J.W. Pomeroy. 2016. Changes to flood peaks of a mountain river: implications for analysis of the 2013 flood in the Upper Bow River, Canada. Hydrological Processes 30:4657-73. doi: 10.1002/hyp.10957.

See Also

booth_plot

Examples

```
threshold <- 0.9*max(W05AA008$Flow)  # arbitrary threshold
my_peaks <- get_peaks(W05AA008, threshold)
str(my_peaks)
```

`get_wscstation` *Reads station information from a data file produced by ECDE Retrieves station information for an individual Water Survey of Canada site, adds a text string at position 21 that combines key elements for a title.*

Description

Reads station information from a data file produced by ECDE Retrieves station information for an individual Water Survey of Canada site, adds a text string at position 21 that combines key elements for a title.

Usage

```
get_wscstation(stnID, stn)
```

Arguments

`stnID`	A Water Survey of Canada station number
`stn`	a data frame of station information from ECDataExplorer. The data frame '`HYDAT_list`' is supplied with this package.

Value

Returns a data frame with 21 variables

- Station StationID
- StationName Station Name
- HYDStatus Active or Discontinued
- Prov Province
- Latitude
- Longitude
- DrainageArea km2
- Years # of years with data
- From Start Year
- To End Year
- Reg. Regulated
- Flow if TRUE/Yes
- Level if TRUE/Yes
- Sed if TRUE/Yes
- OperSched Continuous or Seasonal
- RealTime if TRUE/Yes

- RHBN if TRUE/Yes is in the reference hydrologic basin network
- Region if TRUE/Yes is in the reference hydrologic basin network
- Datum if TRUE/Yes is in the reference hydrologic basin network
- Operator if TRUE/Yes is in the reference hydrologic basin network
- Station_lname Added field combines ID,Name,Province and if RHBN an * is added

Author(s)

Paul Whitfield <paul.h.whitfield@gmail.com>

Examples

```
df <- HYDAT_list
s_info <- get_wscstation("05BB001", df)
title <- s_info[21]
print(title)
```

`HYDAT_list` *HYDAT_list*

Description

A dataframe of station information, as extracted from the EC Data Explorer

Usage

```
HYDAT_list
```

Format

A dateframe with 7791 rows and 20 columns.

Source

Water Survey of Canada

Variables:

- Station StationID
- StationName Station Name
- HYDStatus Active or Discontinued
- Prov Province
- Latitude
- Longitude
- DrainageArea km2

- Years # of years with data
- From Start Year
- To End Year
- Reg. Regulated
- Flow if TRUE/Yes
- Level if TRUE/Yes
- Sed if TRUE/Yes
- OperSched Continuous or Seasonal
- RealTime if TRUE/Yes
- RHBN if TRUE/Yes is in the reference hydrologic basin network
- Region if TRUE/Yes is in the reference hydrologic basin network
- Datum if TRUE/Yes is in the reference hydrologic basin network
- Operator if TRUE/Yes is in the reference hydrologic basin network
- Station_lname Added field combines ID,Name,Province and if RHBN an * is added

---

`hydrograph_plot` *Plot hydrographs*

Description

Creates a hydrograph plot for simulated, observed, and inflow hydrograph series, including precipitation if provided. The secondary y axis will be used to plot the precip time series. The function assumes that the supplied time series have the same length and duration in time. If this is not true, then the defined period or period calculated from the first available flow series will be used to determine the plotting limits in time. If the data is take from output from the **Raven** model, this is not a concern. The supplied time series should be in **xts** format, which can be obtained directly by using the `hyd.extract` function in the package **RavenR** . Note that a plot title is purposely omitted in order to allow the automatic generation of plot titles.

Usage

```
hydrograph_plot(flows = NULL, precip = NULL, prd = NULL,
  winter_shading = FALSE, range_mult_flow = NULL,
  range_mult_precip = 1.5, flow_labels = NULL,
  ylabel = "Flow [m3/s]", precip_label = "Precipitation [mm]",
  leg_pos = NULL, leg_box = NULL, zero_axis = T,
  plot_mode = "base")
```

Arguments

flows	data frame of flows to plot
precip	data frame of precipitation values to plot
prd	period to use in plotting
winter_shading	optionally adds a transparent cyan shading for the December 1st to March 31st period in each year that is plotted. Default is FALSE.
range_mult_flow	range multiplier for max value in hydrograph. This is useful in preventing overlap if precip is also plotted. This value should not be less than 1.0, otherwise the values will be cutoff in the plot.
range_mult_precip	range multiplier for max value in precipitation plot (default 1.5)
flow_labels	string vector of labels for flow values
ylabel	text label for y-axis of the plot (default 'Flow [m3/s]')
precip_label	text label for precipitation y-axis (default 'Precipitation [mm]')
leg_pos	string specifying legend placement on plot e.g. 'topleft', 'right', etc., and is consistent with the legend function options. If NULL, the function will place the legend left, if precip added, on the topleft otherwise).
leg_box	boolean on whether to put legend in an opaque white box or not. If NULL (the default), the function will automatically not use a white box and leave the background of the legend transparent.
zero_axis	fixes the y axis to start exactly at zero (default TRUE). By default, R will plot the values with a small buffer for presentation. Be warned that if this option is set to TRUE, the minimum value is set to zero without checking if any flow values are less than zero. This option should not be used for reservoir stage plotting, since most reservoir stage is typically reported as an elevation.
plot_mode	plot mode as 'base' or 'ggplot'. Currently only 'base' plot type is supported, 'ggplot' is under construction.

Value

Returns TRUE if the function is executed properly.

Author(s)

Robert Chlumsky <rchlumsk@gmail.com>

Examples

```
# example with synthetic random data
dd <- seq.Date(as.Date("2010-10-01"), as.Date("2013-09-30"),by = 1)
x <- abs(rnorm(length(dd)))
y <- abs(rnorm(length(dd))) * x
df <- data.frame("Date" = dd, x, y)
myprd <- "2011-10-01/2012-09-30"
```

```
precip <- data.frame("Date" = dd," precip" = abs(rnorm(length(dd))) * 10)

# basic hydrograph plot
hydrograph_plot(flows = df, winter_shading = FALSE)

# with different labels
hydrograph_plot(flows = df, winter_shading = FALSE, flow_labels = c("simulated", "observed"))

# with a few more options turned on
hydrograph_plot(flows = df, precip = precip)

# increase the plot ranges to separate flows and precip; add a legend box
hydrograph_plot(flows = df, precip = precip, range_mult_flow = 1.7,
range_mult_precip = 2, leg_box = TRUE)
```

`polar_plot` *Polar plot of daily streamflows*

Description

Produces a polar plot similar to that used in *Whitfield and Cannon, 2000*. It uses output from the function `binned_MannWhitney` or a data structure created using the function `polar_plot_prep`.

Usage

```
polar_plot(bmw, lcol1 = c("black", "gray50"), lcol2 = c("black",
  "gray50"), lfill = c("yellow", "green"), lsig = c("red", "blue"))
```

Arguments

`bmw`	output from binned_MannWhitney
`lcol1`	line colour, default is c("black","gray50")
`lcol2`	point colour, default is c("black","gray50")
`lfill`	fill colour, default is c("yellow","green")
`lsig`	significance symbol colour, default is ("red","blue")

Value

No value is returned; a standard R graphic is created.

Author(s)

Paul Whitfield <paul.h.whitfield@gmail.com>

References

Whitfield, P.H. and A.J. Cannon. 2000. Polar plotting of seasonal hydrologic and climatic data. Northwest Science 74: 76-80.

Whitfield, P.H., Cannon, A.J., 2000. Recent variations in climate and hydrology in Canada. Canadian Water Resources Journal 25: 19-65.

See Also

binned_MannWhitney polar_plot_prep

Examples

```
range1 <- c(1970,1979)
range2 <- c(1990,1999)
b_MW <- binned_MannWhitney(W05AA008, step = 5, range1, range2,
ptest = 0.05, station_ID = "05AA008", station_name = "Crowsnest River at Frank")
polar_plot(b_MW)
```

---

polar_plot_prep *Creates a data structure to be passed to* polar_plot.

Description

Could be used to move data from a different type of analysis different to the binned_MannWhitney function which uses flows. The two series need to be of the same length and their length is related to the step size. For examples 73 periods links to 5 day periods.

Usage

```
polar_plot_prep(station, plot_title, step, x0, x1, stat, prob, test_s,
  variable = "discharge", bin_method = "unstated",
  test_method = "unstated", lline1 = "Period 1", lline2 = "Period 2",
  pvalue = 0.05)
```

Arguments

station	Typically a station number
plot_title	Polar plot title - usually a station name
step	The number of days binned
x0	Time series of length n for a single seasonal cycle
x1	Time series of length n for a single seasonal cycle
stat	Time series of length n for statistical test value for each bin
prob	Time series of length n of probability of test value
test_s	Vector with values of -1, 0, 1 for significance, -1 negative, 1 positive, 0 not significant

variable	Name of variable plotted. Default is "discharge"
bin_method	Default is "unstated"
test_method	Default is "unstated"
lline1	Names of first period, default is "Period 1"
lline2	Names of second period, default is "Period 2"
pvalue	Value of p used. Default is 0.05

Value

Returns a list containing:

StationID ID of station

Station_lname Name of station

variable Name of variable

bin_width Smoothing time step

range1 range1 years

range2 range2 years

p_used p value used

fail TRUE if test failed due to missing values

bin_method method used for binning

test_method Mann-Whitney U

series a data frame containing:

period period numbers i.e. 1:365/step

period1 median values for each bin in period 1

period2 median values for each bin in period 2

mwu Mann Whitney U-statistic for each bin between the two periods

prob probability of U for each period

code significance codes for each bin

Author(s)

Paul Whitfield <paul.h.whitfield@gmail.com>

References

Whitfield, P.H. and A.J. Cannon. 2000. Polar plotting of seasonal hydrologic and climatic data. Northwest Science 74: 76-80.

Whitfield, P.H., Cannon, A.J., 2000. Recent variations in climate and hydrology in Canada. Canadian Water Resources Journal 25: 19-65.

See Also

binned_MannWhitney polar_plot

read_ECDE_flows *Reads a file of WSC daily flows from ECDataExplorer Reads in a file WSC daily flows as returned from the program ECDataExplorer, and omits the last 3 lines as these contain the data disclaimer.*

Description

Reads a file of WSC daily flows from ECDataExplorer Reads in a file WSC daily flows as returned from the program ECDataExplorer, and omits the last 3 lines as these contain the data disclaimer.

Usage

```
read_ECDE_flows(filename)
```

Arguments

filename	Datafile retrieved from ECDataExplorer

Value

Returns a dataframe with the last three rows removed and the Date as Date

Author(s)

Paul Whitfield <paul.h.whitfield@gmail.com>

Examples

```
mfile <- system.file("extdata", "04JD005_Daily_Flow_ts.csv", package = "CSHShydRology")
mdata <- read_ECDE_flows(mfile)
```

regime_plot *Plots the regime of daily streamflows*

Description

Produces a regime hydrograph similar to that in the reference. It shows the flow quantiles for each day of the year and the maximum and minimum. Parameters can be set to change colours and fix the y scale to allow plots of same scale to be produced.

Usage

```
regime_plot(date, flow, title = "", wyear = 1, colour = TRUE,
  mx = 1)
```

Arguments

date	Vector of dates
flow	Vector of daily streamflows. Must be the same length as date
title	Text to be used as the graph title
wyear	Beginning month of water year. USe wyear = 10 for October water year, wyear = 1 for calendar year
colour	Logical. If TRUE plot is in colour, if FALSE plot is grayscale
mx	The maximum y value; if mx = 1 then maximum value of the flows is used to set the maximum y-axis value. The value of mx can be specified to produce a series of plots with the same scale.

Value

No value is returned; a standard R graphic is created.

Author(s)

Paul Whitfield

References

MacCulloch, G. and P. H. Whitfield (2012). Towards a Stream Classification System for the Canadian Prairie Provinces. Canadian Water Resources Journal 37: 311-332.

Examples

```
regime_plot(W05AA008$Date, W05AA008$Flow, title = "05AA008", colour = TRUE, wyear = 10)
```

---

slice *Converts doy or dwy into a factor that is used to bin data*

Description

Whenever the number of bins does not divide in 365 evenly a message is printed showing the number of bins created and the number of days added to the last bin. Simply put, slice is used to convert doy into a factor which is a number of bins per year. A year can be converted into any number of bins; slice does it based upon a number of days. So when you send it am array of doy it slices that into bins of the desired width. For example, if the step is 5. They 365/5 gives 73 bins and becasue of leap years there might be one extra day added every four years to the final bin.

To illustrate: doy: 1 2 3 4 5 6 7 8 9 10 11 12 Bin: 1 1 1 1 1 2 2 2 2 2 3 3

Usage

```
slice(doy, step)
```

Arguments

doy	A vector of the day of calendar year for the dataset
step	Width of bin in days

Value

Returns a vector of bin numbers that is used as a factor for each day in the dataset prints a message indicating the handling of partial bins

Author(s)

Paul Whitfield <paul.h.whitfield@gmail.com>

See Also

binned_MannWhitney, raster_trend

Examples

```
doy <- c(1:365)
# first 30 days are 1, 31-60 are 2 etc
dice <- slice(doy, 30)
plot(doy, dice)
```

StatisticalHydrology-methods

Statistical analysis functions

Description

These functions perform statistical analyses

binned_MannWhitney Compares two time periods of data using Mann-Whitney test

fdcurve Finds flow exceedence probabilities

get_peaks Finds peak flows over a specified threshold

sub_set_Years *Helper function for selecting points for an axis so not all are necessary*

Description

Sub-samples a vector every n places. Many times there are so many years the labels on the plot overlap. This function returns the position and label for the subset. The function can be used on any type of simple array.

Usage

```
sub_set_Years(years, n)
```

Arguments

years	a vector of years
n	sample size

Value

a list containing:

position array of axis positions

label array of labels

Author(s)

Paul Whitfield <paul.h.whitfield@gmail.com>

Examples

```
myears <- c(1900:2045)
myears <- sub_set_Years(myears, 20)
myears

a <- LETTERS
my_alpha <- sub_set_Years(a, 5)
my_alpha
```

Visualization-methods *Visualization functions*

Description

These functions are primarily intended for graphing, although some analyses may also be done.

booth_plot Plot of peaks over a threshold

flow_raster Raster plot of streamflows

flow_raster_qa Raster plot of streamflows with WSC quality flags

flow_raster_trend Raster plot and simple trends of observed streamflows

hydrograph_plot Plots hydrographs and/or precipitation

polar_plot Polar plot of daily streamflows

regime_plot Plots the regime of daily streamflows

W05AA008 *W05AA008*

Description

A dataframe of Water Survey of Canada (WSC) daily flows for station W05AA008, CROWSNEST RIVER AT FRANK Alberta.

Usage

```
W05AA008
```

Format

A dateframe with 25252 rows and 5 columns spanning the period 1910-2013.

Source

Water Survey of Canada

Variables:

- ID StationID
- PARAMParameter 1=Flow, 2=Level
- DateR date
- FlowDaily flow in m^3/s
- SYMWater Survey FLags A, B, D, E

wtr_yr *Designation of the water year*

Description

Display water year

Usage

```
wtr_yr(dates, start_month = 10)
```

Arguments

dates	A vector of dates with actual year
start_month	Month in which the year starts (defaults to October)

Value

Year starting in start_month

Source

http://stackoverflow.com/questions/27626533/r-create-function-to-add-water-year-column

Examples

```
date <- seq(as.Date("1910/1/1"), as.Date("1912/1/1"), "days")
wtr_yr_date <- wtr_yr(dates=date, start_month=10)
data.frame(wtr_yr_date,date)
```

Index