Package 'MESHr'

December 5, 2018

Type Package

Title pre- and post processing for MESH

Version 0.1.6

Date 2018-12-05

Author Al Pietroniro, Environment Canada

Maintainer Kevin Shook <kevin.shook@usask.ca>

Description This package contains functions for pre- and post- processing data for the MESH model.

Depends R (>= 3.1)

Imports
grid, ggplot2, stringr, knitr, reshape2, hydroGOF, raster, rts, readr, stats, hydroTSM, sp, methods

License GPL-3

LazyData true

URL https://github.com/CentreForHydrology/MESHr

RoxygenNote 6.1.0

NeedsCompilation no

VignetteBuilder knitr

R topics documented:

MESHr-package 2
basinLapseRates 2
basinPrecipEvapRunoffPlot 3
basinRunoffPlot 4
basinSnowPlot 5
basinSnowRainPondedPlot 5
basinSoilWaterIcePlot 6
basinSoilWaterPlot 7
basinStoragePlot 8
basinStorageVariablesPlot 8
basinWaterBalancePlot 10
distribPrecip 11

findRecord . . . 12
gridPrecip . . . 13
hydroStats . . . 14
MESH_streamflows . . . 16
parseNums . . . 16
parseText . . . 17
PBIAS . . . 18
plotDataCompleteness . . . 18
read_AEP_csv . . . 19
read_MESH_OutputTimeseries_csv . . . 21
read_r2c_raster . . . 22
read_r2c_shed . . . 23
read_tb0 . . . 24
simpleHydrograph . . . 25
var_present . . . 27
win.eol . . . 28
write_r2c_shed . . . 28
write_tb0 . . . 29

Index **32**

`MESHr-package` *Functions for MESH pre- and post- processing*

Description

The intent of this package is to contain functions do do common tasks for MESH modelling, such as reading output, plotting, and assessing model quality. The first functions will use .csv files - support for netCDF will be added.

References

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

`basinLapseRates` *Calculates basin-wide lapse rates*

Description

This function calculates basin-wide historical lapse rates, by month and hour of day. The intent is to produce a file of rates that can be used for interpolation, when there are insufficient values to determine lapse rates from measured air temperatures. The lapse rates are determined as the slope of a linear regression of delta air temperature vs delta elevation for each time step. The delta air temperature is the difference in air temperature between each station's value and that of the lowest elevation station. The delta elevation is the difference between each station's elevation and that of the lowest-elevation station. Therefore the lapse rate is in K/m.

Usage

```
basinLapseRates(temps = NULL, elevs = NULL)
```

Arguments

temps	Required. A time series data frame of air temperatures in Celsius or K, as returned by `read_tb0`. The first column must be `datetime`, which is a POSIXct value. Each station's elevation will be in a separate column.
elevs	Required. A data frame of station elevations. Note that the first column must contain the station names (which must be the same as in the air temperatures), and the second column must contain the elevation (in m).

Value

If successful, returns a data frame with 24 rows (one for each hour), and 12 columns (one for each month) of the lapse ratge

Examples

```
## Not run:
lapse_rates <- basinLapseRates(temperatures, elevations)
## End(Not run)
```

---

basinPrecipEvapRunoffPlot

Plots basin precipitation evaporation and runoff

Description

Plots basin precipitation evaporation and runoff

Usage

```
basinPrecipEvapRunoffPlot(basinWaterBalance)
```

Arguments

basinWaterBalance	Required. Data frame to be plotted. As read in by `read_MESH_OutputTimeseries_csv`.

Value

Returns a **ggplot2** line plot of the variable values (mm).

Author(s)

Kevin Shook

See Also

read_MESH_OutputTimeseries_csv basinStoragePlot basinSoilWaterIcePlot

Examples

```
## Not run:
waterBalance <- read_MESH_OutputTimeseries_csv("Basin_average_water_balance.csv")
p <- basinPrecipEvapRunoffPlot(waterBalance)
## End(Not run)
```

---

basinRunoffPlot *Plots basin runoff components*

Description

Plots basin runoff components

Usage

```
basinRunoffPlot(basinWaterBalance, cumul = FALSE)
```

Arguments

basinWaterBalance	Required. Data frame to be plotted. As read in by read_MESH_OutputTimeseries_csv.
cumul	Optional. If FALSE, then interval values are plotted. If TRUE (the default) then cumulative values are plotted. Note that the cumulative values are determined by summing the interval values *not* by plotting the MESH cumulative variables.

Value

Returns a **ggplot2** line plot of the variable values (mm).

Author(s)

Kevin Shook

See Also

read_MESH_OutputTimeseries_csv basinStoragePlot basinSoilWaterIcePlot

Examples

```
## Not run:
waterBalance <- read_MESH_OutputTimeseries_csv("Basin_average_water_balance.csv")
p <- basinRunoffPlot(waterBalance)
## End(Not run)
```

basinSnowPlot *Plots basin snow water equivalent*

Description

Plots basin snow water equivalent

Usage

```
basinSnowPlot(basinWaterBalance)
```

Arguments

basinWaterBalance
: Required. Data frame to be plotted. As read in by `read_MESH_OutputTimeseries_csv`.

Value

Returns a **ggplot2** stacked line plot of the basin SWE (mm).

Author(s)

Kevin Shook

See Also

`read_MESH_OutputTimeseries_csv` `basinStoragePlot`

Examples

```
## Not run:
waterBalance <- read_MESH_OutputTimeseries_csv("Basin_average_water_balance.csv")
p <- basinSnowPlot(waterBalance)
## End(Not run)
```

basinSnowRainPondedPlot *Plots basin snow and rain in the canopy and ponded water*

Description

Plots basin snow and rain in the canopy and ponded water

Usage

```
basinSnowRainPondedPlot(basinWaterBalance)
```

Arguments

basinWaterBalance
: Required. Data frame to be plotted. As read in by `read_MESH_OutputTimeseries_csv`.

Value

Returns a **ggplot2** line plot of the basin variables (mm).

Author(s)

Kevin Shook

See Also

`read_MESH_OutputTimeseries_csv` `basinStoragePlot`

Examples

```
## Not run:
waterBalance <- read_MESH_OutputTimeseries_csv("Basin_average_water_balance.csv")
p <- basinSnowRainPondedPlot(waterBalance)
## End(Not run)
```

---

basinSoilWaterIcePlot *Plots basin soil water and ice*

Description

Plots basin soil water and ice

Usage

```
basinSoilWaterIcePlot(basinWaterBalance, layers = c(1, 2))
```

Arguments

basinWaterBalance
: Required. Data frame to be plotted. As read in by `read_MESH_OutputTimeseries_csv`.

layers
: Optional. A vector of the layers to be plotted. By default layers 1 and 2 are used.

Value

Returns a **ggplot2** line plot of the variable values (mm) faceted by Layer number. The name of the variable is `layer`, so you can change change the facetting.

Author(s)

Kevin Shook

See Also

read_MESH_OutputTimeseries_csv basinStoragePlot

Examples

```
## Not run:
waterBalance <- read_MESH_OutputTimeseries_csv("Basin_average_water_balance.csv")
p <- basinSoilWaterIcePlot(waterBalance)
## End(Not run)
```

---

basinSoilWaterPlot *Plots basin soil liquid water*

Description

Plots basin soil liquid water

Usage

```
basinSoilWaterPlot(basinWaterBalance, layers = c(1, 2, 3, 4, 5, 6))
```

Arguments

basinWaterBalance	Required. Data frame to be plotted. As read in by read_MESH_OutputTimeseries_csv.
layers	Optional. A vector of the layers to be plotted. By default layers 1 though 6 are used.

Value

Returns a **ggplot2** line plot of the variable values (mm).

Author(s)

Kevin Shook

See Also

read_MESH_OutputTimeseries_csv basinStoragePlot basinSoilWaterIcePlot

Examples

```
## Not run:
waterBalance <- read_MESH_OutputTimeseries_csv("Basin_average_water_balance.csv")
p <- basinSoilWaterPlot(waterBalance)
## End(Not run)
```

basinStoragePlot *Plots total basin storage*

Description

Plots total basin storage

Usage

```
basinStoragePlot(basinWaterBalance)
```

Arguments

basinWaterBalance
: Required. Data frame to be plotted. As read in by read_MESH_OutputTimeseries_csv.

Value

Returns a **ggplot2** line plot of the value of STG (mm).

Author(s)

Kevin Shook

See Also

read_MESH_OutputTimeseries_csv basinStorageVariablesPlot basinSoilWaterIcePlot

Examples

```
## Not run:
waterBalance <- read_MESH_OutputTimeseries_csv("Basin_average_water_balance.csv")
p <- basinStoragePlot(waterBalance)
## End(Not run)
```

basinStorageVariablesPlot
Plots basin water balance storage components

Description

Creates a **ggplot2** stacked area plot of specified water balance storage components. The variables plotted may include:

variable definition

SNCAN Snow component of precipitation intercepted by the canopy

RCAN Rain component of precipitation intercepted by the canopy mm or kg m-2 of water AVG

SNO Snow water equivalent (SWE) of the snow mass

ZPND Depth of water ponded at the surface

LQWS Water equivalent of the volumetric liquid water content of the soil

FZWS Water equivalent of the volumetric frozen water content of the soil

ALWS Water equivalent of the volumetric liquid and frozen water contents of the soil, sum of LQWS and FZWS

Usage

```
basinStorageVariablesPlot(basinWaterBalance, varNames = "",
  layers = c(1, 2, 3, 4, 5, 6))
```

Arguments

basinWaterBalance	Required. Data frame to be plotted. As read in by `read_MESH_OutputTimeseries_csv`.
varNames	Optional. A vector of the names of the variables to be plotted. If not specified, all of the variables listed above will be plotted.
layers	Optional. A vector of the layers to be plotted. By default layers 1 through 6 are used.

Value

Returns a **ggplot2** stacked area time plot of the variable values (mm).

Author(s)

Kevin Shook

See Also

`read_MESH_OutputTimeseries_csv`

Examples

```
## Not run:
waterBalance <- read_MESH_OutputTimeseries_csv("Basin_average_water_balance.csv")
p <- basinStorageVariablesPlot(waterBalance)
# the plot can have a restricted date range
startDate <- as.Date("2005-10-01", format = "%Y-%m-%d")
```

```
endDate <- as.Date("2006-09-30", format = "%Y-%m-%d")
library(ggplot2)
p <- p + xlim(startDate, endDate)
p
# you can also change the colours used, either by
using a defined scale, or by manually specifying them
# This example uses the colours in the package viridis
# which scales from dark to light by reversing the direction
library(viridis)
p <- p +  scale_fill_viridis(discrete = TRUE, direction = -1)
p

## End(Not run)
```

`basinWaterBalancePlot` *Plots basin precipitation cumulative water balance.*

Description

As with the `basinRunoffPlot`, the cumulative values of precipitation, evaporation and runoff (and optionally, delta storage) are computed by the function, rather than by using the MESH variables. This allows the plot to be used on a sub-set of the basin output data.

Usage

```
basinWaterBalancePlot(basinWaterBalance, accumulate_delta_storage = TRUE)
```

Arguments

`basinWaterBalance`
: Required. Data frame to be plotted. As read in by `read_MESH_OutputTimeseries_csv`. Note that because the value of `DTSG` (delta storage) can be negative, you need to set a threshold value much smaller than zero when you read in the values.

`accumulate_delta_storage`
: Optional. If `TRUE` (the default), the delta storage is accumulated from the beginning of the data set. If `FALSE`, the delta storage values in the file are *not* accumulated as they are assumed to be cumualtive values.

Value

Returns a **ggplot2** line plot of the variable values (mm).

Author(s)

Kevin Shook

See Also

`read_MESH_OutputTimeseries_csv` `basinStoragePlot` `basinSoilWaterIcePlot`

Examples

```
## Not run:
waterBalance <- read_MESH_OutputTimeseries_csv("Basin_average_water_balance.csv",
missingValueThreshold = -1e6)
p <- basinWaterBalancePlot(waterBalance)
## End(Not run)
```

`distribPrecip` *Distributes precipitation in time*

Description

MESH requires all forcing data to have the same time interval, which prevents the use of precipitation data reported at lower frequencies than the model time step. This function distributes low-freqency precipitation (e.g. daily) according to a set of high frequency precipitation (e.g. hourly). The high-frequency values are summed to have the same time intervals as the low-frequency data. The ratios of low/high freqency precipitations are determined for each time step, and these ratios are multiplied by the high-frequency data. Where the high-frequency total precipitation is zero, the low-frequency data is spread evenly over the high-frequency interval.

Usage

```
distribPrecip(LFprecip = NULL, HFprecip = NULL,
  zero_missing_HF = TRUE, period_threshold = 48)
```

Arguments

`LFprecip`	Required. A data frame of low temporal frequency (e.g. daily) precipitation. The first column must be a POSIXct date/time called `datetime`. The second column must be the precipitation.
`HFprecip`	Required. A data frame of high temporal frequency (e.g. hourly) precipitation. The first column must be a POSIXct date/time called `datetime`. The second column must be the precipitation.
`zero_missing_HF`	Optional. If `TRUE` (the default) missing high-frequency precipitation values are replaced with zeros.
`period_threshold`	Length of maximum infilled period in hours. Sequences of missing low-frequency values exceeding this length will be set to `NA_real_`, and the next low-frequency value will be assigned to the next high frequency interval. This is usually required when a gauge is not reported for a seasonal period.

Value

Returns a data frame of the adjusted high-frequency precipitation, with the variables `datetime` and `distributedP`. Note that the last date/time in the returned data corresponds to the final value in the low-frequency data.

Author(s)

Kevin Shook

See Also

read_tb0

Examples

```
## Not run: distributed <- distribPrecip(myPrecip[, c(1, 5)],
adjacentP)

## End(Not run)
```

---

findRecord *Finds record in r2c data specified by string*

Description

Finds record in r2c data specified by string

Usage

```
findRecord(recordLines, string, ignore.case = TRUE)
```

Arguments

recordLines	Requred. Vector of lines from r2c file.
string	Required. Record name to searh for.
ignore.case	Optional. If TRUE (the default), then case is ignored.

Value

Returns trimmed record.

Author(s)

Kevin Shook

gridPrecip *Grids station precipitation*

Description

Grids station interval precipitation values, so that they can be used as MESH inputs. The gridding method is performed by the **hydroTSM** function `hydrokrige`, using the IDW (inverse distance weighting) algorithm.

Usage

```
gridPrecip(precip = NULL, shed_raster = NULL, IDW_file = NULL,
  quiet = TRUE)
```

Arguments

`precip`	Required. A list containing 3 elements: 1. the header meta data, 2. the column meta data, and 3. the precipitation values (in mm). These values are returned automatically by the **MESHr** command `read_tb0`.
`shed_raster`	Required. A `RasterBrick` object describing the MESH basin. This can be created using the **MESHr** command `read_r2c_shed` with the parameter `as_rasters = TRUE`.
`IDW_file`	Required. Output file which holds gridded precipitation for all time steps.
`quiet`	Optional. If `TRUE` (the default) messages are suppressed. If `FALSE`, the time interval and messages from each gridding are listed.

Value

If unsuccessful, returns `FALSE`. If successful, returns `TRUE` and the gridded precipitation values are written to the `IDW_file`. Note that each interval's precipitation is written as it is gridded. This saves on memory, but can be quite slow.

Examples

```
## Not run:
hourly_precip_file <- "Red_Deer_all_hourly_precip_new.tb0"
precip <- read_tb0(hourly_precip_file, values_only = FALSE, timezone = "Etc/GMT+7", NAvalue = -0.1)
shedfile <- "RedDeer_MESH_drainage_database.r2c"
shed_raster <- read_r2c_shed(shedfile, as_rasters = TRUE, values_only = TRUE)
IDW_file <- "RedDeerPrecip.idw"
gridPrecip(precip, shed_raster, IDW_file)
## End(Not run)
```

`hydroStats` *Calculates Goodness of Fit statistics for MESH output*

Description

This function is a wrapper for the function `gof` in the package **hydroGOF**. It computes several Goodness of Fit statistics for each station. The output of this function may be used on its own, or to annotate a hydrograph.

Usage

```
hydroStats(MESHvals, stationNames = "", calStart = "", calEnd = "",
  removeMissing = TRUE, doSpearman = FALSE, doPBFDC = FALSE,
  digits = 2, j = 1, norm = "sd", s = c(1, 1, 1),
  method = c("2009", "2012"), lQ.thr = 0.7, hQ.thr = 0.2)
```

Arguments

`MESHvals`	Required. A data frame of output from a MESH run, as produced by `readOutputTimeseriesCSV`.
`stationNames`	Optional. A vector of strings holding station names. If specified, the station names will be used in the returned data frame, otherwise the MESH station numbers will be used.
`calStart`	Optional. The start date of the calibration period. Must be a string in the format 'yyyy-mm-dd'. If specified, values on and after this date will be designated as the `Calibration` period. The remaining values will be designated as the `Validation` period.
`calEnd`	Optional. The start date of the calibration period. Must be a string in the format 'yyyy-mm-dd'. If specified, values on and after this date will be designated as the `Calibration` period. The remaining values will be designated as the `Validation` period.
`removeMissing`	Required. Should rows with missing values be removed before statistics are calculted? Default is `TRUE`
`doSpearman`	Optional. Should Spearman correlation be computed? Default is `FALSE`.
`doPBFDC`	Optional. Should percent bias of slope of the midsegment of the FDC be computed? Default is `FALSE`.
`digits`	Optional. The number of decimal places for rounding goodness of fit statistics. If `0` the default, then it is not set. Default value is `2`. Note that percentages like NRMSE and PBIAS will only use a maximum of one decimal place.
`j`	Optional. Argument passed to the `mNSE` function in **hydroGOF**.
`norm`	Optional. Argument passed to the `nrmse` function in **hydroGOF**.
`s`	Optional. Argument passed to the `KGE` function in **hydroGOF**.
`method`	Optional. Argument passed to the `KGE` function in **hydroGOF**.
`lQ.thr`	Optional. Argument passed to the `pbiassfdc` function in **hydroGOF**.
`hQ.thr`	Optional. Argument passed to the `pbiassfdc` function in **hydroGOF**.

Value

Returns a data frame with the following variables: #'

station station name or number

me Mean Error

mae Mean Absolute Error

mse Mean Squared Error

rmse Root Mean Square Error

nrmse Normalized Root Mean Square Error (-100% <= nrms <= 100%)

PBIAS Percent Bias

pbiasfdc PBIAS in the slope of the midsegment of the Flow Duration Curve, if selected

RSR Ratio of RMSE to the Standard Deviation of the Observations, RSR = rms / sd(obs). (0 <= RSR <= +Inf)

rSD Ratio of Standard Deviations, rSD = sd(sim) / sd(obs)

NSE Nash-Sutcliffe Efficiency (-Inf <= NSE <= 1)

mNSE Modified Nash-Sutcliffe Efficiency

rNSE Relative Nash-Sutcliffe Efficiency

d Index of Agreement (0 <= d <= 1)

d1 Modified Index of Agreement

rd Relative Index of Agreement

cp Persistence Index (0 <= PI <= 1)

r Pearson Correlation coefficient (-1 <= r <= 1)

r.Spearman Spearman Correlation coefficient (-1 <= r.Spearman <= 1), if selected

R2 Coefficient of Determination (0 <= R2 <= 1).
Gives the proportion of the variance of one variable that is predictable from the other variable

bR2 R2 multiplied by the coefficient of the regression line between `sim` and `obs`
(0 <= bR2 #' <= 1)

KGE Kling-Gupta efficiency between `sim` and `obs`
(0 <= KGE <= 1)

VE Volumetric efficiency between `sim` and `obs`
(-Inf <= VE <= 1)

If the calibration period is specified, then statistics will be computed separately for the Calibration and Validation periods. The period names and dates will be specified in additional columns.

Author(s)

Kevin Shook

See Also

`simpleHydrograph` `gof`

Examples

```
stats <- hydroStats(MESH_streamflows)
stats$NSE
periodStats <- hydroStats(MESH_streamflows, calEnd = "2010-01-01")
periodStats[,1:7]
```

`MESH_streamflows` *MESH streamflow data*

Description

A data frame containing MESH outputs for the Simonette river, as read in using the function `read_MESH_OutputTimeSeries_csv`.

Usage

```
MESH_streamflows
```

Format

A data frame with 4381 rows and 5 variables (including the datetime):

DATE date and time as an R date object

QOMEAS1 measured flows at station 1

QOSIM1 simulated flows at station 1

QOMEAS2 measured flows at station 2

QOSIM2 simulated flows at station 2

Source

This data iwas obtained by running MESH.

`parseNums` *Parses a string containing numbers*

Description

Parses a string containing numbers

Usage

```
parseNums(numString)
```

Arguments

numString	Required. A character string containing numbers separated by any number of spaces.

Value

Returns a numeric vector.

Author(s)

Kevin Shook

Examples

```
parseNums(' 1  2 3   4     5 ')
```

---

parseText *Parses a string containing several sub-strings*

Description

Parses a string containing several sub-strings

Usage

```
parseText(textString)
```

Arguments

textString	Required. A character string containing strings separated by any number of spaces.

Value

Returns a character vector.

Author(s)

Kevin Shook

Examples

```
parseText(' red  green    blue      black')
```

PBIAS *Compute bias as a percentage*

Description

Compute bias as a percentage

Usage

```
PBIAS(obs, sim)
```

Arguments

obs	Observed values as a numeric vector.
sim	Simulated values values as a numeric vector.

Value

If successful returns the percentage of bias as an integer. If unsuccessful, returns the value FALSE.

Author(s)

Muluneh A. Mekonnen

Examples

```
obs <- runif(100)
sim <- runif(100)
PBIAS(obs, sim)
```

plotDataCompleteness *Time plots of variable completeness*

Description

Time plots of variable completeness

Usage

```
plotDataCompleteness(MESHdata)
```

Arguments

MESHdata	Required. A data frame of MESH time series, as returned by read_tb0

Value

Returns a **ggplot2** wrapped facetted plot of data completeness (as a line) for each station. Missing periods are indicated by gaps in the line. The plot is facetted by the variable `station` - so you can change the facetting.

Author(s)

Kevin Shook

See Also

`read_tb0`

Examples

```
## Not run:
qvals <- read_tb0("MESH_input_streamflow.tb0", NAvalue = -0.01, values_only = TRUE)
p <- plotDataCompleteness(qvals)
# change facetting
# the y-axis is unimportant, so the plots can be squished vertically
library(ggplot2)
p <- p + facet_wrap(~station, ncol = 2)
## End(Not run)
```

---

read_AEP_csv *Reads csv file produced by Alberta Environment and Parks*

Description

Reads csv file produced by Alberta Environment and Parks

Usage

```
read_AEP_csv(AEPfile = "", timezone = "", values_only = TRUE)
```

Arguments

`AEPfile`	Required. Name of AEP file to be read in.
`timezone`	Optional. The name of the timezone of the data as a character string. If the `timezone` is not specified, your default value (i.e. your time zone) will be used. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. Under Windows or OSX, you can use '`etc/GMT+6`' or '`etc/GMT+7`' for Central Standard and Mountain Standard time. Under Linux you should use '`Etc/GMT+6`' or '`Etc/GMT+7`'.
`values_only`	optional. If `TRUE` (the default), only the time series values will be returned. If FALSE, the meta data will also be returned.

Value

Returns the time series data as a data frame, with the POSIXct variable `datetime` as the time stamp. Note that the time series interval may be irregular. If `values_only = TRUE`, then the returned value will be a list conisting of the time series data frame and the header meta data as a list, with the variables `values` and `header_meta`, respectively. The meta data are:

variable type

station_site character

station_name character

station_number character

parameter_name character

parameter_type character

parameter_type_name character

time_series_name character

time_series_unit character

longitude numeric

latitude numeric

Author(s)

Kevin Shook

See Also

`read_MESH_OutputTimeseries_csv`

Examples

```
## Not run:
precip <-  read_AEP_csv("05CA805 Skoki Lodge - PC - C.Merged - All.csv",
values_only = FALSE)
# show values
head(precip$values)
# show latitude
precip$header_meta$latitude
## End(Not run)
```

read_MESH_OutputTimeseries_csv *Reads MESH output .csv file containing timeseries*

Description

Reads a file containing any output from a MASH model into a standard R data frame. The names of the varables will be trimmed to remove leading and trailing spaces, and the time variables are combined into a single R date or datetime.

Usage

```
read_MESH_OutputTimeseries_csv(outputFile, timezone = "",
  missingValueThreshold = -0.1)
```

Arguments

outputFile	Required. Name of MESH output file. Must be a .csv file.
timezone	Not required for daily time series. Required for sub-daily time series. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. You can use 'etc/GMT+6' or 'etc/GMT+7' for Central Standard and Mountain Standard time, respectively. DO NOT use 'America/Regina' as the time zone, as it includes historical changes between standard and daylight savings time.
missingValueThreshold	Optional. Any value smaller than this value will be set to NA_real_ when the file is imported. The default value is -0.1 to prevent zero values from being affected.

Value

If successful, returns a data frame. The first columns will be called 'DATE' for daily values, and will contain a standard R date. For sub-daily timeseries the first column will be called 'DATETIME' and will contain a standard POSIXct date/time. If unsuccessful, returns the value FALSE.

Author(s)

Kevin Shook

See Also

simpleHydrograph

Examples

```
## Not run:
timezone <- 'etc/GMT+6'
outfile <- "Basin_average_water_balance_ts.csv"
output <- read_MESH_OutputTimeseries_csv(outfile, timezone)
## End(Not run)
```

`read_r2c_raster` *Reads r2c file to raster brick*

Description

This function reads a file containing a time series of 2D values, which is output from a MESH model. It is not intended to read in a file describing a drainage basin. For that purpose, you should be using the function `read_r2c_shed`. This function returns eiher a **raster** `brick` or an **rts** `rts` object, which is a timeseries raster. Each Frame in the original file becomes a separate layer. 'The name of each layer in the **raster** `brick` is set to the time stamp of each Frame. Because the layer names are standard R variables, they must obey the rules for variable names, inclusing beginning with a character, and not containing spaces. These rule will change the layer names if you are not careful.

Usage

```
read_r2c_raster(r2cFile, NAvalue = NULL, as_rts = FALSE,
  timezone = "", layerNameFormat = NULL)
```

Arguments

`r2cFile`	Required. Name of `r2c` file containing time series.
`NAvalue`	Optional. If specified, values smaller than `NAvalue` will be set to `NA_real_`
`as_rts`	Optional. If `TRUE`, the returned value will be a **rts** object, which allows the creation of 1-D time series, and for temporal aggregation. If `FALSE` (the default) a standard **raster** `brick` object is returned, which is better for simple plotting of the layers.
`timezone`	Optional. If the `r2cFile` contains date values for each Frame, then the Frame times are returned as R dates. If there are hours and seconds, then they will be converted to `POSIXct` datetime values. In this case, you may want to specify the `timezone` of the data. If the `timezone` is not specified, your default value will be used.
`layerNameFormat`	Optional. Sets the layer names when returning the **raster** `brick` to avoid conflicting with the R variable rules.

Value

Returns eiher a **raster** `brick` or an **rts** `rts` object.

Author(s)

Kevin Shook

See Also

rts read_r2c_shed

Examples

```
## Not run:
temps <- read_r2c_raster("TA_M.r2c", NAvalue = 0, as_rts = FALSE, layerNameFormat = "%b_%Y")
# convert air temps from K to C
temps <- temps - 273.15
plot(temps)
# create an animation and save it as a file
library(animation)
saveGIF(animate(temps, n = 1))
## End(Not run)
```

---

read_r2c_shed *Reads r2c file of a MESH watershed*

Description

This function reads in a file containing the layers which define a MESH watershed. To read in a file of timeseries, use the function r2c2raster.

Usage

```
read_r2c_shed(r2cFile = "", values_only = TRUE, as_rasters = TRUE)
```

Arguments

r2cFile	Required. Name of r2c file.
values_only	Optional. If TRUE (the default), then only the values are returned, either as a raster brick (useful for plotting) or as a 3D array (useful for analysis). If FALSE, then a *list* will be returned, containing the 1) the data, 2) the metadata (the variable names, types and units) are returned for each layer, and 3) the r2c file header lines.
as_rasters	Optional. If TRUE, the layers will be returned as as raster brick. If FALSE, they will be returned as an array.

Value

Returns eiher an array or a raster brick of values, and optionally, the meta data and file header.

Author(s)

Kevin Shook

See Also

read_r2c_raster

Examples

```
## Not run:
# read in basin as a raster brick
basin <-  r2c2basin("MESH_drainage_database.r2c")
# read in as an array
basin_array <- r2c2basin("MESH_drainage_database.r2c", as_rasters = FALSE)
# get meta data as well
basin_array <- r2c2basin("MESH_drainage_database.r2c", values_only = FALSE,
as_rasters = FALSE)

## End(Not run)
```

---

read_tb0 *Reads a MESH tb0 file*

Description

Reads in a tb0 file. The time series values, if present, will be stored in a data frame. Optionally, the meta data will be stored as lists. The meta data are of 2 types, the header values, which refer to the entire file and column values, which pertain to individual columns.

Usage

```
read_tb0(tb0File = "", values_only = TRUE, timezone = "",
  NAvalue = NULL)
```

Arguments

tb0File	Required. The name of the file to be read.
values_only	Optional. If TRUE (the default), only the time series values will be returned. If FALSE, the meta data will also be returned. Note that if the value is set to TRUE, and there are no time series values in the file, as for a reservoir file, then an error will result.
timezone	Optional. The data time series have POSIXct datetime values. You may want to specify the timezone of the data. If the timezone is not specified, your default value (i.e. your time zone) will be used.
NAvalue	Optional. If specified, values smaller than NAvalue will be set to NA_real_

Value

Returns time series as a data frame. If meta data are specified, they are returned as a list of header data and a data frame of column meta data. In this case all three sets of data are combined in a single

Author(s)

Kevin Shook

See Also

read_r2c_raster read_MESH_OutputTimeseries_csv

Examples

```
## Not run:
qvals <- read_tb0("MESH_input_streamflow.tb0", NAvalue = -0.01, values_only = TRUE)

## End(Not run)
```

---

simpleHydrograph *Creates hydrograph from MESH output*

Description

Creates a **ggplot** hydrograph from MESH output. This function *only* uses values from a single MESH data frame (as read in using `readOutputTimeseriesCSV`), so does not work with outside sources, such as WSC files. Because this function returns a **ggplot** object, you can change its format in any way you like. The plots produced may be faceted using the commands `facet_wrap` or `facet_grid`.

Usage

```
simpleHydrograph(MESHvals, stationNames = "", byStation = TRUE,
  byYear = FALSE, meas = TRUE, sim = TRUE, calStart = "",
  calEnd = "")
```

Arguments

`MESHvals`	Required. A data frame of output from a MESH run, as produced by `read_MESH_OutputTimeseries_csv`
`stationNames`	Optional. A vector of strings holding station names. If specified, the station names will be used in the plots. Otherwise the MESH station numbers will be used.
`byStation`	Optional. If `TRUE` (the default) then the plots will be coloured according to the station names. You may want to set this to `FALSE` if you are facetting by station name.

byYear	Optional. If TRUE then the plots will be able to be facetted by year. Note that this means that the dates are all plotted using the year 2000, so you will see strange results if you set this to TRUE and don't facet by year. Default is FALSE
meas	Optional. Should the measured values be plotted? Default is TRUE. If FALSE, they will be omitted.
sim	Optional. Should the simulated values be plotted? Default is TRUE. If FALSE, they will be omitted.
calStart	Optional. The start date of the calibration period. Must be a string in the format 'yyyy-mm-dd'. If specified, values on and after this date will be designated as the Calibration period. The remaining values will be designated as the Validation period.
calEnd	Optional. The start date of the calibration period. Must be a string in the format 'yyyy-mm-dd'. If specified, values on and after this date will be designated as the Calibration period. The remaining values will be designated as the Validation period.

Value

If successful, returns a **ggplot2** object. If unsuccessful, returns FALSE. The object can be facetted by the name of the station (the variable is called station). If the option byYear = TRUE, then the object can be facetted by the variable YEAR.

Note

Specifying the calibration start and/or end dates will allow the resulting plot to be facetted by the variable period.

Author(s)

Kevin Shook

See Also

read_MESH_OutputTimeseries_csv hydroStats

Examples

```
# plot hydrograph of all data on single graph
p1 <- simpleHydrograph(MESH_streamflows)
p1
# add station names, and replot
stations <- c("Station1", "Station2")
p2 <- simpleHydrograph(MESH_streamflows, stationNames = stations)
p2
# remove colouring by station, and facet, changing the axis label format
p3 <- simpleHydrograph(MESH_streamflows, stationNames = stations, byStation = FALSE)
# load in all of ggplot2 to modify plots
library(ggplot2)
p3 <- p3 + facet_wrap(~station, nrow = 2) + scale_x_date(date_labels = "%Y")
```

```
p3
# plot by year, then facet
p4 <- simpleHydrograph(MESH_streamflows, stationNames = stations, byYear = TRUE)
p4 <- p4 + facet_wrap(~YEAR, scales = "free_y")
p4
# remove colouring for stations, and facet by station and year
p5 <- simpleHydrograph(MESH_streamflows, stationNames = stations, byStation = FALSE, byYear = TRUE)
p5 <- p5 + facet_grid(YEAR~station, scales = "free_y")
# change colours
plotcols <- c("red", "blue")
p5 <- p5 + scale_colour_manual(values = plotcols)
p5
```

`var_present` *Find if variable is present in a data frame*

Description

Find if variable is present in a data frame

Usage

```
var_present(dataframe, variable)
```

Arguments

`dataframe`	Required. Dataframe to be checked
`variable`	Required. Variable name as a character string.

Value

Returns `TRUE` or `FALSE`

Examples

```
## Not run: var_present(values, "datetime")
```

win.eol *Gets the Windows end of line characters*

Description

Finds the end of line (eol) characters required for writing Windows files, such as CRHM obs files. No parameters are required. This is an internal **CRHMr** function and should *never* need to be called directly.

Usage

```
win.eol()
```

Value

Returns the Windows end of line characters (cr and lf).

Note

This function is used to make the creation of Windows-specific files work on all platforms. CRHM requires its obs and project files to use the Windows end of line characters, which are expressed differently on UNIX-based operating systems such as Linux and OSX.

Author(s)

Kevin Shook

Examples

```
windowsEndOfLine <- win.eol()
```

write_r2c_shed *Write MESH watershed data to r2c file*

Description

Write MESH watershed data to r2c file

Usage

```
write_r2c_shed(basin, header, r2cFile = "")
```

Arguments

basin	Required. The values defining the basin parameters, as read in by read_r2c_shed. Can either be and array or a raster brick.
header	Required. The header lines from the basin r2c file, as read in by read_r2c_shed
r2cFile	Required. The file to be written.

Value

If successful, returns TRUE. If unsucessful, returns FALSE

Author(s)

Kevin Shook

See Also

read_r2c_shed

Examples

```
## Not run:
# read in basin
shed <- read_r2c_shed("MESH_drainage_database.r2c", values_only = FALSE,
as_rasters = FALSE)
# write as another file
write_r2c_shed(shed$basin, shed$header_lines, "new_basin.r2c")

## End(Not run)
```

---

write_tb0 *Writes a MESH tb0 file*

Description

Writes a MESH tb0 file

Usage

```
write_tb0(values = NULL, column_meta = NULL, header = NULL,
  NAvalue = -1, tb0File = "")
```

Arguments

values	Optional. A data frame of the values to be written, if the file is to be a time seired. Note that the first column must be called datetime and must be a POSIXct date/time.
column_meta	Required. A data frame containing the following columns #' **columnUnits** required **columnType** required **columnName** required **columnLocationX** required **columnLocationY** required **DA** optional **coeff1** optional **coeff2** optional **coeff3** optional **coeff4** optional **coeff5** optional
header	Required. A list containing the following variables **filetype** optional, default is tb0 ASCII EnSim 1.0 **datatype** optional, default is Time Series **application** optional, default is EnSimHydrologic **version** optional, default is default is 2.1.23 **written_by** optional, default is default is MESHr **creation_date** optional, default is current date/time **source_file** optional, default is nothing **name** required **projection** required, character string **ellipsoid** required, character string **start_time** required if values are not specified **delta_t** required if values are not specified **attributeunits** optional, default is nothing **unitconversion** optional, default is nothing
NAvalue	Optional. Value to be used for NA_real_ values in the .tb0 file. The default value is -1, which is not suitable for air temperatures.
tb0File	Required. Name of file to be written.

Value

If successful, returns TRUE. If unsuccessful, returns FALSE

Author(s)

Kevin Shook

See Also

read_tb0

Examples

```
## Not run:  write_tb0(values, column_meta_data, header, "MESH_values.ts0")
```

Index

∗Topic **datasets**
MESH_streamflows, 16

basinLapseRates, 2
basinPrecipEvapRunoffPlot, 3
basinRunoffPlot, 4
basinSnowPlot, 5
basinSnowRainPondedPlot, 5
basinSoilWaterIcePlot, *4*, 6, *7, 8*, *10*
basinSoilWaterPlot, 7
basinStoragePlot, *4–7*, 8, *10*
basinStorageVariablesPlot, *8*, 8
basinWaterBalancePlot, 10

distribPrecip, 11

findRecord, 12

gof, *15*
gridPrecip, 13

hydroStats, 14, *26*

MESH_streamflows, 16
MESHr-package, 2

parseNums, 16
parseText, 17
PBIAS, 18
plotDataCompleteness, 18

read_AEP_csv, 19
read_MESH_OutputTimeseries_csv, *4–10*, *20*, 21, *25, 26*
read_r2c_raster, 22, *24, 25*
read_r2c_shed, *23*, 23, *29*
read_tb0, *12*, *19*, 24, *31*
rts, *23*

simpleHydrograph, *15*, *21*, 25

var_present, 27

win.eol, 28
write_r2c_shed, 28
write_tb0, 29