Package 'wasim'

February 20, 2015

Type Package

Title Visualisation and analysis of output files of the hydrological model WASIM

Version 1.1.2

Date 2011-12-27

Author Dominik Reusser, Till Francke

Maintainer Dominik Reusser <reusser@pik-potsdam.de>

Description
Helpful tools for data processing and visualisation of results of the hydrological model WASIM-ETH.

Depends R (>= 1.8.0), MASS, qualV, tiger, fast

License GPL-2

Repository CRAN

Date/Publication 2011-12-30 11:23:17

NeedsCompilation no

R topics documented:

`wasim-package` *Visualisation and analysis of output files of the hydrological model WASIM*

Description

Visualisation and analysis of output files of the hydrological model WASIM

Details

Package:	wasim
Type:	Package
Version:	1.0
Date:	2008-12-16
License:	GPL-2

Author(s)

Dominik Reusser

Maintainer: Dominik Reusser <dreusser@uni-potsdam.de>

Examples

```
  unzip(system.file("weisseritz.zip", package="wasim"))

d.datum <- read.dates(file= "weisseritz/qgesww100.txt")  # Datumsreihe heisst jetzt d.datum
d.wasim.out <- read.results("", "weisseritz",ts.length=21937,subcatchments=2, ending="ww100.txt")    # Datensatz
d.meas.all<-read.table("weisseritz/ww100-rh.txt",header=TRUE, na.strings ="999", skip=4)  # gemessene Daten einl
d.meas<-d.meas.all$Ammelsdorf[69716:91652]
p.storage(d.wasim.out, xdata=d.datum, measured=d.meas)

sd.grid <- read.grid( file= "weisseritz/sd__ww100.grd")  # Datumsreihe heisst jetzt d.datum
p.grid(sd.grid,
        color.palette=topo.colors,
```

```
      zlim=c(-20,20),
        main="Saturation deficit at the Weisseritz")

#for english x-lab in Germany/Switzerland:
Sys.setlocale(category = "LC_ALL", locale = "en")
```

`assemble.date` *Internal function used in read.dates to convert the date format to POSIX*

Description

Internal function

Usage

```
assemble.date(table)
```

Arguments

`table`	object read with read.table

Value

A list with POSIX values

Author(s)

Dominik Reusser

`bias` *Calculate objective functions between two vectors*

Description

The function calculates the bias between two time series

Usage

```
bias(a, b, cond = rep(TRUE, NROW(a)))
rmse(a, b, cond = rep(TRUE, NROW(a)))
mean_error(a, b, cond = rep(TRUE, NROW(a)))
```

Arguments

a	First vector
b	Second vector
cond	Vector of boolean, indicating which cases to exclude

Value

The method describes a scalar the bias between the vectors

Author(s)

Dominik Reusser

---

color.factor *Create colors with intensity according to the magnitude of a value*

Description

Create colors with intensity according to the magnitude of a value

Usage

```
color.factor(color, value, max)
```

Arguments

color	The base color(s) to use
value	A vector of values
max	The maximum value represented by full intensity

Value

A vector of colors, one entry for each value

Author(s)

Dominik Reusser

Examples

```
data <- 1:10
cols=color.factor("red", data, max=10)
plot(data, col=cols)


cols=color.factor(c("red","green","blue"), data, max=10)
plot(data, col=cols)
```

convert_q_units *Convert discharge units*

Description

converts discharge data of a given file (infile) in WASIM-format into output file (outfile) converting discharge from [m3/s] to [mm] (convert_to="specific") or vice versa (convert_to="absolute") using the file gauges_area_file containing the catchment areas and a timestep of timestep_sec seconds

Usage

```
convert_q_units (infile, outfile, gauges_area_file, convert_to, timestep_sec)
```

Arguments

infile	name of file holding discharge data in WASIM format
outfile	name of file to produce containing the converted discharge
gauges_area_file	name of file containing tab-separated data (GAUGE FOREIGN_ID SUBBAS_ID AREA_SUBBAS_KM2 AREA_UPSTREAM_KM2)
convert_to	"specific": convert m3/s to mm; "absolute": convert mm to m3/s
timestep_sec	length of timestep in inout file in seconds

Value

no return value

Author(s)

Till Francke

See Also

read.results

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.
```

data.types — *Types of data in wasim output files and the files prefix*

Description

~ still to write ~

Usage

```
data.types
```

Format

data.frame

diff_diff — *Count time steps with wrong direction in dynamics*

Description

`diff_diff` counts the time steps for two time series during which the first shows an increase and the second shows a decrease.

Usage

```
diff_diff(x, y)
```

Arguments

`x`	Modelled time series or array with dimension `c(number_series, dim(y))`
`y`	Measured time series

Value

The count

Author(s)

Dominik Reusser

Examples

```
data(models_dawson,package="wasim")
diff_diff(models_dawson[,1],models_dawson[,2])

do.call(rbind,lapply(models_dawson, FUN =diff_diff, y=models_dawson[,1]))
```

example.peaks *Synthetic peak errors*

Description

A number of synthetic peak errors used for testing performance measures and similar

Usage

```
data(example.peaks)
```

Format

The format for example.peaks is: num [1:12, 1:91] 0.1346 0.1346 0.1846 0.0846 0.1346 ...

The format for the reference.peak is: num [1:91] 0.135 0.134 0.134 0.134 0.134 ...

Examples

```
data(example.peaks)
str(example.peaks)
str(reference.peak)
plot(reference.peak,type="line")
lines(example.peaks[,1], lty=2)
require(tiger)
diagnostic_dawson(measured = reference.peak, modelled = example.peaks[1,])
## maybe str(peaks) ; plot(peaks) ...
```

extract_wasim_params *Extract Parameters from WaSiM-Control-File*

Description

retrieves desired parameters (currently limited to soil_model and snow_model) from WASIM control file

Usage

```
extract_wasim_params(wasim_param_file,param_list)
```

Arguments

wasim_param_file	Path to WASIM control file (see details).
param_list	List of parameters to be extracted from the file. Currently supported:(all lowercase) m,tkorr,kkorr,kd,hmax,kh,cmelt,t0r,t0,c0

Details

The ordinary WASIM control file (*.ctrl) may be used as wasim_param_file. However, if this files uses internal variables (starting with \$), it is more safe to use the file *.\$\$\$ instead (produced after calling WASIM).

Value

Dataframe params contains the fields specified in param_list and their respective values.

Author(s)

Till Francke

---

f.score *Calculate objective functions for sensitive parts of a time series*

Description

Calculate objective functions for sensitive parts of a time series. ToDo: write better description.

Usage

```
f.score(data, sens, measured, parameter, criterion, limit)
```

Arguments

data	a matrix of modeled time series
sens	result from sensitivity
measured	measured time series
parameter	vector of indizes, indicating which parameters to include in the analysis
criterion	1: rmse, 2: Nash-Sutcliffe coeffiecient of efficiency, 3: Nash-Sutcliffe coeffiecient of efficiency with more weight on high flows
limit	limit, above which the sensitivity must be in order for the data to be included in the analysis.

Value

A vector of results of the objective function, one for each time series

Author(s)

Dominik Reusser

Examples

```
#ToDo: Write an example
```

highlight *Highlight an area in a plot*

Description

Draws a number of colored rectangles into a plot.

Usage

```
highlight(x1,x2,yrange,col="yellow")
```

Arguments

x1	A vector of x values from which to start rectangles
x2	A vector of x values at which to end rectangles
yrange	yrange the rectangel covers
col	Drawing color

Value

Used for its side functions in a plot.

Author(s)

Dominik Reusser

k_hyd *Hydrological recession constant*

Description

This function calculates the local hydrological recession constant for each point in a time series. The function returns NA for periods with increasing discharge.

Usage

```
k_hyd(x)
```

Arguments

x	discharge time serie

Value

Vector of recession constants.

Author(s)

Dominik Reusser

References

Blume Recession Paper

Examples

```
data(example.peaks,package="wasim")

k_hyd(reference.peak)
```

`line_sumrain` *Draw line of cumulative rainfall*

Description

Add a line with cumulative rainfall data to a plot. The maximum value of the plot has to be passed. If the cumulative rainfall exceeds this maximum, it will start again at 0.

Usage

```
line_sumrain(xdata, cum_sum_rain, theMax, ...)
```

Arguments

`xdata`	Values ploted on the x axis, usually read with `read.dates`
`cum_sum_rain`	Vector with the cumulative rain
`theMax`	The maximum of the y-axis
`...`	additional arguments passed to `lines`

Value

Used for its side effect

Author(s)

Dominik Reusser

See Also

See also `p.flow_comp`

Examples

```
#Uniform rain
plot(1:100, rep(1, 100),ylim=c(10,0),type="h", xlab="time", ylab="rainfall")
line_sumrain(xdata=1:100, cum_sum_rain=(1:100), theMax=10)
```

`max_rel_diff` *Calculate the maximum ration of the derivatives of two time series*

Description

Calculate the maximum ration of the derivatives of two time series

Usage

```
max_rel_diff(x, y)
```

Arguments

`x`	Modelled time series or array with dimension `c(number_series, dim(y))`
`y`	Measured time series

Value

```
max(diff(x) / diff(y), na.rm=TRUE)
```

Author(s)

Dominik Reusser

Examples

```
data(models_dawson,package="wasim")
max_rel_diff(models_dawson[,1],models_dawson[,2])

do.call(rbind,lapply(models_dawson, FUN =max_rel_diff, y=models_dawson[,1]))
```

`models_dawson` *Synthetic peak errors from Dawson 2007*

Description

A number of synthetic peak errors used for testing performance measures and similar

Usage

```
data(models_dawson)
```

Format

A data frame with 160 observations on the following 5 variables.

`Observed` a numeric vector

`Model.A...Naive` a numeric vector

`Model.B...Low.Flow` a numeric vector

`Model.C...Noisy` a numeric vector

`Model.D...High.Flow` a numeric vector

Source

https://co-public.lboro.ac.uk/cocwd/HydroTest/index.html

References

Dawson, C. W.; Abrahart, R. J. & See, L. M. HydroTest: A web-based toolbox of evaluation metrics for the standardised assessment of hydrological forecasts Environmental Modelling & Software, 2007 , 22 , 1034-1052

Examples

```
data(models_dawson)
plot(models_dawson[,1])
lines(models_dawson[,2])
diagnostic_dawson(modelled=models_dawson[,2], measured=models_dawson[,1], use_qualV=TRUE)
do.call(rbind,lapply(models_dawson,FUN=diagnostic_dawson, measured=models_dawson[,1]))
```

myRead.table — *Read a wasim result file into a table with given length*

Description

This function reads an output-txt File from WaSiM. It is usually called from other functions. Standard colum names are generated. Date and time are removed from the data object (see read.dates). Finaly, the time series are expanded with NA to fill a certain length (If simulations stoped to early).

Usage

```
myRead.table(name, subcatchments, has_stat, ts.length, na.values)
```

Arguments

name	File name to read
ts.length	Number of expected data points for the time series. The resulting table is guaranteed to have this length
subcatchments	Number of subcatchments simulated
has_stat	boolean indicating wheter statistics column exists
na.values	String which identifies NA

Value

A table of dimension dim=c(subcatchments+1,ts.length). The last column contains the statistics column if existing.

Author(s)

Dominik Reusser

p.grid — *Read and plot a grid-file output by WaSiM*

Description

These two functions help to read and plot grid-files created by WaSiM. You need the gridasci-executable if to convert binary grid data to ascii grid data.

Usage

```
p.grid(grid, ...)
read.grid(file, record_size=4, endian="little")
```

Arguments

grid	A grid read with the read.grid function
...	Additional arguments passed to plot
file	WASIM grid output file in ASCII or binary format.
record_size	number of bytes used to represent a float value in the binary grid. Usually 4, but may be platform-dependent.
endian	see readBin

Details

The file is opened as a binary grid unless the infile has one of the following extensions: *.asc, *.ascii, *.txt. Nodata values are represented by NAs.

Value

read.grid returns a list:

head	dataframe of header information containing the fields ncols, nrows, xllcorner, yllcorner, cellsize, nodata_value.
tab	The actual grid

Author(s)

Dominik Reusser, Till Francke

References

WaSiM-ETH manual

See Also

read.results, p.flow_comp for more WaSiM related function. filled.contour for plotting options

Examples

```
unzip(system.file("weisseritz.zip", package="wasim"))
sd.grid <- read.grid( file= "weisseritz/sd__ww100.grd")  # Datumsreihe heisst jetzt d.datum
p.grid(sd.grid,
      color.palette=topo.colors,
    zlim=c(-20,20),
      main="Saturation deficit at the Weisseritz")
```

p.storage *Plot flow components / storage fillage / balance (integrals) of WaSiM model run*

Description

For each component listed in storage, the time series for this component is plotted in a separate window together with the measured discharge.

Usage

```
p.flow_comp(data, xdata = 1:NROW(data), measured = NULL,
                  data.names = data.types$beschreibung_en[csubset],
                  csubset = c(10, 29, 6, 8, 9, 5, 11), crain = 7, ylab =
                  "flow [mm/h]", c.flow_com = NULL, l.flow_com = NULL,
                  interflow_correction = TRUE, interflow_row = 11,
                  baseflow_row = 8, legend.position = "right", ...)
p.storage(data, xdata, measured, data.names =
                  data.types$beschreibung_en, storage = c(18, 20, 22,
                  24, 26), catchment = 1, mfrow = c(2, 3), ...)
p.balance(data, xdata, measured, plot = TRUE, flows = c(7, 10,
                  3, 1), storage = c(18, 20, 22, 24, 26), catchment = 1,
                  xlab = "time", ...)
```

Arguments

data	Data as read with read.results
xdata	Values ploted on the x axis, usually read with read.dates
measured	Observed flow data, ploted as reference. Can be read with read.observations
data.names	Names of data series used for legends and y-axes
storage	Vector or integers, indicating which data series to use as storages
plot	Boolean, indicating whether to plot the integrated flow components
csubset	vector of indices indicating which components to plot
crain	index of the column in data which carries rain information
interflow_correction	Boolean indicating whether baseflow should be subtracted from interflow (because WaSiM reports the sum of the two as interflow)
interflow_row	index of the column in data which contains interflow information
baseflow_row	index of the column in data which contains baseflow information
ylab	label to use for y-axis
xlab	label to use for x-axis
mfrow	mfrow parameter for plots

legend.position	position parameter for legend command
c.flow_com	vector of colors to use for the different flow components
l.flow_com	vector of line widths to use for the different flow components
catchment	Which subcatchment should be ploted (column index from data)
flows	numeric vector indicating which flow components to plot (indices from data)
...	Additional arguments passed to plot

Value

Used for its function of creating a plot

Author(s)

Dominik Reusser

Examples

```
  unzip(system.file("weisseritz.zip", package="wasim"))
d.datum <- read.dates(file= "weisseritz/qgesww100.txt")  # Datumsreihe heisst jetzt d.datum
d.wasim.out <- read.results("", "weisseritz",ts.length=21937,subcatchments=2, ending="ww100.txt")   # Datensatz
d.meas.all<-read.table("weisseritz/ww100-rh.txt",header=TRUE, na.strings ="999", skip=4)  # gemessene Daten einl
d.meas<-d.meas.all$Ammelsdorf[69716:91652]
p.storage(d.wasim.out, xdata=d.datum, measured=d.meas)
```

---

read.results *read.results*

Description

Import WASIM simulation results

Usage

```
read.results(record, path = "wasim/output",
                  ts.length, subcatchments, data.types.prefix =
                  data.types$prefix, data.types.has_stat =
                  data.types$has_stat, ending, endings = rep(ending,
                  NROW(data.types.prefix)), na.values = c("999",
                  "999.00"), read.files.nr = 1:nrow(data.types),
                  remove.missing = FALSE)

read.dates(file=NULL,na.strings=c("999", "999.00","-9999.00"),sep="\t", skip=3)

read.observations(filelist, path = "wasim/input", gauge_names = NULL,
                  date_modelled = NULL, na.strings = c("999", "999.00",
                  "9999", "9999.00", "9999.000000", "-9999", "-9999.0",
                  "-9999.00"), sep = "")
```

Arguments

file	WASIM output file the date information is read from. If not specified, the function tries to open the first available file from the globaldata.types\$filename.
record	subdirectory/ string to be appended to path (eg. for multiple runs, etc.)
path	path the output files reside in (no ending slash)
ts.length	number of records to be read
subcatchments	number of subcatchments (columns) in WASIM-output file (without statistic column)
data.types.prefix	prefixes for filenames (better use default)
data.types.has_stat	indicator whether files contain an extra column for summary statistics
ending	suffix for filenames
endings	suffixes for filenames. Use this if endings differ for various files.
read.files.nr	numeric vector indicating which entries from data.types.prefix to process
na.values	list of strings for indicating nodata values in the WASIM-files
na.strings	Strings in input file to be interpreted as nodata (NA)
sep	field separator in input file
remove.missing	boolean indication whether to remove an entry from the default data.types if a file was not found
skip	number of lines to skip from the header
filelist	list of input files to be considered
gauge_names	names of gauges that should be read (NULL:read all)
date_modelled	timespan that will be extracted

Details

read.results causes error messages: "...line n hasn't m elements..." The global variable data.types\$has_stat specifies an statistics column in one of the output files where there is none.

"...has length 14 expecting length 15 - filling ..." The global variable data.types\$has_stat specifies no statistics column in one of the output files where there seems to be one.

Value

date_vec	vector of POSIX-dates

Author(s)

Dominik Reusser, Till Francke

See Also

p.storage

Examples

```
  unzip(system.file("weisseritz.zip", package="wasim"))
d.datum <- read.dates(file= "weisseritz/qgesww100.txt")  # Datumsreihe heisst jetzt d.datum
str(d.datum)

d.wasim.out <- read.results("", "weisseritz",ts.length=21937,subcatchments=2, ending="ww100.txt")   # Datensatz

str(d.wasim.out)
```

`rerange` *Linear transformation of data*

Description

The function performs a linear transformation of the data, such that afterwards range(data)=c(theMin,theMax).

Usage

```
rerange(data, min.goal = 0, max.goal = 1, min.data =
                 min(data), max.data = max(data), center = NA)
```

Arguments

`data`	vector with the `data` to transform
`min.goal`	new minimum value
`max.goal`	new maximum value
`min.data`	old minimum value
`max.data`	old maximum value
`center`	which old value should become the new center ((max.goal + min.goal) / 2)

Value

vector with the transformed data

Author(s)

Dominik Reusser

Examples

```
rerange(data=1:20)
rerange(data=1:30, center=5)
```

span *Calculate the span/range of a time series*

Description

This function has been implemented for a simpler call of apply.

Usage

```
span(x)
```

Arguments

x	A vector

Value

```
diff(range(x,na.rm=TRUE))
```

Author(s)

Dominik Reusser

Examples

```
data(models_dawson,package="wasim")
span(models_dawson[,1])
```

Index