Package 'dynatopmodel'

January 19, 2018

Type Package

Title Implementation of the Dynamic TOPMODEL Hydrological Model

Version 1.2.1

Date 2018-01-19

Author Peter Metcalfe, Keith Beven and Jim Freer

Maintainer Peter Metcalfe <p.metcalfe@lancaster.ac.uk>

Description A native R implementation and enhancement of the Dynamic TOPMODEL semi-distributed hydrological model. Includes some preprocessing, utility and routines for displaying outputs.

Depends R (>= 3.1),

Imports deSolve, maptools, rgdal, rgeos, zoo, xts, sp, raster, lubridate, topmodel, methods, grDevices, stats, utils, graphics, tools

License GPL-2

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-01-19 14:37:01 UTC

R topics documented:

aggregate_obs 2
aggregate_xts 3
approx.pe.ts 4
brompton 5
build_chans 6
build_layers 8
build_routing_table 9
dev.reset 10
discretise 11

disp_output 13
dynatopmodel 14
get.disp.par 16
get.run.par 17
Mode 17
new_guid 18
NSE 18
run.dtm 19
time_at_peak 22
time_to_peak 23
upslope.area 24

Index **26**

`aggregate_obs` *Resample observation data at a new time interval*

Description

Takes a list of time series and resample to a new interval.

Usage

```
aggregate_obs(obs, dt, is.rate = TRUE)
```

Arguments

`obs`	List of times series (zoo) objects with a POSIXct index.
`dt`	New time interval, hours.
`is.rate`	If TRUE then these are rates i.e m/hr. Otherwise they are absolute values across the interval and are scaled before return by a factor equal to the ratio of the old interval to the new interval.

Details

Time series of observation data are often of different temporal resolutions, however the input to most hydrological models, as is the case with the Dynamic TOPMODEL, requires those data at the same interval. This provides a method to resample a collection of such data to a single interval.

Value

The list of observations resampled at the new interval.

Examples

```
# Resample Brompton rainfall and PE data to 15 minute intervals
require(dynatopmodel)
data("brompton")

obs <- aggregate_obs(list("rain"=brompton$rain, "pe"=brompton$pe), dt=15/60)

# check totals for Sept - Oct 2012
sum(obs$rain*15/60, na.rm=TRUE)
sum(brompton$rain, na.rm=TRUE)
```

`aggregate_xts` *Resample observation data at a new time interval*

Description

Takes a list of time series and resample to a new interval.

Usage

```
aggregate_xts(ser, dt, fun = mean)
```

Arguments

`ser`	xts Times series to aggregate
`dt`	numeric New time interval, hours
`fun`	function Function applied to aggregate the series to the new interval

Details

Time series of observation data are often of different temporal resolutions, however the input to most hydrological models, as is the case with the Dynamic TOPMODEL, requires those data at the same interval. This provides a method to resample a collection of such data to a single interval.

Value

Time series resampled at the new interval #' @return The list of observations resampled at the new interval.

Examples

```
# Resample Brompton rainfall and PE data to 15 minute intervals
require(dynatopmodel)
data("brompton")

rain <- aggregate_xts(brompton$rain, dt=15/60)
```

`approx.pe.ts` *Create sinsuiodal time series of potential evapotranspiration input*

Description

Create sinsuiodal time series of potential evapotranspiration input

Usage

```
approx.pe.ts(start, end, dt = 1, emin = 0, emax = 5/1000)
```

Arguments

`start`	Start time of returned series in a format that can be coerced into a POSIXct instance. Defaults to start of rainfall data
`end`	End time for returned series in a format that can be coerced into a POSIXct instance. Defaults to end of rainfall data.
`dt`	Time interval in hours
`emin`	Minimum daily PE total (m or mm)
`emax`	Maximum daily PE total (m or mm)

Details

Dynamic TOPMODEL requires a time series of potential evapotranspiration in order to calculate and remove actual evapotranspiration from the root zone during a run. Many sophisticated physical models have been developed for estimating PE and AE, including the Priestly-Taylor (Priestley and Taylor, 1972) and Penman-Monteith (Montieth, 1965) methods. These, however, require detailed meteorological data such as radiation input and relative humidities that are, in general, difficult to obtain. Calder (1983) demonstrated that a simple approximation using a sinusoidal variation in potential evapotranspiration to be a good approximation to more complex schemes.

If the insolation is also taken to vary sinusoidally through the daylight hours then, ignoring diurnal meteorological variations, the potential evapotranspiration during daylight hours for each year day number can be calculated (for the catchment's latitude). Integration over the daylight hours allows the daily maximum to be calculated and thus a sub-daily series generated.

Value

Time series (xts) of potential evapotranspiration ([L]/[T]) covering the given time range and at the desired interval in m or mm/hr

References

Beven, K. J. (2012). Rainfall-runoff modelling : the primer. Chichester, UK, Wiley-Blackwell.

Calder, I. R. (1986). A stochastic model of rainfall interception. Journal of Hydrology, 89(1), 65-71.

Examples

```
## Not run:
# Create PE data for 2012 for use in the Brompton test case

require(dynatopmodel)

data("brompton")

# Generate time series at hourly and 15 minute intervals
pe.60 <- approx.pe.ts("2012-01-01", "2012-12-31", dt=1)
pe.15 <- approx.pe.ts("2012-01-01", "2012-12-31", dt=0.25)

# Check annual totals - should be around 900mm
sum(pe.60)*1000
sum(pe.15*0.25)*1000

# Check maximum daily total on the 1st of July
sum(pe.60["2012-07-01"])*1000
sum(pe.15["2012-07-01"]*0.25)*1000

## End(Not run)
```

---

brompton *Topographic and observation data for running Dynamic TOPMODEL.*

Description

Brompton is a small agricultural catchment in N.Yorkshire, UK. Its eastern edges rise in the North Yorks Moors and it drains southwards, becoming North Beck before joining the Wiske in Northallerton.

In the late 19th century the area upstream of Water End was drained and turned over to arable cultivation and has since suffered from infrequent, but severe, flooding due to intense rainfall from weather systems moving in from the North Sea.The last event hat flooded the village was in November 2012; flooding was narrowly avoided in the storms of December 2015 and in a convective event in July 2017.

The catchment exhibits high land-channel connectivity due to heavily-modified natural channels and extensive artificial drainage, both surface and subsurface. It has a homogenous land cover, with almost all its area comprising class 1 arable grassland and crops. The terrain is undulating with slightly acid, base-rich loam and clay soils predominating. Distances from the channel appear to exert most influence over the catchment response. It is hypothesised that this is due to the influence of the field drainage.

Usage

```
data(brompton)
```

Format

An environment comprising the DEM, river network, observed flows from stage data reconstructed at the EA gauge at Water End, and hydrometric (rainfall and pe) necessary to run the model.

References

Metcalfe P. (2016). Case study 2. Brompton runoff attenuation modelling. In Hankin, B., Burgess-Gamble, L., Bentley, S., Rose, S. (Eds.). How to model and map catchment processes when flood risk management planning. Science report SC120015/R1, Environment Agency, Bristol, UK.

Metcalfe, P., Beven, K., Hankin, B., & Lamb, R. (2017). A modelling framework for evaluation of the hydrological impacts of nature-based approaches to flood risk management, with application to in-channel interventions across a 29 km^2 scale catchment in the United Kingdom. Hydrological Processes, 31(9), 1734-1748.

See Also

discretise

run.dtm

Examples

```
require(dynatopmodel)
data(brompton)

require(raster)

# Show it
sp::plot(brompton$dem)
```

---

build_chans *Construct a raster of channel locations from vector or topographic data*

Description

The discretise methods both requires a raster defining the locations of the channel cells and the proportion of each river cell occupied by the channel. A detailed river network (DRN) may be available in vector format and can be used to compute this. If not, the channel location can be inferred from a spatially-distributed metric, typically the topographic wetness index.

Usage

```
build_chans(dem, drn, chan.width = 1, atb = NULL, buffer = 10,
  atb.thresh = 0.8, single.chan = TRUE)
```

Arguments

dem	raster Elevation raster (DEM) using a projected coordinate system (e.g UTM) and regular grid spacing. Not required if atb raster supplied.
drn	SpatialLines Detailed river network (DRN) in vector (ESRI Shapefile) format. Not required if atb raster supplied.
chan.width	numeric Vector of channel widths, in m, for each reach defined in the DRN. Will be recycled if shorter than the number of channels
atb	raster Optional raster used as criteria for locating the channel. Typically the value of the topographic wetness index (TWI) determined from the elevations. Should be in a projected coordinate system (e.g UTM) and use a regular grid spacing. For the TWI to be meaningful this raster should have a resolution of a least 30m. It can be calculated using the upslope.area method applied to the DEM and atb=TRUE.
buffer	If using a vector input then buffer the DRN by this width to capture all river cells.
atb.thresh	If atb supplied and DRN null then this specifies the threshold value above which cells are identified as containing part of the channel network
single.chan	If using a vector input then the first raster layer returned contains either 1 for a river cell or NA for a non-river cell. Otherwise the values are the line ids of the channel vector, that typically identify individual reaches

Value

A two-band raster with the same dimensions as the elevation or ATB raster whose first layer comprises non-zero cells where identified with the channel and whose second layer holds the proportions of those cells occupied by the channel.

References

Kirkby, M. (1975). Hydrograph modelling strategies. In Peel, R., Chisholm, Michael, Haggett, Peter, & University of Bristol. Department of Geography. (Eds.). Processes in physical and human geography : Bristol essays. pp. 69-90. London: Heinemann Educational.

Examples

```
## Not run:

require(dynatopmodel)
data("brompton")

chan.rast <- build_chans(dem=brompton$dem, drn=brompton$drn, buff=5, chan.width=2)
# show it
sp::plot(chan.rast[[1]], col="green", legend=FALSE)

## End(Not run)
```

build_layers *Construct basic landscape layer data for Dynamic TOPMODEL run*

Description

Given an elevation raster this function will create a basic multi-band raster that can be used to run Dynamic TOPMODEL after applying a suitable discretisation. It comprises the supplied elevations with the addition of upslope contributing area and topographic wetness index (TWI).

Usage

```
build_layers(dem, fill.sinks = TRUE, deg = 0.1)
```

Arguments

dem	Elevation raster using a projected coordinate system (e.g UTM) and regular grid spacing. Should have a resolution of a least 30m for the TWI to be meaningful.
fill.sinks	If TRUE (default) then run a sinkfill before calculating the upslope area and TWI.
deg	Threshold intercell slope to determine sinks (degrees).

Value

A multi-band raster (stack) comprising, in order, the elevations, upslope area and topographic wetness index values.

Author(s)

Peter Metcalfe

Examples

```
## Not run:
require(dynatopmodel)
data("brompton")

# Upslope area and wetness index for Brompton catchment
layers <- build_layers(brompton$dem)

sp::plot(layers, main=c("Elevation AMSL (m)", "Upslope area (log(m^2/m))", "TWI ((log(m^2/m))"))

## End(Not run)
```

`build_routing_table` *Generate a network routing table*

Description

Generates a network width table for a catchment. When passed to the run.dtm routine this will be used to route channel flows to the outlet during a Dynamic TOPMODEL run.

Usage

```
build_routing_table(dem, chans = NULL, outlet = NULL, breaks = 5,
  len.fun = flow.lens)
```

Arguments

`dem`	Elevation raster using a projected coordinate system (e.g UTM) and a regular grid spacing. Areas outside the catchment should be set to NA
`chans`	Optional raster of the same dimensions and resolution as the DEM. Non-zero cells in this raster are considered to contain a river channel. If not supplied then flowpaths from the entire catchment area are considered.
`outlet`	Index of cell or cells identified with the catchment outlet
`breaks`	Number of distance intervals
`len.fun`	For large rasters the flow.len function can be very slow and many paths fail to reach a single outlet cell. This applies a simple straight line distance to the outlet to obtain a rough approximation.

Details

Dynamic TOPMODEL routes channel flow to the outlet by a network-width approach (see Beven, 2012, pp. 97-97). A time-delay histogram is produced using the table. When any flow is distributed to the channel "unit" it is immediately redistributed across future time steps according to the proportions found in the histogram. These flows are then added to future outputs from the model.

Value

A two-column data.frame. Its first column is the average flow distance to the outlet, in m, the second the proportions of the catchment channel network within each distance category.

Author(s)

Peter Metcalfe

References

Beven, K. J. (2012). Rainfall-runoff modelling : the primer. Chichester, UK, Wiley-Blackwell.

Examples

```
## Not run:
# Create a routing table for the Brompton test case and show histogram

data(brompton)

tab <- build_routing_table(brompton$dem,
  chans=brompton$reaches,
  breaks=5)
barplot(tab[,2]*100, xlab="Mean flow distance to outlet (m)",
ylab="Network Width %", names.arg=tab[,1])

## End(Not run)
```

`dev.reset` *Reset device parameters*

Description

Reset some display parameters of the active device, or make one active while setting its parameters

Usage

```
dev.reset(dev = dev.cur(), ...)
```

Arguments

`dev`	numeric Device ID. Defaults to currently active device. If another valid device this will be activated. If not an existing device a new device will be opened.
`...`	Named list with any other valid graphics parameters to apply

Value

Updated parameters of the active device

See Also

par

`discretise` *Discrete a catchment into hydrological response units (HRUs)*

Description

Discrete a catchment into a set hydrological response units (HRUs) according to any number of landscape layers and cuts

Usage

```
discretise(layers, chans, cuts = list(a = 10), area.thresh = 2/100,
  order.by = names(cuts)[[1]], riv.cells.na = FALSE, renumber = FALSE,
  order = FALSE, burn.hrus = NULL, chan.width = 5, remove.areas = TRUE,
  hrus = NULL)
```

Arguments

`layers`	A multi-band raster (stack) comprising the catchment data. This should be in a projected coordinate system (or none) and have regular cells. The first layer should be the elevation raster, and subsequent (named) layers should supply the landscape data drawn in to create the discretisation
`chans`	Raster containing channel reach locations, of the same dimensions and resolution of the DEM and other catchment layers. The reaches should be numbered sequentially and any areas not containing part of the channel should be NA. If a second band is supplied with values 0-1 then this is taken to be the proportion of the corresponding non-zero cell occupied by the channel. If this layer is not present then the proportion is inferred from the channel width as p = min(1, chan.width/xres(dem))
`cuts`	A list of cuts of the form layer_name=number. Each name should correspond to a layer name in the layers parameter.
`area.thresh`	Minimum area for response units, expressed as a percentage of the catchment plan area, excluding channel cells. Areas smaller than this are aggregated with adjacent areas until exceeding the threshold area
`order.by`	Name of layer whose values will be use to sort the response units, in decreasing order. Defaults to the name of the first cut
`riv.cells.na`	Boolean Remove river cells from discretisation (default FALSE)
`renumber`	Boolean Renumber HRUs after discretisation so that their IDS are in numerical order (default TRUE)
`order`	Boolean Reorder HRUs after discretisation so that those with highest TWI come first (approximate to distances from channel). Default FALSE
`burn.hrus`	list Named list of geometries (supplied as rasters) to burn into discretisation of HRUs that will be stamped onto the classification. Overrides any classification already defined
`chan.width`	Channel width, in same units as DEM. Only used if chans doesn't contain a layer to specify the proportion of each river cell comprised of the channel.

remove.areas	Boolean Whether to remove areas that fall into classes with smaller areal contribution than the supplied threshold (default FALSE)
hrus	list Unused, maintained for backward compatibility

Details

This applies the given cuts to the supplied landscape layers to produce areal groupings of the catchment.

Value

A list comprising the following:

groups A data frame whose rows comprising the names, plan area and model parameters of each response unit. See Beven and Freer (2001) and Metcalfe et al. (2015) for a description of these parameters

weights Flux distribution (weighting) matrix for routing of subsurface flow downslope through the response units. If n is the number of response units (including channel "unit(s)") this is an n x n matrix. Row sums should thus always add to 1. The elements of the i-th row give the proportion of flow directed from response unit i to the other units

cuts list Cuts applied to produce these HRUs

area.thresh Area threshold specified

layers Multi-band raster comprising the the original rasters that the specified cuts were applied to produce the discretisation; the channel network

chans The channel raster

hru The resultant response unit locations

Examples

```
# Landcover and soils are fairly homogenous throughout the Brompton catchment.
# Due to the extensive artifical sybsurface drainage discharging directly into
# the channel it is hypothesied that the storm response is largely mostly controlled
# by proximity to the network. A simple discretisation according to flow distance
# from the nearest channel thus appears to capture the dynamics without introducing
# unnecessary complexity.
## Not run:
require(dynatopmodel)

data(brompton)

chans <- build_chans(brompton$dem, drn=brompton$drn, chan.width=2)
sort by distance from the channel network, but want areas closest the channel to come first
layers <- addLayer(brompton$dem, 2000-brompton$flowdists)
disc <- discretise(layers, cuts=c(flowdists=10), chans=chans, area.thresh=0.5/100)
rm(chans)
rm(layers)
write.table(disc$groups, sep="\t", row.names=FALSE)

## End(Not run)
```

disp_output *Display output of a Dynamic TOPMODEL run*

Description

Simple output of the results of a simulation.

Usage

```
disp_output(qsim, rain, evap = NULL, qobs = NULL, tm = NULL,
  start = min(index(qsim)), end = max(index(qsim)),
  par = get.disp.par(lwd.rain = 3, qint = 0.1), show.maxima = FALSE,
  freq = "day", pch.qsim = "+", pch.obs = "*", ...)
```

Arguments

qsim	Time series of simulated discharges.
rain	Time series of rainfall, at same interval as simulated values.
evap	Time series of evapotranspiration (optional), at same interval as simulated values.
qobs	Time series of evapotranspiration (optional), at same interval as simulated values.
tm	Display a vertical line at this time in the simulation. If NULL no line will be drawn.
start	Start time for plot in a format interpretable as POSIXct date time. Defaults start of simulated discharges.
end	End time for plot in a format interpretable as POSIXct date time. Defaults to end of simulated discharges.
par	Parameters controlling display output. A default set may be obtained through a call to disp.par.
show.maxima	Boolean Whether to show the daily maxima as points
freq	character Period for which maxima are identified, day by default
pch.qsim	character Symbol for plotting maxima of simulations, if stipulated
pch.obs	character Symbol for plotting maxima of observations, if these are supplied
...	Any further named parameters will be treated as graphics parameters and applied throughout the plot.

Details

This will render the hydrograph, any observations, actual evapotranspiration, if supplied, and the rainfall hyetograph.

Author(s)

Peter Metcalfe

See Also

disp.par

Examples

```
## Not run:
# Show the output of the storm simulation, overriding label colours and vertical axis limits.
require(dynatopmodel)

data(brompton)

x11()
with(brompton$storm.run, disp_output(evap=ae*1000,qobs=qobs*1000,
                                     qsim=qsim*1000, rain=rain*1000,
                                  max.q=2, cex.main=1, col.axis="slategrey", las.time=1))

## End(Not run)
```

`dynatopmodel` *Implementation of the Dynamic TOPMODEL hydrological model.*

Description

A native R implementation and enhancement of Dynamic TOPMODEL, an extension to the semi-distributed hydrological model TOPMODEL. It includes some digital terrain analysis functions for discretisation of catchments by topographic indexes and other geo-referenced layers containing relevant landscape data.

TOPMODEL (Beven & Kirkby, 1979) is a well-established and widely used hydrological model that implements a spatial aggregation strategy ("discretisation") in order to reduce its computational demands. Hydrological similar areas identified by the discretisation procedure are referred to as hydrological response units (HRUs). Beven and Freer (2001) introduced a "dynamic" variant that addressed some of the limitations of the original TOPMODEL but which retained its computational and parametric efficiency. In particular, the original assumption of a quasi-steady water table was replaced by time-dependent kinematic routing between and within HRUs. This allows a more flexible discretisation approach, whereby any type of landscape data can be used to identify the HRUs.

With the introduction of a single new parameter SDmax specifying the maximum storage deficit before downslope flow out of a HRU ceases, variable upslope drainage areas can be simulated. This allows application to arid catchments subject to seasonal drying of upslope areas.

The 2001 version was implemented in FORTRAN and it, and its source code, have not been made generally available. It has been applied in a number of studies (see Metcalfe et al, 2015). A modified version with chemical stores attached to each HRU was implemented by Page et al. (2007) and applied to modelling the Cl signal in the Hafren, Mid-Wales. This new version, described in detail

in Metcalfe et al. (2015), retains the core dynamics of the FORTRAN implementation but makes use of data storage and vectorisation features of the R language to allow efficient scaling of the problem domain. This version was utilised by Metcalfe et al. (2017) to supply hillslope runoff to a new hydraulic channel routing model for evaluation of the effectiveness of arrays of in-channel barriers on mitigating flood risk.

A new, semi-distributed surface routing algorithm has been introduced in this version. This allows examination of surface storages as they move downslope and specification of different effective velocities throughout each unit. It also allows for a modified routing matrix that can reflect situations where the surface flow pathways differ from the topography. These could be used to simulate a unit associated with one or more surface features designed to intercept storm runoff, such as excavated ponds or bunds (see Hankin et al., 2016, 2017; Metcalfe, 2017).

The preprocessing routines supplied incorporate handling of geo-referenced spatial data to allows it to integrate with modern GIS through industry-standard file formats, such as GEOTiff and ESRI Shapefiles.

References

Beven, K. J. and M. J. Kirkby (1979). A physically based variable contributing area model of basin hydrology. Hydrol. Sci. Bull 24(1): 43-69.

Beven, K. J. and J. Freer (2001). A Dynamic TOPMODEL. Hydrological Processes 15(10): 1993-2011.

Hankin, B., Craigen I., Chappell, N., Metcalfe, P., Page, T. (2016). The Rivers Trust Life-IP Natural Course Project: Strategic Investigation of Natural Flood Management in Cumbria.Technical Report. Available at http://naturalcourse.co.uk/uploads/2017/04/2016s4667-Rivers-Trust-Life-IP-NFM-Opportunities-Technical-Report-v8.0.pdf. Rivers Trust, Callington, Cornwall, UK.

Hankin, B., Metcalfe, P., Johnson, D., Chappell, N., Page, T., Craigen, I., Lamb, R., Beven, K. (2017). Strategies for Testing the Impact of Natural Flood Risk Management Measures. In Hromadka, T. & Rao, P. (eds). Flood Risk Management. InTech, Czech Republic. ISBN 978-953-51-5526-3.

Metcalfe, P. (2017). Development of a modelling framework for integrated flood risk management (Doctoral dissertation). Lancaster University, UK.

Metcalfe, P., Beven, K., & Freer, J. (2015). Dynamic TOPMODEL: a new implementation in R and its sensitivity to time and space steps. Environmental Modelling & Software, 72, 155-172.

Metcalfe, P., Beven, K., Hankin, B., & Lamb, R. (2017). A modelling framework for evaluation of the hydrological impacts of nature?based approaches to flood risk management, with application to in-channel interventions across a 29 km2 scale catchment in the United Kingdom. Hydrological Processes, 31(9), 1734-1748.

Page, T., Beven, K. J., Freer, J., & Neal, C. (2007). Modelling the chloride signal at Plynlimon, Wales, using a modified dynamic TOPMODEL incorporating conservative chemical mixing (with uncertainty). Hydrological Processes, 21(3), 292-307.

See Also

`discretise`

`run.dtm`

`get.disp.par` *Default list of parameters to control the graphical output during model simulation and via disp.output*

Description

list of parameters to control the graphical output during model simulation. Parameters with names corresponding to the graphical parameters returned by par() will be applied to the plot.

Usage

```
get.disp.par(...)
```

Arguments

...	Further named arguments supplied will overwrite default values

Value

List of parameters with default values. These include: graphics.show Whether to show graphic output. Default TRUE graphics.window.length Width of display window in hours. Default is 120 days. graphics.interval Interval between refreshing the graphical output, in hours. max.q Max discharge (mm/hr) for display max.rain numeric Max rainfall (mm/hr) for display lwd.rain numeric Line size for rainfall plot int.q Interval between ticks / line on the y axis, in mm/hr int.time Period between ticks on the time axis, a numerical value in hours or one of "day", "week", "month" prop Proportion of screen occupied by the rainfall hyteograph cex Overall scaling factor of the plot las.time Alignment of time axis labels xmar Size of margin on right and left of plot, inches ymar Size of margin above and below plot col Colours for plot lines: in order, simulated values, observed values

Note

In this version many of the options are now obsolete. The most relevant are graphics.show and graphics.interval.

Author(s)

Peter Metcalfe

Examples

```
# Enable graphics output and set display interval to 6 hours
par <- get.disp.par(graphics.show=TRUE,
       graphics.interval=6)
```

get.run.par — *get.run.par Initalise model run parameters. Note this function is maintained for backward compatibility only*

Description

get.run.par Initalise model run parameters. Note this function is maintained for backward compatibility only

Usage

```
get.run.par(tms = NULL, dt = NULL, units = "secs", ...)
```

Arguments

tms	xts time series or an object that has a POSIXct index
dt	Numeric Time step in seconds
units	string Units for dt.
...	Any other parameters returned by get.run.par

Details

The returned value includes a simulation times calculated from the supplied time range and interval

Value

Structure to maintain run information.

Mode — *Mode - modal value*

Description

Mode - modal value

Usage

```
Mode(x)
```

Arguments

x	Numeric. A vector of numerical values whose mode is wanted

Value

The modal value

Note

Capitalisation is to distinguish method name from base::mode

`new_guid` *Get one or more GUIDs*

Description

A GUID (globally unique ID) can be used to identify objects such as files uniquely on the system. This method is based on the system time + plus a random number sampled from a uniform distribution

Usage

```
new_guid(n = 1, sep = "-", max = 1e+05)
```

Arguments

`n`	numeric Number of guids to return
`sep`	character Separator character
`max`	numeric Maximum for randomised element (default 1e6)

Value

A new GUID based on the system time + plus a random number in 0 to max

---

`NSE` *Nash Sutcliffe Efficiency of a model's output against observations*

Description

Returns the the NSE (NSE, Nash and Sutcliffe, 1970) of the simulated values against the given observations.

Usage

```
NSE(qsim, qobs, digits = 2)
```

Arguments

`qsim`	Time series or vector of simulated values
`qobs`	Time series or vector of observations
`digits`	No. decimal places in returned value

Value

A number <= 1 indicating the goodness of fit of the simulated series against observations (1= perfect fit). Values of >0.8 are generally regarded as "behavioural"

Author(s)

Peter Metcalfe

References

Nash, J., & Sutcliffe, J. V. (1970). River flow forecasting through conceptual models part I-A discussion of principles. Journal of hydrology, 10(3), 282-290.

Examples

```
## Not run:
require(dynatopmodel)

data(brompton)

# Goodness of fit for the storm simulation

NSE(brompton$storm.run$qsim, brompton$storm.run$qobs)

## End(Not run)
```

`run.dtm` *Run Dynamic TOPMODEL against hydrometric data and a catchment discretisation*

Description

Run Dynamic TOPMODEL against hydrometric data and a catchment discretisation

Usage

```
run.dtm(groups, weights, rain, routing, upstream_inputs = NULL, qobs = NULL,
  qt0 = 1e-04, pe = NULL, dt = NULL, ntt = 2, ichan = 1,
  Wsurf = weights, Wover = weights, i.out = ichan[1], dqds = NULL,
  sim.start = NA, sim.end = NA, disp.par = get.disp.par(), ...)
```

Arguments

`groups`	Data frame of areal group definitions along with their hydrological parameters (see Metcalfe et al., 2015)
`weights`	If the discretisation has n groups, this holds the n x n flux distribution (weighting) matrix defining downslope

rain	A time series of rainfall data in m/hr. One column per gauge if multiple gauges used.
routing	data.frame Channel routing table comprises a two-column data.frame or matrix. Its first column should be average flow distance to the outlet in m, the second the proportions of the catchment channel network within each distance category. Can be generated by make.routing.table
upstream_inputs	xts A list of any upstream hydrographs in addition to hillslope runoff feeding into the river network
qobs	Optional time series of observation data
qt0	Initial specific discharge (m/hr)
pe	Time series of potential evapotranspiration, at the same time step as rainfall data
dt	Time step (hours). Defaults to the interval used by the rainfall data
ntt	Number of inner time steps used in subsurface routing algorithm
ichan	Integer index of the "channel" group. Defaults to 1
Wsurf	matrix Surface routing matrix. Defines routing of overland flow downslope between units. By default identical to subsurface routing matrix by default, but can be altered to reflect modified connectivity of certain areas with the hillslope
Wover	matrix Optional surface overflow routing matrix. Defines routing of overland flow from a unit that has run out of surface excess storage capacity. Identical to surface routing matrix by default. Can be altered to reflect an overflow channel for a runoff storage area, for example.
i.out	For multi-channel systems, the index of the outlet reach
dqds	Function to supply a custom flux-storage relationship as the kinematic wave celerity. If not supplied then exponential relationship used.
sim.start	Optional start time for simulation in any format that can be coerced into a POSIXct instance. Defaults to start of rainfall data
sim.end	Optional end time of simulation in any format that can be coerced into a POSIXct instance. Defaults to end of rainfall data
disp.par	List of graphical routing parameters. A set of defaults are retrieved by calling disp.par()
...	Any further arguments will be treated as graphical parameters as documented in get.disp.par

Details

The grouping (HRU) table may be generated by the discretise method and includes each indexed channel as separate group. See Metcalfe et al. (2015) for descriptions of the parameters maintained in this table.

Evapotranspiration input can be generated using the approx.pe.ts method

Value

qsim: time series of specific discharges (m/hr) at the specified time interval. can be converted to absolute discharges by multiplying by catch.area

catch.area: the catchment area in m^2, calculated from the areas in the groups table

data.in: a list comprising the parameters supplied to the call

datetime sim.start Start of simulation

sim.end datetime End time of simulation

fluxes: a list comprising, for each response unit the specific base flows qbf, specific upslope inputs qin, drainage fluxes quz, and any overland flow qof, all in m/hr

storages: a list comprising, for each response unit, root zone and unsaturated storage, total storage deficit and surface storages (all m)

Note

If rain, pe or observation data differ in time period, use aggregate_xts to coerce the relevant series to the desired time interval

Author(s)

Peter Metcalfe

References

Metcalfe, P., Beven, K., & Freer, J. (2015). Dynamic TOPMODEL: a new implementation in R and its sensitivity to time and space steps. Environmental Modelling & Software, 72, 155-172.

See Also

aggregate_xts

discretise

Examples

```
## Not run:
require(dynatopmodel)
data(brompton)

# Examine the November 2012 event that flooded the village (see Metcalfe et al., 2017)
sel <- "2012-11-23 12:00::2012-12-01"
# Precalculated discretisation
disc <- brompton$disc
groups <- disc$groups
rain <- brompton$rain[sel]
# to 15 minute intervals
rain <- disaggregate_xts(rain, dt = 15/60)
# Reduce PE, seems a bit on high side and resulted in a weighting factor for the rainfall
pe <- brompton$pe[sel]/2
qobs <- brompton$qobs[sel]
```

```
# Here we apply the same parameter values to all groups.
# we could also consider a discontinuity at the depth of subsurface drains (~1m)
# or in areas more remote from the channel that do not contribute fast subsurface
# flow via field drainage
groups <- disc$groups
groups$m <- 0.0044
# Simulate impermeable clay soils
groups$td <-  33
groups$ln_t0 <- 1.15
groups$srz_max <- 0.1
qobs <- brompton$qobs[sel]
qt0 <- as.numeric(qobs[1,])
# initial root zone storage - almost full due to previous event
groups$srz0 <- 0.98
# Quite slow channel flow, which might be expected with the shallow and reedy
# low bedslope reaches with very rough banks comprising the major channel
groups$vchan <- 400
groups$vof <- 50
# Rain is supplied at hourly intervals: convert to 15 minutes
rain <- disaggregate_xts(rain, dt = 15/60)
weights <- disc$weights
# Output goes to a new window
graphics.off()
x11()

# Initial discharge from the observations
qt0 <- as.numeric(qobs[1,])

# Run the model across the November 2012 storm event
# using a 15 minute interval
run <- run.dtm(groups=groups,
               weights=weights,
               rain=rain,
               pe=pe,
               qobs=qobs,
               qt0=qt0,
               routing=brompton$routing,
               graphics.show=TRUE, max.q=2.4)

## End(Not run)
```

`time_at_peak` *Time of maximum observation*

Description

Determine the time of the maximum item in the supplied time series.

Usage

```
time_at_peak(ts, icol = 1)
```

Arguments

ts	Time series
icol	Column index if a multi-column time series

Author(s)

Peter Metcalfe

Peter Metcalfe

Examples

```
require(dynatopmodel)

data(brompton)

with(brompton$storm.run, time_at_peak(qsim))
```

---

time_to_peak *Time between the peak rainfall and the peak discharge*

Description

Return the lag, in hours, between the peak in the rainfall record and that of the discharge

Usage

```
time_to_peak(rain, qsim, units = "hour")
```

Arguments

rain	Time series of rainfall.
qsim	Time series of discharges.
units	Units in which to return the value

Author(s)

Peter Metcalfe

See Also

time_at_peak

Examples

```
require(dynatopmodel)

data(brompton)

with(brompton$storm.run, time_to_peak(rain, qsim))
```

`upslope.area` *Upslope contributing area and wetness index calculation*

Description

Determine upslope contributing area based on an elevation raster and, optionally, compute the topographic wetness index.

Usage

```
upslope.area(dem, log = TRUE, atb = FALSE, deg = 0.1, fill.sinks = TRUE)
```

Arguments

`dem`	raster Elevation raster (in m), using a projected coordinate system with identical x and y resolutions.
`log`	Boolean Return the natural log of the values.
`atb`	Boolean If TRUE, include both the upslope contributing area and the topographic wetness index $ln(a/tan(beta))$. Otherwise calculate just the upslope area.
`deg`	numeric Minimum intercell slope to identify with a sink (degrees).
`fill.sinks`	Fill sinks before calculation using the threshold angle given by deg.

Note

This is a wrapper to the function implemented in the TOPMODEL package by Wouter Buytaert.

Author(s)

Peter Metcalfe and Wouter Buytaert

References

Quinn, P. FALSE., Beven, K. J., & Lamb, R. (1995). The In (a/tan/beta) index: How to calculate it and how to use it within the Topmodel framework. Hydrological processes, 9(2), 161-182.

Examples

```
## Not run:
require(dynatopmodel)
data(brompton)

a.atb <- upslope.area(brompton$dem, atb=TRUE)
sp::plot(a.atb, main=c("Upslope area (log(m^2/m))", "TWI log(m^2/m)"))

## End(Not run)
```

Index

∗Topic **datasets**
brompton, 5

aggregate_obs, 2
aggregate_xts, 3
approx.pe.ts, 4

brompton, 5
build_chans, 6
build_layers, 8
build_routing_table, 9

dev.reset, 10
discretise, *6*, 11, *15*
disp_output, 13
dynatopmodel, 14
dynatopmodel-package *(*dynatopmodel*)*, 14

get.disp.par, 16
get.run.par, 17

Mode, 17

new_guid, 18
NSE, 18

run.dtm, *6*, *15*, 19

time_at_peak, 22
time_to_peak, 23

upslope.area, 24