Package 'SWAT2R'

January 14, 2011

Type Package

Title Input/Output functions for the hydrological model SWAT

Version 0.1.5-0

Date 2011-01-14

Author Mauricio Zambrano-Bigiarini

Maintainer Mauricio Zambrano-Bigiarini <mauricio.zambrano@ing.unitn.it>

Description Library for creating input files for and reading the outputs from the Soil & Water Assessment Tool hydrological model (SWAT,http://www.brc.tamus.edu/swat/)

License GPL (>=2)

Depends R (>= 2.9.0), Hmisc, foreign

Suggests zoo, hydroTSM, hydroGOF, sp, scatterplot3d

URL http://www.r-project.org

LazyLoad yes

R topics documented:

SWAT2R-package *SWAT2R*

Description

Functions for analysing the results of the Soil and Water Assessment Tool (SWAT) hydrological model.

Details

Package:	SWAT2R
Type:	Package
Version:	0.1.5-0
Date:	2011-01-13
License:	What license is it under?
LazyLoad:	yes

~~ An overview of how to use the package, including the most important functions ~~

Author(s)

Mauricio Zambrano Bigiarini <mauricio.zambrano@ing.unitn.it>

Maintainer: Mauricio Zambrano Bigiarini <mauricio.zambrano@ing.unitn.it>

References

Arnold, J. G., R. Srinivasan, R. S. Muttiah, and J. R.Williams (1998), Large area hydrologic modeling and assessment part I: Model development, Journal of the American Water Resources Association, 34(1), 73-89

Arnold, J. G., and N. Fohrer (2005), SWAT2000: current capabilities and research opportunities in applied watershed modelling, Hydrological Processes, 19(3), 563-572, doi:10.1002/hyp.5611

Neitsch, S. L., J. G. Arnold, J. R. Kiniry, and J. R. Williams (2005a), Soil and Water Assessment Tool Theoretical Documentation Version 2005, Grassland, Soil and Water Research Laboratory; Agricultural Research Service 808 East Blackland Road; Temple, Texas 76502; Blackland Research \& Extension Center; Texas Agricultural Experiment Station 720 East Blackland Road Temple, Texas 76502, USA

Neitsch, S. L., J. G. Arnold, J. R. Kiniry, R. Srinivasan, and J. R. Williams (2005b), Soil and Water Assessment Tool Input/Output file documentation, Version 2005, Grassland, Soil andWater Research Laboratory; Agricultural Research Service 808 East Blackland Road; Temple, Texas 76502; Blackland Research \& Extension Center; Texas Agricultural Experiment Station 720 East Blackland Road Temple, Texas 76502, USA.

See Also

hydroTSM

cms2mm *m3/s -> mm*

Description

This function transforms a numeric vector or streamflows from cubic meter per second (m3/s) into millimetres (mm), by using the draining area of the catchment along with the sampling frequency of the discharges.

Usage

```
cms2mm(x, AreaKm2)
```

Arguments

x	zoo object with streamflow values in cubic meter per second (m3/s). Its sampling frequency have to be in c('daily', 'monthly' or 'annual')
AreaKm2	Draining area, in squared kilometers (km2)

Details

streamflow in millimetres (mm) per year, month or day, depending on the sampling frequency of x. q_mm = q_cms * 86400 * ndays / (1000 * AreaKm2)

Author(s)

Mauricio Zambrano-Bigiarini, <mauricio.zambrano@ing.unitn.it>

---

dates2juliandays *dates2juliandays*

Description

This function converts a vector of days into a vector of Julian days, but in the format required for SWAT hydrological model, in the sense that each julian day goes only from 1 to 365/366, so, every year the origin for counting the julian day is set to the 31-Dec of the previous year, in a way that the 01-January takes 1 as its Julian day

Usage

```
dates2juliandays(dates, verbose)
```

Arguments

dates	vector containing the dates that will be converted into Julian days in SWAT format
verbose	logical; if TRUE, progress messages are printed

Author(s)

Mauricio Zambrano-Bigiarini, <mauricio.zambrano@ing.unitn.it>

df2swat *df2swat*

Description

data.frame (precipitation, temperature, flow) into DBF file
This function reads a data.frame (preferably created with the '.CHE_daily_analysis' function of the 'hydroTSM' package) with daily time series of precipitation or temperature and generates as many single DBF files as gauge stations are in the original data.frame

Usage

```
df2swat(var.type, x.ts, x.gis, output.drty, Date= "Date", date.fmt="%Y-%m-%d", i
```

Arguments

var.type	character with the name of the target variable that will be analysed. Valid values are: -) "Precipitation" -) "Temperature" -) "Flow"
x.ts	data.frame with the time series that will be written into DBF format. This file HAS to HAVE the following format: -) 1st column : called 'Date' with all the dates in which we have measurements -) Other columns: have to have the name of the corresponding station (e.g., "P9001", "T9000E"), and all the rows have to have the measured value for the corresponing date. So, each column corresponds to the time series of a single station
x.gis	OPTIONAL. Only required when 'var.type' is "Temperature", because the computation of the Maximum and Minimum daily Temperarure needs the elevation in each station. data.frame with the spatial information that is needed to write the DBF file. This file HAS to HAVE , at least, 2 columns, indicated by the values given to 'id' and 'elevation'
output.drty	character indicating the path of the directory in which the DFB files will be written. If the user leaves this empty this parametr, the working directory is used.
Date	character indicating the field name in 'x.ts' that stores the dates of the measurements in each station.
date.fmt	character indicating the date format of the dates stored in the 'Date' field
id	OPTIONAL. Only required when 'var.type' is "Temperature". character indicating the field name in 'x.gis' that stores the ID of each station, with a 'T' at the begining of the ID (e.g., "T9001", "T9000E"),
elevation	OPTIONAL. Only required when 'var.type' is "Temperature". character indicating the field name in 'x.gis' that stores the elevation of each station, [masl].

Author(s)

Mauricio Zambrano-Bigiarini, <mauricio.zambrano@ing.unitn.it>

diy *Days in a Year*

Usage

```
diy(year)
```

Arguments

```
year
```

Examples

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

## The function is currently defined as
function (year)
{
    if (class(year) == "character")
        year <- as.numeric(year)
    ndays <- numeric(length(year))
    ndays[1:length(year)] <- sapply(1:length(year), function(j,
        x) {
        t1 <- as.Date(paste(x[j], "-01-01", sep = ""))
        t2 <- as.Date(paste(x[j], "-12-31", sep = ""))
        ndays[j] <- length(seq(from = t1, to = t2, by = "days"))
    }, x = year)
    return(ndays)
  }
```

obs2swatcup *obs2swatcup*

Description

This function generates a file with observed values of streamflows for being used as input to SWAT-CUP (http://www.eawag.ch/organisation/abteilungen/siam/software/swat/index_EN)

Usage

```
obs2swatcup(x, tstep="daily", from, to, date.fmt="%Y-%m-%d", fname="Observations
```

Arguments

x	zoo object with flow measurement(s) in 'm3/s'. It could be 'daily' or 'monthly'
tstep	time step in which are stored the values of 'x'. Valid values are: 'daily, 'monthly'
from	Character indicating the starting date for the values stored in all the files that will be read. It HAs to be in the format indicated by 'date.fmt'
to	Character indicating the starting date for the values stored in all the files that will be read. It HAs to be in the format indicated by 'date.fmt'
date.fmt	Character indicating the date format in which you provide 'from' and 'to', e.g. "
fname	Filename (without extension) of the output file. The number of observations will be added at the end of the filename
dates	OPTIONAL (not implmented yet). Vector with the dates to be used for converting a raw 'x' values into a 'zoo' object
version	integer indicating the version of SWAT-CUP for which the file has to be written
var.id	OPTIONAL. Only needed when 'version==3'. Character that have to be added as prefix before the year and day/month as second column in the observation file

Author(s)

Mauricio Zambrano-Bigiarini, <mauricio.zambrano@ing.unitn.it>

References

http://www.eawag.ch/organisation/abteilungen/siam/software/swat/downloads/Manual_SwatCup.pdf

---

qbeh2ecdfs *Q behavioural -> Empirical CDFs*

Description

This function computes ECDFs for user-defined quantiles of the streamflows simulated with behavioural parameter sets.
Steps:
1) it computes un-weighted quantiles (e.g., Q5, Q50, Q95) for the streamflows simulated with EACH behavioural parameter set.
2) it computes ECDFs for each desired quantile, by weighting the quantiles of each behavioural parameter set by its corresponding less-formal likelihood.

Usage

```
qbeh2ecdfs(Qbeh, qobs, weights, byrow=FALSE, quantiles.desired= c(0.05, 0.5, 0.9
           quantiles.labels= c("Q5", "Q50", "Q95"), ylab="Probability",
           col="blue", leg.cex=1.2, verbose=TRUE, ...)
```

Arguments

Qbeh	matrix, which columns (by default) contains the behavioural values to be used for the computation of the weighted quantiles.
qobs	Numeric with the observed values, used for putting its weighted quantiles as a legend in the plot.
weights	numeric vector, with the values of the weights to be used for computing the quantiles. Omitting the weights argument or specifying NULL or a zero-length vector will result in the usual unweighted estimates.
byrow	logical, indicating if the computations have to be made for each column or for each row of x. When the simulated values obtained with different behavioural parameter sets are stored in columns, byrow must be *TRUE*. When the simulated values obtained with different behavioural parameter sets are stored in rows, byrow must be *FALSE*.
quantiles.desired	numeric vector, with the quantiles to be computed. Default value is *c(.025, .5, .975)* (=> 2.5%, 50%, 97.5%)
quantiles.labels	character vector, with the names to be given to quantiles.desired. Default value is *c("Q5", "Q50", "Q95")*
ylab	A title for the y axis. See plot.
col	integer indicating the version of SWAT-CUP for which the file has to be written
leg.cex	Character expansion factor *relative* to current 'par("cex")'. Used for text, and provides the default for 'pt.cex' and 'title.cex'. Default value = 1.2
verbose	logical; if TRUE, progress messages are printed
...	further arguments passed to the plot function or from other methods.

Author(s)

Mauricio Zambrano-Bigiarini, <mauricio.zambrano@ing.unitn.it>

References

http://www.eawag.ch/organisation/abteilungen/siam/software/swat/downloads/Manual_SwatCup.pdf

---

read.autocal *read.autocal*

Description

Function for reading the 'autocalX.out' resulting after the autocalibration process in SWAT 2005, and giving as output a data.frame with the columns of the read file.

Usage

```
read.autocal(file, tstep, out.type="Q", outID=NA, verbose=TRUE)
```

Arguments

file	the name of the file which the data are to be read from. If it does not contain an _absolute_ path, the file name is _relative_ to the current working directory, 'getwd()'. Tilde-expansion is performed where supported.
tstep	Time step used for the simulation that created the 'autocalX.out' file. It must be one of the following values: c("daily", "monthly", "annual"),
out.type	Type of results that have to be read. It must be one of the following values: -) "Q" : only results related to water quantity are read (first 3 columns): c("YEAR", "DAY", "FLOWcms") -) "Q+Sed" : only results related to water quantity AND sediments are read (first 4 columns): The previously mentioned 3 along with c("SEDCONCmg/kg") -) "Q+Sed+WQ": all the columns of the 'autocalX.out' are read
outID	Currently not used....
verbose	logical; if TRUE, progress messages are printed

Note

This function has not been exhaustively tested

Author(s)

Mauricio Zambrano-Bigiarini, <mauricio.zambrano@ing.unitn.it>

---

read.aXX.aut *read.aXX.aut*

Description

Function for reading the 'aXX.aut' resulting after the autocalibration process in SWAT 2005, and giving as output a data.frame with the columns of the read file.

Usage

```
read.aXX.aut(file, tstep, out.type="Q", autID=NA, verbose=TRUE)
```

Arguments

file	the name of the file which the data are to be read from. If it does not contain an _absolute_ path, the file name is _relative_ to the current working directory, 'getwd()'. Tilde-expansion is performed where supported.
tstep	Time step used for the simulation that created the 'aXX.aut' file. It must be one of the following values: c("daily", "monthly", "annual"),
out.type	Type of results that have to be read. It must be one of the following values: -) "Q" : only results related to water quantity are read (first 3 columns): c("YEAR", "DAY", "FLOWcms") -) "Q+Sed" : only results related to water quantity AND sediments are read (first 4 columns): The previously mentioned 3 along with c("SEDCONCmg/kg") -) "Q+Sed+WQ": all the columns of the 'autocalX.out' are read
autID	Currently not used....
verbose	logical; if TRUE, progress messages are printed

Note

This function has not been exhaustively tested

Author(s)

Mauricio Zambrano-Bigiarini, <mauricio.zambrano@ing.unitn.it>

`read.out.hru` *read.out.hru*

Description

Function for reading the 'output.hru' files of SWAT 2005 (regarding HRUs), and giving as output a data.frame with the columns of the read file.

Usage

```
read.out.hru(file="output.hru", tstep, out.type="Q", hruID=NA, verbose=TRUE)
```

Arguments

`file`	the name of the file which the data are to be read from. If it does not contain an _absolute_ path, the file name is _relative_ to the current working directory, 'getwd()'. Tilde-expansion is performed where supported.
`tstep`	Time step used for the simulation that created the 'output.hru' file. It must be one of the following values: c("daily", "monthly", "annual"),
`out.type`	Type of results that have to be read. It must be one of the following values: -) "Q" : only results related to water quantity are read (first 35 columns): c("LULC", "HRU", "GIS", "SUB", "MGT", "MON", "AREAkm2", "PRECIPmm", "SNOFALLmm", "SNOMELTmm", "IRRmm", "PETmm", "ETmm", "SW_INITmm", "SW_ENDmm", "PERCmm", "GW_RCHGmm", "DA_RCHGmm", "REVAPmm", "SA_IRRmm", "DA_IRRmm", "SA_STmm", "DA_STmm", "SURQ_GENmm", "SURQ_CNTmm", "TLOSSmm", "LATQmm", "GW_Qmm", "WYLDmm", "DAILYCN", "TMP_AVdgC", "TMP_MXdgC", "TMP_MNdgC", "SOL_TMPdgC", "SOLARMJ/m2") -) "Q+Sed" : only results related to water quantity AND sediments are read (first 37 columns): The previously mentioned 35 along with c("SYLDt/ha", "USLEt/ha") -) "Q+Sed+WQ": all the columns of the 'output.hru' are read
`hruID`	OPTIONAL. Integer with the number of the HRU for wich the results will be provided. If this argument is not provided, the results will be given for all the reaches in 'output.hru'
`verbose`	logical; if TRUE, progress messages are printed

Author(s)

Mauricio Zambrano-Bigiarini, <mauricio.zambrano@ing.unitn.it>

`read.out.rch` *read.out.rch*

Description

Function for reading the 'output.rch' files of SWAT 2005 (regarding reaches), and giving as output a data.frame with the columns of the read file.

This 'output.rch' file has 9 rows representing the header, and 1 colum with the text 'REACH', and 43 additional colums with results regarding water quantity, sediments, and water quality

Usage

```
read.out.rch(file="output.rch", tstep, out.type="Q", rchID=NA, verbose=TRUE)
```

Arguments

`file`	the name of the file which the data are to be read from. If it does not contain an _absolute_ path, the file name is _relative_ to the current working directory, 'getwd()'. Tilde-expansion is performed where supported.
`tstep`	Time step used for the simulation that created the 'output.rch' file. It must be one of the following values: c("daily", "monthly", "annual"),
`out.type`	Type of results that have to be read. It must be one of the following values: -) "Q" : only results related to water quantity are read (first 8 columns): c("RCH", "GIS", "MON", "DrAREAkm2", "FLOW_INcms", "FLOW_OUTcms", "EVAPcms", "TLOSScms") -) "Q+Sed" : only results related to water quantity AND sediments are read (first 11 columns): The previously mentioned 8 along with c("SED_INtons", "SED_OUTtons", "SEDCONCmg/kg") -) "Q+Sed+WQ": all the columns of the 'output.rch' are read
`rchID`	OPTIONAL. Integer with the number of the reach for wich the results will be provided. If this argument is not provided, the results will be given for all the reaches in 'output.rch'
`verbose`	logical; if TRUE, progress messages are printed

Author(s)

Mauricio Zambrano-Bigiarini, <mauricio.zambrano@ing.unitn.it>

`read.out.rsv` *read.out.rsv*

Description

Function for reading the 'output.rsv' files of SWAT 2005 (regarding reservoirs), and giving as output a data.frame with the columns of the read file.

Usage

```
read.out.rsv(file="output.rsv", tstep, out.type="Q", rsvID=NA, verbose=TRUE)
```

Arguments

file	the name of the file which the data are to be read from. If it does not contain an _absolute_ path, the file name is _relative_ to the current working directory, 'getwd()'. Tilde-expansion is performed where supported.
tstep	Time step used for the simulation that created the 'output.rch' file. It must be one of the following values: c("daily", "monthly", "annual"),
out.type	Type of results that have to be read. It must be one of the following values: -) "Q" : only results related to water quantity are read (first 8 columns): c("RES", "MON", "VOLUMEm3", "FLOW_INcms", "FLOW_OUTcms", "PRECIPm3", "EVAPm3", "SEEPAGEm3") -) "Q+Sed" : only results related to water quantity AND sediments are read (first 11 columns): The previously mentioned 8 along with c("SED_INtons", "SED_OUTtons", "SED_CONCppm") -) "Q+Sed+WQ": all the columns of the 'output.rsv' are read
rsvID	OPTIONAL. Integer with the number of the reservoir for wich the results will be provided. If this argument is not provided, the results will be given for all the reaches in 'output.rsv'
verbose	logical; if TRUE, progress messages are printed

Author(s)

Mauricio Zambrano Bigiarini <mauricio.zambrano@ing.unitn.it>

---

read.out.sub *read.out.sub*

Description

Function for reading the 'output.sub' files of SWAT 2005 (regarding subbasins), and giving as output a data.frame with the columns of the read file.

Usage

```
read.out.sub(file="output.sub", tstep, out.type="Q", subID=NA, verbose=TRUE)
```

Arguments

file	the name of the file which the data are to be read from. If it does not contain an _absolute_ path, the file name is _relative_ to the current working directory, 'getwd()'. Tilde-expansion is performed where supported.
tstep	Time step used for the simulation that created the 'output.sub' file. It must be one of the following values: c("daily", "monthly", "annual"),

out.type	Type of results that have to be read. It must be one of the following values: -) "Q" : only results related to water quantity are read (first 13 columns): c("SUB", "GIS", "MON", "AREAkm2", "PRECIPmm", "SNOMELTmm", "PETmm", "ETmm", "SWmm", "PERCmm", "SURQmm", "GW_Qmm", "WYLDmm") -) "Q+Sed" : only results related to water quantity AND sediments are read (first 14 columns): The previously mentioned 13 along with "SYLDt/ha" -) "Q+Sed+WQ": all the columns of the 'output.sub' are read
subID	OPTIONAL. Integer with the number of the subbasin for wich the results will be provided. If this argument is not provided, the results will be given for all the reaches in 'output.sub'
verbose	logical; if TRUE, progress messages are printed

Author(s)

Mauricio Zambrano Bigiarini <mauricio.zambrano@ing.unitn.it>

---

read.param *Read Parameters*

Description

This function reads the files:
-) 'modelpara.out' created by the GLUE algorithm of SWAT-CUP,
-) 'modelpara.beh' created by the GLUE algorithm of SWAT-CUP,
-) 'goal.sf2' created by the SUFI-2 algorithm of SWAT-CUP
-) 'goal.pso' created by the PSO algorithm of SWAT-CUP

Usage

```
read.param(file,
           header=FALSE,
           skip=0,
           param.cols=2:11,
           param.names=c("CN2.mgt", "RCHRG_DP.gw", "GWQMN.gw", "GW_REVAP.gw",
                         "REVAPMN.gw", "SOL_K.sol", "SOL_Z.sol", "GW_DELAY.gw",
                         "ALPHA_BF.gw", "SOL_AWC.sol"),
           of.col=12, of.name="NSeff", na.strings="-9999",
           plot=TRUE, beh.thr=NA, beh.col="red", beh.lty=1, beh.lwd=2,
           nrow=2, ylab=of.name, col="black", cex=0.5, pch=19, verbose=TRUE, ...
```

Arguments

file	character, with the name (including path) of the file which the data are to be read from.
header	a logical value indicating whether the file contains the names of the variables as its first line. If missing, the value is determined from the file format: header is set to TRUE if and only if the first row contains one fewer field than the number of columns.
skip	integer: the number of lines of the data file to skip before beginning to read data.

param.cols	numeric, with the number of the columns in file that store the values of each parameter
param.names	character with the name of the parameters defined by param.cols
of.col	numeric, with the number of the column in file that store the values of objective function
of.name	character, with the name that will be given to the column of.col
na.strings	character, with the string which is to be interpreted as NA values. read.table
plot	logical, indicating if a dotty plot with the parameter values vs the objective function is to be produced.
beh.thr	OPTIONAL. Only used when plot=TRUE. numeric, with the value to be used for drawing a horizontal line for separating behavioural from non behavioural parameter sets.
beh.col	OPTIONAL. Only used when plot=TRUE. character, with the color to be used for drawing a horizontal line for separating behavioural from non behavioural parameter sets.
beh.lty	OPTIONAL. Only used when plot=TRUE. numeric, with the type to be used for drawing a horizontal line for separating behavioural from non behavioural parameter sets.
beh.lwd	OPTIONAL. Only used when plot=TRUE. numeric, with the width to be used for drawing a horizontal line for separating behavioural from non behavioural parameter sets.
nrow	OPTIONAL. Only used when plot=TRUE. numeric, with the number of rows to be used for splitting the drawing window.
ylab	OPTIONAL. Only used when plot=TRUE. character, with the label for the 'y' axis.
col	OPTIONAL. Only used when plot=TRUE. character, representing the color to be used for drawing the points of the dotty plots
cex	OPTIONAL. Only used when plot=TRUE. numeric, representing the values controlling the size of text and points with respect to the default
pch	OPTIONAL. Only used when plot=TRUE. numeric with the type of symbol to be used for drawing the points of the dotty plots. (e.g., 1: white circle; 9: white rhombus with a cross inside)
verbose	logical; if TRUE, progress messages are printed
...	OPTIONAL. Only used when plot=TRUE. further arguments passed to the plot command or from other methods.

Author(s)

Mauricio Zambrano-Bigiarini, <mauricio.zambrano@ing.unitn.it>

References

SWAT-CUP: http://www.eawag.ch/organisation/abteilungen/siam/software/swat/index_EN.

`TmeanElev2Tmax` *TmeanElev2Tmax*

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
TmeanElev2Tmax(date, Elevation, Tmean)
```

Arguments

`date`

`Elevation`

`Tmean`

Author(s)

Mauricio Zambrano-Bigiarini, <mauricio.zambrano@ing.unitn.it>

`TmeanElev2Tmin` *TmeanElev2Tmin*

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
TmeanElev2Tmin(date, Elevation, Tmean)
```

Arguments

`date`

`Elevation`

`Tmean`

Author(s)

Mauricio Zambrano-Bigiarini, <mauricio.zambrano@ing.unitn.it>

wquantile *Q behavioural -> Empirical CDFs*

Description

This function computes weighted quantiles of each column (by default, or for each row if specified by the user) of a matrix/data.frame

It is a wrapper to the `wtd.quantiles` function of the **Hmisc** package, specially thought for a matrix containing streamflows simulated by different (behavioural) parameter sets

Usage

```
wquantile(x, weights=NULL, byrow=FALSE, probs=c(.025, .5, .975),
          normwt=TRUE, verbose=TRUE)
```

Arguments

`x`	numeric or matrix to be used for the computation of the weighted quantiles.
`weights`	numeric vector, with the values of the weights to be used for computing the quantiles. See `wtd.quantiles`. Omitting the `weights` argument or specifying `NULL` or a zero-length vector will result in the usual unweighted estimates.
`byrow`	logical, indicating if the computations have to be made for each column or for each row of `x`. When the simulated values obtained with different behavioural parameter sets are stored in columns, `byrow` must be *`TRUE`*. When the simulated values obtained with different behavioural parameter sets are stored in rows, `byrow` must be *`FALSE`*.
`probs`	numeric vector, with the quantiles to be computed. `wtd.quantiles`. Default value is *`c(.025, .5, .975)`* (=> 2.5%, 50%, 97.5%)
`normwt`	See `wtd.quantiles`. Specify 'normwt=TRUE' to make 'weights' sum to 'length(x)' after deletion of NAs.
`verbose`	logical; if TRUE, progress messages are printed

Author(s)

Mauricio Zambrano-Bigiarini, <mauricio.zambrano@ing.unitn.it>

See Also

`wtd.quantiles`