Package ‘dendRoAnalyst’

July 3, 2020

Type Package

Date 2020-07-03

Title A Tool for Processing and Analyzing Dendrometer Data

Version 0.1.1

Maintainer Sugam Aryal <sugam.aryal@fau.de>

Description There are various functions for managing and cleaning data before the application of different approaches. This includes identifying and erasing sudden jumps in dendrometer data not related to environmental change, identifying the time gaps of recordings, and changing the temporal resolution of data to different frequencies. Furthermore, the package calculates daily statistics of dendrometer data, including the daily amplitude of tree growth. Various approaches can be applied to separate radial growth from daily cyclic shrinkage and expansion due to uptake and loss of stem water. In addition, it identifies periods of consecutive days with user-defined climatic conditions in daily meteorological data, then check what trees are doing during that period.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 2.10), boot, pspline, zoo, graphics, grDevices, stats, base

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Sugam Aryal [aut, cre, dtc],
Martin Häusser [aut],
Jussi Grießinger [aut],
Ze-Xin Fan [aut],
Achim Bräuning [aut, dgs]

Repository CRAN

Date/Publication 2020-07-03 19:00:07 UTC

R topics documented:

`clim.twd` *Calculating relative growth change during no-rain periods.*

Description

This function calculates the number and the location of climatically adverse periods within a climate time series. The user can define a duration and threshold of these conditions. The function also provides the relative radial/circumferencial change during each adverse period for the original or normalized data. See Raffelsbauer et al., (2019) for more details.

Usage

```
clim.twd(
  df,
  Clim,
  dailyValue = "max",
  thresholdClim = 0,
  thresholdDays = 2,
  norm = F,
  showPlot
)
```

Arguments

`df`	dataframe with first column containing date and time in the format `yyyy-mm-dd HH:MM:SS` and the dendrometer data in following columns.
`Clim`	dataframe with the first column containing doy and second column containing corresponding climate data.

`dailyValue` either *'max', 'min'*, or *'mean'* for selecting the daily resampled value. Default is *'max'*. See `dendro.resample` for details.

`thresholdClim` numeric, the theshold for the respective climatic parameter. E.g. if climatic data is precipitation then days, where precipitation is below or equal to this value, are considered as adverse climate. Dafault is 0.

`thresholdDays` numeric, the minimum number of consecutive adverse days to be considered for analysis. For example, `thresholdDays`=2 means the relative radial/circumferential change is calculated for adverse periods lasting for more than 2 days. Default is 2.

`norm` logical, if `TRUE` the function uses normalized data instead of original dataset. Default is `FALSE`.

`showPlot` logical, if `TRUE`, generates plots.

Value

A dataframe containing the respective periods, relative radial/circumference change for each tree, the ID for each period and their beginning and end.

References

Raffelsbauer V, Spannl S, Peña K, Pucha-Cofrep D, Steppe K, Bräuning A (2019) Tree Circumference Changes and Species-Specific Growth Recovery After Extreme Dry Events in a Montane Rainforest in Southern Ecuador. Front Plant Sci 10:342. https://doi.org/10.3389/fpls.2019.00342

Examples

```
library(dendRoAnalyst)
data(gf_nepa17)
data(ktm_rain17)
relative_dry_growth<-clim.twd(df=gf_nepa17, Clim=ktm_rain17, dailyValue='max', showPlot=TRUE)
1

head(relative_dry_growth,10)
```

`daily.data` *Calculation of daily statistics for dendrometer data*

Description

This function calculates various statistics of dendrometer data on a daily basis. The daily statistics includes the daily maximum and minimum with their corresponding times and daily amplitude (difference between daily maximum and minimum). See King et al. (2013) for details.

Usage

```
daily.data(df, TreeNum)
```

Arguments

df	dataframe with first column containing date and time in the format yyyy-mm-dd HH:MM:SS and the dendrometer data in following columns.
TreeNum	numerical value indicating the tree to be analysed. E.g. '1' refers to the first dendrometer data column in *df*.

Value

A dataframe with the daily statistics of the dendrometer data that contains:

Columns	**Description**
DOY	The day of year.
min	The minimum value record for the corresponding day.
Time_min	The time when minimum value recorded for the corresponding day.
max	The maximum value record for the corresponding day.
Time_max	The time when maximum value recorded for the corresponding day.
mean	The daily average value of the dendrometer reading.
median	The daily median value of the dendrometer reading.
amplitude	The difference between daily maximum and daily minimum.

References

King G, Fonti P, Nievergelt D, Büntgen U, Frank D (2013) Climatic drivers of hourly to yearly tree radius variations along a 6°C natural warming gradient. Agricultural and Forest Meteorology 168:36–46. https://doi.org/10.1016/j.agrformet.2012.08.002

Examples

```
library(dendRoAnalyst)
data(nepa17)
daily_stats<-daily.data(df=nepa17, TreeNum=1)
head(daily_stats,10)
```

dendro.resample *Resampling temporal resolution of dendrometer data*

Description

This function is designed to change the temporal resolution of data. Depending on the objective, the user can define either maximum, minimum, or mean values to resample data in hourly, daily, weekly or monthly frequency.

Usage

```
dendro.resample(df, by, value)
```

Arguments

df — dataframe with first column containing date and time in the format `yyyy-mm-dd HH:MM:SS`.

by — either *H, D, W* or *M* to resample data into hourly, daily, weekly or monthly resolution.

value — either *max, min* or *mean* for the resampling value.

Value

Dataframe with resampled data.

Examples

```
library(dendRoAnalyst)
data(nepa17)
# To resample monthly with maximum value
resample_M<-dendro.resample(df=nepa17[,1:2], by='M', value='max')
head(resample_M,10)
```

---

dendro.truncate *Truncation of the dendrometer data*

Description

This function is helpful to truncate dendrometer data for a user-defined period.

Usage

```
dendro.truncate(df, CalYear, DOY)
```

Arguments

df — dataframe with the first column named date and time in the format `yyyy-mm-dd HH:MM:SS`.

CalYear — numerical value or array of two elements for the desired year of calculation.

DOY — numerical value or array of two elements representing the day of year. If we provide an array instead of a single value for `CalYear` and a single value for `DOY`, it truncates data from the `DOY` of the first `CalYear` to the same `DOY` of the second `CalYear`. Conversely, if we provide one value for `CalYear` and an array of two elements for `DOY` truncates the data form first `DOY` to second `DOY` within the same `CalYear`. Finally, if we provide an array with two values for both `DOY` and `CalYear`, it truncates data from the first `DOY` of the first `CalYear` to the second `DOY` of second `CalYear`.

Value

A dataframe with the truncated data for the defined periods.

Examples

```
library(dendRoAnalyst)
data(nepa)
#Extracting data from doy 20 to 50 in 2017.
trunc1<-dendro.truncate(df=nepa, CalYear=2017, DOY=c(20,50))
head(trunc1,10)
```

gf_nepa17 *Dendrometer data of Kathmandu for 2017 with gap filled*

Description

The dendrometer data from three Chir pine tree collected in hourly resolution for 2017.

Usage

```
gf_nepa17
```

Format

A data frame with 8760 rows and 3 variables:

`Time` datetime time of data recording

`T2` double reading for first tree

`T3` double reading for second tree

jump.locator *Removing artefacts due to manual adjustments of dendrometers*

Description

Dendrometers generally have limited memory capacity beyond which it stops recording. To keep the measurement ongoing, they should be adjusted periodically, which can cause positive or negative jumps in the data. This function locates these artefacts and interactively adjusts them one by one.

Usage

```
jump.locator(df, TreeNum, v)
```

Arguments

`df`	dataframe with first column containing date and time in the format `yyyy-mm-dd HH:MM:SS` and the dendrometer data in following columns.
`TreeNum`	numerical value indicating the tree to be analysed. E.g. '1' refers to the first dendrometer data column in *df*.
`v`	numerical value which is considered as artefact. E.g. `v=1` implies that if the difference to the consecutive data point is more than 1 or less than -1, it will be considered as an artefact.

Value

A dataframe containing jump-free dendrometer data.

Examples

```
library(dendRoAnalyst)
data(nepa)
jump_free_nepa<-jump.locator(df=nepa, TreeNum=1 ,v=1)
head(jump_free_nepa,10)
```

`ktm_rain17` *Daily rainfall data of Kathmandu for 2017.*

Description

This file contains daily rainfall data of Kathmandu. The source of this data is 'Government of Nepal, Department of Hydrology and Meteorology'.

Usage

```
ktm_rain17
```

Format

A data frame with 365 rows and 2 variables:

`DOY` double Day of year.

`rainfall` double rainfall in millimeters

Source

http://www.mfd.gov.np/city?id=31/

nepa *Dendrometer data from Kathmandu*

Description

Dendrometer data from three Chir pine trees collected in hourly resolution for 2 years.

Usage

```
nepa
```

Format

A data frame with 14534 rows and 3 variables:

`Time` datetime time of data recording

`T2` double reading for first tree

`T3` double reading for second tree

nepa17 *Dendrometer data of Kathmandu for 2017*

Description

Dendrometer data from three Chir pine tree collected in hourly resolution for 2017.

Usage

```
nepa17
```

Format

A data frame with 8753 rows and 3 variables:

`Time` datetime time of data recording

`T2` double reading for first tree

`T3` double reading for second tree

`nepa2` *Dendrometer data from Kathmandu version 2*

Description

Dendrometer data from three Chir pine trees collected in hourly resolution for 2 years with separated time.

Usage

```
nepa2
```

Format

A data frame with 14534 rows and 8 variables:

`year` numeric year of data recording

`month` numeric months of data recording

`day` numeric days of data recording

`hours` numeric hours of data recording

`minutes` numeric minutes of data recording

`seconds` numeric seconds of data recording

`T2` double reading for first tree

`T3` double reading for second tree

`network.interpolation` *Interpolation of NA values using the dendrometer network*

Description

A function to interpolate the missing data of a dendrometer with the help of other dendrometers from the same site, provided they have the same measurement period and temporal resolution.

Usage

```
network.interpolation(df, referenceDF, niMethod)
```

Arguments

`df` dataframe with first column containing date and time in the format `yyyy-mm-dd HH:MM:SS` and dendrometer data in the second column and onward. The data gaps must be filled with `NA` using the gap.interpolation function.

`referenceDF` dataframe with other dendrometers to be used as reference for the interpolation. The more dendrometers are included, the more robust will be the interpolation.

`niMethod` string, either *'linear'* or *'proportional'* for interpolation method.

Value

A dataframe with NA values replaced by interpolated data.

Examples

```
library(dendRoAnalyst)
data("gf_nepa17")
df1<-gf_nepa17
# Creating an artificial reference dataset.
df2<-cbind(gf_nepa17,gf_nepa17[,2:3],gf_nepa17[,2:3])
# Creating gaps in dataset by replacing some of the reading with NA in dataset.
df1[40:50,3]<-NA
# Using proportional interpolation method.
df1_NI<-network.interpolation(df=df1, referenceDF=df2, niMethod='proportional')
head(df1_NI,10)
```

phase.sc *Application of the stem-cycle approach to calculate different phases, their duration and to plot them.*

Description

This function analyses the dendrometer data using Stem-cycle approach (Downs et al. 1999; Deslauriers et al. 2011). A function that defines three phases: 1) Shrinkage, when the dendrometer reading is less than previous reading, 2) Expansion, when current reading is more than previous reading and 3) Increment, when current reading is higher than the previous maximum. Additionally, it calculates various statistics for each phase.

Usage

```
phase.sc(
  df,
  TreeNum,
  smoothing = NULL,
  outputplot = FALSE,
  days,
  cols = c("#fee8c8", "#fdbb84", "#e34a33"),
  phNames = c("Shrinkage", "Expansion", "Increment"),
  cex = NULL,
  cex.axis = NULL,
  cex.legend = NULL,
  font.axis = NULL,
  col.axis = NULL,
  ...
)
```

Arguments

`df` dataframe with first column containing date and time in the format `yyyy-mm-dd HH:MM:SS`. It should contain data with constant temporal resolution for best results.

`TreeNum` numerical value indicating the tree to be analysed. E.g. '1' refers to the first dendrometer data column in *df*.

`smoothing` numerical value from 1 to 12 which indicates the length of the smoothing spline, i.e. 1 = 1 hour and 12 = 12 hours. Default is `NULL` for no smoothing.The function `smooth.Pspline` is used for smoothing the data.

`outputplot` logical, to `plot` the phase diagram.

`days` array with initial and final day for plotting. E.g. *c(a,b)*, where a = initial date and b = final date.

`cols` array with three elements representing colors for shrinking, expanding and increasing phases respectively.

`phNames` array with three elements for three different phases. Default is **"Shrinkage", "Expansion" and "Increment"**.

`cex` numeric, for the size of the points. Default is `NULL`.

`cex.axis` numeric, for the size of the axis tick labels. Default is `NULL`.

`cex.legend` numeric, for the size of the legend labels. Default is `NULL`.

`font.axis` numeric, for the font type of the axis tick labels. Default is `NULL`.

`col.axis` color names, for the color of the axis tick labels. Default is `NULL`.

`...` other graphical parameters.

Value

A list of two dataframes. The first dataframe *SC_cycle* with cyclic phases along with various statistics and the second dataframe *SC_phase* with assigned phases for each data point.The dataframe *SC_cycle* contains the beginning, end, duration, magnitude and rate of each phase. The dataframe *SC_phase* contains time and corresponding phases during that time. The contents of *SC_cycle* are:

Columns	**Description**
`Phase`	Cyclic phases. 1, 2, and 3 for Shrinkage, Expansion, and Increment respectively.
`start`	Time when the corresponding phase starts.
`end`	Time when the corresponding phase ends.
`Duration_h`	Duration of the corresponding phase in hours.
`Duration_m`	Duration of the corresponding phase in minutes.
`Magnitude`	Radial/circumferential change during the corresponding phase in millimeters.
`rate`	Rate of Radial/circumferential change in micrometers per hour.
`DOY`	Day of year for the corresponding phase.

References

Deslauriers A, Rossi S, Turcotte A, Morin H, Krause C (2011) A three-step procedure in SAS to analyze the time series from automatic dendrometers. Dendrochronologia 29:151–161. https://doi.org/10.1016/j.dendro.2011.01

Downes G, Beadle C, Worledge D (1999) Daily stem growth patterns in irrigated Eucalyptus globulus and E. nitens in relation to climate. Trees 14:102–111. https://doi.org/10.1007/PL00009752

Examples

```
library(dendRoAnalyst)
data(gf_nepa17)
sc.phase<-phase.sc(df=gf_nepa17, TreeNum=1, smoothing=12, outputplot=TRUE, days=c(150,160))
head(sc.phase[[1]],10)
head(sc.phase[[2]],10)
```

phase.zg — *Application of the zero-growth approach to calculate different phases, their duration and to plot them.*

Description

This function analyses data using the zero-growth approach. Initially, it divides the data in two categories: 1) Tree water deficiency (TWD), i.e. the reversible shrinkage and expansion of the tree stem when the current reading is below the previous maximum and, 2) Increment (GRO), the irreversible expansion of the stem when the current reading is above the previous maximum. Then it calculates the TWD for each data point as the difference between the modelled "growth line" and the observed measurement. See Zweifel et. al.,(2016) for details.

Usage

```
phase.zg(
  df,
  TreeNum,
  outputplot,
  days,
  linearCol = "#2c7fb8",
  twdCol = "#636363",
  twdFill = NULL,
  twdFillCol = "#f03b20",
  xlab = "DOY",
  ylab1 = "Stem size variation [mm]",
  ylab2 = "TWD [mm]",
  twdYlim = NULL,
  cex.axis = NULL,
  cex.lab = NULL,
  font.lab = NULL,
  col.lab = NULL,
  font.axis = NULL,
  col.axis = NULL
)
```

Arguments

`df` dataframe with first column containing date and time in the format `yyyy-mm-dd HH:MM:SS`. It should contain data with constant temporal resolution for best results.

`TreeNum` numerical value indicating the tree to be analysed. E.g. '1' refers to the first dendrometer data column in *df*.

`outputplot` logical, to `plot` the phase diagram.

`days` array with initial and final day for plotting. E.g. *c(a,b)*, where a = initial date and b = final date.

`linearCol` color for the modelled curve.

`twdCol` color for the TWD curve.

`twdFill` filling method for the area under the TWD curve. Equivalent to `density` argument of the `polygon` function in the **graphics** package of R. Default value is `NULL`.

`twdFillCol` color to fill the area under the TWD curve.

`xlab` string, x label of the `plot`.

`ylab1` string, y label of the upper `plot`.

`ylab2` string, y label of the lower `plot`.

`twdYlim` numeric, to define the limit of the y-axis of the lower plot. Default is `NULL`, which automatically adjusts the y-axis limit.

`cex.axis` numeric, for the size of the axis tick labels. Default is `NULL`.

`cex.lab` numeric, for the size of the axis labels. Default is `NULL`.

`font.lab` numeric, for the font type of the axis labels. Default is `NULL`.

`col.lab` color names, for the color of the axis labels. Default is `NULL`.

`font.axis` numeric, for the font type of the axis tick labels. Default is `NULL`.

`col.axis` color names, for the color of the axis tick labels. Default is `NULL`.

Value

A list of two dataframes. The first dataframe *ZG_cycle* contains the cyclic phases along with various statistics and the second dataframe *ZG_phase* with assigned phases for each data point.The contents of *ZG_cycle* are:

Columns	**Description**
`DOY`	Day of year for the corresponding phase.
`Phase`	TWD for tree water deficit and GRO for irreversible expansion.
`start`	Time when the corresponding phase starts.
`end`	Time when the corresponding phase ends.
`Duration_h`	Duration of the corresponding phase in hours.
`Magnitude`	Radial/circumferential change in millimeters.
`rate`	Rate of Radial/circumferential change in micrometers per hour.
`Max.twd`	Maximum TWD recorded for the corresponding TWD phase.
`Max.twd.time`	Time of occurrence of maximum TWD value for each TWD phase.
`Avg.twd`	Average of TWD values for each TWD phase.
`STD.twd`	Standard deviation of TWD values for each TWD phase.

References

Zweifel R, Haeni M, Buchmann N, Eugster W (2016) Are trees able to grow in periods of stem shrinkage? New Phytol 211:839–849. https://doi.org/10.1111/nph.13995

Examples

```
library(dendRoAnalyst)
data(gf_nepa17)
zg.phase<-phase.zg(df=gf_nepa17, TreeNum=1, outputplot=TRUE, days=c(150,160))
head(zg.phase[[1]],10)
head(zg.phase[[2]],10)
```

spline.interpolation *Detection and spline interpolation of missing values in dendrometer data.*

Description

This function detects gap(s) in time series, inserts the missing rows based on the provided temporal resolution and assings NA values to the corresponding value. If required the NA values can be replaced by spline interpolation using na.spline of the package **zoo**.

Usage

```
spline.interpolation(df, resolution, fill = FALSE)
```

Arguments

df	dataframe with first column containing date and time in the format yyyy-mm-dd HH:MM:SS and following columns with dendrometer data for the same temporal resolution and time period.
resolution	integer, indicating the resolution of dendrometer data in **minutes**.
fill	logical, if TRUE it fills the NA values using spline interpolation. Default is FALSE.

Value

A dataframe containing the dendrometer data including gaps filled with either NA or interpolated values.

Examples

```
library(dendRoAnalyst)
data(nepa17)
gf_nepa17<-spline.interpolation(df=nepa17, resolution=60)
head(gf_nepa17,10)
```

twd.maxima *Locating the maxima of TWD periods*

Description

This function detects the TWD phases, including their beginning (TWDb), using the phase.zg function. Then it calculates the number, time of occurance (Tm) and value of every local maximum within each TWD phase. In addition it calculates the time difference between 'TWDb' and each 'Tm' within each TWD phase.

Usage

```
twd.maxima(df, TreeNum, smoothing = 5, showPlot = T, days = c(150, 160), ...)
```

Arguments

`df`	dataframe with first column containing date and time in the format `yyyy-mm-dd HH:MM:SS`. It should contain data with constant temporal resolution for best results.
`TreeNum`	numerical value indicating the tree to be analysed. E.g. '1' refers to the first dendrometer data column in *df*.
`smoothing`	numerical value from 1 to 12 which indicates the length of the smoothing spline, i.e. 1 = 1 hour and 12 = 12 hours. Default is 5.
`showPlot`	logical, if `TRUE`, it generates a plot. Default is `TRUE`.
`days`	array with initial and final day for plotting. E.g. *c(a,b)*, where a=initial date and b=final date. Default is *c(150,160)*.
`...`	additional graphical parameter incuded in `phase.zg`.

Value

A dataframe with statistics of maxima in each TWD phase.

Examples

```
library(dendRoAnalyst)
data(gf_nepa17)
df1=gf_nepa17[2500:3500,]
twd_max<-twd.maxima(df=df1, TreeNum=2, showPlot=FALSE)
head(twd_max,10)
```

Index