Package ‘dendRoAnalyst’
March 19, 2024

Type Package
Date 2024-03-19
Title A Tool for Processing and Analyzing Dendrometer Data
Version 0.1.5
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 7.3.1
Depends R (>= 3.5.0), stats, base, tidyverse
Imports dplyr, ggplot2, lubridate, readxl, tibble, tidyr, zoo, forecast, mgcv, minpack.lm, pspline, stringr
Suggests knitr, rmarkdown, testthat (>= 3.0.0)
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 2024-03-19 16:20:05 UTC
Config/testthat/edition 3
R topics documented:

clim.twd ................................................................. 2
daily.data ............................................................... 3
dendro.resample ......................................................... 4
dendro.truncate ......................................................... 5
dm.detrend ............................................................... 6
dm.fit.gompertz ......................................................... 7
dm.na.interpolation ..................................................... 7
gf_nePA17 ............................................................... 8
i.jump.locator ........................................................... 9
jump.locator ............................................................. 9
ktm_rain17 ............................................................... 10
mean_detrended.dm .................................................... 11
mov.cor.dm ............................................................. 11
nePA ....................................................................... 12
nePA17 ................................................................. 13
nePA2 ................................................................. 13
network.interpolation ................................................ 14
phase.sc ............................................................... 15
phase.zg ............................................................... 16
plot_mov.cor .......................................................... 17
plot_SC_output ......................................................... 18
plot_ZG_output ........................................................ 19
read.dendrometer ..................................................... 20
twd.maxima ............................................................ 20

Index 22

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/circumferential 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 = "<10",
  thresholdDays = ">5",
  showPlot = TRUE
)
**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 `Date` in `yyyy-mm-dd` and second column containing corresponding climate data.

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

- **thresholdClim**
  - string, the threshold 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. Default is `'<10'`.

- **thresholdDays**
  - string, 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 `'>5'`.

- **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**


**Examples**

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

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.
dendro.resample

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:

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>The day of year in &quot;yyyy-mm-dd&quot;.</td>
</tr>
<tr>
<td>Min</td>
<td>The minimum value record for the corresponding day.</td>
</tr>
<tr>
<td>Time_min</td>
<td>The time when minimum value recorded for the corresponding day.</td>
</tr>
<tr>
<td>Max</td>
<td>The maximum value record for the corresponding day.</td>
</tr>
<tr>
<td>Time_max</td>
<td>The time when maximum value recorded for the corresponding day.</td>
</tr>
<tr>
<td>mean</td>
<td>The daily average value of the dendrometer reading.</td>
</tr>
<tr>
<td>median</td>
<td>The daily median value of the dendrometer reading.</td>
</tr>
<tr>
<td>amplitude</td>
<td>The difference between daily maximum and daily minimum.</td>
</tr>
<tr>
<td>Remarks</td>
<td>&quot;*&quot; if Time_max &gt; Time_min otherwise &quot;.&quot;</td>
</tr>
</tbody>
</table>

References


Examples

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

dendro.resample Resampling temporal resolution of dendrometer and climate 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.
**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 from 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.
dm.detrend

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)
```

---

<table>
<thead>
<tr>
<th>dm.detrend</th>
<th>Detrend the dendrometer data</th>
</tr>
</thead>
</table>

Description

This function detrends the dendrometer data either using first difference or using gam function from mgcv package or the Gompertz function.

Usage

```
dm.detrend(df, method = "gam", CalYear)
```

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.
- **method**: either `"f_diff"`, `"gam"` or `"gomp"` indicating the method to detrend the dendrometer data column in `df`.
- **CalYear**: numeric for year of calculation. If `df` has more than one year, assigning `CalYear` truncates the data of only that year.

Value

A dataframe with the detrended dendrometer series.

Examples

```
library(dendRoAnalyst)
data(gf_nepa17)
detrended<-dm.detrend(df=nepa17, method='f_diff', CalYear=2017)
head(detrended,10)
```
**dm.fit.gompertz**  
*Fitting gompertz function on annual dendrometer data*

**Description**

This function models the annual growth of dendrometer data using gompertz function.

**Usage**

```r
dm.fit.gompertz(df, CalYear, TreeNum, f_derivative = F)
```

**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.
- `CalYear`: numeric for year of calculation. If `df` has more than one year, assigning `CalYear` truncates the data of only that year.
- `TreeNum`: numerical value indicating the tree to be analysed. E.g. '1' refers to the first dendrometer data column in `df`.
- `f_derivative`: logical if yes returns first derivative of gompertz curve.

**Value**

A dataframe with the modelled dendrometer series.

**Examples**

```r
library(dendRoAnalyst)
data(gf_nepa17)
gomp_fitted<-dm.fit.gompertz(df=gf_nepa17, TreeNum = 1, CalYear=2017)
head(gomp_fitted,10)
```

---

**dm.na.interpolation**  
*Detection and 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 assigns NA values to the corresponding value. If required the NA values can be replaced by spline interpolation using `na.spline` of the package `zoo` or seasonal interpolation considering the seasonality of the daily pattern using `na.interp` of the package `forecast`.

**Usage**

```r
dm.na.interpolation(df, resolution, fill = FALSE, method = "spline")
```
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`.
- **method**: string, 'spline' for the spline interpolation or 'seasonal' for the seasonal interpolation.

Value

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

Examples

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

---

**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
**i.jump.locator**

Removing artefacts due to manual adjustments of dendrometers interactively

**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**

```r
i.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.

---

**jump.locator**

Removing artefacts due to manual adjustments of dendrometers automatically for more than one 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 adjusts them. Unlike in `i.jump.locator`, it can handle dataset with more than one dendrometers.

**Usage**

```r
jump.locator(df, 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.

- **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**

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

**Description**

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

**Usage**

```r
ktm_rain17
```

**Format**

A data frame with 365 rows and 2 variables:

- **TIME** Date in YYYY-MM-DD format.
- **rainfall** double rainfall in millimeters

**Source**

mean_detrended.dm

Calculate mean of detrended dendrometer data.

Description

This function calculate the mean detrended series of dendrometer data. It is usefull to make a single averaged detrended dendrometer series of a species in a site. Further, it provides option for removing first order autocorrelation from dendrometer series using auto.arima function of "forecast" package.

Usage

mean_detrended.dm(detrended.dm, ac1.remove = T, robust.mean = T)

Arguments

detrended.dm  dataframe output data frame of dm.detrend function.
ac1.remove    logical if TRUE removes first order autocorrelation.
robust.mean   logical if TRUE calculates robust mean.

Value

A data frame with the mean of detrended dendrometer series.

Examples

library(dendRoAnalyist)
data(gf_nepa17)
detrended<-dm.detrend(df=nepa17, method='f_diff', CalYear=2017)
m_detre <- mean_detrended.dm(detrended)
head(m_detre,10)

mov.cor.dm     Calculating running correlation between dendrometer data and daily climate.

Description

This function calculates running correlation between dendrometer series and provided climate variables. Users can choose methods such as pearson, kendall and spearman, see cor.test for further information.

Usage

mov.cor.dm(df, Clim, TreeNum, win_size, cor_method = "pearson")
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 `Date in yyyy-mm-dd` and second column containing corresponding climate data.
- **TreeNum**: numerical value indicating the tree to be analysed. E.g. '1' refers to the first dendrometer data column in `df`.
- **win_size**: numerical, the running days windows of which the correlation is to be calculated. The minimum value is 18.
- **cor_method**: string, method to be applied during correlation calculation. One of the following: pearson, kendall and spearman

Value

A list of dataframes containing the correlation and significant value between dendrometer series and provided climates.

Examples

```r
library(dendRoAnalyst)
data(gf_nepa17)
data(ktm_rain17)
out_corr<-mov.cor.dm(df=gf_nepa17, Clim=ktm_rain17, TreeNum=1, win_size=21)
head(out_corr[1],10)
```

---

**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
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

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])
colnames(df2) <- c('Time','T1','T2','T3','T4','T5','T6')
# 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)
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)

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.

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:

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


Examples

```r
library(dendRoAnalyst)
data(gf_nepal17)
sc.phase<-phase.sc(df=gf_nepal17, TreeNum=1, smoothing=12)
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.

The severity value of each TWD was introduced in version 0.1.4 of the package.

**Usage**

```r
phase.zg(df, TreeNum)
```

**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*.

**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

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOY</td>
<td>Day of year for the corresponding phase.</td>
</tr>
<tr>
<td>Phase</td>
<td>TWD for tree water deficit and GRO for irreversible expansion.</td>
</tr>
<tr>
<td>start</td>
<td>Time when the corresponding phase starts.</td>
</tr>
<tr>
<td>end</td>
<td>Time when the corresponding phase ends.</td>
</tr>
<tr>
<td>Duration_h</td>
<td>Duration of the corresponding phase in hours.</td>
</tr>
<tr>
<td>Magnitude</td>
<td>Radial/circumferential change in millimeters.</td>
</tr>
<tr>
<td>rate</td>
<td>Rate of Radial/circumferential change in micrometers per hour.</td>
</tr>
<tr>
<td>Max.twd</td>
<td>Maximum TWD recorded for the corresponding TWD phase.</td>
</tr>
<tr>
<td>twd.severity</td>
<td>The severity of the individual TWD period (see description below).</td>
</tr>
<tr>
<td>Max.twd.time</td>
<td>Time of occurrence of maximum TWD value for each TWD phase.</td>
</tr>
<tr>
<td>Avg.twd</td>
<td>Average of TWD values for each TWD phase.</td>
</tr>
<tr>
<td>STD.twd</td>
<td>Standard deviation of TWD values for each TWD phase.</td>
</tr>
</tbody>
</table>

### References


### Examples

```r
library(dendRoAnalyst)
data(gf_nepa17)
zg.phase<-phase.zg(df=gf_nepa17[1:600,, TreeNum=1])
head(zg.phase[[1]],10)
head(zg.phase[[2]],10)
```

---

#### Description

This function plots the moving correlation i.e., output of `mov.cor.dm`.

#### Usage

```r
plot_mov.cor(mov.cor.output, sig.only = T, ci = 0.95, clim_vars = "all")
```

#### Arguments

- **mov.cor.output**: list the output of `mov.cor.dm` function.
- **sig.only**: logical TRUE to plot only significant correlation.
- **ci**: numeric confidence interval
- **clim_vars**: array of climate variables or "all" for all climate variables.
Value

A plot with correlation.

Examples

```r
library(dendRoAnalyst)
data(gf_nepa17)
data(ktm_rain17)
out_corr<-mov.cor.dm(df=gf_nepa17, Clim=ktm_rain17, TreeNum=1, win_size=21)
plot_mov.cor(mov.cor.output=out_corr, sig.only=TRUE, ci=0.95)
```

Description

This function plots the stem cycle of dendrometer data.

Usage

```r
plot_SC_output(
  SC_output,
  DOY,
  Year,
  cols = c("#fee8c8", "#fdbb84", "#e34a33"),
  phNames = c("Shrinkage", "Expansion", "Increment")
)
```

Arguments

- **SC_output**: list the output of `phase.sc` function.
- **DOY**: array with initial and final day for plotting. E.g. `c(a,b)`, where `a` = initial date and `b` = final date.
- **Year**: array for indicating year for plotting.
- **cols**: array of three elements: colour for each phases.
- **phNames**: array with three elements for three different phases. Default is "Shrinkage", "Expansion" and "Increment".

Value

A plot with different phases.
**Examples**

```r
library(dendRoAnalyst)
data(gf_nepa17)
sc.phase<-phase.sc(df=gf_nepa17, TreeNum=1, smoothing=12)
plot_SC_output(SC_output=sc.phase,DOY=c(50,60), Year=2017)
```

---

**plot_ZG_output**

*Plotting output of zero-growth approach (i.e., output of phase.zg function).*

**Description**

This function plots the GRO and TWD of dendrometer data for a defined time period.

**Usage**

```r
plot_ZG_output(ZG_output, DOY, Year)
```

**Arguments**

- **ZG_output**: list the output of `phase.zg` function.
- **DOY**: array with initial and final day for plotting. E.g. `c(a,b)`, where `a` = initial date and `b` = final date.
- **Year**: array for indicating year for plotting.

**Value**

A plot with GRO and TWD in two different subplots.

**Examples**

```r
library(dendRoAnalyst)
data(gf_nepa17)
zg.phase<-phase.zg(df=gf_nepa17, TreeNum=1)
plot_ZG_output(ZG_output=zg.phase,DOY=c(50,51), Year=2017)
```
**read.dendrometer**  
*Reading dendrometer data.*

**Description**
This function reads dendrometer data from .csv or .txt or .xlsx files. This function automatically recognizes the date time format of the first column of the data frame and changes it to "yyyy-mm-dd HH:MM:SS format".

**Usage**
read.dendrometer(file, sep = NULL, dec = NULL)

**Arguments**
- `file`: string file name or path of the file.
- `sep`: string the separator of the files. Only if they are different than the standard separators such as tab for .txt file and comma for .csv file.
- `dec`: the character used in the file for decimal points.

**Value**
A dataframe with the dendrometer data:

---

**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 occurrence (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)

**Arguments**
- `df`: data frame 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.
twd.maxima

Value

A data frame 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)
head(twd_max,10)
Index

* datasets
  
gf_nepa17, 8
ktm_rain17, 10
nepa, 12
nepa17, 13
nepa2, 13

clim.twd, 2
daily.data, 3
dendro.resample, 3, 4
dendro.truncate, 5
dm.detrend, 6
dm.fit.gompertz, 7
dm.na.interpolation, 7
gf_nepa17, 8
i.jump.locator, 9, 9
jump.locator, 9
ktm_rain17, 10
mean_detrended.dm, 11
mgcv, 6
mov.cor.dm, 11
na.interp, 7
na.spline, 7
nepa, 12
nepa17, 13
nepa2, 13
network.interpolation, 14

phase.sc, 15
phase.zg, 16
plot_mov.cor, 17
plot_SC_output, 18
plot_ZG_output, 19
read.dendrometer, 20

smooth.Pspline, 15
twd.maxima, 20

22