

# Package: TDPanalysis (via r-universe)

August 21, 2024

**Type** Package

**Title** Granier's Sap Flow Sensors (TDP) Analysis

**Version** 1.0

**Date** 2020-02-28

**Author** Maxime Durand

**Maintainer** Maxime Durand <durand1211@gmail.com>

**Description** Set of functions designed to help in the analysis of TDP sensors. Features includes dates and time conversion, weather data interpolation, daily maximum of tension analysis and calculations required to convert sap flow density data to sap flow rates at the tree and plot scale (For more information see : Granier (1985) <DOI:10.1051/forest:19850204> & Granier (1987) <DOI:10.1093/treephys/3.4.309>).

**Imports** stats, plyr, graphics

**Depends** R (>= 2.10)

**Encoding** UTF-8

**LazyData** true

**License** GPL-2

**RoxygenNote** 7.0.2

**NeedsCompilation** no

**Repository** https://mxdurand.r-universe.dev

**RemoteUrl** https://github.com/mxdurand/tdpanalysis

**RemoteRef** HEAD

**RemoteSha** d0ef201a522a49c550913c5edcd2519cfe61d8ea

## Contents

date.to.DOY . . . . .	2
datetime . . . . .	3
remove.fun . . . . .	3

SpFl . . . . .	4
SpWd_Area_calc . . . . .	5
tens.to.sapflow . . . . .	5
timecont . . . . .	6
Tmax.find . . . . .	7
Tmax.mean . . . . .	7
Tmaxplot . . . . .	8
Wat.transp . . . . .	9
<b>Index</b>	<b>10</b>

---

date.to.DOY	<i>Date conversion</i>
-------------	------------------------

---

## Description

Convert dates from the DD/MM/YYYY format to day of the year (DOY)

## Usage

```
date.to.DOY(dates, format = "dd/mm/yyyy")
```

## Arguments

dates	Vector with dates to convert.
format	Format of the date (support DD/MM/YYYY MM/DD/YYYY and YYYY/MM/DD).

## Value

Return a vector containing the corresponding DOY.

## Examples

```
dates = c("01/01/2000", "03/03/2000", "03/03/1999")
date.to.DOY(dates=dates)
```

---

datetime	<i>Time &amp; dates conversion</i>
----------	------------------------------------

---

**Description**

Convert DOY and time into a single numerical variable

**Usage**

```
datetime(dates, Time)
```

**Arguments**

dates	Vector with dates in the DOY format.
Time	Vector with time

**Details**

time vector should be numerical (e.g. as outputed by the time.to.cont function)

**Value**

Return a vector containing DOY and time as a single numerical variable

**Examples**

```
dates = c(102,102,102,102,103,103,103,103)
Time = c(22, 22.5, 23, 23.5, 0, 0.5, 1, 1.5)
datetime(dates=dates, Time=Time)
```

---

remove.fun	<i>Remove unwanted dates</i>
------------	------------------------------

---

**Description**

Remove all data for the corresponding date argument

**Usage**

```
remove.fun(df, dates)
```

**Arguments**

df	Data frame containing a DOY column named "DOY".
dates	Character vector containing the DOY to remove from the data frame.

**Details**

This function is primarily used to remove days for which Tmax is too extreme.

**Value**

Return the inputed data frame without the date corresponding the the "dates" argument.

**Examples**

```
DOY = c(rep(102, times=10), rep(103, times=10))
ID = c(rep("A", times=5), rep("B", times=5), rep("A", times=5), rep("B", times=5))
Tmax = c(rep(2.5, times=5), rep(2.7, times=5), rep(3.2, times=5), rep(3.4, times=5))
df <- data.frame(DOY, ID, Tmax)
dates = c("103")
remove.fun(df=df, dates=dates)
```

---

SpFl

*Sap flow dataset*

---

**Description**

Exemple dataset exemple for the TDPanalysis package

**Usage**

SpFl

**Format**

An object of class `data.frame` with 432 rows and 4 columns.

**Details**

"DATE" is dates in dd/mm/yyyy format. "TIME" is time in hh:mm:ss format, "ID" is sub-groups and "tension" is the measured tension from the TDP probe.

---

SpWd_Area_calc	<i>Sapwood area calculation</i>
----------------	---------------------------------

---

**Description**

Calculate sapwood area based on diameter, heartwood diameter and sapwood fraction

**Usage**

```
SpWd_Area_calc(diam, SpWd_frac = 1, HtWd_diam = 0)
```

**Arguments**

diam	Vector with diameter.
SpWd_frac	Numerical (from 0 to 1). Indicate the fraction of the diameter which is sapwood
HtWd_diam	Vector with diameter of the heartwood.

**Details**

If SpWD\_frac and HtWd\_diam are both entered, the function will return an error. Units of "diam" and "HtWd\_diam" should be the same.

**Value**

Return a numerical vector containing the sapwood area

**Examples**

```
diam = c(12,14,16,13,15)
SpWd_Area_calc(diam=diam, SpWd_frac=0.2)
```

---

tens.to.sapflow	<i>Convert tension into sap flow density</i>
-----------------	--

---

**Description**

Use the Granier formula to convert tension into sap flow density using daily or mean Tmax

**Usage**

```
tens.to.sapflow(tension, Tmax)
```

**Arguments**

tension	Vector with tension.
Tmax	Vector with corresponding maximums of tension.

**Value**

Return a numerical vector containing the sap flow density

**References**

Granier A. 1985. A new method of sap flow measurement in tree stems. *Annales Des Sciences Forestieres* 42(2): 193-200.

Granier A. 1987. Evaluation of transpiration in a douglas-fir stand by means of sap flow measurements. *Tree Physiology* 3(4): 309-319.

**Examples**

```
Tmax = c(rep(2.5, times=5), rep(2.7, times=5), rep(3.2, times=5), rep(3.4, times=5))
tension = c(5:25)
tens.to.sapflow(tension=tension, Tmax=Tmax)
```

---

timecont

*Time conversion*


---

**Description**

Convert time from the HH:MM:SS format to a numerical

**Usage**

```
timecont(Time, sep = ":")
```

**Arguments**

Time	Vector with time to convert.
sep	Character element containing regular expression(s) to use to splitting.

**Details**

time vector should be in the HH:MM:SS format.

**Value**

Return a vector containing the corresponding time.

**Examples**

```
Time = c("14:30:00", "20:45:00", "05:00:00")
timecont(Time=Time)
```

---

Tmax.find	<i>Find Tmax</i>
-----------	------------------

---

**Description**

Find the daily maximum of tension

**Usage**

```
Tmax.find(tension, dates, ID)
```

**Arguments**

tension	Vector with tension.
dates	Vector with dates in the DOY format.
ID	Character vector for specifying which group the tension is assigned to (e.g. trees)

**Value**

Return a vector containing daily Tmax for each group specified in the ID argument

**Examples**

```
tension = c(1:20)
dates = c(rep(102, times=10), rep(103, times=10))
ID = c(rep("A", times=5), rep("B", times=5), rep("A", times=5), rep("B", times=5))
Tmax.find(tension=tension, dates=dates, ID=ID)
```

---

Tmax.mean	<i>Calculate a mean of Tmax</i>
-----------	---------------------------------

---

**Description**

Calculate a mean Tmax for each sub-group

**Usage**

```
Tmax.mean(df)
```

**Arguments**

df	Data frame containing all Tmax for each sub-group.
----	--

**Details**

The data frame should contain a column named "Tmax" which all Tmax and a column named "ID" to identify which Tmax belong to which sug-group.

**Value**

Return the inputed data frame with a new column names "Tmax\_mean".

**Examples**

```
ID = c(rep("A", times=5), rep("B", times=5), rep("A", times=5), rep("B", times=5))
Tmax = c(rep(2.5, times=5), rep(2.7, times=5), rep(3.2, times=5), rep(3.4, times=5))
DOY = c(rep(102, times=10), rep(103, times=10))
df <- data.frame(DOY, ID, Tmax)
Tmax.mean(df)
```

---

Tmaxplot

*Plot the Tmax*


---

**Description**

Plot the Tmax with indications of extreme values

**Usage**

```
Tmaxplot(df)
```

**Arguments**

df                      Data frame containing Tmax, identification of sub-groups and DOY.

**Details**

The dataframe should contain at least 3 columns named "Tmax" (daily maximums of tension), "DOY" (day of the year) and "ID" (sub-groups). The red horizontal lines represents 3 times the inter-quartile range ( $3 \times \text{IQR}$ ) of all the Tmax of the data. The blue horizontal line represent the  $1.5 \times \text{IQR}$  without the Tmax outside the red lines.

**Value**

Return a plot of Tmax by days for each sub-group

**Examples**

```
DOY = c(rep(102, times=10), rep(103, times=10))
ID = c(rep("A", times=5), rep("B", times=5), rep("A", times=5), rep("B", times=5))
Tmax = c(rep(0.7512, times=5), rep(0.7359, times=5), rep(0.7644, times=5), rep(0.7666, times=5))
df <- data.frame(DOY, ID, Tmax, stringsAsFactors = FALSE)
Tmaxplot(df)
```



---

Wat.transp	<i>Calculate daily transpiration</i>
------------	--------------------------------------

---

**Description**

Calculate daily transpiration for each sub-group inputed

**Usage**

```
Wat.transp(Sapflow, days, ID)
```

**Arguments**

Sapflow	Vector with sap flow.
days	Vector containg the days for which to calculate transpiration
ID	Character vector containing identification for each sub-group

**Details**

!!Beware of the units!! The Granier formula usually convert tension into sap flow density (in kg.dm<sup>-2</sup>.h<sup>-1</sup>). So, you should first convert sap flow density into sap flow (in kg.h<sup>-1</sup>). Moreover, if you take measurment every 30 minutes sap flow should be corrected by dividing the value by 2.

**Value**

Return a data frame with transpiration for each day and sub-group inputed

**Examples**

```
ID = c(rep("A", times=5), rep("B", times=5), rep("A", times=5), rep("B", times=5))
Sapflow = c(rep(2.5, times=5), rep(2.7, times=5), rep(3.2, times=5), rep(3.4, times=5))
days = c(rep(102, times=10), rep(103, times=10))
Wat.transp(Sapflow=Sapflow, days=days, ID=ID)
```

# Index

## \* datasets

SpFl, 4

date.to.DOY, 2

datetime, 3

remove.fun, 3

SpFl, 4

SpWd\_Area\_calc, 5

tens.to.sapflow, 5

timecont, 6

Tmax.find, 7

Tmax.mean, 7

Tmaxplot, 8

Wat.transp, 9