Package ‘pollen’

October 14, 2022

Type Package
Title Analysis of Aerobiological Data
Version 0.82.0

Depends R (>= 2.10)
Imports lubridate, purrr, dplyr
Suggests covr, ggplot2, knitr, tidyr, rmarkdown, testthat
URL https://nowosad.github.io/pollen/
BugReports https://github.com/Nowosad/pollen/issues
License MIT + file LICENSE
LazyData TRUE
VignetteBuilder knitr
RoxygenNote 7.1.2
NeedsCompilation no
Author Jakub Nowosad [aut, cre] (<https://orcid.org/0000-0002-1057-3721>)
Maintainer Jakub Nowosad <nowosad.jakub@gmail.com>
Repository CRAN
Date/Publication 2021-12-03 12:20:02 UTC

R topics documented:

base_temp ................................................................. 2
gdd ................................................................. 3
gdd_data ............................................................. 4
Description

This function determines a base temperature ("tbase") based on the mean temperature of the entire season and the number of days of the ith planting to reach a given developmental stage under study. It allows to use one of four methods to calculate tbase, including: (1) the least standard deviation in GDD (Magoon and Culpepper, 1932; Stier, 1939) - "sd_gdd"; (2) the least standard deviation in days (Arnold, 1959) - "sd_day"; (3) the coefficient of variation in days (Nuttonson, 1958) - "cv_day"; (4) the regression coefficient (Hoover, 1955) - "y_i".

Usage

base_temp(tavg, d, type)

Arguments

tavg the mean temperature of the entire season for the plantings (a numerical vector, where one value is a planting)
d the number of days of the ith planting to reach a given developmental stage under study (e.g., flowering) (a numerical vector, where one value is a planting)
type either "sd_gdd", "sd_day", "cv_day", or "y_i". For the explanation of each type, see the Yang et al. 1995

Value

a numeric value representing base temperature that could be then used, for example, in GDD calculations

References


See Also
[gdd()] for calculation of growing degree days (GDD)

Examples

```r
library(pollen)
tavg <- c(25, 20, 15, 10)
d <- c(6, 11, 16, 21)
base_temp(tavg = tavg, d = d, type = "sd_gdd")
base_temp(tavg = tavg, d = d, type = "sd_day")
base_temp(tavg = tavg, d = d, type = "cv_day")
base_temp(tavg = tavg, d = d, type = "y_i")
```

---

**gdd**

_Growing Degree Days Function_

**Description**

This function calculates growing degree days (GDD) using the average of the daily maximum and minimum temperatures, a base temperature and a maximum base temperature

**Usage**

gdd(tmax, tmin, tbase, tbase_max, type = "C")

**Arguments**

tmax
  daily maximum temperature
tmin
  daily minimum temperature
tbase
  base temperature
tbase_max
  maximum base temperature
type
  either "B", "C", or "D". The default is "C". Type "B" - The heat units are calculated based on the difference between the mean daily temperature and the threshold ('tbase'). In the case when the value of 'tmin' is lower than 'tbase', then it is replaced by 'tbase'. Type "C" - same as type "B" and when the value of 'tmax' is larger than 'tbase_max', then it is replaced by 'tbase_max'. Type "D" - same as type "B" and when the value of 'tmax' is larger than 'tbase_max', then no heat units are added.
**gdd_data**

**Value**

a numeric vector with GDD values

**References**


**See Also**

[base_temp()] for determining a base temperature

**Examples**

```r
set.seed(25)
da <- data.frame(tmax = runif(100, 6, 10), tmin = runif(100, 4, 6))
gdd(tmax = da$tmax, tmin = da$tmin, tbase = 5, tbase_max = 30)
```

---

**gdd_data**  
Exemplary dataset for GDD calculations

**Description**

gdd_data A dataset containing a synthetic data of day, tmax (daily maximum temperature), and tmin (daily minimum temperature)

**Format**

A data frame with 100 rows and 3 variables:

- day
- tmax
- tmin
A Outliers Replacer Function

Description

This function finds outliers in pollen time-series and replaces them with background values.

Usage

outliers_replacer(value, date, threshold = 5, sum_percent = 100)

Arguments

- **value**: pollen concentration values
- **date**: dates
- **threshold**: a number indicating how many times outlying value needs to be larger than the background to be replaced (default is 5)
- **sum_percent**: a sum_percent parameter

Value

a new data.frame object with replaced outliers

References


Examples

data(pollen_count)
df <- subset(pollen_count, site=='Shire')
new_df <- outliers_replacer(df$birch, df$date)
identical(df, new_df)

library('purrr')
new_pollen_count <- pollen_count %>% split(., .site) %>%
map_df(~outliers_replacer(value=.hazel, date=.date, threshold=4))
pollen_count A dataset containing a synthetic data of alder, birch, and hazel pollen count in four locations ('Oz', 'Shire', 'Atlantis', 'Hundred Acre Wood') between 2007 and 2016

Format
A data frame with 8352 rows and 5 variables:
- site
- date
- alder
- birch
- hazel

pollen_index A Pollen Index Function

Description
This function calculates the Pollen Index (PI), which is implemented as the average amount of annual pollen collected based on the input data.

Usage
pollen_index(value, date)

Arguments
value pollen concentration values
date dates

Examples
data(pollen_count)
df <- subset(pollen_count, site == 'Oz')
pollen_index(value = df$birch, date = df$date)
pollen_season  
A Pollen Season Function

Description
This function calculates the start and the end of pollen season for each year

Usage
pollen_season(value, date, method, threshold = NULL)

Arguments
- value: pollen concentration values
- date: dates
- method: the pollen season method - "90", "95", "98", "Mesa", "Jager", "Lejoly", or "Driessen"
- threshold: a threshold value used for the "Driessen" method

Value
a data.frame object with year, date of pollen season start and date of pollen season end

References


Lejoly-Gabriel and Leuschner: 1983, Comparison of air-borne pollen at Louvain-la-Neuve (Belgium) and Basel (Switzerland) during 1979 and 1980, Grana 22, 59-64.

Examples

data(pollen_count)
df <- subset(pollen_count, site=='Qz')
pollen_season(value=df$birch, date=df$date, method="95")

df2 <- subset(pollen_count, site=='Atlantis')
pollen_season(value=df2$alder, date=df2$date, method="95")

library('purrr')
pollen_count %>% split(., .$site) %>%
  map_df(~pollen_season(value=.hazel, date=.date, method="95"), .id="site")
Index

* pollen
  pollen_index, 6

base_temp, 2

gdd, 3
  gdd_data, 4

outliers_replacer, 5

pollen_count, 6
  pollen_index, 6
  pollen_season, 7