Package ‘CSIndicators’

October 22, 2022

Title Climate Services’ Indicators Based on Sub-Seasonal to Decadal Predictions

Version 0.0.2

Description Set of generalised tools for the flexible computation of climate related indicators defined by the user. Each method represents a specific mathematical approach which is combined with the possibility to select an arbitrary time period to define the indicator. This enables a wide range of possibilities to tailor the most suitable indicator for each particular climate service application (agriculture, food security, energy, water management...).

This package is intended for sub-seasonal, seasonal and decadal climate predictions, but its methods are also applicable to other time-scales, provided the dimensional structure of the input is maintained. Additionally, the outputs of the functions in this package are compatible with ‘CSTools’.

This package was developed in the context of H2020 MED-GOLD (776467) and S2S4E (776787) projects. Lledó et al. (2019) <doi:10.1016/j.renene.2019.04.135>.

Depends R (>= 3.6.0)

Imports multiApply (>= 2.1.1), s2dv, stats, ClimProjDiags

Suggests testthat, CSTools, knitr, markdown, rmarkdown

VignetteBuilder knitr

License Apache License 2.0

URL https://earth.bsc.es/gitlab/es/csindicators/

BugReports https://earth.bsc.es/gitlab/es/csindicators/-/issues

Encoding UTF-8

RoxygenNote 7.2.0

NeedsCompilation no

Author Eva Rifà [cre],
Nuria Perez-Zanon [aut] (<https://orcid.org/0000-0001-8568-3071>),
Chou Chihchung [aut],
Llörenç Lledó [aut],
González-Reviriego Nube [ctb],
Marcos Raül [ctb],
AbsToProbs is a function that transforms ensemble forecast into probabilities. The Cumulative Distribution Function of a forecast is used to obtain the probabilities of each value in the ensemble. If multiple initializations (start dates) are provided, the function will create the Cumulative Distribution Function excluding the corresponding initialization.
AbsToProbs

Usage

AbsToProbs(
  data,
  dates = NULL,
  start = NULL,
  end = NULL,
  time_dim = "time",
  memb_dim = "member",
  sdate_dim = "sdate",
  ncores = NULL
)

Arguments

data A multidimensional array with named dimensions.
dates A vector of dates or a multidimensional array of dates with named dimensions matching the dimensions on parameter 'data'. By default it is NULL, to select a period this parameter must be provided.
start An optional parameter to define the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
end An optional parameter to define the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
time_dim A character string indicating the name of the temporal dimension. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object data$data can be specified. This dimension is required to subset the data in a requested period.
memb_dim A character string indicating the name of the dimension in which the ensemble members are stored.
sdate_dim A character string indicating the name of the dimension in which the initialization dates are stored.
ncores An integer indicating the number of cores to use in parallel computation.

Value

A multidimensional array with named dimensions containing the probabilites in the element data.

Examples

exp <- array(rnorm(216), dim = c(dataset = 1, member = 2, sdate = 3, ftime = 9, lat = 2, lon = 2))
exprobs <- AbsToProbs(exp)
data <- array(rnorm(5 * 2 * 61 * 1),
  c(member = 5, sdate = 2, ftime = 61, lon = 1))
Dates <- c(seq(as.Date("01-05-2000", format = "%d-%m-%Y"),
AccumulationExceedingThreshold

Accumulation of a variable when Exceeding (not exceeding) a Threshold

Description

The accumulation (sum) of a variable in the days (or time steps) that the variable is exceeding (or not exceeding) a threshold during a period. The threshold provided must be in the same units than the variable units, i.e. to use a percentile as a scalar, the function Threshold or QThreshold may be needed. Providing mean daily temperature data, the following agriculture indices for heat stress can be obtained by using this function:

• GDDSummation of daily differences between daily average temperatures and 10°C between April 1st and October 31st

Usage

AccumulationExceedingThreshold(
  data,
  threshold,
  op = ">",
  diff = FALSE,
  dates = NULL,
  start = NULL,
  end = NULL,
  time_dim = "time",
  na.rm = FALSE,
  ncores = NULL
)

Arguments

data A multidimensional array with named dimensions.
threshold a multidimensional array with named dimensions in the same units as parameter 'data' and with the common dimensions of the element 'data' of the same length.
op An operator ‘>’ (by default), ‘<’, ‘>=’ or ‘<=’.
diff A logical value indicating whether to accumulate the difference between data and threshold (TRUE) or not (FALSE by default).
**dates**
A vector of dates or a multidimensional array of dates with named dimensions matching the dimensions on parameter 'data'. By default it is NULL, to select a period this parameter must be provided.

**start**
An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.

**end**
An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.

**time_dim**
A character string indicating the name of the dimension to compute the indicator. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object data$data can be specified.

**na.rm**
A logical value indicating whether to ignore NA values (TRUE) or not (FALSE).

**ncores**
An integer indicating the number of cores to use in parallel computation.

**Value**
A multidimensional array with named dimensions containing the indicator in the element data.

**Examples**

```r
# Assuming data is already (tasmax + tasmin)/2 - 10
data <- array(rnorm(5 * 3 * 214 * 2, mean = 25, sd = 3),
c(memb = 5, sdate = 3, time = 214, lon = 2))
Dates <- c(seq(as.Date("01-05-2000", format = "%d-%m-%Y"),
as.Date("30-11-2000", format = "%d-%m-%Y"), by = 'day'),
  seq(as.Date("01-05-2001", format = "%d-%m-%Y"),
as.Date("30-11-2001", format = "%d-%m-%Y"), by = 'day'),
  seq(as.Date("01-05-2002", format = "%d-%m-%Y"),
as.Date("30-11-2002", format = "%d-%m-%Y"), by = 'day'))
GDD <- AccumulationExceedingThreshold(data, threshold = 0, start = list(1, 4),
  end = list(31, 10))
```

---

**CST_AbsToProbs**  
_Transform ensemble forecast into probabilities_

**Description**

The Cumulative Distribution Function of a forecast is used to obtain the probabilities of each value in the ensemble. If multiple initializations (start dates) are provided, the function will create the Cumulative Distribution Function excluding the corresponding initialization.
Usage

CST_AbsToProbs(
  data,
  start = NULL,
  end = NULL,
  time_dim = "ftime",
  memb_dim = "member",
  sdate_dim = "sdate",
  ncores = NULL
)

Arguments

  data          An 's2dv_cube' object as provided function CST_Load in package CSTools.
  start         An optional parameter to define the initial date of the period to select from the
                 data by providing a list of two elements: the initial date of the period and the
                 initial month of the period. By default it is set to NULL and the indicator is
                 computed using all the data provided in data.
  end           An optional parameter to define the final date of the period to select from the
                 data by providing a list of two elements: the final day of the period and the final
                 month of the period. By default it is set to NULL and the indicator is computed
                 using all the data provided in data.
  time_dim      A character string indicating the name of the temporal dimension. By default,
                 it is set to 'ftime'. More than one dimension name matching the dimensions
                 provided in the object data$data can be specified. This dimension is required
                 to subset the data in a requested period.
  memb_dim      A character string indicating the name of the dimension in which the ensemble
                 members are stored.
  sdate_dim     A character string indicating the name of the dimension in which the initialization
                 dates are stored.
  ncores        An integer indicating the number of cores to use in parallel computation.

Value

  An 's2dv_cube' object containing the probabilities in the element data.

Examples

exp <- NULL
exp$data <- array(rnorm(216), dim = c(dataset = 1, member = 2, sdate = 3,          
                          ftime = 9, lat = 2, lon = 2))
class(exp) <- 's2dv_cube'
exp_probs <- CST_AbsToProbs(exp)
exp$data <- array(rnorm(5 * 3 * 214 * 2),
                  c(member = 5, sdate = 3, ftime = 214, lon = 2))
exp$Dates[[1]] <- c(seq(as.Date("01-05-2000", format = "%d-%m-%Y"),
                       as.Date("30-11-2000", format = "%d-%m-%Y"), by = 'day'),
                       seq(as.Date("01-05-2001", format = "%d-%m-%Y"),
                       as.Date("30-11-2002", format = "%d-%m-%Y"),
                       by = 'day'))
CST_AccumulationExceedingThreshold

Accumulation of a variable when Exceeding (not exceeding) a Threshold

Description

The accumulation (sum) of a variable in the days (or time steps) that the variable is exceeding (or not exceeding) a threshold during a period. The threshold provided must be in the same units than the variable units, i.e. to use a percentile as a scalar, the function Threshold or QThreshold may be needed. Providing mean daily temperature data, the following agriculture indices for heat stress can be obtained by using this function:

- GDDSummation of daily differences between daily average temperatures and 10°C between April 1st and October 31st

Usage

CST_AccumulationExceedingThreshold(
    data,
    threshold,
    op = ">",
    diff = FALSE,
    start = NULL,
    end = NULL,
    time_dim = "ftime",
    na.rm = FALSE,
    ncores = NULL
)

Arguments

data An 's2dv_cube' object as provided by function CST_Load in package CSTools.
threshold An 's2dv_cube' object as output of a 'CST_' function in the same units as parameter 'data' and with the common dimensions of the element 'data' of the same length. A single scalar is also possible.
op An operator ‘>’ (by default), ‘<’, ‘>=’ or ‘<=’.
diff A logical value indicating whether to accumulate the difference between data and threshold (TRUE) or not (FALSE by default).
CST_MergeRefToExp

Description

Some indicators are defined for specific temporal periods (e.g.: summer from June 21st to September 21st). If the initialization forecast date is later than the one required for the indicator (e.g.: July 1st), the user may want to merge past observations, or other references, to the forecast (or hindcast) to compute the indicator. The function MergeObs2Exp takes care of this steps. If the forecast simulation doesn’t cover the required period because it is initialized too early (e.g.: Initialization on November 1st the forecast covers until the beginning of June next year), a climatology (or other references) could be added at the end of the forecast lead time to cover the desired period (e.g.: until the end of summer).

Usage

CST_MergeRefToExp(
  data1,
  data2,
  start1,
  end1,
  time_dim,
  na.rm = FALSE,
  ncores = 1)

Value

A 's2dv_cube' object containing the indicator in the element data.

Examples

exp <- NULL
exp$data <- array(rnorm(216)*200, dim = c(dataset = 1, member = 2, sdate = 3,
  ftime = 9, lat = 2, lon = 2))
class(exp) <- 's2dv_cube'
DOT <- CST_AccumulationExceedingThreshold(exp, threshold = 280)

CST_MergeRefToExp

Merge a Reference To Experiments

start
An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.

end
An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.

time_dim
A character string indicating the name of the dimension to compute the indicator. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object data$data can be specified.

na.rm
A logical value indicating whether to ignore NA values (TRUE) or not (FALSE).

ncores
An integer indicating the number of cores to use in parallel computation.
end1,
start2,
end2,
time_dim = "ftime",
sdate_dim = "sdate",
ncores = NULL
)

Arguments

data1 An 's2dv_cube' object as provided function CST_Load in package CSTools.
data2 An 's2dv_cube' object as provided function CST_Load in package CSTools.
start1 A list to define the initial date of the period to select from data1 by providing a list of two elements: the initial date of the period and the initial month of the period.
end1 A list to define the final date of the period to select from data1 by providing a list of two elements: the final day of the period and the final month of the period.
start2 A list to define the initial date of the period to select from data2 by providing a list of two elements: the initial date of the period and the initial month of the period.
end2 A list to define the final date of the period to select from data2 by providing a list of two elements: the final day of the period and the final month of the period.
time_dim A character string indicating the name of the temporal dimension. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object data$data can be specified. This dimension is required to subset the data in a requested period.
sdate_dim A character string indicating the name of the dimension in which the initialization dates are stored.
ncores An integer indicating the number of cores to use in parallel computation.

Value

A 's2dv_cube' object containing the indicator in the element data.

Examples

data_dates <- c(seq(as.Date("01-07-1993", "%d-%m-%Y", tz = 'UTC'),
as.Date("01-12-1993","%d-%m-%Y", tz = 'UTC'), "day"),
    seq(as.Date("01-07-1994", "%d-%m-%Y", tz = 'UTC'),
        as.Date("01-12-1994","%d-%m-%Y", tz = 'UTC'), "day"))
dim(data_dates) <- c(ftime = 154, sdate = 2)
data <- NULL
data$data <- array(1:(2*154*2), c(ftime = 154, sdate = 2, member= 2))
data$data$start <- data_dates
class(data) <- 's2dv_cube'
ref_dates <- seq(as.Date("01-01-1993", "%d-%m-%Y", tz = 'UTC'),
    as.Date("01-12-1994","%d-%m-%Y", tz = 'UTC'), "day")
dim(ref_dates) <- c(ftime = 350, sdate = 2)
```r
ref <- NULL
def <- array(1001:1700, c(ftime = 350, sdate = 2))
def$data <- ref$Dates$start <- ref_dates
class(ref) <- 's2dv_cube'
new_data <- CST_MergeRefToExp(data1 = ref, data2 = data,
    start1 = list(21, 6), end1 = list(30, 6),
    start2 = list(1, 7), end2 = list(21, 9))
```

---

**CST_PeriodAccumulation**

*Period Accumulation on 's2dv_cube' objects*

---

### Description

Period Accumulation computes the sum (accumulation) of a given variable in a period. Providing precipitation data, two agriculture indices can be obtained by using this function:

- **SprR** - Spring Total Precipitation: The total precipitation from April 21th to June 21st
- **HarR** - Harvest Total Precipitation: The total precipitation from August 21st to October 21st

### Usage

```r
CST_PeriodAccumulation(
da, 
start = NULL, 
end = NULL, 
time_dim = "ftime", 
na.rm = FALSE, 
ncores = NULL
)
```

### Arguments

- **data**
  - An 's2dv_cube' object as provided function CST_Load in package CSTools.

- **start**
  - An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.

- **end**
  - An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.

- **time_dim**
  - A character string indicating the name of the dimension to compute the indicator. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object data$data can be specified.

- **na.rm**
  - A logical value indicating whether to ignore NA values (TRUE) or not (FALSE).

- **ncores**
  - An integer indicating the number of cores to use in parallel computation.
CST_PeriodMean

Value

A 's2dv_cube' object containing the indicator in the element data.

Examples

```r
exp <- NULL
class(exp) <- "s2dv_cube"
exp$data <- array(rnorm(216)*200, dim = c(dataset = 1, member = 2, sdate = 3, ftime = 9, lat = 2, lon = 2))
exp$data <- array(rnorm(5 * 3 * 214 * 2), c(memb = 5, sdate = 3, ftime = 214, lon = 2))
exp$Dates[[1]] <- c(seq(as.Date("01-05-2000", format = "%d-%m-%Y"), as.Date("30-11-2000", format = "%d-%m-%Y"), by = "day"), seq(as.Date("01-05-2001", format = "%d-%m-%Y"), as.Date("30-11-2001", format = "%d-%m-%Y"), by = "day"), seq(as.Date("01-05-2002", format = "%d-%m-%Y"), as.Date("30-11-2002", format = "%d-%m-%Y"), by = "day"))
exp$data <- array(rnorm(216)*200, dim = c(dataset = 1, member = 2, sdate = 3, ftime = 9, lat = 2, lon = 2))
exp$Dates[[1]] <- c(seq(as.Date("01-05-2000", format = "%d-%m-%Y"), as.Date("30-11-2000", format = "%d-%m-%Y"), by = "day"), seq(as.Date("01-05-2001", format = "%d-%m-%Y"), as.Date("30-11-2001", format = "%d-%m-%Y"), by = "day"), seq(as.Date("01-05-2002", format = "%d-%m-%Y"), as.Date("30-11-2002", format = "%d-%m-%Y"), by = "day"))
TP <- CST_PeriodAccumulation(exp)
SprR <- CST_PeriodAccumulation(exp, start = list(21, 4), end = list(21, 6))
HarR <- CST_PeriodAccumulation(exp, start = list(21, 8), end = list(21, 10))
```

CST_PeriodMean

Period Mean on 's2dv_cube' objects

Description

Period Mean computes the average (mean) of a given variable in a period. Providing temperature data, two agriculture indices can be obtained by using this function:

- **GST**: Growing Season average Temperature: The average temperature from April 1st to October 31st
- **SprTX**: Spring Average Maximum Temperature: The average daily maximum temperature from April 1st to May 31st

Usage

```r
CST_PeriodMean(
  data,
  start = NULL,
  end = NULL,
  time_dim = "ftime",
  na.rm = FALSE,
  ncores = NULL
)
```
### Arguments

- **data**: An 's2dv_cube' object as provided function CST_Load in package CSTools.
- **start**: An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
- **end**: An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
- **time_dim**: A character string indicating the name of the dimension to compute the indicator. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object data$data can be specified.
- **na.rm**: A logical value indicating whether to ignore NA values (TRUE) or not (FALSE).
- **ncores**: An integer indicating the number of cores to use in parallel computation.

### Value

An 's2dv_cube' object containing the indicator in the element data.

### Examples

```r
exp <- NULL
exp$data <- array(rnorm(45), dim = c(member = 7, ftime = 8))
class(exp) <- 's2dv_cube'
exp$Dates$start <- c(seq(as.Date("01-07-1993", "%d-%m-%Y", tz = "UTC"),
                          as.Date("01-08-1993","%d-%m-%Y", tz = "UTC"), "day"),
                          seq(as.Date("01-07-1994", "%d-%m-%Y", tz = "UTC"),
                              as.Date("01-08-1994","%d-%m-%Y", tz = "UTC"), "day"))
SA <- CST_PeriodMean(exp)
```

### Description

From the user's perspective, an absolute threshold can be very useful for a specific needs (e.g.: grape variety). However, this absolute threshold could be transformed to a relative threshold in order to get its frequency in a given dataset. Therefore, the function QThreshold returns the probability of an absolute threshold. This is done by computing the Cumulative Distribution Function of a sample and leaving one out. The sample used will depend on the dimensions of the data provided and the dimension names provided in sdate_dim and memb_dim parameters:
Usage

CST_QThreshold(  
data,  
threshold,  
start = NULL,  
end = NULL,  
time_dim = "ftime",  
memb_dim = "member",  
sdate_dim = "sdate",  
ncores = NULL  
)

Arguments

data
An 's2dv_cube' object as provided function CST_Load in package CSTools.

threshold
An 's2dv_cube' object as output of a 'CST_' function in the same units as parameter 'data' and with the common dimensions of the element 'data' of the same length. A single scalar is also possible.

start
An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.

end
An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.

time_dim
A character string indicating the name of the temporal dimension. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object data$data can be specified. This dimension is required to subset the data in a requested period.

memb_dim
A character string indicating the name of the dimension in which the ensemble members are stored.

sdate_dim
A character string indicating the name of the dimension in which the initialization dates are stored.

ncores
An integer indicating the number of cores to use in parallel computation.

Details

- If a forecast (hindcast) has dimensions member and start date, and both must be used in the sample, their names should be passed in sdate_dim and memb_dim.
- If a forecast (hindcast) has dimensions member and start date, and only start date must be used in the sample (the calculation is done in each separate member), memb_dim can be set to NULL.
- If a reference (observations) has start date dimension, the sample used is the start date dimension.
- If a reference (observations) doesn’t have start date dimension, the sample used must be specified in sdate_dim parameter.
Value

An 's2dv_cube' object containing the probability of an absolute threshold in the element data.

Examples

```r
threshold <- 26
exps <- NULL
exp$data <- array(abs(rnorm(112)*26), dim = c(member = 7, sdate = 8, ftime = 2))
class(exp) <- 's2dv_cube'
exp_probs <- CST_QThreshold(exp, threshold)
exp$data <- array(rnorm(5 * 3 * 214 * 2), c(member = 5, sdate = 3, ftime = 214, lon = 2))
exp$Dates[[1]] <- c(seq(as.Date("01-05-2000", format = "%d-%m-%Y"),
                   as.Date("30-11-2000", format = "%d-%m-%Y"), by = 'day'),
                   seq(as.Date("01-05-2001", format = "%d-%m-%Y"),
                   as.Date("30-11-2001", format = "%d-%m-%Y"), by = 'day'),
                   seq(as.Date("01-05-2002", format = "%d-%m-%Y"),
                   as.Date("30-11-2002", format = "%d-%m-%Y"), by = 'day'))
exp_probs <- CST_QThreshold(exp, threshold)
```

---

CST_SelectPeriodOnData

*Select a period on Data on 's2dv_cube' objects*

Description

Auxiliary function to subset data for a specific period.

Usage

```r
CST_SelectPeriodOnData(data, start, end, time_dim = "ftime", ncores = NULL)
```

Arguments

data  An 's2dv_cube' object as provided function CST_Load in package CSTools.

start A parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period.

end A parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period.

time_dim A character string indicating the name of the dimension to compute select the dates. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object data$data can be specified.

ncores An integer indicating the number of cores to use in parallel computation.
Value

A 's2dv_cube' object containing the subset of the object data$data during the period requested from start to end.

Examples

```r
exp <- NULL
exp$data <- array(rnorm(5 * 3 * 214 * 2),
c(memb = 5, sdate = 3, ftime = 214, lon = 2))
ex$Dates$start <- c(seq(as.Date("01-05-2000", format = "%d-%m-%Y"),
as.Date("30-11-2000", format = "%d-%m-%Y"), by = 'day'),
seq(as.Date("01-05-2001", format = "%d-%m-%Y"),
as.Date("30-11-2001", format = "%d-%m-%Y"), by = 'day'),
seq(as.Date("01-05-2002", format = "%d-%m-%Y"),
as.Date("30-11-2002", format = "%d-%m-%Y"), by = 'day'))
class(exp) <- 's2dv_cube'
Period <- CST_SelectPeriodOnData(exp, start = list(21, 6), end = list(21, 9))
```

CST_Threshold

**Absolute value of a relative threshold (percentile)**

Description

Frequently, thresholds are defined by a percentile that may correspond to a different absolute value depending on the variable, gridpoint and also julian day (time). This function calculates the corresponding value of a percentile given a dataset.

Usage

```r
CST_Threshold(
  data, 
  threshold, 
  start = NULL, 
  end = NULL, 
  time_dim = "ftime", 
  memb_dim = "member", 
  sdate_dim = "sdate", 
  na.rm = FALSE, 
  ncores = NULL
)
```

Arguments

data An 's2dv_cube' object as provided function CST_Load in package CSTools.
threshold A single scalar or vector indicating the relative threshold(s).
start

An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.

end

An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.

time_dim

A character string indicating the name of the temporal dimension. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object data$data can be specified. This dimension is required to subset the data in a requested period.

memb_dim

A character string indicating the name of the dimension in which the ensemble members are stored. When set to NULL, threshold is computed for individual members.

sdate_dim

A character string indicating the name of the dimension in which the initialization dates are stored.

na.rm

A logical value indicating whether to ignore NA values (TRUE) or not (FALSE).

ncores

An integer indicating the number of cores to use in parallel computation.

Value

An 's2dv_cube' object containing the corresponding values of a percentile in the element data.

Examples

```r
threshold <- 0.9
exp <- NULL
exp$data <- array(rnorm(5 * 3 * 214 * 2),
                  dim = c(member = 5, sdate = 3, ftime = 214, lon = 2))
exp$Dates$start <- c(seq(as.Date("01-05-2000", format = "%d-%m-%Y"),
                            as.Date("30-11-2000", format = "%d-%m-%Y"), by = "day"),
                       seq(as.Date("01-05-2001", format = "%d-%m-%Y"),
                            as.Date("30-11-2001", format = "%d-%m-%Y"), by = "day"),
                       seq(as.Date("01-05-2002", format = "%d-%m-%Y"),
                            as.Date("30-11-2002", format = "%d-%m-%Y"), by = "day"))
class(exp) <- 's2dv_cube'
exp_probs <- CST_Threshold(exp, threshold, start = list(21, 4), end = list(21, 6))
```

Total Spell Time Exceeding Threshold
**CST_TotalSpellTimeExceedingThreshold**

**Description**

The number of days (when daily data is provided) that are part of a spell (defined by its minimum length e.g. 6 consecutive days) that exceed (or not exceed) a threshold are calculated with `TotalSpellTimeExceedingThreshold`. This function allows to compute indicators widely used in Climate Services, such as:

- **WSDI** Warm Spell Duration Index that count the total number of days with at least 6 consecutive days when the daily temperature maximum exceeds its 90th percentile.

This function requires the data and the threshold to be in the same units. The 90th percentile can be translate into absolute values given a reference dataset using function `Threshold` or the data can be transform into probabilites by using function `AbsToProbs`. See section @examples.

**Usage**

```r
CST_TotalSpellTimeExceedingThreshold(
  data,
  threshold,
  spell,
  op = ">",
  start = NULL,
  end = NULL,
  time_dim = "ftime",
  ncores = NULL
)
```

**Arguments**

- **data**: An 's2dv_cube' object as provided by function `CST_Load` in package CSTools.
- **threshold**: An 's2dv_cube' object as output of a 'CST_' function in the same units as parameter 'data' and with the common dimensions of the element 'data' of the same length. A single scalar is also possible. If `tim_dim` is in the dimension (with the same length as `data`), the comparison will be done day by day.
- **spell**: A scalar indicating the minimum length of the spell.
- **op**: An operator `>` (by default), `<`, `>=` or `<=`.
- **start**: An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in `data`.
- **end**: An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in `data`.
- **time_dim**: A character string indicating the name of the dimension to compute the indicator. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object `data$data` can be specified.
- **ncores**: An integer indicating the number of cores to use in parallel computation.
Value

An ‘s2dv_cube’ object containing the indicator in the element data.

See Also

[Threshold()] and [AbsToProbs()].

Examples

```r
exp <- NULL
ev$data <- array(rnorm(5 * 3 * 214 * 2)*23, 
  c(member = 5, sdate = 3, ftime = 214, lon = 2))
ev$Dates$start <- c(seq(as.Date("01-05-2000", format = "%d-%m-%Y"),
  as.Date("30-11-2000", format = "%d-%m-%Y"), by = 'day'),
  seq(as.Date("01-05-2001", format = "%d-%m-%Y"),
  as.Date("30-11-2001", format = "%d-%m-%Y"), by = 'day'),
  seq(as.Date("01-05-2002", format = "%d-%m-%Y"),
  as.Date("30-11-2002", format = "%d-%m-%Y"), by = 'day'))
class(exp) <- 's2dv_cube'
TTSET <- CST_TotalSpellTimeExceedingThreshold(exp, threshold = 23, spell = 3)
```

---

CST_TotalTimeExceedingThreshold

*Total Time of a variable Exceeding (not exceeding) a Threshold*

Description

The Total Time of a variable exceeding (or not) a Threshold returns the total number of days (if the data provided is daily, or the corresponding units to the data frequency provided) that a variable is exceeding a threshold during a period. The threshold provided must be in the same units than the variable units, i.e. to use a percentile as a scalar, the function AbsToProbs or QThreshold may be needed (see examples). Providing maximum temperature daily data, the following agriculture indices for heat stress can be obtained by using this function:

- **SU35**: Total count of days when daily maximum temperatures exceed 35°C in the seven months from the start month given (e.g. from April to October for start month of April).
- **SU36**: Total count of days when daily maximum temperatures exceed 36 between June 21st and September 21st
- **SU40**: Total count of days when daily maximum temperatures exceed 40 between June 21st and September 21st
- **Spr32**: Total count of days when daily maximum temperatures exceed 32 between April 21st and June 21st
CST_TotalTimeExceedingThreshold

Usage

CST_TotalTimeExceedingThreshold(
  data,
  threshold,
  op = ">",
  start = NULL,
  end = NULL,
  time_dim = "ftime",
  na.rm = FALSE,
  ncores = NULL
)

Arguments

data An 's2dv_cube' object as provided by function CST_Load in package CSTools.
threshold An 's2dv_cube' object as output of a 'CST_' function in the same units as parameter data and with the common dimensions of the element data of the same length (e.g. an array with the same lengths of longitude and latitude). A single scalar is also possible (for the case of comparing all grid points with the same scalar).
op An operator '>' (by default), '<', '>=', or '<='. 
start An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
end An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
time_dim A character string indicating the name of the dimension to compute the indicator. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object data$data can be specified.
na.rm A logical value indicating whether to ignore NA values (TRUE) or not (FALSE).
ncores An integer indicating the number of cores to use in parallel computation.

Value

An 's2dv_cube' object containing the indicator in the element data.

Examples

exp <- NULL
exp$data <- array(abs(rnorm(5 * 3 * 214 * 2)*280),
  c(member = 5, sdate = 3, ftime = 214, lon = 2))
exp$Dates$start <- c(seq(as.Date("01-05-2000", format = "%d-%m-%Y"),
  as.Date("30-11-2000", format = "%d-%m-%Y"), by = 'day'),
  seq(as.Date("01-05-2001", format = "%d-%m-%Y"),
as.Date("30-11-2001", format = "%d-%m-%Y"), by = 'day'),
seq(as.Date("01-05-2002", format = "%d-%m-%Y"),
as.Date("30-11-2002", format = "%d-%m-%Y"), by = 'day'))
class(exp) <- 's2dv_cube'
DOT <- CST_TotalTimeExceedingThreshold(exp, threshold = 280)

CST_WindCapacityFactor

Wind capacity factor on s2dv_cube objects

Description

Wind capacity factor computes the wind power generated by a specific wind turbine model under
specific wind speed conditions, and expresses it as a fraction of the rated capacity (i.e. maximum
power) of the turbine.

It is computed by means of a tabular power curve that relates wind speed to power output. The	

Usage

CST_WindCapacityFactor(
  wind,
  IEC_class = c("I", "I/II", "II", "II/III", "III"),
  start = NULL,
  end = NULL,
  time_dim = "ftime",
  ncores = NULL
)

Arguments

wind        An s2dv_cube object with instantaneous wind speeds expressed in m/s.
IEC_class   A string indicating the IEC wind class (see IEC 61400-1) of the turbine to be
            selected. Classes 'I', 'II' and 'III' are suitable for sites with an annual mean
            wind speed of 10, 8.5 and 7.5 m/s respectively. Classes 'I/II' and 'II/III'
            indicate intermediate turbines that fit both classes. More details of the five tur-
            bines and a plot of its power curves can be found in Lledó et al. (2019).
start       An optional parameter to defined the initial date of the period to select from the
            data by providing a list of two elements: the initial date of the period and the
            initial month of the period. By default it is set to NULL and the indicator is
            computed using all the data provided in data.
end         An optional parameter to defined the final date of the period to select from the
            data by providing a list of two elements: the final day of the period and the final
            month of the period. By default it is set to NULL and the indicator is computed
            using all the data provided in data.
CST_WindPowerDensity

**time_dim**
A character string indicating the name of the dimension to compute the indicator. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object data$data can be specified.

**ncores**
An integer indicating the number of cores to use in parallel computation for temporal subsetting.

**Value**
An s2dv_cube object containing the Wind Capacity Factor (unitless).

**Author(s)**
Llorenç Lledó, <llledo@bsc.es>

**References**


**Examples**
```r
wind <- array(rweibull(n = 100, shape = 2, scale = 6), c(member = 10, lat = 2, lon = 5))
wind <- CSTools::s2dv_cube(data = wind, lat = c(40, 41), lon = 1:5,
                   Variable = list(varName = 'sfcWind', level = 'Surface'),
                   Datasets = 'synthetic', when = Sys.time(),
                   Dates = list(start = '1990-01-01 00:00:00', end = '1990-01-01 00:00:00'),
                   source_file = NA)
WCF <- CST_WindCapacityFactor(wind, IEC_class = "III")
```

---

**CST_WindPowerDensity**

*Wind power density on s2dv_cube objects*

**Description**

Wind Power Density computes the wind power that is available for extraction per square meter of swept area.

It is computed as 0.5*ro*wspd^3. As this function is non-linear, it will give inaccurate results if used with period means.
CST_WindPowerDensity

Usage

CST_WindPowerDensity(
    wind,
    ro = 1.225,
    start = NULL,
    end = NULL,
    time_dim = "ftime",
    ncores = NULL
)

Arguments

wind An s2dv_cube object with instantaneous wind speeds expressed in m/s obtained from CST_Load or s2dv_cube functions from CSTools package.

ro A scalar, or alternatively a multidimensional array with the same dimensions as wind, with the air density expressed in kg/m^3. By default it takes the value 1.225, the standard density of air at 15°C and 1013.25 hPa.

start An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.

eend An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.

time_dim A character string indicating the name of the dimension to compute the indicator. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object data$data can be specified.

ncores An integer indicating the number of cores to use in parallel computation for temporal subsetting.

Value

An s2dv_cube object containing Wind Power Density expressed in W/m^2.

Author(s)

Llorenç Lledó, <llledo@bsc.es>

Examples

wind <- array(rweibull(n = 100, shape = 2, scale = 6), c(member = 10, lat = 2, lon = 5))
wind <- CSTools::s2dv_cube(data = wind, lat = c(40, 41), lon = 1:5,
    Variable = list(varName = 'sfcWind', level = 'Surface'),
    Datasets = 'synthetic', when = Sys.time(),
    Dates = list(start = '1990-01-01 00:00:00', end = '1990-01-01 00:00:00'),
    source_file = NA)

WPD <- CST_WindPowerDensity(wind)
**MergeRefToExp**

Merge a Reference To Experiments

### Description

Some indicators are defined for specific temporal periods (e.g.: summer from June 21st to September 21st). If the initialization forecast date is later than the one required for the indicator (e.g.: July 1st), the user may want to merge past observations, or other reference, to the forecast (or hindcast) to compute the indicator. The function `MergeObs2Exp` takes care of this steps.

### Usage

```r
MergeRefToExp(
  data1,
  dates1,
  start1,
  end1,
  data2,
  dates2,
  start2,
  end2,
  time_dim = "time",
  sdate_dim = "sdate",
  ncores = NULL
)
```

### Arguments

- **data1**: A multidimensional array with named dimensions.
- **dates1**: a vector of dates or a multidimensional array of dates with named dimensions matching the dimensions on parameter 'data1'.
- **start1**: A list to define the initial date of the period to select from data1 by providing a list of two elements: the initial date of the period and the initial month of the period.
- **end1**: A list to define the final date of the period to select from data1 by providing a list of two elements: the final day of the period and the final month of the period.
- **data2**: A multidimensional array with named dimensions.
- **dates2**: A vector of dates or a multidimensional array of dates with named dimensions matching the dimensions on parameter 'data2'.
- **start2**: A list to define the initial date of the period to select from data2 by providing a list of two elements: the initial date of the period and the initial month of the period.
- **end2**: A list to define the final date of the period to select from data2 by providing a list of two elements: the final day of the period and the final month of the period.
- **time_dim**: "time".
- **sdate_dim**: "sdate".
- **ncores**: NULL.
time_dim  A character string indicating the name of the temporal dimension. By default, it is set to ‘ftime’. More than one dimension name matching the dimensions provided in the object data$data can be specified. This dimension is required to subset the data in a requested period.

sdate_dim  A character string indicating the name of the dimension in which the initialization dates are stored.

cores  An integer indicating the number of cores to use in parallel computation.

Value
A multidimensional array with named dimensions.

Examples

```r
data_dates <- c(seq(as.Date("01-07-1993", "%d-%m-%Y", tz = "UTC"),
  as.Date("01-12-1993","%d-%m-%Y", tz = "UTC"), "day"),
  seq(as.Date("01-07-1994", "%d-%m-%Y", tz = "UTC"),
  as.Date("01-12-1994","%d-%m-%Y", tz = "UTC"), "day"))
dim(data_dates) <- c(time = 154, sdate = 2)
ref_dates <- seq(as.Date("01-01-1993", "%d-%m-%Y", tz = "UTC"),
  as.Date("01-12-1994","%d-%m-%Y", tz = "UTC"), "day")
dim(ref_dates) <- c(time = 350, sdate = 2)
ref <- array(1001:1700, c(time = 350, sdate = 2))
data <- array(1:(2*154*2), c(time = 154, sdate = 2, member= 2))
new_data <- MergeRefToExp(data1 = ref, dates1 = ref_dates, start1 = list(21, 6),
  end1 = list(30, 6), data2 = data, dates2 = data_dates,
  start2 = list(1, 7), end = list(21, 9))
```
Arguments

- **data**: A multidimensional array with named dimensions.
- **dates**: A vector of dates or a multidimensional array of dates with named dimensions matching the dimensions on parameter 'data'. By default it is NULL, to select a period this parameter must be provided.
- **start**: An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
- **end**: An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
- **time_dim**: A character string indicating the name of the dimension to compute the indicator. By default, it is set to 'time'. More than one dimension name matching the dimensions provided in the object `data$data` can be specified.
- **na.rm**: A logical value indicating whether to ignore NA values (TRUE) or not (FALSE).
- **ncores**: An integer indicating the number of cores to use in parallel computation.

Value

A multidimensional array with named dimensions containing the indicator in the element data.

Examples

```r
exp <- array(rnorm(216)*200, dim = c(dataset = 1, member = 2, sdate = 3,
                                           ftime = 9, lat = 2, lon = 2))
TP <- PeriodAccumulation(exp, time_dim = 'ftime')
data <- array(rnorm(5 * 3 * 214 * 2),
               c(memb = 5, sdate = 3, time = 214, lon = 2))
# ftime tested
Dates <- c(seq(as.Date("01-05-2000", format = "%d-%m-%Y"),
               as.Date("30-11-2000", format = "%d-%m-%Y"), by = 'day'),
           seq(as.Date("01-05-2001", format = "%d-%m-%Y"),
               as.Date("30-11-2001", format = "%d-%m-%Y"), by = 'day'),
           seq(as.Date("01-05-2002", format = "%d-%m-%Y"),
               as.Date("30-11-2002", format = "%d-%m-%Y"), by = 'day'))
SprR <- PeriodAccumulation(data, dates = Dates, start = list(21, 4), end = list(21, 6))
HarR <- PeriodAccumulation(data, dates = Dates, start = list(21, 8), end = list(21, 10))
```
Period Mean computes the average (mean) of a given variable in a period. Providing temperature data, two agriculture indices can be obtained by using this function:

- **GST** Growing Season average Temperature: The average temperature from April 1st to October 31st
- **SprTX** Spring Average Maximum Temperature: The average daily maximum temperature from April 1st to May 31st

### Usage

```r
PeriodMean(
  data,
  dates = NULL,
  start = NULL,
  end = NULL,
  time_dim = "time",
  na.rm = FALSE,
  ncores = NULL
)
```

### Arguments

- **data**
  
  A multidimensional array with named dimensions.

- **dates**
  
  A vector of dates or a multidimensional array of dates with named dimensions matching the dimensions on parameter ‘data’. By default it is NULL, to select a period this parameter must be provided.

- **start**
  
  An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in `data`.

- **end**
  
  An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in `data`.

- **time_dim**
  
  A character string indicating the name of the dimension to compute the indicator. By default, it is set to ‘ftime’. More than one dimension name matching the dimensions provided in the object `data$data` can be specified.

- **na.rm**
  
  A logical value indicating whether to ignore NA values (TRUE) or not (FALSE).

- **ncores**
  
  An integer indicating the number of cores to use in parallel computation.
Value

A multidimensional array with named dimensions containing the indicator in the element data.

Examples

```r
exp <- array(rnorm(56), dim = c(member = 7, ftime = 8))
SA <- PeriodMean(exp, time_dim = 'ftime')
```

QThreshold

Transform an absolute threshold into probabilities

Description

From the user’s perspective, an absolute threshold can be very useful for a specific needs (e.g.: grape variety). However, this absolute threshold could be transformed to a relative threshold in order to get its frequency in a given dataset. Therefore, the function QThreshold returns the probability of an absolute threshold. This is done by computing the Cumulative Distribution Function of a sample and leaving-one-out. The sample used will depend on the dimensions of the data provided and the dimension names provided in sdate_dim and memb_dim parameters:

- If a forecast (hindcast) has dimensions member and start date, and both must be used in the sample, their names should be passed in sdate_dim and memb_dim.
- If a forecast (hindcast) has dimensions member and start date, and only start date must be used in the sample (the calculation is done in each separate member), memb_dim can be set to NULL.
- If a reference (observations) has start date dimension, the sample used is the start date dimension.
- If a reference (observations) doesn’t have start date dimension, the sample used must be specified in sdate_dim parameter.

Usage

```r
QThreshold(
  data,
  threshold,
  dates = NULL,
  start = NULL,
  end = NULL,
  time_dim = "time",
  memb_dim = "member",
  sdate_dim = "sdate",
  ncores = NULL
)
```
Arguments

- **data**: A multidimensional array with named dimensions.
- **threshold**: A multidimensional array with named dimensions in the same units as parameter 'data' and with the common dimensions of the element 'data' of the same length.
- **dates**: A vector of dates or a multidimensional array of dates with named dimensions matching the dimensions on parameter 'data'. By default it is NULL, to select a period this parameter must be provided.
- **start**: An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
- **end**: An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
- **time_dim**: A character string indicating the name of the temporal dimension. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object data$data can be specified. This dimension is required to subset the data in a requested period.
- **memb_dim**: A character string indicating the name of the dimension in which the ensemble members are stored.
- **sdate_dim**: A character string indicating the name of the dimension in which the initialization dates are stored.
- **ncores**: An integer indicating the number of cores to use in parallel computation.

Value

A multidimensional array with named dimensions containing the probability of an absolute threshold in the element data.

Examples

```r
threshold = 25
data <- array(rnorm(5 * 3 * 20 * 2, mean = 26),
c(member = 5, sdate = 3, time = 20, lon = 2))
threshold_q <- QThreshold(data, threshold)
```

---

**SelectPeriodOnData**

*Select a period on Data on multidimensional array objects*

Description

Auxiliary function to subset data for a specific period.
SelectPeriodOnDates

Usage

SelectPeriodOnData(data, dates, start, end, time_dim = "ftime", ncores = NULL)

Arguments

data A multidimensional array with named dimensions.
dates A vector of dates or a multidimensional array of dates with named dimensions.
start An optional parameter to defined the initial date of the period to select from the
    data by providing a list of two elements: the initial date of the period and the
    initial month of the period.
end An optional parameter to defined the final date of the period to select from the
    data by providing a list of two elements: the final day of the period and the final
    month of the period.
time_dim A character string indicating the name of the dimension to compute select the
dates. By default, it is set to ‘ftime’. More than one dimension name matching
the dimensions provided in the object data$data can be specified.
ncores An integer indicating the number of cores to use in parallel computation.

Value

A multidimensional array with named dimensions containing the subset of the object data during
the period requested from start to end.

Examples

data <- array(rnorm(5 * 3 * 214 * 2),
c(memb = 5, sdate = 3, ftime = 214, lon = 2))
Dates <- c(seq(as.Date("01-05-2000", format = "%d-%m-%Y"),
as.Date("30-11-2000", format = "%d-%m-%Y"), by = 'day'),
seq(as.Date("01-05-2001", format = "%d-%m-%Y"),
as.Date("30-11-2001", format = "%d-%m-%Y"), by = 'day'),
seq(as.Date("01-05-2002", format = "%d-%m-%Y"),
as.Date("30-11-2002", format = "%d-%m-%Y"), by = 'day'))
dim(Dates) <- c(ftime = 214, sdate = 3)
Period <- SelectPeriodOnData(data, Dates, start = list(21, 6), end = list(21, 9))

SelectPeriodOnDates

Select a period on Dates

Description

Auxiliary function to subset dates for a specific period.

Usage

SelectPeriodOnDates(dates, start, end, time_dim = "ftime", ncores = NULL)
Threshold

Arguments

dates A vector of dates or a multidimensional array of dates with named dimensions.
start An optional parameter to define the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period.
end An optional parameter to define the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period.
time_dim A character string indicating the name of the dimension to compute select the dates. By default, it is set to 'time'. More than one dimension name matching the dimensions provided in the object data$data can be specified.
ncores An integer indicating the number of cores to use in parallel computation.

Value

A multidimensional array with named dimensions containing the subset of the vector dates during the period requested from start to end.

Examples

Dates <- c(seq(as.Date("01-05-2000", format = "%d-%m-%Y"),
    as.Date("30-11-2000", format = "%d-%m-%Y"), by = 'day'),
    seq(as.Date("01-05-2001", format = "%d-%m-%Y"),
    as.Date("30-11-2001", format = "%d-%m-%Y"), by = 'day'),
    seq(as.Date("01-05-2002", format = "%d-%m-%Y"),
    as.Date("30-11-2002", format = "%d-%m-%Y"), by = 'day'))
Period <- SelectPeriodOnDates(Dates, start = list(21, 6), end = list(21, 9))

Threshold

Absolute value of a relative threshold (percentile)

Description

Frequently, thresholds are defined by a percentile that may correspond to a different absolute value depending on the variable, gridpoint and also julian day (time). This function calculates the corresponding value of a percentile given a dataset.

Usage

Threshold(
    data,
    threshold,
    dates = NULL,
    start = NULL,
    end = NULL,
    time_dim = "time",
)
Threshold

```r
memb_dim = "member",
sdate_dim = "sdate",
na.rm = FALSE,
ncores = NULL
```

**Arguments**

- **data**: A multidimensional array with named dimensions.
- **threshold**: A single scalar or vector indicating the relative threshold(s).
- **dates**: A vector of dates or a multidimensional array of dates with named dimensions matching the dimensions on parameter 'data'. By default it is NULL, to select a period this parameter must be provided.
- **start**: An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
- **end**: An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
- **time_dim**: A character string indicating the name of the temporal dimension. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object data$data can be specified. This dimension is required to subset the data in a requested period.
- **memb_dim**: A character string indicating the name of the dimension in which the ensemble members are stored. When set it to NULL, threshold is computed for individual members.
- **sdate_dim**: A character string indicating the name of the dimension in which the initialization dates are stored.
- **na.rm**: A logical value indicating whether to ignore NA values (TRUE) or not (FALSE).
- **ncores**: An integer indicating the number of cores to use in parallel computation.

**Value**

A multidimensional array with named dimensions containing the corresponding values of a percentile in the element data.

**Examples**

```r
threshold <- 0.9

data <- array(rnorm(25 * 3 * 214 * 2, mean = 26),
              c(member = 25, sdate = 3, time = 214, lon = 2))
thres_q <- Threshold(data, threshold)

data <- array(rnorm(1 * 3 * 214 * 2),
              c(member = 1, sdate = 3, time = 214, lon = 2))
res <- Threshold(data, threshold)
```
TotalSpellTimeExceedingThreshold

Total Spell Time Exceeding Threshold

Description

The number of days (when daily data is provided) that are part of a spell (defined by its minimum length e.g. 6 consecutive days) that exceed (or not exceed) a threshold are calculated with TotalSpellTimeExceedingThreshold. This function allows to compute indicators widely used in Climate Services, such as:

- **WSDI Warm Spell Duration Index** that count the total number of days with at least 6 consecutive days when the daily temperature maximum exceeds its 90th percentile.

This function requires the data and the threshold to be in the same units. The 90th percentile can be translate into absolute values given a reference dataset using function Threshold or the data can be transform into probabilities by using function AbsToProbs. See section @examples.

Usage

```r
TotalSpellTimeExceedingThreshold(
  data, 
  threshold, 
  spell, 
  op = ">", 
  dates = NULL, 
  start = NULL, 
  end = NULL, 
  time_dim = "time", 
  ncores = NULL
)
```

Arguments

- **data** A multidimensional array with named dimensions.
- **threshold** A multidimensional array with named dimensions in the same units as parameter 'data' and with the common dimensions of the element 'data' of the same length. If `time_dim` is in the dimension (with the same length as `data`), the comparison will be done day by day.
- **spell** A scalar indicating the minimum length of the spell.
- **op** An operator ‘>’ (by default), ‘<’, ‘>=’ or ‘<=’.
- **dates** A vector of dates or a multidimensional array of dates with named dimensions matching the dimensions on parameter 'data'. By default it is NULL, to select a period this parameter must be provided.
- **start** An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
**TotalTimeExceedingThreshold**

Total Time of a variable Exceeding (not exceeding) a Threshold

#### Description

The Total Time of a variable exceeding (or not) a Threshold returns the total number of days (if the data provided is daily, or the corresponding units to the data frequency provided) that a variable is exceeding a threshold during a period. The threshold provided must be in the same units than the variable units, i.e. to use a percentile as a threshold, the function `Threshold` or `QThreshold` may be needed (see examples). Providing maximum temperature daily data, the following agriculture indices for heat stress can be obtained by using this function:

- **SU35** Total count of days when daily maximum temperatures exceed 35°C
- **SU36** Total count of days when daily maximum temperatures exceed 36 between June 21st and September 21st
- **SU40** Total count of days when daily maximum temperatures exceed 40 between June 21st and September 21st
- **Spr32** Total count of days when daily maximum temperatures exceed 32 between April 21st and June 21st

#### Details

This function considers NA values as the end of the spell. For a different behaviour consider to modify the 'data' input by substituting NA values by values exceeding the threshold.

#### Value

A multidimensional array with named dimensions containing the indicator in the element data.

#### See Also

[Threshold()] and [AbsToProbs()].

#### Examples

```r
data <- array(rnorm(120), c(member = 1, sdate = 2, time = 20, lat = 4))
threshold <- array(rnorm(4), c(lat = 4))
total <- TotalSpellTimeExceedingThreshold(data, threshold, spell = 6)
```

---

**TotalTimeExceedingThreshold**

Total Time of a variable Exceeding (not exceeding) a Threshold

---

end An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.

time_dim A character string indicating the name of the dimension to compute the indicator. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object `data$data` can be specified.

ncores An integer indicating the number of cores to use in parallel computation.
TotalTimeExceedingThreshold

Usage

TotalTimeExceedingThreshold(
  data,
  threshold,
  op = ">",
  dates = NULL,
  start = NULL,
  end = NULL,
  time_dim = "time",
  na.rm = FALSE,
  ncores = NULL
)

Arguments

data A multidimensional array with named dimensions.
threshold A multidimensional array with named dimensions in the same units as parameter
data and with the common dimensions of the element data of the same length
(e.g. an array with the same lengths of longitude and latitude). A single scalar
is also possible (for the case of comparing all grid points with the same scalar).
op A operator '>' (by default), '<', '>=' or '<='. 
dates A vector of dates or a multidimensional array of dates with named dimensions
matching the dimensions on parameter 'data'. By default it is NULL, to select a
period this parameter must be provided.
start An optional parameter to defined the initial date of the period to select from the
data by providing a list of two elements: the initial date of the period and the
initial month of the period. By default it is set to NULL and the indicator is
computed using all the data provided in data.
end An optional parameter to defined the final date of the period to select from the
data by providing a list of two elements: the final day of the period and the final
month of the period. By default it is set to NULL and the indicator is computed
using all the data provided in data.
time_dim A character string indicating the name of the dimension to compute the indicator.
By default, it is set to 'time'. More than one dimension name matching the
dimensions provided in the object data$data can be specified.
na.rm A logical value indicating whether to ignore NA values (TRUE) or not (FALSE).
ncores An integer indicating the number of cores to use in parallel computation.

Value

A multidimensional array with named dimensions containing the indicator in the element data.

Examples

exp <- array(abs(rnorm(5 * 3 * 214 * 2)*280),
             c(member = 5, sdate = 3, ftime = 214, lon = 2))
Wind capacity factor computes the wind power generated by a specific wind turbine model under specific wind speed conditions, and expresses it as a fraction of the rated capacity (i.e. maximum power) of the turbine.

It is computed by means of a tabular power curve that relates wind speed to power output. The tabular values are interpolated with a linear piecewise approximating function to obtain a smooth power curve. Five different power curves that span different IEC classes can be selected (see below).

**Usage**

```r
WindCapacityFactor(
  wind,
  IEC_class = c("I", "I/II", "II", "II/III", "III"),
  dates = NULL,
  start = NULL,
  end = NULL,
  time_dim = "time",
  ncores = NULL
)
```

**Arguments**

- **wind**
  A multidimensional array, vector or scalar with instantaneous wind speeds expressed in m/s.

- **IEC_class**
  A string indicating the IEC wind class (see IEC 61400-1) of the turbine to be selected. Classes 'I', 'II' and 'III' are suitable for sites with an annual mean wind speed of 10, 8.5 and 7.5 m/s respectively. Classes 'I/II' and 'II/III' indicate intermediate turbines that fit both classes. More details of the five turbines and a plot of its power curves can be found in Lledó et al. (2019).

- **dates**
  A vector of dates or a multidimensional array of dates with named dimensions matching the dimensions on parameter 'data'. By default it is NULL, to select a period this parameter must be provided.

- **start**
  An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.

- **end**
  An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
WindPowerDensity

Description

Wind Power Density computes the wind power that is available for extraction per square meter of swept area. It is computed as \(0.5*\rho*wind^3\). As this function is non-linear, it will give inaccurate results if used with period means.

Usage

```r
WindPowerDensity(
  wind,
  ro = 1.225,
  dates = NULL,
  start = NULL,
  end = NULL,
  time_dim = "time",
  ncores = NULL
)
```

Value

An array with the same dimensions as wind, containing the Wind Capacity Factor (unitless).

Author(s)

Llorenç Lledó. <llledo@bsc.es>

References


Examples

```r
wind <- rweibull(n = 100, shape = 2, scale = 6)
WCF <- WindCapacityFactor(wind, IEC_class = "III")
```
WindPowerDensity

Arguments

- **wind**: A multidimensional array, vector or scalar with instantaneous wind speeds expressed in m/s.
- **ro**: A scalar, or alternatively a multidimensional array with the same dimensions as wind, with the air density expressed in kg/m^3. By default it takes the value 1.225, the standard density of air at 15ºC and 1013.25 hPa.
- **dates**: A vector of dates or a multidimensional array of dates with named dimensions matching the dimensions on parameter 'data'. By default it is NULL, to select a period this parameter must be provided.
- **start**: An optional parameter to defined the initial date of the period to select from the data by providing a list of two elements: the initial date of the period and the initial month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
- **end**: An optional parameter to defined the final date of the period to select from the data by providing a list of two elements: the final day of the period and the final month of the period. By default it is set to NULL and the indicator is computed using all the data provided in data.
- **time_dim**: A character string indicating the name of the dimension to compute the indicator. By default, it is set to 'ftime'. More than one dimension name matching the dimensions provided in the object data$data can be specified.
- **ncores**: An integer indicating the number of cores to use in parallel computation for temporal subsetting.

Value

An array with the same dimensions as wind, containing Wind Power Density expressed in W/m^2.

Author(s)

Llorenç Lledó, <llledo@bsc.es>

Examples

```r
wind <- rweibull(n = 100, shape = 2, scale = 6)
WPD <- WindPowerDensity(wind)
```
Index

AbsToProbs, 2
AccumulationExceedingThreshold, 4

CST_AbsToProbs, 5
CST_AccumulationExceedingThreshold, 7
CST_MergeRefToExp, 8
CST_PeriodAccumulation, 10
CST_PeriodMean, 11
CST_QThreshold, 12
CST_SelectPeriodOnData, 14
CST_Threshold, 15
CST_TotalSpellTimeExceedingThreshold, 16
CST_TotalTimeExceedingThreshold, 18
CST_WindCapacityFactor, 20
CST_WindPowerDensity, 21

MergeRefToExp, 23
PeriodAccumulation, 24
PeriodMean, 26

QThreshold, 27
SelectPeriodOnData, 28
SelectPeriodOnDates, 29

Threshold, 30
TotalSpellTimeExceedingThreshold, 32
TotalTimeExceedingThreshold, 33

WindCapacityFactor, 35
WindPowerDensity, 36