

Package ‘CSIndicators’

May 18, 2023

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

Version 1.0.1

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](https://doi.org/10.1016/j.renene.2019.04.135)>.

Depends R (>= 3.6.0)

Imports multiApply (>= 2.1.1), stats

Suggests testthat, CSTools, knitr, markdown, rmarkdown

VignetteBuilder knitr

License GPL-3

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],
Llorenç Lledó [aut],
González-Reviriego Nube [ctb],
Marcos Raül [ctb],

Palma Lluís [ctb],
 An-Chi Ho [ctb],
 BSC-CNS [cph]

Maintainer Eva Rifà <eva.rifarovira@bsc.es>

Repository CRAN

Date/Publication 2023-05-18 18:10:02 UTC

R topics documented:

AbsToProbs	2
AccumulationExceedingThreshold	4
CST_AbsToProbs	6
CST_AccumulationExceedingThreshold	7
CST_MergeRefToExp	9
CST_PeriodAccumulation	10
CST_PeriodMean	12
CST_QThreshold	13
CST_SelectPeriodOnData	15
CST_Threshold	16
CST_TotalSpellTimeExceedingThreshold	17
CST_TotalTimeExceedingThreshold	19
CST_WindCapacityFactor	20
CST_WindPowerDensity	22
MergeRefToExp	23
PeriodAccumulation	25
PeriodMean	26
QThreshold	28
SelectPeriodOnData	29
SelectPeriodOnDates	30
Threshold	31
TotalSpellTimeExceedingThreshold	33
TotalTimeExceedingThreshold	34
WindCapacityFactor	36
WindPowerDensity	38
Index	40

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

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

Arguments

<code>data</code>	A multidimensional array with named dimensions.
<code>dates</code>	An optional parameter containing 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. All common dimensions with 'data' need to have the same length.
<code>start</code>	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.
<code>end</code>	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.
<code>time_dim</code>	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 <code>data\$data</code> can be specified. This dimension is required to subset the data in a requested period.
<code>memb_dim</code>	A character string indicating the name of the dimension in which the ensemble members are stored.
<code>sdate_dim</code>	A character string indicating the name of the dimension in which the initialization dates are stored.
<code>ncores</code>	An integer indicating the number of cores to use in parallel computation.

Value

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

Examples

```
exp <- array(rnorm(216), dim = c(dataset = 1, member = 2, sdate = 3,
                               ftime = 9, lat = 2, lon = 2))
exp_probs <- AbsToProbs(exp)
data <- array(rnorm(5 * 3 * 61 * 1),
```

```

      c(member = 5, sdate = 3, ftime = 61, lon = 1))
Dates <- c(seq(as.Date("01-05-2000", format = "%d-%m-%Y"),
              as.Date("30-06-2000", format = "%d-%m-%Y"), by = 'day'),
          seq(as.Date("01-05-2001", format = "%d-%m-%Y"),
              as.Date("30-06-2001", format = "%d-%m-%Y"), by = 'day'),
          seq(as.Date("01-05-2002", format = "%d-%m-%Y"),
              as.Date("30-06-2002", format = "%d-%m-%Y"), by = 'day'))
dim(Dates) <- c(ftime = 61, sdate = 3)
exp_probs <- AbsToProbs(data, dates = Dates, start = list(21, 4),
                       end = list(21, 6))

```

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 = "ftime",
  na.rm = FALSE,
  ncores = NULL
)

```

Arguments

`data` A multidimensional array with named dimensions.

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$attrs$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(exp$attrs$Dates) <- c(ftime = 214, sdate = 3)
exp_probs <- CST_AbsToProbs(data = exp, start = list(21, 4), end = list(21, 6))

```

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

<code>data</code>	An 's2dv_cube' object as provided by function <code>CST_Load</code> in package <code>CSTools</code> .
<code>threshold</code>	If only one threshold is used, it can be an 's2dv_cube' object or a multidimensional array with named dimensions. It must be in the same units and with the common dimensions of the same length as parameter 'data'. It can also be a vector with the same length of 'time_dim' from 'data' or a scalar. If we want to use two thresholds: it can be a vector of two scalars, a list of two vectors with the same length of 'time_dim' from 'data' or a list of two multidimensional arrays with the common dimensions of the same length as parameter 'data'. If two thresholds are used, parameter 'op' must be also a vector of two elements.
<code>op</code>	An operator '>' (by default), '<', '>=' or '<='. If two thresholds are used it has to be a vector of a pair of two logical operators: <code>c('<', '>')</code> , <code>c('<', '>=')</code> , <code>c('<=', '>')</code> , <code>c('<=', '>=')</code> , <code>c('>', '<')</code> , <code>c('>', '<=')</code> , <code>c('>=', '<')</code> , <code>c('>=', '<=')</code> .
<code>diff</code>	A logical value indicating whether to accumulate the difference between data and threshold (TRUE) or not (FALSE by default). It can only be TRUE if a unique threshold is used.
<code>start</code>	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.
<code>end</code>	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.
<code>time_dim</code>	A character string indicating the name of the dimension to compute the indicator. By default, it is set to 'ftime'. It can only indicate one time dimension.
<code>na.rm</code>	A logical value indicating whether to ignore NA values (TRUE) or not (FALSE).
<code>ncores</code>	An integer indicating the number of cores to use in parallel computation.

Value

An 's2dv_cube' object containing the aggregated values in the element data with dimensions of the input parameter 'data' except the dimension where the indicator has been computed.

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*

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 hind-cast) 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,
  start2,
  end2,
  time_dim = "ftime",
  sdate_dim = "sdate",
  ncores = NULL
)
```

Arguments

<code>data1</code>	An 's2dv_cube' object as provided function <code>CST_Load</code> in package <code>CSTools</code> .
<code>data2</code>	An 's2dv_cube' object as provided function <code>CST_Load</code> in package <code>CSTools</code> .
<code>start1</code>	A list to define the initial date of the period to select from <code>data1</code> by providing a list of two elements: the initial date of the period and the initial month of the period.
<code>end1</code>	A list to define the final date of the period to select from <code>data1</code> by providing a list of two elements: the final day of the period and the final month of the period.
<code>start2</code>	A list to define the initial date of the period to select from <code>data2</code> by providing a list of two elements: the initial date of the period and the initial month of the period.
<code>end2</code>	A list to define the final date of the period to select from <code>data2</code> by providing a list of two elements: the final day of the period and the final month of the period.
<code>time_dim</code>	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 <code>data\$data</code> 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$data <- array(1:(2*154*2), c(ftime = 154, sdate = 2, member= 2))
data$attrs$Dates<- data_dates
class(data) <- 's2dv_cube'
ref_dates <- seq(as.Date("01-01-1993", "%d-%m-%Y", tz = 'UTC'),
  as.Date("01-12-1994", "%d-%m-%Y", tz = 'UTC'), "day")
dim(ref_dates) <- c(ftime = 350, sdate = 2)
ref <- NULL
ref$data <- array(1001:1700, c(ftime = 350, sdate = 2))
ref$attrs$Dates <- 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:

- SprRSpring Total Precipitation: The total precipitation from April 21st to June 21st
- HarRHarvest Total Precipitation: The total precipitation from August 21st to October 21st

Usage

```
CST_PeriodAccumulation(
  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

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

Examples

```

exp <- NULL
exp$data <- array(rnorm(216)*200, dim = c(dataset = 1, member = 2, sdate = 3,
    ftime = 9, lat = 2, lon = 2))
class(exp) <- 's2dv_cube'
TP <- CST_PeriodAccumulation(exp)
exp$data <- array(rnorm(5 * 3 * 214 * 2),
    c(memb = 5, sdate = 3, ftime = 214, lon = 2))
exp$attrs$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 <- CST_PeriodAccumulation(exp, start = list(21, 4), end = list(21, 6))
dim(SprR$data)
head(SprR$attrs$Dates)
HarR <- CST_PeriodAccumulation(exp, start = list(21, 8), end = list(21, 10))
dim(HarR$data)
head(HarR$attrs$Dates)

```

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:

- GSTGrowing Season average Temperature: The average temperature from April 1st to October 31st
- SprTXSpring Average Maximum Temperature: The average daily maximum temperature from April 1st to May 31st

Usage

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

Arguments

<code>data</code>	An 's2dv_cube' object as provided function CST_Load in package CSTools.
<code>start</code>	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.
<code>end</code>	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.
<code>time_dim</code>	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 <code>data\$data</code> can be specified.
<code>na.rm</code>	A logical value indicating whether to ignore NA values (TRUE) or not (FALSE).
<code>ncores</code>	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$data <- array(rnorm(45), dim = c(member = 7, ftime = 8))
class(exp) <- 's2dv_cube'
exp$attrs$Dates <- 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)

```

CST_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:

Usage

```

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

```

Arguments

<code>data</code>	An 's2dv_cube' object as provided function <code>CST_Load</code> in package <code>CSTools</code> .
<code>threshold</code>	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.
<code>start</code>	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 <code>NULL</code> and the indicator is computed using all the data provided in <code>data</code> .

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 <code>data\$data</code> 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

```
threshold <- 26
exp <- 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$attrs$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'))
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

```
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

```
exp <- NULL
exp$data <- array(rnorm(5 * 3 * 214 * 2),
                 c(memb = 5, sdate = 3, ftime = 214, lon = 2))
exp$attrs$Dates <- c(seq(as.Date("01-05-2000", format = "%d-%m-%Y"),
                        as.Date("30-11-2000", format = "%d-%m-%Y"), by = 'day'),
                    seq(as.Date("01-05-2001", format = "%d-%m-%Y"),
                        as.Date("30-11-2001", format = "%d-%m-%Y"), by = 'day'),
                    seq(as.Date("01-05-2002", format = "%d-%m-%Y"),
                        as.Date("30-11-2002", format = "%d-%m-%Y"), by = 'day'))
dim(exp$attrs$Dates) <- c(ftime = 214, sdate = 3)
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

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

Arguments

<code>data</code>	An 's2dv_cube' object as provided function <code>CST_Load</code> in package <code>CSTools</code> .
<code>threshold</code>	A single scalar or vector indicating the relative threshold(s).
<code>start</code>	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 <code>NULL</code> and the indicator is computed using all the data provided in <code>data</code> .
<code>end</code>	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 <code>NULL</code> and the indicator is computed using all the data provided in <code>data</code> .
<code>time_dim</code>	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 <code>data\$data</code> can be specified. This dimension is required to subset the data in a requested period.
<code>memb_dim</code>	A character string indicating the name of the dimension in which the ensemble members are stored. When set it to <code>NULL</code> , threshold is computed for individual members.
<code>sdate_dim</code>	A character string indicating the name of the dimension in which the initialization dates are stored.
<code>na.rm</code>	A logical value indicating whether to ignore NA values (<code>TRUE</code>) or not (<code>FALSE</code>).
<code>ncores</code>	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

```
threshold <- 0.9
exp <- NULL
exp$data <- array(rnorm(5 * 3 * 214 * 2),
                  c(member = 5, sdate = 3, ftime = 214, lon = 2))
exp$attrs$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'))
class(exp) <- 's2dv_cube'
exp_probs <- CST_Threshold(exp, threshold, start = list(21, 4), end = list(21, 6))
```

CST_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

```
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	If only one threshold is used, it can be an 's2dv_cube' object or a multidimensional array with named dimensions. It must be in the same units and with the common dimensions of the same length as parameter 'data'. It can also be a vector with the same length of 'time_dim' from 'data' or a scalar. If we want to use two thresholds: it can be a vector of two scalars, a list of two vectors with the same length of 'time_dim' from 'data' or a list of two multidimensional arrays with the common dimensions of the same length as parameter 'data'. If two thresholds are used, parameter 'op' must be also a vector of two elements.
spell	A scalar indicating the minimum length of the spell.
op	An operator '>' (by default), '<', '>=' or '<='. If two thresholds are used it has to be a vector of a pair of two logical operators: c('<', '>'), c('<', '>='), c('<=', '>'), c('<=', '>='), c('>', '<'), c('>', '<='), c('>=', '<'), c('>=', '<=')).
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 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'. It can only indicate one time dimension.
ncores	An integer indicating the number of cores to use in parallel computation.

Value

An 's2dv_cube' object containing the number of days that are part of a spell within a threshold in element data.

See Also

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

Examples

```
exp <- NULL
exp$data <- array(rnorm(5 * 3 * 214 * 2)*23,
                 c(member = 5, sdate = 3, ftime = 214, lon = 2))
exp$attrs$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'))
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. It returns the total number of days (if the data provided is daily, or the corresponding units of the data frequency) that a variable is exceeding a threshold during a period. The threshold provided must be in the same units as 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

Usage

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

Arguments

<code>data</code>	An 's2dv_cube' object as provided by function <code>CST_Load</code> in package <code>CSTools</code> .
<code>threshold</code>	If only one threshold is used, it can be an 's2dv_cube' object or a multidimensional array with named dimensions. It must be in the same units and with the common dimensions of the same length as parameter 'data'. It can also be a vector with the same length of 'time_dim' from 'data' or a scalar. If we want to use two thresholds: it can be a vector of two scalars, a list of two vectors

with the same length of 'time_dim' from 'data' or a list of two multidimensional arrays with the common dimensions of the same length as parameter 'data'. If two thresholds are used, parameter 'op' must be also a vector of two elements.

op	An operator '>' (by default), '<', '>=' or '<='. If two thresholds are used it has to be a vector of a pair of two logical operators: c('<', '>'), c('<', '>='), c('<=', '>'), c('<=', '>='), c('>', '<'), c('>', '<='), c('>=', '<'), c('>=', '<=')).
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 dimension to compute the indicator. By default, it is set to 'ftime'. It can only indicate one time dimension.
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 in element data the total number of the corresponding units of the data frequency that a variable is exceeding a threshold during a period.

Examples

```
exp <- NULL
exp$data <- array(abs(rnorm(5 * 3 * 214 * 2)*280),
                  c(member = 5, sdate = 3, ftime = 214, lon = 2))
exp$attrs$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'))
class(exp) <- 's2dv_cube'
DOT <- CST_TotalTimeExceedingThreshold(exp, threshold = 280)
```

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 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

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

Arguments

<code>wind</code>	An <code>s2dv_cube</code> object with instantaneous wind speeds expressed in m/s.
<code>IEC_class</code>	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).
<code>start</code>	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.
<code>end</code>	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.
<code>time_dim</code>	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 <code>data\$data</code> can be specified.
<code>ncores</code>	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, <l1lledo@bsc.es>

References

Lledó, Ll., Torralba, V., Soret, A., Ramon, J., & Doblas-Reyes, F. J. (2019). Seasonal forecasts of wind power generation. *Renewable Energy*, 143, 91–100. <https://doi.org/10.1016/j.renene.2019.04.135>

International Standard IEC 61400-1 (third ed.) (2005)

Examples

```
wind <- NULL
wind$data <- array(rweibull(n = 100, shape = 2, scale = 6),
                  c(member = 10, lat = 2, lon = 5))
wind$coords <- list(lat = c(40, 41), lon = 1:5)
variable <- list(varName = 'sfcWind',
                 metadata = list(sfcWind = list(level = 'Surface')))
wind$attrs <- list(Variable = variable, Datasets = 'synthetic',
                  when = Sys.time(), Dates = '1990-01-01 00:00:00')
class(wind) <- 's2dv_cube'
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 $0.5 * \rho * \text{wspd}^3$. As this function is non-linear, it will give inaccurate results if used with period means.

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

Author(s)

Llorenç Lledó, <lledo@bsc.es>

Examples

```

wind <- NULL
wind$data <- array(rweibull(n = 100, shape = 2, scale = 6),
                  c(member = 10, lat = 2, lon = 5))
wind$coords <- list(lat = c(40, 41), lon = 1:5)
variable <- list(varName = 'sfcWind',
                 metadata = list(sfcWind = list(level = 'Surface')))
wind$attrs <- list(Variable = variable, Datasets = 'synthetic',
                  when = Sys.time(), Dates = '1990-01-01 00:00:00')
class(wind) <- 's2dv_cube'
WCF <- CST_WindPowerDensity(wind)

```

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

```

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

```

Arguments

<code>data1</code>	A multidimensional array with named dimensions.
<code>dates1</code>	A vector of dates or a multidimensional array of dates with named dimensions matching the dimensions on parameter 'data1'.
<code>start1</code>	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.
<code>end1</code>	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.
<code>data2</code>	A multidimensional array with named dimensions.
<code>dates2</code>	A vector of dates or a multidimensional array of dates with named dimensions matching the dimensions on parameter 'data2'.
<code>start2</code>	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.
<code>end2</code>	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.
<code>time_dim</code>	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 <code>data\$data</code> can be specified. This dimension is required to subset the data in a requested period.
<code>sdate_dim</code>	A character string indicating the name of the dimension in which the initialization dates are stored.
<code>ncores</code>	An integer indicating the number of cores to use in parallel computation.

Value

A multidimensional array with named dimensions.

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(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),
  time_dim = 'time')

```

PeriodAccumulation *Period Accumulation on multidimensional array 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:

- SprRSpring Total Precipitation: The total precipitation from April 21th to June 21st
- HarRHarvest Total Precipitation: The total precipitation from August 21st to October 21st

Usage

```

PeriodAccumulation(
  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 '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(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))
```

PeriodMean

Period Mean on multidimensional array 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:

- GSTGrowing Season average Temperature: The average temperature from April 1st to Octobe 31st
- SprTXSpring Average Maximum Temperature: The average daily maximum temperature from April 1st to May 31st

Usage

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

Arguments

<code>data</code>	A multidimensional array with named dimensions.
<code>dates</code>	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.
<code>start</code>	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.
<code>end</code>	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.
<code>time_dim</code>	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 <code>data\$data</code> can be specified.
<code>na.rm</code>	A logical value indicating whether to ignore NA values (TRUE) or not (FALSE).
<code>ncores</code>	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(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

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

Arguments

<code>data</code>	A multidimensional array with named dimensions.
<code>threshold</code>	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.
<code>dates</code>	A vector of dates or a multidimensional array of dates with named dimensions matching the dimensions on parameter 'data'. By default it is <code>NULL</code> , 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

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

SelectPeriodOnData *Select a period on Data on multidimensional array objects*

Description

Auxiliary function to subset data for a specific period.

Usage

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

Arguments

data	A multidimensional array with named dimensions with at least the time dimension specified in parameter 'time_dim'. All common dimensions with 'dates' parameter need to have the same length.
dates	An array of dates with named dimensions with at least the time dimension specified in parameter 'time_dim'. All common dimensions with 'data' parameter need to have the same length.
start	A list with two elements to define the initial date of the period to select from the data. The first element is the initial day of the period and the second element is the initial month of the period.
end	A list with two elements to define the final date of the period to select from the data. The first element is the final day of the period and the second element is 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'. Parameters 'data' and 'dates'
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)
```

Arguments

dates	An 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 <code>data\$data</code> 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'))
dim(Dates) <- c(ftime = 214, sdate = 3)
Period <- SelectPeriodOnDates(Dates, start = list(21, 6), end = list(21, 9))
```

Threshold	<i>Absolute value of a relative threshold (percentile)</i>
-----------	--

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

```

Arguments

<code>data</code>	A multidimensional array with named dimensions.
<code>threshold</code>	A single scalar or vector indicating the relative threshold(s).
<code>dates</code>	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.
<code>start</code>	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.
<code>end</code>	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.
<code>time_dim</code>	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 <code>data\$data</code> can be specified. This dimension is required to subset the data in a requested period.
<code>memb_dim</code>	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.
<code>sdate_dim</code>	A character string indicating the name of the dimension in which the initialization dates are stored.
<code>na.rm</code>	A logical value indicating whether to ignore NA values (TRUE) or not (FALSE).
<code>ncores</code>	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

```

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

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

Arguments

data	A multidimensional array with named dimensions.
threshold	If only one threshold is used: it can be a multidimensional array with named dimensions. It must be in the same units and with the common dimensions of the same length as parameter 'data'. It can also be a vector with the same length of 'time_dim' from 'data' or a scalar. If we want to use two thresholds: it can be a vector of two scalars, a list of two vectors with the same length of 'time_dim' from 'data' or a list of two multidimensional arrays with the common dimensions of the same length as parameter 'data'. If two thresholds are used, parameter 'op' must be also a vector of two elements.
spell	A scalar indicating the minimum length of the spell.
op	An operator '>' (by default), '<', '>=' or '<='. If two thresholds are used it has to be a vector of a pair of two logical operators: c('<', '>'), c('<', '>='), c('<=', '>'), c('<=', '>='), c('>', '<'), c('>', '<='), c('>=', '<'), c('>=', '<=')).

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 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 dimension to compute the indicator. By default, it is set to 'ftime'. It can only indicate one time dimension.
ncores	An integer indicating the number of cores to use in parallel computation.

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 number of days that are part of a spell within a threshold with dimensions of the input parameter 'data' except the dimension where the indicator has been computed.

See Also

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

Examples

```
data <- array(rnorm(120), c(member = 1, sdate = 2, ftime = 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

Description

The Total Time of a variable exceeding (or not) a Threshold. It returns the total number of days (if the data provided is daily, or the corresponding units of the data frequency) that a variable is exceeding a threshold during a period. The threshold provided must be in the same units as 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

Usage

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

Arguments

<code>data</code>	A multidimensional array with named dimensions.
<code>threshold</code>	If only one threshold is used: it can be a multidimensional array with named dimensions. It must be in the same units and with the common dimensions of the same length as parameter <code>'data'</code> . It can also be a vector with the same length of <code>'time_dim'</code> from <code>'data'</code> or a scalar. If we want to use two thresholds: it can be a vector of two scalars, a list of two vectors with the same length of <code>'time_dim'</code> from <code>'data'</code> or a list of two multidimensional arrays with the common dimensions of the same length as parameter <code>'data'</code> . If two thresholds are used, parameter <code>'op'</code> must be also a vector of two elements.
<code>op</code>	An operator <code>'>'</code> (by default), <code>'<'</code> , <code>'>='</code> or <code>'<='</code> . If two thresholds are used it has to be a vector of a pair of two logical operators: <code>c('<', '>')</code> , <code>c('<', '>=')</code> , <code>c('<=', '>')</code> , <code>c('<=', '>=')</code> , <code>c('>', '<')</code> , <code>c('>', '<=')</code> , <code>c('>=', '<')</code> , <code>c('>=', '<=')</code> .
<code>dates</code>	A vector of dates or a multidimensional array of dates with named dimensions matching the dimensions on parameter <code>'data'</code> . By default it is <code>NULL</code> , to select a period this parameter must be provided.

<code>start</code>	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.
<code>end</code>	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.
<code>time_dim</code>	A character string indicating the name of the dimension to compute the indicator. By default, it is set to 'ftime'. It can only indicate one time dimension.
<code>na.rm</code>	A logical value indicating whether to ignore NA values (TRUE) or not (FALSE).
<code>ncores</code>	An integer indicating the number of cores to use in parallel computation.

Value

A multidimensional array with named dimensions containing the total number of the corresponding units of the data frequency that a variable is exceeding a threshold during a period.

Examples

```
exp <- array(abs(rnorm(5 * 3 * 214 * 2)*280),
             c(member = 5, sdate = 3, ftime = 214, lon = 2))
DOT <- TotalTimeExceedingThreshold(exp, threshold = 300, time_dim = 'ftime')
```

WindCapacityFactor	<i>Wind capacity factor</i>
--------------------	-----------------------------

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 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

```
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.
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 the Wind Capacity Factor (unitless).

Author(s)

Llorenç Lledó, <l1lledo@bsc.es>

References

Lledó, Ll., Torralba, V., Soret, A., Ramon, J., & Doblas-Reyes, F. J. (2019). Seasonal forecasts of wind power generation. *Renewable Energy*, 143, 91–100. <https://doi.org/10.1016/j.renene.2019.04.135>

International Standard IEC 61400-1 (third ed.) (2005)

Examples

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

WindPowerDensity *Wind power density on multidimensional array objects*

Description

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

It is computed as $0.5 * \rho * \text{wspd}^3$. As this function is non-linear, it will give inaccurate results if used with period means.

Usage

```
WindPowerDensity(
  wind,
  ro = 1.225,
  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.
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.
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, <l11ledo@bsc.es>

Examples

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

Index

AbsToProbs, [2](#)
AccumulationExceedingThreshold, [4](#)

CST_AbsToProbs, [6](#)
CST_AccumulationExceedingThreshold, [7](#)
CST_MergeRefToExp, [9](#)
CST_PeriodAccumulation, [10](#)
CST_PeriodMean, [12](#)
CST_QThreshold, [13](#)
CST_SelectPeriodOnData, [15](#)
CST_Threshold, [16](#)
CST_TotalSpellTimeExceedingThreshold,
[17](#)
CST_TotalTimeExceedingThreshold, [19](#)
CST_WindCapacityFactor, [20](#)
CST_WindPowerDensity, [22](#)

MergeRefToExp, [23](#)

PeriodAccumulation, [25](#)
PeriodMean, [26](#)

QThreshold, [28](#)

SelectPeriodOnData, [29](#)
SelectPeriodOnDates, [30](#)

Threshold, [31](#)
TotalSpellTimeExceedingThreshold, [33](#)
TotalTimeExceedingThreshold, [34](#)

WindCapacityFactor, [36](#)
WindPowerDensity, [38](#)