Package ‘cmsafops’

September 17, 2020

Title Tools for CM SAF NetCDF Data

Version 1.0.0

Description The Satellite Application Facility on Climate Monitoring (CM SAF) is a ground segment of the European Organization for the Exploitation of Meteorological Satellites (EUMETSAT) and one of EUMETSATs Satellite Application Facilities. The CM SAF contributes to the sustainable monitoring of the climate system by providing essential climate variables related to the energy and water cycle of the atmosphere (<http://www.cmsaf.eu>). It is a joint cooperation of eight National Meteorological and Hydrological Services.

The 'cmsafops' R-package provides a collection of R-operators for the analysis and manipulation of CM SAF NetCDF formatted data. Other CF conform NetCDF data with time, longitude and latitude dimension should be applicable, but there is no guarantee for an error-free application.

CM SAF climate data records are provided for free via (<https://wui.cmsaf.eu>). Detailed information and test data are provided on the CM SAF webpage (<http://www.cmsaf.eu/R_toolbox>).

URL https://www.cmsaf.eu

License GPL (>= 3)

Depends R (>= 3.6)

Imports assertthat (>= 0.2.1), fields (>= 10.3), FNN (>= 1.1), ncdf4 (>= 1.17), rainfarmr (>= 0.1), raster (>= 3.0), sp (>= 1.4)

NeedsCompilation no

Repository CRAN

LazyData true

Suggests cmsaf, cmsafvis, spelling (>= 2.1), testthat (>= 2.3)

RoxygenNote 7.1.1

Encoding UTF-8

Language en-US

Author Steffen Kothe [aut, cre]

Maintainer Steffen Kothe <Steffen.Kothe@dwd.de>

Date/Publication 2020-09-17 11:50:07 UTC
topics documented:

- add_grid_info ......................................................... 3
- box_mergetime ....................................................... 4
- change_att ........................................................... 6
- cmsaf.add ............................................................ 7
- cmsaf.addc ........................................................... 9
- cmsaf.cat ............................................................. 11
- cmsaf.div ............................................................ 13
- cmsaf.divc ........................................................... 14
- cmsaf.mul ........................................................... 16
- cmsaf.mulc ........................................................... 18
- cmsaf.sub ........................................................... 20
- cmsaf.subc ........................................................... 21
- cmsafops ............................................................. 23
- dayrange ............................................................. 25
- divdpm ............................................................... 26
- extract.level ......................................................... 27
- extract.period ....................................................... 29
- fldmax ............................................................... 31
- fldmean .............................................................. 32
- fldmin ............................................................... 34
- get_time ............................................................. 35
- levbox_mergetime .................................................... 36
- mon.anomaly ......................................................... 38
- monmax ............................................................... 39
- monmean ............................................................. 41
- monmin ............................................................... 42
- monsd ............................................................... 44
- monsum .............................................................. 45
- muldpm ............................................................... 46
- multimonmean ....................................................... 48
- multimonsum ....................................................... 50
- ncinfo ............................................................... 51
- read_ncvar ......................................................... 53
- remap ............................................................... 54
- seas.anomaly ......................................................... 56
- seasmean ............................................................ 57
- seassum ............................................................. 59
- sellonlatbox ....................................................... 60
- selmon ............................................................. 62
- selperiod ........................................................... 64
- selpoint ............................................................ 65
- selpoint.multi ...................................................... 67
- seltime ............................................................. 69
- selyear ............................................................ 71
- timcumsum ......................................................... 72
- timmax ............................................................. 73
add_grid_info

Description

Adds a standard longitude/latitude grid to a file which is based on a different grid.

Usage

\texttt{add\_grid\_info(infile, auxfile, outfile, overwrite = FALSE, verbose = FALSE)}

Arguments

- \texttt{infile} Character containing file name or path of input file.
- \texttt{auxfile} Character containing file name or path of auxiliary file.
- \texttt{outfile} Character containing file name or path of output file. If NULL, the input file is directly edited instead.
- \texttt{overwrite} Logical; should existing output file be overwritten? If \texttt{outfile} is NULL, this parameter is ignored.
- \texttt{verbose} logical; if TRUE, progress messages are shown

Details

No existing data is changed. The additional grid info is added as two additional variables (lon and lat).
box_mergetime

Function to combine NetCDF files and simultaneously cut a region (and level).

Description

This function selects a region (and optionally a level) from a bunch of CM SAF NetCDF files that match the same pattern of the filename, and writes the output to a new file. If no longitude and latitude values are given, files are only merged. All input files have to have the same grid and the same variable. The reference time of the output file is determined by the first input file.

Usage

box_mergetime(
  var,
  path,
  pattern,
  outfile,
  lon1 = -180,
  lon2 = 180,
  lat1 = -90,
  lat2 = 90,
  level = NULL,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE
)

Arguments

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>var</td>
<td>Name of NetCDF variable (character).</td>
</tr>
<tr>
<td>path</td>
<td>The directory of input NetCDF files without / at the end (character).</td>
</tr>
<tr>
<td>pattern</td>
<td>A part of the filename, which is the same for all desired input files (character). The pattern has to be a character string containing a regular expression.</td>
</tr>
<tr>
<td>outfile</td>
<td>Filename of output NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>lon1</td>
<td>Longitude of lower left corner (numeric).</td>
</tr>
<tr>
<td>lon2</td>
<td>Longitude of upper right left corner (numeric).</td>
</tr>
<tr>
<td>lat1</td>
<td>Latitude of lower left corner (numeric).</td>
</tr>
<tr>
<td>lat2</td>
<td>Latitude of upper right corner (numeric). Longitude of upper right corner (numeric).</td>
</tr>
<tr>
<td>level</td>
<td>Number of level that should be extracted (integer) or NULL.</td>
</tr>
</tbody>
</table>

See Also

Other data manipulation functions: box_mergetime(), levbox_mergetime(), remap()
**box_mergetime**

- **nc34**  
  NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.

- **overwrite**  
  logical; should existing output file be overwritten?

- **verbose**  
  logical; if TRUE, progress messages are shown

**Value**

A NetCDF file including the merged time series of the selected region is written. The resulting file uses the meta data of the first input file.

**See Also**

Other data manipulation functions: `add_grid_info()`, `levbox_mergetime()`, `remap()`

**Examples**

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- c(as.Date("2000-01-01"), as.Date("2001-02-01"))
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data1 <- array(250:350, dim = c(21, 21, 1))
data2 <- array(230:320, dim = c(21, 21, 1))

## create two simple example NetCDF files
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
               vals = time[1], unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file_n1.nc"), vars)
ncvar_put(ncnew, var1, data1)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncclose(ncnew)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
               vals = time[2], unlim = TRUE)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file_n2.nc"), vars)
ncvar_put(ncnew, var1, data2)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncclose(ncnew)
```

## Cut a region and merge both example CM SAF NetCDF files into one output file. Get path information of working directory with getwd() command.

```r
box_mergetime("SIS", tempdir(), "CMSAF_example_file_n",
  file.path(tempdir(),"CMSAF_example_file_box_mergetime.nc"), 8, 12, 48, 52)
```

```r
unlink(c(file.path(tempdir(),"CMSAF_example_file_n1.nc"),
  file.path(tempdir(),"CMSAF_example_file_n2.nc"),
  file.path(tempdir(),"CMSAF_example_file_box_mergetime.nc")))
```

---

### change_att

**Change attributes of a NetCDF variable.**

#### Description

This function can change the name, standard_name, long_name, units, _FillValue and missing_value of a variable. There is no separate outfile, thus use this function with care. The values for v_name, s_name, l_name, u_name, F_val and m_val are optional and will only be changed if they are given. If an attribute is not defined yet, it is added by the function.

#### Usage

```r
change_att(
  var, 
  infile, 
  v_name = NULL, 
  s_name = NULL, 
  l_name = NULL, 
  u_name = NULL, 
  F_val = NULL, 
  m_val = NULL, 
  val_prec = "double", 
  verbose = FALSE 
)
```

#### Arguments

- `var`: Name of NetCDF variable (character).
- `infile`: Filename of input NetCDF file. This may include the directory (character).
- `v_name`: New variable name (character).
- `s_name`: New standard name (character).
- `l_name`: New long name (character).
- `u_name`: New units name (character).
- `F_val`: New fill value (numeric).
- `m_val`: New missing value (numeric).
- `val_prec`: Precision of the FillValue and missing value (character). Default is double.
- `verbose`: logical; if TRUE, progress messages are shown.
Value

The variable information within the infile NetCDF is changed.

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time, unlim = TRUE)
var1 <- ncvar_def("Data1", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Change the variable and standard name of the example CM SAF NetCDF file:
change_att("Data1", file.path(tempdir(),"CMSAF_example_file.nc"), v_name = "SIS", s_name = "surface_downwelling_shortwave_flux_in_air")
unlink(file.path(tempdir(),"CMSAF_example_file.nc"))
```

---

**Description**

The function adds the fields of infile1 to the fields of infile2. Infiles have to have the same spatial and temporal dimension or one infile can contain only one timestep. The outfile uses the meta data of infile1.
Usage

cmsaf.add(
  vari1,
  vari2,
  infile1,
  infile2,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE
)

Arguments

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vari1</td>
<td>Name of variable in infile1 (character).</td>
</tr>
<tr>
<td>vari2</td>
<td>Name of variable in infile2 (character).</td>
</tr>
<tr>
<td>infile1</td>
<td>Filename of first input NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>infile2</td>
<td>Filename of second input NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>outfile</td>
<td>Filename of output NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>nc34</td>
<td>NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.</td>
</tr>
<tr>
<td>overwrite</td>
<td>logical; should existing output file be overwritten?</td>
</tr>
<tr>
<td>verbose</td>
<td>logical; if TRUE, progress messages are shown</td>
</tr>
</tbody>
</table>

Value

A NetCDF file including the added fields of infile1 and infile2 is written.

See Also

Other mathematical operators: cmsaf.addc(), cmsaf.divc(), cmsaf.div(), cmsaf.mulc(), cmsaf.mul(), cmsaf.subc(), cmsaf.sub(), divdpm(), muldpm()

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
latt <- seq(45, 55, 0.5)
time <- c(as.Date("2000-01-01"), as.Date("2001-02-01"))
origin <- as.Date("1983-01-01 00:00:00")
```
time <- as.numeric(difftime(time, origin, units = "hour"))
data1 <- array(250:350, dim = c(21, 21, 1))
data2 <- array(230:320, dim = c(21, 21, 1))

## create two example NetCDF files
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time[1], unlim = TRUE)
var1 <- ncvvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file_1.nc"), vars)
ncvar_put(ncnew, var1, data1)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time[2], unlim = TRUE)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file_2.nc"), vars)
ncvar_put(ncnew, var1, data2)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Add the fields of both example CM SAF NetCDF files and write the
## result into one output file.
cmsaf.add("SIS", "SIS", file.path(tempdir(),"CMSAF_example_file_1.nc"),
file.path(tempdir(),"CMSAF_example_file_2.nc"),
file.path(tempdir(),"CMSAF_example_file_add.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file_1.nc"),
file.path(tempdir(),"CMSAF_example_file_2.nc"),
file.path(tempdir(),"CMSAF_example_file_add.nc")))

cmsaf.addc

Add a constant to a dataset.

Description

This function adds a given constant number to each element of a dataset.

Usage

cmsaf.addc(
  var,
  const = 0,
  infile,
  outfile,
nc34 = 4,
overwrite = FALSE,
verbose = FALSE
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>var</td>
<td>Name of NetCDF variable (character).</td>
</tr>
<tr>
<td>const</td>
<td>Constant number (numeric).</td>
</tr>
<tr>
<td>infile</td>
<td>Filename of input NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>outfile</td>
<td>Filename of output NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>nc34</td>
<td>NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.</td>
</tr>
<tr>
<td>overwrite</td>
<td>logical; should existing output file be overwritten?</td>
</tr>
<tr>
<td>verbose</td>
<td>logical; if TRUE, progress messages are shown</td>
</tr>
</tbody>
</table>

Value

A NetCDF file including the manipulated data fields of infile is written. Standard output precision is 'double'.

See Also

Other mathematical operators: cmsaf.add(), cmsaf.divc(), cmsaf.div(), cmsaf.mulc(), cmsaf.mul(), cmsaf.subc(), cmsaf.sub(), divdpm(), muldpm()

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(diff(time, time, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time, unlim = TRUE)
```
var1 <- ncvar_def("SIS", "W m^-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Add a given number each dataset element of the example CM SAF NetCDF
## file and write the output to a new file.
cmsaf.addc("SIS", 10, file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_addc.nc"))
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_addc.nc")))

---

**Description**

This function concatenates datasets of an arbitrary number of input files. All input files have to have the same structure with the same variable and different timesteps.

**Usage**

```r
cmsaf.cat(var, infiles, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

**Arguments**

- `var` Name of NetCDF variable (character).
- `infiles` Vector with filenames of input NetCDF files. The file names may include the directory (character).
- `outfile` Filename of output NetCDF file. This may include the directory (character).
- `nc34` NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- `overwrite` logical; should existing output file be overwritten?
- `verbose` logical; if TRUE, progress messages are shown

**Value**

A NetCDF file including the merged time series is written. The resulting file uses the meta data of the first input file.
Examples

## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

```r
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- c(as.Date("2000-01-01"), as.Date("2001-02-01"))
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data1 <- array(250:350, dim = c(21, 21, 1))
data2 <- array(230:320, dim = c(21, 21, 1))

## create two simple example NetCDF files
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time[1], unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file_1.nc"), vars)
ncvar_put(ncnew, var1, data1)
cnclose(ncnew)

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time[2], unlim = TRUE)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file_2.nc"), vars)
ncvar_put(ncnew, var1, data2)
cnclose(ncnew)

## Cut a region and merge both example CM SAF NetCDF files into one output file. Get path information of working directory with getwd() command.
wd <- getwd()
cmsaf.cat("SIS", c(file.path(tempdir(),"CMSAF_example_file_1.nc"),
file.path(tempdir(),"CMSAF_example_file_2.nc")),
file.path(tempdir(),"CMSAF_example_file_cat.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file_1.nc"),
file.path(tempdir(),"CMSAF_example_file_2.nc"),
file.path(tempdir(),"CMSAF_example_file_cat.nc")))
```
### Description

The function divides the fields of infile1 by the fields of infile2. Infiles have to have the same spatial and temporal dimension or one infile can contain only one timestep. The outfile uses the meta data of infile1.

### Usage

```r
.cmsaf.div(
  vari1, 
  vari2, 
  infile1, 
  infile2, 
  outfile, 
  nc34 = 4, 
  overwrite = FALSE, 
  verbose = FALSE
)
```

### Arguments

- `vari1`: Name of variable in infile1 (character).
- `vari2`: Name of variable in infile2 (character).
- `infile1`: Filename of first input NetCDF file. This may include the directory (character).
- `infile2`: Filename of second input NetCDF file. This may include the directory (character).
- `outfile`: Filename of output NetCDF file. This may include the directory (character).
- `nc34`: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- `overwrite`: logical; should existing output file be overwritten?
- `verbose`: logical; if TRUE, progress messages are shown

### Value

A NetCDF file including the divided fields of infile1 and infile2 is written.

### See Also

Other mathematical operators: `cmsaf.addc()`, `cmsaf.add()`, `cmsaf.divc()`, `cmsaf.mulc()`, `cmsaf.mul()`, `cmsaf.subc()`, `cmsaf.sub()`,
Examples

## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

```r
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- c(as.Date("2000-01-01"), as.Date("2001-02-01"))
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data1 <- array(250:350, dim = c(21, 21, 1))
data2 <- array(230:320, dim = c(21, 21, 1))

## create two example NetCDF files
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
               vals = time[1], unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file_1.nc"), vars)
ncvar_put(ncnew, var1, data1)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
               vals = time[2], unlim = TRUE)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file_2.nc"), vars)
ncvar_put(ncnew, var1, data2)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Divide the fields of both example CM SAF NetCDF files and write the result into one output file.
cmsaf.div("SIS", "SIS", file.path(tempdir(), "CMSAF_example_file_1.nc"),
          file.path(tempdir(), "CMSAF_example_file_2.nc"))

unlink(c(file.path(tempdir(), "CMSAF_example_file_1.nc"),
         file.path(tempdir(), "CMSAF_example_file_2.nc"),
         file.path(tempdir(), "CMSAF_example_file_div.nc")))
```
**Description**
This function divides each element of a dataset by a given constant number.

**Usage**

cmsaf.divc(
  var,
  const = 1,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE
)

**Arguments**
- **var**: Name of NetCDF variable (character).
- **const**: Constant number (numeric).
- **infile**: Filename of input NetCDF file. This may include the directory (character).
- **outfile**: Filename of output NetCDF file. This may include the directory (character).
- **nc34**: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- **overwrite**: logical; should existing output file be overwritten?
- **verbose**: logical; if TRUE, progress messages are shown

**Value**
A NetCDF file including the manipulated data fields of `infile` is written. Standard output precision is 'double'.

**See Also**
Other mathematical operators: `cmsaf.addc()`, `cmsaf.add()`, `cmsaf.div()`, `cmsaf.mulc()`, `cmsaf.mul()`, `cmsaf.subc()`, `cmsaf.sub()`, `divdpm()`, `muldpm()`

**Examples**

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
```
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time, unlim = TRUE)
var1 <- ncvvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Divide each dataset element of the example CM SAF NetCDF file by a
## given number and write the output to a new file.
cmsaf.divc("SIS", 100, file.path(tempdir(),"CMSAF_example_file.nc"),
    file.path(tempdir(),"CMSAF_example_file_divc.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
    file.path(tempdir(),"CMSAF_example_file_divc.nc")))

---

cmsaf.mul

Multiply the fields of two input NetCDF files.

Description

The function multiplies the fields of infile1 and infile2. Infiles have to have the same spatial and
temporal dimension or one infile can contain only one timestep. The outfile uses the meta data of
infile1.

Usage

cmsaf.mul(
    vari1,
    vari2,
    infile1,
    infile2,
    outfile,
    nc34 = 4,
    overwrite = FALSE,
    verbose = FALSE
)
Arguments

vari1   Name of variable in infile1 (character).
vari2   Name of variable in infile2 (character).
infile1 Filename of first input NetCDF file. This may include the directory (character).
infile2 Filename of second input NetCDF file. This may include the directory (character).
outfile Filename of output NetCDF file. This may include the directory (character).
nc34    NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
overwrite logical; should existing output file be overwritten?
verbose  logical; if TRUE, progress messages are shown

Value

A NetCDF file including the multiplied fields of infile1 and infile2 is written.

See Also

Other mathematical operators: cmsaf.addc(), cmsaf.add(), cmsaf.divc(), cmsaf.div(), cmsaf.mulc(), cmsaf.subc(), cmsaf.sub(), divdpm(), muldpm()

Examples

## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- c(as.Date("2000-01-01"), as.Date("2001-02-01"))
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data1 <- array(250:350, dim = c(21, 21, 1))
data2 <- array(230:320, dim = c(21, 21, 1))

## create two example NetCDF files

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time[1], unlim = TRUE)
var1 <- ncvart_def("SIS", "W m\(^{-2}\)", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file_1.nc"), vars)
ncvar_put(ncnew, var1, data1)
## Multiply the fields of both example CM SAF NetCDF files and write the result into one output file.

cmsaf.mul("SIS", "SIS", file.path(tempdir(),"CMSAF_example_file_1.nc"),
file.path(tempdir(),"CMSAF_example_file_2.nc"),
file.path(tempdir(),"CMSAF_example_file_mul.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file_1.nc"),
file.path(tempdir(),"CMSAF_example_file_2.nc"),
file.path(tempdir(),"CMSAF_example_file_mul.nc")))

---

cmsaf.mulp

### Multiply data with a constant.

**Description**

This function multiplies each element of a dataset with a given constant number.

**Usage**

```r
cmsaf.mulp(
  var,
  const = 1,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE
)
```

**Arguments**

- `var` Name of NetCDF variable (character).
- `const` Constant number (numeric).
- `infile` Filename of input NetCDF file. This may include the directory (character).
- `outfile` Filename of output NetCDF file. This may include the directory (character).
nc34  NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.

overwrite  logical; should existing output file be overwritten?

verbose  logical; if TRUE, progress messages are shown

Value

A NetCDF file including the manipulated data fields of infile is written. Standard output precision is 'double'.

See Also

Other mathematical operators: cmsaf.addc(), cmsaf.add(), cmsaf.divc(), cmsaf.div(), cmsaf.mul(), cmsaf.subc(), cmsaf.sub(), divdpm(), muldpm()

Examples

## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncclose(ncnew)

## Multiply each dataset element of the example CM SAF NetCDF file by a given number and write the output to a new file.

cmsaf.mulc("SIS", 10, file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_mulc.nc"))
Subtract the fields of two input NetCDF files.

Description

The function subtracts the fields of infile2 from the fields of infile1. Infiles have to have the same spatial and temporal dimension or one infile can contain only one timestep. The outfile uses the meta data of infile1.

Usage

```r
cmsaf.sub(
  vari1, vari2, infile1, infile2, outfile, nc34 = 4,
  overwrite = FALSE, verbose = FALSE
)
```

Arguments

- `vari1`: Name of variable in infile1 (character).
- `vari2`: Name of variable in infile2 (character).
- `infile1`: Filename of first input NetCDF file. This may include the directory (character).
- `infile2`: Filename of second input NetCDF file. This may include the directory (character).
- `outfile`: Filename of output NetCDF file. This may include the directory (character).
- `nc34`: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- `overwrite`: logical; should existing output file be overwritten?
- `verbose`: logical; if TRUE, progress messages are shown

Value

A NetCDF file including the subtracted fields of infile1 and infile2 is written.

See Also

Other mathematical operators: `cmsaf.addc()`, `cmsaf.add()`, `cmsaf.divc()`, `cmsaf.div()`, `cmsaf.mulc()`, `cmsaf.mul()`, `cmsaf.subc()`, `divdpm()`, `muldpm()`
Examples

## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- c(as.Date("2000-01-01"), as.Date("2001-02-01"))
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data1 <- array(250:350, dim = c(21, 21, 1))
data2 <- array(230:320, dim = c(21, 21, 1))

## create two example NetCDF files

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time[1], unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file_1.nc"), vars)
ncvar_put(ncnew, var1, data1)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time[2], unlim = TRUE)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file_2.nc"), vars)
ncvar_put(ncnew, var1, data2)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Subtract the fields of both example CM SAF NetCDF files and write the result into one output file.
cmsaf.sub("SIS", "SIS", file.path(tempdir(), "CMSAF_example_file_1.nc"), file.path(tempdir(), "CMSAF_example_file_2.nc"), file.path(tempdir(), "CMSAF_example_file_sub.nc"))

unlink(c(file.path(tempdir(), "CMSAF_example_file_1.nc"), file.path(tempdir(), "CMSAF_example_file_2.nc"), file.path(tempdir(), "CMSAF_example_file_sub.nc")))

---

cmsaf.subc

Subtract a constant from a dataset.
Description

This function subtracts a given constant number from each element of a dataset.

Usage

```r
cmsaf.subc(
  var,
  const = 0,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE
)
```

Arguments

- `var`: Name of NetCDF variable (character).
- `const`: Constant number (numeric).
- `infile`: Filename of input NetCDF file. This may include the directory (character).
- `outfile`: Filename of output NetCDF file. This may include the directory (character).
- `nc34`: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- `overwrite`: logical; should existing output file be overwritten?
- `verbose`: logical; if TRUE, progress messages are shown

Value

A NetCDF file including the manipulated data fields of `infile` is written. Standard output precision is 'double'.

See Also

Other mathematical operators: `cmsaf.addc()`, `cmsaf.add()`, `cmsaf.divc()`, `cmsaf.div()`, `cmsaf.mulc()`, `cmsaf.mul()`, `cmsaf.sub()`, `divdpm()`, `muldpm()`

Examples

```r
# Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

# create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
```
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
cnclose(ncnew)

## Subtract a given number from each dataset element of the example CM SAF NetCDF file and write the output to a new file.
cmsaf.subc("SIS", 10, file.path(tempdir(), "CMSAF_example_file.nc"),
file.path(tempdir(), "CMSAF_example_file_subc.nc"))

unlink(c(file.path(tempdir(), "CMSAF_example_file.nc"),
file.path(tempdir(), "CMSAF_example_file_subc.nc")))

cmsafops

cmsafops: A package for analyzing and manipulating CM SAF NetCDF formatted data.

Description

The 'cmsafops' functions are manipulating NetCDF input files and write the result in a separate output file. The functions were designed and tested for CM SAF NetCDF data, but most of the functions can be applied to other NetCDF data, which use the CF convention and time, latitude, longitude dimensions. As interface to NetCDF data the ncdf4 package is used.

Toolbox

The CM SAF R Toolbox is a user-friendly shiny app in the cmsaf package, which helps to apply 'cmsafops' operators.

Mathematical operators

cmsaf.add, cmsaf.addc, cmsaf.div, cmsaf.divc, cmsaf.mul, cmsaf.mulp, cmsaf.sub, cmsaf.subc, 
divdpm, muldpm
Daily statistics
dayrange, ydaymean

Monthly statistics
mon.anomaly, monmax, monmean, monmin, monsd, monsum, multimonmean, multimonsum, ymonmax,
ynmonmean, ymonmin, ymonsd, ymonsum

Seasonal statistics
seas.anomaly, seasmean, seassum, yseasmax, yseasmean, yseasmin, yseassd

Annual statistics
year.anomaly, yearmean, yearsum

Temporal operators
timmax, timmean, timmin, timpctl, timsd, timsum, trend

Spatial operators
fldmax, fldmean, fldmin, wfeldmean

Selection and removal functions
extract.level, extract.period, sellonlatbox, selmon, selperiod, selpoint.multi, selpoint,
seltime, selyear

Data manipulation
box_mergetime, levbox_mergetime, add_grid_info, remap

Other functions
cmsaf.cat, get_time, ncinfo, read_ncvar

Author(s)
Maintainer: Steffen Kothe <Steffen.Kothe@dwd.de>
Contact: CM SAF Team <contact.cmsaf@dwd.de>

References
http://www.cmsaf.eu/R_toolbox
Kothe, S.; Hollmann, R.; Pfeifroth, U.; Träger-Chatterjee, C.; Trentmann, J. The CM SAF R Tool-
box—A Tool for the Easy Usage of Satellite-Based Climate Data in NetCDF Format. ISPRS Int. J.
This function determines the diurnal range.

Description

The function calculates the difference of maximum and minimum values of hourly data from a single CM SAF NetCDF input file.

Usage

dayrange(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)

Arguments

var Name of NetCDF variable (character).

infile Filename of input NetCDF file. This may include the directory (character).

outfile Filename of output NetCDF file. This may include the directory (character).

nc34 NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.

overwrite logical; should existing output file be overwritten?

verbose logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of the diurnal range is written (character).

See Also

Other daily statistics: ydaymean()

Examples

## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(IS0date(2000, 1, 1), IS0date(2000, 1, 6), "hours")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(diff(tIme, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 121))
## create example NetCDF

```r
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", 
                vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)
```

## Determine the diurnal range of the example CM SAF NetCDF file and write the output to a new file.

```r
dayrange("SIS", file.path(tempdir(),"CMSAF_example_file.nc"), 
         file.path(tempdir(),"CMSAF_example_file_dayrange.nc"))
```

```r
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
         file.path(tempdir(),"CMSAF_example_file_dayrange.nc"))
```

---

### divdpm

**Divide by days per month.**

**Description**

This function divides each timestep of a time series by the number of days of the corresponding month. This can be useful to convert units, such as millimeters (mm) to monthly millimeters per day (mm/d). Leap-years are included.

**Usage**

```r
divdpm(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

**Arguments**

- `var`: Name of NetCDF variable (character).
- `infile`: Filename of input NetCDF file. This may include the directory (character).
- `outfile`: Filename of output NetCDF file. This may include the directory (character).
- `nc34`: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- `overwrite`: logical; should existing output file be overwritten?
- `verbose`: logical; if TRUE, progress messages are shown

**Value**

A NetCDF file including a time series of the length of infile is written.
See Also

Other mathematical operators: `cmsaf.addc()`, `cmsaf.add()`, `cmsaf.divc()`, `cmsaf.div()`, `cmsaf.mulc()`, `cmsaf.mul()`, `cmsaf.subc()`, `cmsaf.sub()`, `muldpm()`

Examples

## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

```r
library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
               vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Divide each timestep of the example CM SAF NetCDF file by the number of days per month and write the output to a new file.

divdpm("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
       file.path(tempdir(),"CMSAF_example_file_divdpm.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
        file.path(tempdir(),"CMSAF_example_file_divdpm.nc")))
```

extract.level

Extract levels from 4-dimensional NetCDF files.

Description

This function extracts one or all levels of a 4-dimensional NetCDF file. A level is defined as a dimension, which does not correspond to longitude, latitude or time. The user can choose either one specific level (given by an integer) or all levels (level = "all").
Usage

```r
extract.level(
  var,
  infile,
  outfile,
  level = 1,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE
)
```

Arguments

- **var**: Name of NetCDF variable (character).
- **infile**: Filename of input NetCDF file. This may include the directory (character).
- **outfile**: Filename of output NetCDF file. This may include the directory (character).
- **level**: Number of level (default = 1) or all levels (level = "all") (numeric or character).
- **nc34**: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- **overwrite**: logical; should existing output file be overwritten?
- **verbose**: logical; if TRUE, progress messages are shown

Value

A NetCDF file including the selected level is written. In case of level = "all" all levels are written in separate NetCDF files and outfile names are expanded by ",_level" and the level number.

See Also

Other selection and removal functions: `extract.period()`, `sellonlatbox()`, `selmon()`, `selperiod()`, `selpoint.multi()`, `selpoint()`, `seltime()`, `selyear()`

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
height <- seq(0, 1000, 100)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
```
extract.period

Remove a time period.

Description

This function deletes a time period between a given start and end date from a time series. If start
and end are the same, only this date will be removed.

Usage

extract.period(
  var,
  start,
  end,
  infile,
  outfile,
  nc34 = 4,
extract.period

overwrite = FALSE,
verbose = FALSE
)

Arguments

var Name of NetCDF variable (character).
start Start date as character in form of 'YYYY-MM-DD' (e.g., '2001-12-31').
end End date as character in form of 'YYYY-MM-DD' (e.g., '2014-01-01').
infile Filename of input NetCDF file. This may include the directory (character).
outfile Filename of output NetCDF file. This may include the directory (character).
nc34 NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
overwrite logical; should existing output file be overwritten?
verbose logical; if TRUE, progress messages are shown

Value

A NetCDF file excluding the selected time period is written.

See Also

Other selection and removal functions: extract.level(), sellonlatbox(), selmon(), selperiod(), selpoint.multi(), selpoint(), seltime(), selyear()

Examples

## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Remove a 13-months period of the example CM SAF NetCDF file and write
## the output to a new file.
extract.period("SIS", "2001-01-01", "2002-01-01",
file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_extract.period.nc"))
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_extract.period.nc")))

---

### fldmax

**Determine the spatial maximum**

**Description**

The function determines the maximum value of each timestep from data of a single NetCDF file. The input file should contain a time series of 2D-data.

**Usage**

```r
fldmax(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

**Arguments**

- `var` Name of NetCDF variable (character).
- `infile` Filename of input NetCDF file. This may include the directory (character).
- `outfile` Filename of output NetCDF file. This may include the directory (character).
- `nc34` NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- `overwrite` logical; should existing output file be overwritten?
- `verbose` logical; if TRUE, progress messages are shown

**Value**

A NetCDF file including a time series of maximum values is written.

**See Also**

Other spatial operators: `fldmean()`, `fldmin()`, `wfldmean()`
Examples

### Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: https://wui.cmsaf.eu/

```r
library(ncdf4)
```

```r
## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))
```

```r
## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
cncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
cncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
cnclose(ncnew)
```

```r
## Determine the maximum values of the example CM SAF NetCDF file and write the output to a new file.

fldmax("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_fldmax.nc"))
```

```r
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_fldmax.nc")))
```

---

**fldmean**  
Determine the spatial mean

### Description

The function determines the mean value of each timestep from data of a single NetCDF file. The input file should contain a time series of 2D-data.

### Usage

```r
fldmean(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```
### Arguments

- **var** Name of NetCDF variable (character).
- **infile** Filename of input NetCDF file. This may include the directory (character).
- **outfile** Filename of output NetCDF file. This may include the directory (character).
- **nc34** NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- **overwrite** logical; should existing output file be overwritten?
- **verbose** logical; if TRUE, progress messages are shown

### Value

A NetCDF file including a time series of spatial means is written.

### See Also

Other spatial operators: `fldmax()`, `fldmin()`, `wfldmean()`

### Examples

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the spatial means of the example CM SAF NetCDF file and write the output to a new file.
```
 fldmin("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
 file.path(tempdir(),"CMSAF_example_file fldmean.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
 file.path(tempdir(),"CMSAF_example_file fldmean.nc")))

---

**fldmin**

Determine the spatial minimum.

---

**Description**

The function determines the minimum value of each timestep from data of a single NetCDF file. The input file should contain a time series of 2D-data.

**Usage**

```r
fldmin(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

**Arguments**

- **var**: Name of NetCDF variable (character).
- **infile**: Filename of input NetCDF file. This may include the directory (character).
- **outfile**: Filename of output NetCDF file. This may include the directory (character).
- **nc34**: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- **overwrite**: logical; should existing output file be overwritten?
- **verbose**: logical; if TRUE, progress messages are shown

**Value**

A NetCDF file including a time series of minimum values is written.

**See Also**

Other spatial operators: `fldmax()`, `fldmean()`, `wfldmean()`

**Examples**

```r
## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncd4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
```
\begin{verbatim}
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", 
               vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the minimum values of the example CM SAF NetCDF file and 
## write the output to a new file.
fldmin("SIS", file.path(tempdir(),"CMSAF_example_file.nc"), 
       file.path(tempdir(),"CMSAF_example_file_fldmin.nc"))
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"), 
        file.path(tempdir(),"CMSAF_example_file_fldmin.nc")))
\end{verbatim}

\section*{get\_time}

\textit{Convert time steps to POSIXct.}

\section*{Description}

Times in NetCDF data are generally given in form of a time step and a time unit. This function
uses both information to convert them to POSIXct time values. For the unit 'months since' an
approximation of 30.4375 d is used!

\section*{Usage}

\footnotesize
\begin{verbatim}
get_time(time.unit, time.step)
\end{verbatim}

\section*{Arguments}

\begin{itemize}
  \item \texttt{time.unit} \hspace{1cm} Time unit, which is conform to the CF convention (character).
  \item \texttt{time.step} \hspace{1cm} Time steps in form of a numeric or integer vector.
\end{itemize}

\section*{Value}

Time in form of POSIXct is returned. Default time zone is UTC.
Examples

```r
get_time("hours since 1987-01-01", 249109)
get_time("days since 1987-01-01", 9109)
```

levbox_mergetime

*Function to combine NetCDF files and simultaneously cut a region and level.*

Description

This function selects a region and a level from a bunch of CM SAF NetCDF files that match the same pattern of the filename, and writes the output to a new file. If no longitude and latitude values are given, files are only merged. All input files have to have the same rectangular grid and the same variable. The reference time of the output file is determined by the first input file.

Usage

```r
levbox_mergetime(
  var,
  level = 1,
  path,
  pattern,
  outfile,
  lon1 = -180,
  lon2 = 180,
  lat1 = -90,
  lat2 = 90,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE
)
```

Arguments

- **var**: Name of NetCDF variable (character).
- **level**: Number of level that should be extracted (integer).
- **path**: The directory of input NetCDF files without / at the end (character).
- **pattern**: A part of the filename, which is the same for all desired input files (character). The pattern has to be a character string containing a regular expression.
- **outfile**: Filename of output NetCDF file. This may include the directory (character).
- **lon1**: Longitude of lower left corner (numeric).
- **lon2**: Longitude of upper right left corner (numeric).
- **lat1**: Latitude of lower left corner (numeric).
- **lat2**: Latitude of upper right corner (numeric). Longitude of upper right corner (numeric).
nc34  NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
overwrite  logical; should existing output file be overwritten?
verbose  logical; if TRUE, progress messages are shown

Value

A NetCDF file including the merged time series of the selected region is written. The output NetCDF file contains only the selected level.

See Also

Other data manipulation functions: add_grid_info(), box_mergetime(), remap()

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- c(as.Date("2000-01-01"), as.Date("2001-02-01"))
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
level <- c(1:5)
data1 <- array(250:350, dim = c(21, 21, 5, 1))
data2 <- array(230:320, dim = c(21, 21, 5, 1))

## create two example NetCDF files
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
l <- ncdim_def(name = "level", units = "1", vals = level)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time[1], unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, l, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file_n1.nc"), vars)
ncvar_put(ncnew, var1, data1)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
ncatt_put(ncnew, "level", "standard_name", "level", prec = "text")
nc_close(ncnew)

t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time[2], unlim = TRUE)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file_n2.nc"), vars)
```
mon.anomaly

### Determine monthly anomalies

**Description**

The function subtracts from each timestep of a time series the corresponding multi-year monthly mean. To get monthly anomalies, the input file should contain monthly mean values.

**Usage**

```r
mon.anomaly(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

**Arguments**

- `var` Name of NetCDF variable (character).
- `infile` Filename of input NetCDF file. This may include the directory (character).
- `outfile` Filename of output NetCDF file. This may include the directory (character).
- `nc34` NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- `overwrite` logical; should existing output file be overwritten?
- `verbose` logical; if TRUE, progress messages are shown

**Value**

A NetCDF file including a time series of differences is written.

**See Also**

Other monthly statistics: `monmax()`, `monmean()`, `monmin()`, `monsd()`, `monsum()`, `multimonmean()`, `multimonsum()`, `ymonmax()`, `ymonmean()`, `ymonmin()`, `ymonsd()`, `ymonsum()`
Examples

## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hours"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the monthly anomalies of the example CM SAF NetCDF file and write the output to a new file.

mon.anomaly("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_mon.anomaly.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_mon.anomaly.nc")))

### monmax

**Determine monthly maxima.**

**Description**

The function determines monthly maximum values from data of a single CM SAF NetCDF input file. This function is applicable to 3-dimensional NetCDF data.

**Usage**

```
monmax(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```
Arguments

- **var**: Name of NetCDF variable (character).
- **infile**: Filename of input NetCDF file. This may include the directory (character).
- **outfile**: Filename of output NetCDF file. This may include the directory (character).
- **nc34**: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- **overwrite**: logical; should existing output file be overwritten?
- **verbose**: logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of monthly maxima is written.

See Also

Other monthly statistics: `mon.anomaly()`, `monmean()`, `monmin()`, `monsd()`, `monsum()`, `multimonmean()`, `multimonsum()`, `ymonmax()`, `ymonmean()`, `ymonmin()`, `ymonsd()`, `ymonsum()`

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2000-03-31"), "days")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 91))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
               vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the monthly maximum of the example CM SAF NetCDF file and
```
## monmean

Determine monthly means

### Description

The function determines monthly mean values from data of a single CM SAF NetCDF input file. This function is applicable to 3-dimensional NetCDF data.

### Usage

```r
monmean(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>var</code></td>
<td>Name of NetCDF variable (character).</td>
</tr>
<tr>
<td><code>infile</code></td>
<td>Filename of input NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td><code>outfile</code></td>
<td>Filename of output NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td><code>nc34</code></td>
<td>NetCDF version of output file. If <code>nc34 = 3</code> the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.</td>
</tr>
<tr>
<td><code>overwrite</code></td>
<td>logical; should existing output file be overwritten?</td>
</tr>
<tr>
<td><code>verbose</code></td>
<td>logical; if TRUE, progress messages are shown</td>
</tr>
</tbody>
</table>

### Value

A NetCDF file including a time series of monthly means is written.

### See Also

Other monthly statistics: `mon.anomaly()`, `monmax()`, `monmin()`, `mons()`, `monsum()`, `multimonmean()`, `multimonsum()`, `ymonmax()`, `ymonmean()`, `ymonmin()`, `ymonsum()`, `ymonsd()`.

### Examples

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data
```
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2000-03-31"), "days")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 91))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m\(^{-2}\)", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the monthly mean of the example CM SAF NetCDF file and
## write the output to a new file.
monmean("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_monmean.nc"))
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_monmean.nc")))

---

**monmin**

**Determine monthly minima**

**Description**

The function determines monthly minimum values from data of a single CM SAF NetCDF input file. This function is applicable to 3-dimensional NetCDF data.

**Usage**

```r
monmin(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

**Arguments**

- **var** Name of NetCDF variable (character).
- **infile** Filename of input NetCDF file. This may include the directory (character).
- **outfile** Filename of output NetCDF file. This may include the directory (character).
- **nc34** NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
monmin

overwrite logical; should existing output file be overwritten?
verbose logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of monthly minima is written.

See Also

Other monthly statistics: mon.anomaly(), monmax(), monmean(), monsd(), monsum(), multimonmean(), multimonsum(), ymonmax(), ymonmean(), ymonmin(), ymonsd(), ymonsum()

Examples

## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lad <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2000-03-31"), "days")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 91))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
               vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the monthly minimum of the example CM SAF NetCDF file and
## write the output to a new file.
monmin("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
       file.path(tempdir(),"CMSAF_example_file_monmin.nc"))
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
       file.path(tempdir(),"CMSAF_example_file_monmin.nc")))
monsd  

**Determine monthly standard deviations**

**Description**

The function determines monthly standard deviation values from data of a single CM SAF NetCDF input file. This function is applicable to 3-dimensional NetCDF data.

**Usage**

```r
monsd(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

**Arguments**

- `var`  Name of NetCDF variable (character).
- `infile`  Filename of input NetCDF file. This may include the directory (character).
- `outfile`  Filename of output NetCDF file. This may include the directory (character).
- `nc34`  NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- `overwrite`  logical; should existing output file be overwritten?
- `verbose`  logical; if TRUE, progress messages are shown

**Value**

A NetCDF file including a time series of monthly minima is written.

**See Also**

Other monthly statistics: `mon.anomaly()`, `monmax()`, `monmean()`, `monmin()`, `monsum()`, `multimonmean()`, `multimonsum()`, `ymonmax()`, `ymonmean()`, `ymonmin()`, `ymonsd()`, `ymonsum()`

**Examples**

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2000-03-31"), "days")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 91))
```
## Create Example NetCDF

```r
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", 
               vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
cnclose(ncnew)
```

## Determine the Monthly Standard Deviation of the Example CM SAF NetCDF

```r
# Determine the monthly standard deviation of the example CM SAF NetCDF
# file and write the output to a new file.
monsd("SIS", file.path(tempdir(),"CMSAF_example_file.nc"), 
      file.path(tempdir(),"CMSAF_example_file_monsd.nc"))
```

```r
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
        file.path(tempdir(),"CMSAF_example_file_monsd.nc")))
```

---

### Description

The function determines monthly sums from data of a single CM SAF NetCDF input file. This function is applicable to 3-dimensional NetCDF data.

### Usage

```r
monsum(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

### Arguments

- **var**: Name of NetCDF variable (character).
- **infile**: Filename of input NetCDF file. This may include the directory (character).
- **outfile**: Filename of output NetCDF file. This may include the directory (character).
- **nc34**: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- **overwrite**: logical; should existing output file be overwritten?
- **verbose**: logical; if TRUE, progress messages are shown

### Value

A NetCDF file including a time series of monthly sums is written.
### muldpm

**Multiply by days per month.**

### Description

This function multiplies each timestep of a time series by the number of days of the corresponding month. This can be useful to convert units, such as monthly millimeters per day (mm/d) to millimeters (mm). Leap-years are included.

### See Also

Other monthly statistics: `mon.anomaly()`, `monmax()`, `monmean()`, `monmin()`, `monsd()`, `multimonmean()`, `multimonsum()`, `ymonmax()`, `ymonmean()`, `ymonmin()`, `ymonsd()`, `ymonsum()`
muldpm

Usage

muldpm(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)

Arguments

var Name of NetCDF variable (character).
infile Filename of input NetCDF file. This may include the directory (character).
outfile Filename of output NetCDF file. This may include the directory (character).
nc34 NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
overwrite logical; should existing output file be overwritten?
verbose logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of the length of infile is written.

See Also

Other mathematical operators: cmsaf.addc(), cmsaf.add(), cmsaf.divc(), cmsaf.div(), cmsaf.mulc(),
 cmsaf.mul(), cmsaf.subc(), cmsaf.sub(), divdpm()

Examples

## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
            vals = time, unlim = TRUE)
var1 <- ncvart_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
cmp_var(ncnew, var1, data)
multimonmean

Determine multi-monthly means

Description

The function determines multi-monthly mean values from data of a single CM SAF NetCDF input file. The months are given as a vector of integers from 1 to 12. This allows means of user-defined seasons.

Usage

multimonmean(
  var,
  month = c(1),
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>var</td>
<td>Name of NetCDF variable (character).</td>
</tr>
<tr>
<td>month</td>
<td>Months which should be averaged, in form of a comma separated vector of integer values from 1 to 12 (integer).</td>
</tr>
<tr>
<td>infile</td>
<td>Filename of input NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>outfile</td>
<td>Filename of output NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>nc34</td>
<td>NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.</td>
</tr>
<tr>
<td>overwrite</td>
<td>logical; should existing output file be overwritten?</td>
</tr>
<tr>
<td>verbose</td>
<td>logical; if TRUE, progress messages are shown</td>
</tr>
</tbody>
</table>
Value

A NetCDF file including a time series of multi-monthly means is written.

See Also

Other monthly statistics: `mon.anomaly()`, `monmax()`, `monmean()`, `monmin()`, `monsd()`, `monsum()`, `multimonsum()`, `ymonmax()`, `ymonmean()`, `ymonmin()`, `ymonsd()`, `ymonsum()`

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
               vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the mean of the monsoon seas from June to September of the
## example CM SAF NetCDF file and write the output to a new file.
multimonmean("SIS", c(6, 7, 8, 9), file.path(tempdir(), "CMSAF_example_file.nc"),
               file.path(tempdir(), "CMSAF_example_file_multimonmean.nc"))

unlink(c(file.path(tempdir(), "CMSAF_example_file.nc"),
        file.path(tempdir(), "CMSAF_example_file_multimonmean.nc")))
```
multimonsum

Determine multi-monthly sums

Description

The function determines multi-monthly sums from data of a single CM SAF NetCDF input file. The months are given as a vector of integers from 1 to 12. This allows sums of user-defined seasons.

Usage

```r
multimonsum(
  var,
  month = c(1),
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE
)
```

Arguments

- **var**: Name of NetCDF variable (character).
- **month**: Months which should be averaged, in form of a comma separated vector of integer values from 1 to 12 (integer).
- **infile**: Filename of input NetCDF file. This may include the directory (character).
- **outfile**: Filename of output NetCDF file. This may include the directory (character).
- **nc34**: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- **overwrite**: logical; should existing output file be overwritten?
- **verbose**: logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of multi-monthly sums is written.

See Also

Other monthly statistics: `mon.anomaly()`, `monmax()`, `monmean()`, `monmin()`, `monsd()`, `monsum()`, `multimonmean()`, `ymonmax()`, `ymonmean()`, `ymonmin()`, `ymonsd()`, `ymonsum()`
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

```r
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(0:150, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
               vals = time, unlim = TRUE)
var1 <- ncvar_def("rain", "mm", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the sum of the monsoon seas from June to September of the example CM SAF NetCDF file and write the output to a new file.
multimonsum("rain", c(6, 7, 8, 9), file.path(tempdir(),"CMSAF_example_file.nc"),
            file.path(tempdir(),"CMSAF_example_file_multimonsum.nc"))
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
        file.path(tempdir(),"CMSAF_example_file_multimonsum.nc")))
```

---

### ncinfo

Get information about the content of a NetCDF file.

#### Description

Shows the content of a NetCDF file in three different detail levels.

#### Usage

```r
ncinfo(infile, info = "s", verbose = FALSE)
```
Arguments

infile Filename of input NetCDF file. This may include the directory (character).
info The output can be: long ('l'), medium ('m') and short ('s') (character). Default is short ('s'). The option 'l' additionally returns a list object with file information.
verbose logical; if TRUE, progress messages are shown

Value

prints the content of the infile NetCDF.

Examples

## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
cnc <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
cncvar_put(cnc, var1, data)
cncatt_put(cnc, "lon", "standard_name", "longitude", prec = "text")
cncatt_put(cnc, "lat", "standard_name", "latitude", prec = "text")
cnc_close(cnc)

## Get information on a medium detail level of the example CM SAF NetCDF file:
ncinfo(file.path(tempdir(),"CMSAF_example_file.nc"), "m")
unlink(file.path(tempdir(),"CMSAF_example_file.nc"))
**read_ncvar**

**Read NetCDF variable.**

**Description**

This simple function reads a variable of a NetCDF file into R.

**Usage**

```r
read_ncvar(var, infile, verbose = FALSE)
```

**Arguments**

- `var`: Name of NetCDF variable (character).
- `infile`: Filename of input NetCDF file. This may include the directory (character).
- `verbose`: logical; if TRUE, progress messages are shown

**Value**

The output is a list object including the variable and the corresponding time variable. The dimension of the chosen variable is most commonly a two or three dimensional array.

**Examples**

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
cvar_put(ncnew, var1, data)
```
remap(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Load the data of variable 'SIS' of the example file into R. To
## access the data use e.g., my.data$SIS
my.data <- read_ncvar("SIS", file.path(tempdir(),"CMSAF_example_file.nc"))
unlink(file.path(tempdir(),"CMSAF_example_file.nc"))

---

remap

Grid interpolation.

---

**Description**

The function interpolates the data of infile1 to the grid of infile2. From infile2 only the grid information is used. By default, a nearest neighbor interpolation provided by `get.knnx` is used. For interpolation between regular grids a simple bilinear interpolation as provided by `interp.surface.grid` as well as a conservative remapping as provided by `remapcon` can be chosen.

**Usage**

```r
remap(
  var,  # Name of NetCDF variable (character).
  infile1,  # Filename of first input NetCDF file. This may include the directory (character). The data of infile1 are interpolated.
  infile2,  # Filename of second input file. This may include the directory (character). The grid information of infile2 are the target grid for the interpolation. This File may also be an ASCII-File containing the grid information.
  outfile,  # Filename of output NetCDF file. This may include the directory (character).
  method = "nearest",  # Method used for remapping (character). Options are "bilinear" for bilinear interpolation, "conservative" for conservative remapping (only for regular grids, respectively) and "nearest" for nearest-neighbor interpolation. Default is "nearest".
  overwrite = FALSE,  # Flag to overwrite the output file (logical).
  verbose = FALSE)  # Print information during execution (logical).
```

**Arguments**

- `var`  
  Name of NetCDF variable (character).
- `infile1`  
  Filename of first input NetCDF file. This may include the directory (character). The data of infile1 are interpolated.
- `infile2`  
  Filename of second input file. This may include the directory (character). The grid information of infile2 are the target grid for the interpolation. This File may also be an ASCII-File containing the grid information.
- `outfile`  
  Filename of output NetCDF file. This may include the directory (character).
- `method`  
  Method used for remapping (character). Options are "bilinear" for bilinear interpolation, "conservative" for conservative remapping (only for regular grids, respectively) and "nearest" for nearest-neighbor interpolation. Default is "nearest".
remap

nc34 NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.

overwrite logical; should existing output file be overwritten?

verbose logical; if TRUE, progress messages are shown

Value

A NetCDF file including the interpolated data of infile1 on the grid of infile2 is written.

See Also

Other data manipulation functions: add_grid_info(), box_mergetime(), levbox_mergetime()

Examples

## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lats <- seq(45, 55, 0.5)
lon2 <- seq(5, 15, 1)
lats2 <- seq(45, 55, 1)
time <- c(as.Date("2000-01-01"), as.Date("2001-02-01"))
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hours since 1983-01-01 00:00:00",
vals = time[1], unlim = TRUE)

data1 <- array(250:350, dim = c(21, 21, 1))
data2 <- array(230:320, dim = c(21, 21, 1))

## create two example NetCDF files

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time[1], unlim = TRUE)

var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)

cnew <- nc_create(file.path(tempdir(),"CMSAF_example_file_1.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon2)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat2)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time[1], unlim = TRUE)

cnew <- nc_create(file.path(tempdir(),"CMSAF_example_file_2.nc"), vars)
seas.anomaly(ncnew, var1, data2)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Interpolate the fields of both example CM SAF NetCDF file 1 to the
## coarser grid of file 2 and write the result into one output file.
remap("SIS", file.path(tempdir(),"CMSAF_example_file_1.nc"),
file.path(tempdir(),"CMSAF_example_file_2.nc"),
file.path(tempdir(),"CMSAF_example_file_remap.nc"))
unlink(c(file.path(tempdir(),"CMSAF_example_file_1.nc"),
file.path(tempdir(),"CMSAF_example_file_2.nc"),
file.path(tempdir(),"CMSAF_example_file_remap.nc")))

seas.anomaly

Determine seasonal anomalies.

Description

The function determines the seasonal means of a time series and subtracts the corresponding multi-
seasonal means to get seasonal anomalies.

Usage

seas.anomaly(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE
)

Arguments

var             Name of NetCDF variable (character).
infile          Filename of input NetCDF file. This may include the directory (character).
outfile         Filename of output NetCDF file. This may include the directory (character).
nc34            NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3
                format (numeric). Default output is NetCDFv4.
overwrite       logical; should existing output file be overwritten?
verbose         logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of seasonal anomalies is written.
seasmean

Determine seasonal means

Description

The function determines seasonal mean values from data of a single CM SAF NetCDF input file. The seasonal mean is only determined if all three months of a season are available. For (north-) winter this are January, February and the December of the previous year (DJF). The other seasons are MAM, JJA, and SON.
seasmean

Usage

seasmean(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)

Arguments

var Name of NetCDF variable (character).
infile Filename of input NetCDF file. This may include the directory (character).
outfile Filename of output NetCDF file. This may include the directory (character).
nc34 NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
overwrite logical; should existing output file be overwritten?
verbose logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of seasonal means is written.

See Also

Other seasonal statistics: seas.anomaly(), seassum(), yseasmax(), yseasmean(), yseasmin(), yseassd()

Examples

## Create an example NetCDF file with a similar structure as used by CM SAFe. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
latt <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the seasonal means of the example CM SAF NetCDF file and
## write the output to a new file.
seasmean("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
          file.path(tempdir(),"CMSAF_example_file_seasmean.nc"))
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
          file.path(tempdir(),"CMSAF_example_file_seasmean.nc")))

### seassum

Determine seasonal sums

**Description**

The function determines seasonal sum values from data of a single CM SAF NetCDF input file. The seasonal sum is only determined if all three months of a season are available. For (north-) winter this are January, February and the December of the previous year (DJF). The other seasons are MAM, JJA, and SON.

**Usage**

seassum(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)

**Arguments**

- **var**: Name of NetCDF variable (character).
- **infile**: Filename of input NetCDF file. This may include the directory (character).
- **outfile**: Filename of output NetCDF file. This may include the directory (character).
- **nc34**: NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- **overwrite**: logical; should existing output file be overwritten?
- **verbose**: logical; if TRUE, progress messages are shown

**Value**

A NetCDF file including a time series of seasonal sums is written.

**See Also**

Other seasonal statistics: seas.anomaly(), seasmean(), yseasmax(), yseasmean(), yseasmin(), yseassd()
Examples

## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

```r
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
cnc <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(cnc, var1, data)
ncatt_put(cnc, "lon", "standard_name", "longitude", prec = "text")
ncc <- nc_close(cnc)

## Determine the seasonal sums of the example CM SAF NetCDF file and write the output to a new file.
seassum("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_seassum.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_seassum.nc"))
```

---

**sellonlatbox**  
Select a region by longitude and latitude.

---

**Description**

This function cuts a region from data of a CM SAF NetCDF file. The region is selected by giving the coordinates of the lower left and upper right corner of a rectangular grid area.

**Usage**

```r
sellonlatbox()
```
sellonlatbox

var,
infile,
outfile,
lon1 = -180,
lon2 = 180,
lat1 = -90,
lat2 = 90,
c34 = 4,
overwrite = FALSE,
verbose = FALSE
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>var</td>
<td>Name of NetCDF variable (character).</td>
</tr>
<tr>
<td>infile</td>
<td>Filename of input NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>outfile</td>
<td>Filename of output NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>lon1</td>
<td>Longitude of lower left corner (numeric).</td>
</tr>
<tr>
<td>lon2</td>
<td>Longitude of upper right corner (numeric).</td>
</tr>
<tr>
<td>lat1</td>
<td>Latitude of lower left corner (numeric).</td>
</tr>
<tr>
<td>lat2</td>
<td>Latitude of upper right corner (numeric).</td>
</tr>
<tr>
<td>nc34</td>
<td>NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.</td>
</tr>
<tr>
<td>overwrite</td>
<td>logical; should existing output file be overwritten?</td>
</tr>
<tr>
<td>verbose</td>
<td>logical; if TRUE, progress messages are shown</td>
</tr>
</tbody>
</table>

Value

A NetCDF file including the selected region is written.

See Also

Other selection and removal functions: `extract.level()`, `extract.period()`, `selmon()`, `selperiod()`, `selpoint.multi()`, `selpoint()`, `seltime()`, `selyear()`

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
```
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Cut a region of the example CM SAF NetCDF file and write the output
## to a new file.
sellonlatbox("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_sellonlatbox.nc"), 8, 12, 48, 52)

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_sellonlatbox.nc")))

---

**selmon**

*Extract a list of months.*

**Description**

This function selects a given list of months from a time series.

**Usage**

```r
selmon(
  var,  
  month = c(1),  
  infile,  
  outfile,  
  nc34 = 4,  
  overwrite = FALSE,  
  verbose = FALSE
)
```

**Arguments**

- **var** Name of NetCDF variable (character).
month    Months, which should be selected, in form of a comma separated vector of integer values from 1 to 12 (integer).
infile   Filename of input NetCDF file. This may include the directory (character).
outfile  Filename of output NetCDF file. This may include the directory (character).
nc34     NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
overwrite logical; should existing output file be overwritten?
verbose  logical; if TRUE, progress messages are shown

Value
A NetCDF file including a time series of the selected month is written.

See Also
Other selection and removal functions: `extract.level()`, `extract.period()`, `sellonlatbox()`, `selperiod()`, `selpoint.multi()`, `selpoint()`, `seltime()`, `selyear()`

Examples
```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
                vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Select all March and June values of the example CM SAF NetCDF file
```
## selperiod

Extract a list of dates.

### Description

This function selects a time period from a time series.

### Usage

```r
selperiod(
  var,
  start,
  end,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE
)
```

### Arguments

- `var` Name of NetCDF variable (character).
- `start` Start date as character in form of 'YYYY-MM-DD' (e.g., '2001-12-31').
- `end` End date as character in form of 'YYYY-MM-DD' (e.g., '2001-12-31').
- `infile` Filename of input NetCDF file. This may include the directory (character).
- `outfile` Filename of output NetCDF file. This may include the directory (character).
- `nc34` NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- `overwrite` logical; should existing output file be overwritten?
- `verbose` logical; if TRUE, progress messages are shown

### Value

A NetCDF file including the selected time period is written.

### See Also

Other selection and removal functions: `extract.level()`, `extract.period()`, `sellonlatbox()`, `selmon()`, `selpoint.multi()`, `selpoint()`, `seltime()`, `selyear()`
Examples

## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

```r
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lng <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lng)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Select a 13-months period of the example CM SAF NetCDF file and write the output to a new file.
seelperiod("SIS", "2001-01-01", "2002-01-01", file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_seelperiod.nc"))
```

## selpoint

Extract data at a given point.

Description

This function extracts all data at a given point. A point is given by a pair of longitude and latitude coordinates. The function will find the closest grid point to the given coordinates and extracts the data for this point. The output file can be optional in NetCDF or csv. The outfile is checked for the correct file extension.
Usage

selpoint(
    var,  
    infile, 
    outfile, 
    lon1 = 0, 
    lat1 = 0, 
    format = "nc", 
    nc34 = 4, 
    overwrite = FALSE, 
    verbose = FALSE
)

Arguments

var     Name of NetCDF variable (character).
infile   Filename of input NetCDF file. This may include the directory (character).
outfile  Filename of output NetCDF file. This may include the directory (character).
lon1     Longitude of desired point (numeric).
lat1     Latitude of desired point (numeric).
format   Intended output format. Options are nc or csv. Default is nc (character).
nc34     NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
overwrite logical; should existing output file be overwritten?
verbose  logical; if TRUE, progress messages are shown

Value

A NetCDF or csv file including the selected point is written. The csv file is tested for use in Excel and includes two columns (Time and Data), which are separated by ‘;’.

See Also

Other selection and removal functions: extract.level(), extract.period(), sellonlatbox(), selmon(), selperiod(), selpoint.multi(), seltime(), selyear()

Examples

## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Select a point of the example CM SAF NetCDF file and write the output
## to a csv-file.
selpoint("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_selpoint.nc"),
8, 48, "csv")

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_selpoint.nc.csv")))

selpoint.multi  Extract data at multiple points.

Description

This function extracts all data at given points. The points are given by a pair of vectors with longitude and latitude coordinates. The function will find the closest grid points to the given coordinates and extracts the data for these points. For each point a separate output file is written. The output-files can be optional in NetCDF or csv. Input can be a single NetCDF file (given by the infile attribute) or a bunch of NetCDF files (given by the path and pattern attributes).

Usage

selpoint.multi(
  var,
  infile,
  path,
  pattern,
  outpath,
  lon1,
Arguments

- **var**: Name of NetCDF variable (character).
- **infile**: Filename of input NetCDF file. This may include the directory (character). Infile is not needed if path and pattern are given.
- **path**: Directory of input files (character). Will not be used if infile is given.
- **pattern**: Pattern that all desired files in the 'path' directory have in common (character).
- **outpath**: Directory where output files will be stored (character).
- **lon1**: Longitude vector of desired points (numeric vector). Must have the same length as lat1.
- **lat1**: Latitude vector of desired points (numeric vector). Must have the same length as lon1.
- **station_names**: Optional vector of names, which will be used for the output files (character vector). Must have the same length as lon1 and lat1.
- **format**: Intended output format. Options are **nc** or **csv**. Default is **nc** (character).
- **nc34**: NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- **verbose**: logical; if TRUE, progress messages are shown.

Value

For each pair of longitude and latitude coordinates one separate NetCDF or csv file including the selected data is written. The csv files are tested for use in Excel and include four columns (Time ; Data ; Longitude ; Latitude), which are separated by ';'. If station_names are defined, the output files will be named according to this vector. Otherwise, the output files will be named as selpoint_longitude_latitude.format. Already existing files will be overwritten in case that station_names are given or renamed (e.g., selpoint1_longitude_latitude.nc) in case that no station_names are given.

See Also

Other selection and removal functions: extract.level(), extract.period(), sellonlatbox(), selmon(), seperiod(), selpoint(), seltime(), selyear()}

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>```
library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
               vals = time, unlim = TRUE)

var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")

vars <- list(var1)

ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Select two points of the example CM SAF NetCDF file and write the
## output to a csv-file.

setpoint.multi(var = "SIS", infile = file.path(tempdir(), "CMSAF_example_file.nc"),
               outpath = tempdir(), lon1 = c(8, 9), lat1 = c(48, 49),
               station_names = c("A", "B"), format = "csv")

unlink(c(file.path(tempdir(), "CMSAF_example_file.nc"), file.path(tempdir(), "A.csv"),
        file.path(tempdir(), "B.csv")))

---

**seltime**

Extract specific timestep.

**Description**

This function selects a given list of times from a time series.

**Usage**

```
seltime(
  var,
  hour_min = c("00:00:00"),
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
)```

verbose = FALSE
)

Arguments

var Name of NetCDF variable (character).
hour_min Times, which should be selected, in form of a vector of character values in
the form of ‘HH:MM:SS’ (e.g. c(‘12:00:00’) (character).
infile Filename of input NetCDF file. This may include the directory (character).
outfile Filename of output NetCDF file. This may include the directory (character).
nc34 NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3
format (numeric). Default output is NetCDFv4.
overwrite logical; should existing output file be overwritten?
verbose logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of the selected times is written.

See Also

Other selection and removal functions: extract.level(), extract.period(), sellonlatbox(),
selmon(), selperiod(), selpoint.multi(), selpoint(), selyear()

Examples

## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(ISODATE(2000, 1, 1), ISODATE(2000, 1, 6), "hours")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 121))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time, unlim = TRUE)
var1 <- ncvvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Select all 12:00 and 21:00 values of the example CM SAF NetCDF file
## and write the output to a new file.
seltime("SIS", c("12:00:00", "21:00:00"), file.path(tempdir(),"CMSAF_example_file.nc"),
   file.path(tempdir(),"CMSAF_example_file_seltime.nc"))
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
   file.path(tempdir(),"CMSAF_example_file_seltime.nc")))

## selyear

### Description

This function selects a given list of years from a time series.

### Usage

```r
selyear(
  var, 
  year = c(2000),
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE
)
```

### Arguments

- **var**: Name of NetCDF variable (character).
- **year**: Year in form of a comma separated vector of integer values (e.g. c(2000,2015)) (integer).
- **infile**: Filename of input NetCDF file. This may include the directory (character).
- **outfile**: Filename of output NetCDF file. This may include the directory (character).
- **nc34**: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- **overwrite**: logical; should existing output file be overwritten?
- **verbose**: logical; if TRUE, progress messages are shown

### Value

A NetCDF file including a time series of the selected years is written.
timcumsum

Accumulate data of NetCDF file.

Description

Computes the accumulation of the given variable over time. The resulting outfile has the same dimensions as the infile.
**Usage**

```r
timcumsum(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  na_replace = "mean",
  verbose = FALSE
)
```

**Arguments**

- `var` Name of variable in `infile` (character).
- `infile` Character containing file name or path of input file.
- `outfile` Character containing file name or path of output file. If NULL, the input file is directly edited instead.
- `nc34` NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- `overwrite` Logical; should existing output file be overwritten? If `outfile` is NULL, this parameter is ignored.
- `na_replace` Replacing NA values with either 'mean' or 'previous' for monthly mean or previous value, respectively (character).
- `verbose` logical; if TRUE, progress messages are shown

**Description**

The function determines all-time maximum values from data of a single CM SAF NetCDF input file. This function is applicable to 3-dimensional NetCDF data.

**Usage**

```r
timmax(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

**Arguments**

- `var` Name of NetCDF variable (character).
- `infile` Filename of input NetCDF file. This may include the directory (character).
- `outfile` Filename of output NetCDF file. This may include the directory (character).
- `nc34` NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- `overwrite` logical; should existing output file be overwritten?
- `verbose` logical; if TRUE, progress messages are shown
Value

A NetCDF file including a time series of all-time maxima is written.

See Also

Other temporal operators: timmean(), timmin(), timpctl(), timsd(), timsum(), trend()

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncd4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lv <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2000-03-31"), "days")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 91))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
   vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the all-time maximum of the example CM SAF NetCDF file and
## write the output to a new file.
timmax("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
   file.path(tempdir(),"CMSAF_example_file_timmax.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
   file.path(tempdir(),"CMSAF_example_file_timmax.nc")))
```

---

**timmean**

Determine all-time mean.
**Description**

The function determines the all-time mean from data of a single CM SAF NetCDF input file and is useful to calculate climatological means. The function limits the timesteps, which are read at once, to avoid RAM overflow.

**Usage**

```
timmean(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

**Arguments**

- `var`: Name of NetCDF variable (character).
- `infile`: Filename of input NetCDF file. This may include the directory (character).
- `outfile`: Filename of output NetCDF file. This may include the directory (character).
- `nc34`: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- `overwrite`: logical; should existing output file be overwritten?
- `verbose`: logical; if TRUE, progress messages are shown

**Value**

A NetCDF file including the temporal mean is written.

**See Also**

Other temporal operators: `timmax()`, `timmin()`, `timpctl()`, `timsd()`, `timsum()`, `trend()`

**Examples**

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
```
vals = time, unlim = TRUE)
var1 <- ncvarg_def("SIS", "W m\(^{-2}\)", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the climatology of the example CM SAF NetCDF file and write
## the output to a new file.
timmean("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
       file.path(tempdir(),"CMSAF_example_file_timmean.nc"))
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
       file.path(tempdir(),"CMSAF_example_file_timmean.nc")))

---

### timmin

**Determine all-time minima.**

**Description**

The function determines all-time minimum values from data of a single CM SAF NetCDF input file. This function is applicable to 3-dimensional NetCDF data.

**Usage**

```r
timmin(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

**Arguments**

- `var`  Name of NetCDF variable (character).
- `infile`  Filename of input NetCDF file. This may include the directory (character).
- `outfile`  Filename of output NetCDF file. This may include the directory (character).
- `nc34`  NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- `overwrite`  logical; should existing output file be overwritten?
- `verbose`  logical; if TRUE, progress messages are shown

**Value**

A NetCDF file including a time series of all-time minima is written.

**See Also**

Other temporal operators: `timmax()`, `timmean()`, `timpctl()`, `timsd()`, `timsum()`, `trend()`
Examples

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2000-03-31"), "days")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 91))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", 
vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the all-time minimum of the example CM SAF NetCDF file and write the output to a new file.
timmin("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_timmin.nc"))
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_timmin.nc")))
```

### timpctl

**Determine percentile over all timesteps.**

**Description**

The function determines a given percentile over all timesteps from data of a single CM SAF NetCDF input file.

**Usage**

```r
timpctl()
```
```r
var,
p = 0.95,
infile,
outfile,
nc34 = 4,
overwrite = FALSE,
verbose = FALSE
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>var</td>
<td>Name of NetCDF variable (character).</td>
</tr>
<tr>
<td>p</td>
<td>Percentile number given as probability within [0, 1] (numeric). Default is 0.95.</td>
</tr>
<tr>
<td>infile</td>
<td>Filename of input NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>outfile</td>
<td>Filename of output NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>nc34</td>
<td>NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.</td>
</tr>
<tr>
<td>overwrite</td>
<td>logical; should existing output file be overwritten?</td>
</tr>
<tr>
<td>verbose</td>
<td>logical; if TRUE, progress messages are shown</td>
</tr>
</tbody>
</table>

Value

A NetCDF file including a time series of all-time seasonal standard deviations is written.

See Also

Other temporal operators: `timmax()`, `timmean()`, `timmin()`, `timsd()`, `timsum()`, `trend()`

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
```

```
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the 90% percentile of the example CM SAF NetCDF file and
## write the output to a new file.
timpctl("SIS", 0.9, file.path(tempdir(), "CMSAF_example_file.nc"),
        file.path(tempdir(), "CMSAF_example_file_timpctl.nc"))
unlink(c(file.path(tempdir(), "CMSAF_example_file.nc"),
        file.path(tempdir(), "CMSAF_example_file_timpctl.nc")))

timsd

Determine all-time standard deviations.

Description

The function determines all-time standard deviation values from data of a single CM SAF NetCDF input file.

Usage

timsd(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)

Arguments

  var      Name of NetCDF variable (character).
  infile   Filename of input NetCDF file. This may include the directory (character).
  outfile  Filename of output NetCDF file. This may include the directory (character).
  nc34     NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
  overwrite logical; should existing output file be overwritten?
  verbose  logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of all-time standard deviations is written.

See Also

Other temporal operators: timmax(), timmean(), timmin(), timpctl(), timsum(), trend()
Examples

## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
  vals = time, unlim = TRUE);
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the all-time seasonal standard deviation of the example CM SAF NetCDF file and write the output to a new file.
timsd("SIS", file.path(tempdir(), "CMSAF_example_file.nc"),
  file.path(tempdir(), "CMSAF_example_file_timsd.nc"))
unlink(c(file.path(tempdir(), "CMSAF_example_file.nc"),
  file.path(tempdir(), "CMSAF_example_file_timsd.nc")))

---

### timsum

**Determine all-time sum.**

**Description**

The function determines the temporal sum from data of a single CM SAF NetCDF input file and is useful to calculate climatological sums. The function limits the timesteps, which are read at once, to avoid RAM overflow.

**Usage**

```r
timsum(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```
Arguments

- **var**
  - Name of NetCDF variable (character).

- **infile**
  - Filename of input NetCDF file. This may include the directory (character).

- **outfile**
  - Filename of output NetCDF file. This may include the directory (character).

- **nc34**
  - NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.

- **overwrite**
  - Logical; should existing output file be overwritten?

- **verbose**
  - Logical; if TRUE, progress messages are shown

Value

A NetCDF file including the temporal sum is written.

See Also

Other temporal operators: `timmax()`, `timmean()`, `timmin()`, `timpctl()`, `timsd()`, `trend()`

Examples

```r
## Create an example NetCDF file with a similar structure as used by CMSAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncclose(ncnew)

## Determine the all-time sum of the example CM SAF NetCDF file and
## write the output to a new file.
```
trend("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_timsum.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_timsum.nc")))

trend

Determine linear trends.

Description

The function determines the trend from data of a single CM SAF NetCDF input file basing on a
simple linear model. Depending on the file size, this function could be very time consuming, thus
there are two available options. Option 1 (default) is using an apply approach and will read the
whole data in once. This option is quite fast, but requires enough memory. Option 2 is using the
same calculation, but reads the data pixel by pixel, which is very slow, but can also be applied for
large data files, which would not fit into the memory at once.

Usage

trend(
  var,
  infile,
  outfile,
  option = 1,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE
)

Arguments

var Name of NetCDF variable (character).
infile Filename of input NetCDF file. This may include the directory (character).
outfile Filename of output NetCDF file. This may include the directory (character).
option The way of data handling. Option = 1 is fast but memory consuming (default).
        Option = 2 is slow, but needs much less memory. Input is either 1 or 2 (numeric).
nc34 NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3
       format (numeric). Default output is NetCDFv4.
overwrite logical; should existing output file be overwritten?
verbose logical; if TRUE, progress messages are shown
Value

A NetCDF file including three data layers is written. One layer (trend1) contains the linear trend multiplied by the number of time steps. In older versions of the package (<= 1.7) the trend was given in the same way as trend1. Another layer (trend2) contains just the calculated linear trend. An additional layer contains a measure for the significance of the calculated trends, which was derived using the 95% confidence interval. The significance is calculated from the lower and upper value of the 95% confidence interval: lower or upper value < 0: sig = 0 (not significant); lower and upper value < 0: sig = -1 (negative significant); lower and upper value > 0: sig = 1 (positive significant).

See Also

Other temporal operators: timmax(), timmean(), timmin(), timpctl(), timsd(), timsum()

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the trend of the example CM SAF NetCDF file and write the
## output to a new file.
trend("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_trend.nc"))
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
```
Determine the weighted spatial mean.

Description

The function determines area weighted mean values from data of a single file. The calculation is based on the `weighted.mean` function of the `raster` package.

Usage

```r
wfldmean(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

Arguments

- `var`: Name of NetCDF variable (character).
- `infile`: Filename of input NetCDF file. This may include the directory (character).
- `outfile`: Filename of output NetCDF file. This may include the directory (character).
- `nc34`: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- `overwrite`: logical; should existing output file be overwritten?
- `verbose`: logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of area weighted spatial means is written.

See Also

Other spatial operators: `fldmax()`, `fldmean()`, `fldmin()`

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(diff(time(time, origin, units = "hour")))
```
ydaymean

Determine multi-year daily means.

Description

The function determines multi-year daily mean values from data of a single CM SAF NetCDF input file.

Usage

ydaymean(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)

Arguments

<table>
<thead>
<tr>
<th>var</th>
<th>Name of NetCDF variable (character).</th>
</tr>
</thead>
<tbody>
<tr>
<td>infile</td>
<td>Filename of input NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>outfile</td>
<td>Filename of output NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>nc34</td>
<td>NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.</td>
</tr>
<tr>
<td>overwrite</td>
<td>logical; should existing output file be overwritten?</td>
</tr>
<tr>
<td>verbose</td>
<td>logical; if TRUE, progress messages are shown</td>
</tr>
</tbody>
</table>

Value

A NetCDF file including a time series of multi-year daily means is written.
year.anomaly  

Determine annual anomalies.

Description

The function determines the annual means of a time series and subtracts the climatology from each mean to get annual anomalies.

See Also

Other daily statistics: dayrange()
year.anomaly

Usage

year.anomaly(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE
)

Arguments

var        Name of NetCDF variable (character).
infile     Filename of input NetCDF file. This may include the directory (character).
outfile    Filename of output NetCDF file. This may include the directory (character).
nc34       NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
overwrite  logical; should existing output file be overwritten?
verbose    logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of annual anomalies is written.

See Also

Other annual statistics: yearmean(), yearsum()

Examples

## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
Determine annual means

The function determines annual mean values from data of a single CM SAF NetCDF input file.

Usage

yearmean(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)

Arguments

var       Name of NetCDF variable (character).
infile    Filename of input NetCDF file. This may include the directory (character).
outfile   Filename of output NetCDF file. This may include the directory (character).
nc34      NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3
          format (numeric). Default output is NetCDFv4.
overwrite  logical; should existing output file be overwritten?
verbose    logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of annual means is written.

See Also

Other annual statistics: year.anomaly(), yearsum()
yearsum

## Description

The function determines annual sums from data of a single CM SAF NetCDF input file.

## Usage

```r
yearsum(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```
Arguments

- **var**: Name of NetCDF variable (character).
- **infile**: Filename of input NetCDF file. This may include the directory (character).
- **outfile**: Filename of output NetCDF file. This may include the directory (character).
- **nc34**: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- **overwrite**: logical; should existing output file be overwritten?
- **verbose**: logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of annual sums is written.

See Also

Other annual statistics: `year.anomaly()`, `yearmean()`

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lam <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lam)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
               vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the annual sums of the example CM SAF NetCDF file and write
## the output to a new file.
yearsum
```
```
Determine multi-year monthly maxima.

Description

The function determines multi-year monthly maximum values from data of a single CM SAF NetCDF input file.

Usage

ymonmax(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)

Arguments

var Name of NetCDF variable (character).
infile Filename of input NetCDF file. This may include the directory (character).
outfile Filename of output NetCDF file. This may include the directory (character).
nc34 NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
overwrite logical; should existing output file be overwritten?
verbose logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of multi-year monthly maxima is written.

See Also

Other monthly statistics: mon.anomaly(), monmax(), monmean(), monmin(), monsd(), monsum(), multimonmean(), multimonsum(), ymonmean(), ymonmin(), ymons(), ymonsum()

Examples

## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the multi-year monthly maximum of the example CM SAF NetCDF
## file and write the output to a new file.
ymonmax("SIS", file.path(tempdir(), "CMSAF_example_file.nc"),
file.path(tempdir(), "CMSAF_example_file_ymonmax.nc"))
unlink(c(file.path(tempdir(), "CMSAF_example_file.nc"),
file.path(tempdir(), "CMSAF_example_file_ymonmax.nc")))

---

### ymonmean

Determine multi-year monthly means.

#### Description

The function determines multi-year monthly mean values from data of a single CM SAF NetCDF input file.

#### Usage

```r
ymonmean(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

#### Arguments

- **var**: Name of NetCDF variable (character).
- **infile**: Filename of input NetCDF file. This may include the directory (character).
- **outfile**: Filename of output NetCDF file. This may include the directory (character).
- **nc34**: NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- **overwrite**: logical; should existing output file be overwritten?
- **verbose**: logical; if TRUE, progress messages are shown
ymonmean

Value

A NetCDF file including a time series of multi-year monthly means is written.

See Also

Other monthly statistics: `mon.anomaly()`, `monmax()`, `monmean()`, `monmin()`, `monsd()`, `monsum()`, `multimonmean()`, `multimonsum()`, `ymonmax()`, `ymonmin()`, `ymonsd()`, `ymonsum()`

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
               vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the multi-year monthly mean of the example CM SAF NetCDF
## file and write the output to a new file.
ymonmean("SIS", file.path(tempdir(), "CMSAF_example_file.nc"),
          file.path(tempdir(), "CMSAF_example_file_ymonmean.nc"))

unlink(c(file.path(tempdir(), "CMSAF_example_file.nc"),
         file.path(tempdir(), "CMSAF_example_file_ymonmean.nc")))
```
ymonmin

Determine multi-year monthly minima.

Description

The function determines multi-year monthly minimum values from data of a single CM SAF NetCDF input file.

Usage

ymonmin(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>var</td>
<td>Name of NetCDF variable (character).</td>
</tr>
<tr>
<td>infile</td>
<td>Filename of input NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>outfile</td>
<td>Filename of output NetCDF file. This may include the directory (character).</td>
</tr>
<tr>
<td>nc34</td>
<td>NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.</td>
</tr>
<tr>
<td>overwrite</td>
<td>logical; should existing output file be overwritten?</td>
</tr>
<tr>
<td>verbose</td>
<td>logical; if TRUE, progress messages are shown</td>
</tr>
</tbody>
</table>

Value

A NetCDF file including a time series of multi-year monthly minima is written.

See Also

Other monthly statistics: `mon.anomaly()`, `monmax()`, `monmean()`, `monmin()`, `monsd()`, `monsum()`, `multimonmean()`, `multimonsum()`, `ymonmax()`, `ymonmean()`, `ymsd()`, `ymonsum()`

Examples

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lats <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))
```
## ymons

### Create Example NetCDF

```r
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m^-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar.Put(ncnew, var1, data)
ncatt.Put(ncnew, "lon", "standard_name", "longitude", prec = "text")
nccat.Put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)
```

## Determine the Multi-Year Monthly Minimum of the Example CM SAF NetCDF File and Write the Output to a New File.

```r
ymonmin("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
         file.path(tempdir(),"CMSAF_example_file_ymonmin.nc"))
```

```r
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
         file.path(tempdir(),"CMSAF_example_file_ymonmin.nc"))
```

---

### ymons Function

**Determine multi-year monthly standard deviations.**

**Description**

The function determines multi-year monthly standard deviation values from data of a single CM SAF NetCDF input file.

**Usage**

```r
ymonsd(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

**Arguments**

- **var**: Name of NetCDF variable (character).
- **infile**: Filename of input NetCDF file. This may include the directory (character).
- **outfile**: Filename of output NetCDF file. This may include the directory (character).
- **nc34**: NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- **overwrite**: logical; should existing output file be overwritten?
- **verbose**: logical; if TRUE, progress messages are shown

**Value**

A NetCDF file including a time series of multi-year monthly standard deviations is written.
See Also

Other monthly statistics: mon.anomaly(), monmax(), monmean(), monmin(), monsd(), monsum(), multimonmean(), multimonsum(), ymonmax(), ymonmean(), ymonmin(), ymonsum()

Examples

## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the multi-year monthly standard deviation of the example CM
## SAF NetCDF file and write the output to a new file.

ymonsd("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_ymonsd.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_ymonsd.nc")))

ymonsum

Determine multi-year monthly sums.

Description

The function determines multi-year monthly sums from data of a single CM SAF NetCDF input file.
ymonsum

Usage

ymonsum(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)

Arguments

var
Name of NetCDF variable (character).

infile
Filename of input NetCDF file. This may include the directory (character).

outfile
Filename of output NetCDF file. This may include the directory (character).

nc34
NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.

overwrite
logical; should existing output file be overwritten?

verbose
logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of multi-year monthly sums is written.

See Also

Other monthly statistics: mon.anomaly(), monmax(), monmean(), monmin(), monsd(), monsum(), multimonmean(), multimonsum(), ymonmax(), ymonmean(), ymonmin(), ymonsd()

Examples

## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(0:150, dim = c(21, 21, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00", vals = time, unlim = TRUE)
var1 <- ncvart_def("rain", "mm", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
cncarp.Put(ncnew, var1, data)
yseasmax

Determine multi-year seasonal maxima.

Description

The function determines multi-year seasonal maximum values from data of a single CM SAF NetCDF input file.

Usage

yseasmax(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)

Arguments

var
Name of NetCDF variable (character).

infile
Filename of input NetCDF file. This may include the directory (character).

outfile
Filename of output NetCDF file. This may include the directory (character).

nc34
NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.

overwrite
logical; should existing output file be overwritten?

verbose
logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of multi-year seasonal maxima is written.

See Also

Other seasonal statistics: seas.anomaly(), seasmean(), seassum(), yseasmean(), yseasmin(), yseassd()
yseasmean

## Examples

```r
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the multi-year seasonal maximum of the example CM SAF
## NetCDF file and write the output to a new file.
yseasmax("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_yseasmax.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_yseasmax.nc")))
```

## yseasmean

**Determine multi-year seasonal means.**

**Description**

The function determines multi-year seasonal mean values from data of a single CM SAF NetCDF input file. The seasonal mean is only determined if all three months of a season are available. For (north-) winter this are January, February and the December of the previous year (DJF). The other seasons are MAM, JJA, and SON.

**Usage**

```r
yseasmean(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```
Arguments

var  Name of NetCDF variable (character).
infile  Filename of input NetCDF file. This may include the directory (character).
outfile  Filename of output NetCDF file. This may include the directory (character).
nc34  NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
overwrite  logical; should existing output file be overwritten?
verbose  logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of multi-year seasonal means is written.

See Also

Other seasonal statistics: seas.anomaly(), seasmean(), seassum(), yseasmax(), yseasmin(), yseassd()

Examples

## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
               vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the multi-year seasonal means of the example CM SAF NetCDF
## yseasmin

Determine multi-year seasonal minima.

### Description

The function determines multi-year seasonal minimum values from data of a single CM SAF NetCDF input file.

### Usage

```r
yseasmin(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)
```

### Arguments

- `var`: Name of NetCDF variable (character).
- `infile`: Filename of input NetCDF file. This may include the directory (character).
- `outfile`: Filename of output NetCDF file. This may include the directory (character).
- `nc34`: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
- `overwrite`: logical; should existing output file be overwritten?
- `verbose`: logical; if TRUE, progress messages are shown

### Value

A NetCDF file including a time series of multi-year seasonal minima is written.

### See Also

Other seasonal statistics: `seas.anomaly()`, `seasmean()`, `seassum()`, `yseasmax()`, `yseasmean()`, `yseassd()`

### Examples

```r
## Create an example NetCDF file with a similar structure as used by CM SAF. The file is created with the ncdf4 package. Alternatively example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
               vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
cnclose(ncnew)

## Determine the multi-year seasonal minimum of the example CM SAF
## NetCDF file and write the output to a new file.

yseasmin("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
         file.path(tempdir(),"CMSAF_example_file_yseasmin.nc"))
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
         file.path(tempdir(),"CMSAF_example_file_yseasmin.nc"))

---

**yseassd**

Determine multi-year seasonal standard deviations.

**Description**

The function determines multi-year seasonal standard deviation values from data of a single CM SAF NetCDF input file.

**Usage**

yseassd(var, infile, outfile, nc34 = 4, overwrite = FALSE, verbose = FALSE)

**Arguments**

- **var**: Name of NetCDF variable (character).
- **infile**: Filename of input NetCDF file. This may include the directory (character).
- **outfile**: Filename of output NetCDF file. This may include the directory (character).
- **nc34**: NetCDF version of output file. If `nc34 = 3` the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
yseassd

overwrite logical; should existing output file be overwritten?
verbose logical; if TRUE, progress messages are shown

Value

A NetCDF file including a time series of multi-year seasonal standard deviations is written.

See Also

Other seasonal statistics: \texttt{seas.anomaly()}, \texttt{seasmean()}, \texttt{seassum()}, \texttt{yseasmx()}, \texttt{yseasmean()}, \texttt{yseasmin()}

Examples

## Create an example NetCDF file with a similar structure as used by CM
## SAF. The file is created with the ncdf4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>
library(ncdf4)

## create some (non-realistic) example data
lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(21, 21, 132))

## create example NetCDF
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create(file.path(tempdir(),"CMSAF_example_file.nc"), vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the multi-year seasonal standard deviation of the example
## CM SAF NetCDF file and write the output to a new file.
yseassd("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_yseassd.nc"))
unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
file.path(tempdir(),"CMSAF_example_file_yseassd.nc")))
Index

* annual statistics
  - year.anomaly, 86
  - yearmean, 88
  - yearsum, 89
* daily statistics
  - dayrange, 25
  - ydaymean, 85
* data manipulation functions
  - add_grid_info, 3
  - box_mergetime, 4
  - levbox_mergetime, 36
  - remap, 54
* datagen
  - cmsafops, 23
* manip
  - cmsafops, 23
* mathematical operators
  - cmsaf.add, 7
  - cmsaf.addc, 9
  - cmsaf.div, 13
  - cmsaf.divc, 14
  - cmsaf.mul, 16
  - cmsaf.mulc, 18
  - cmsaf.sub, 20
  - cmsaf.subc, 21
  - divdpm, 26
  - muldpm, 46
* monthly statistics
  - mon.anomaly, 38
  - monmax, 39
  - monmean, 41
  - monmin, 42
  - monsd, 44
  - monsum, 45
  - multimonmean, 48
  - multimonsum, 50
  - ymonmax, 91
  - ymonmean, 92
  - ymonmin, 94
  - ymonsd, 95
  - ymonsum, 96
* package
  - cmsafops, 23
* seasonal statistics
  - seas.anomaly, 56
  - seasmean, 57
  - seassum, 59
  - yseasmax, 98
  - yseasmean, 99
  - yseasmin, 101
  - yseassd, 102
* selection and removal functions
  - extract.level, 27
  - extract.period, 29
  - sellonlatbox, 60
  - selmon, 62
  - selperiod, 64
  - selpoint, 65
  - selpoint.multi, 67
  - seltime, 69
  - selyear, 71
* spatial operators
  - fldmax, 31
  - fldmean, 32
  - fldmin, 34
  - wfldmean, 84
* spatial
  - cmsafops, 23
* temporal operators
  - timmax, 73
  - timmean, 74
  - timmin, 76
  - timpctl, 77
  - timsd, 79
  - timsum, 80
  - trend, 82
* ts
  - cmsafops, 23
<table>
<thead>
<tr>
<th>Package</th>
<th>Index</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cm SafOps</code></td>
<td>23</td>
<td></td>
</tr>
<tr>
<td><code>add (cm Saf.add)</code></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td><code>add_grid_info</code></td>
<td>3, 5, 24, 37, 55</td>
<td></td>
</tr>
<tr>
<td><code>addc (cm Saf.addc)</code></td>
<td>9</td>
<td></td>
</tr>
<tr>
<td><code>box_merge_time</code></td>
<td>4, 4, 24, 37, 55</td>
<td></td>
</tr>
<tr>
<td><code>cat (cm Saf.cat)</code></td>
<td>11</td>
<td></td>
</tr>
<tr>
<td><code>change_att</code></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td><code>cm Saf package</code></td>
<td>23</td>
<td></td>
</tr>
<tr>
<td><code>cm Saf.add</code></td>
<td>7, 10, 13, 15, 17, 19, 20, 22, 23, 27, 47</td>
<td></td>
</tr>
<tr>
<td><code>cm Saf.addc</code></td>
<td>8, 9, 13, 15, 17, 19, 20, 22, 23, 27, 47</td>
<td></td>
</tr>
<tr>
<td><code>cm Saf.cat</code></td>
<td>11, 24</td>
<td></td>
</tr>
<tr>
<td><code>cm Saf.div</code></td>
<td>8, 10, 13, 15, 17, 19, 20, 22, 23, 27, 47</td>
<td></td>
</tr>
<tr>
<td><code>cm Saf.dvc</code></td>
<td>8, 10, 13, 14, 17, 19, 20, 22, 23, 27, 47</td>
<td></td>
</tr>
<tr>
<td><code>cm Saf.mul</code></td>
<td>8, 10, 13, 15, 16, 19, 20, 22, 23, 27, 47</td>
<td></td>
</tr>
<tr>
<td><code>cm Saf.mulc</code></td>
<td>8, 10, 13, 15, 17, 18, 20, 22, 23, 27, 47</td>
<td></td>
</tr>
<tr>
<td><code>cm Saf.sub</code></td>
<td>8, 10, 13, 15, 17, 19, 20, 22, 23, 27, 47</td>
<td></td>
</tr>
<tr>
<td><code>cm Saf.subc</code></td>
<td>8, 10, 13, 15, 17, 19, 20, 21, 23, 27, 47</td>
<td></td>
</tr>
<tr>
<td><code>cm SafOps</code></td>
<td>23</td>
<td></td>
</tr>
<tr>
<td><code>dayrange</code></td>
<td>24, 25, 86</td>
<td></td>
</tr>
<tr>
<td><code>div (cm Saf.div)</code></td>
<td>13</td>
<td></td>
</tr>
<tr>
<td><code>dvc (cm Saf.dvc)</code></td>
<td>14</td>
<td></td>
</tr>
<tr>
<td><code>divdp</code></td>
<td>8, 10, 13, 15, 17, 19, 20, 22, 23, 26, 47</td>
<td></td>
</tr>
<tr>
<td><code>extract.level</code></td>
<td>24, 27, 30, 61, 63, 64, 66, 68, 70, 72</td>
<td></td>
</tr>
<tr>
<td><code>extract.period</code></td>
<td>24, 28, 29, 61, 63, 64, 66, 68, 70, 72</td>
<td></td>
</tr>
<tr>
<td><code>fldmax</code></td>
<td>24, 29, 31, 33, 34, 84</td>
<td></td>
</tr>
<tr>
<td><code>fldmean</code></td>
<td>24, 31, 32, 34, 84</td>
<td></td>
</tr>
<tr>
<td><code>fldmin</code></td>
<td>24, 31, 33, 34, 84</td>
<td></td>
</tr>
<tr>
<td><code>get.knn</code></td>
<td>54</td>
<td></td>
</tr>
<tr>
<td><code>get_time</code></td>
<td>24, 35</td>
<td></td>
</tr>
<tr>
<td><code>interp.surface.grid</code></td>
<td>54</td>
<td></td>
</tr>
<tr>
<td><code>levbox_merge_time</code></td>
<td>4, 5, 24, 36, 55</td>
<td></td>
</tr>
</tbody>
</table>
timmin, 24, 74, 75, 76, 78, 79, 81, 83
 timpctl, 24, 74–76, 77, 79, 81, 83
 timsd, 24, 74–76, 78, 79, 81, 83
 timsum, 24, 74–76, 78, 79, 80, 83
trend, 24, 74–76, 78, 79, 81, 82

wfldmean, 24, 31, 33, 34, 84

ydaymean, 24, 25, 85
year.anomaly, 24, 86, 88, 90
yearmean, 24, 87, 88, 90
yearsum, 24, 87, 88, 89
ymonmax, 24, 38, 40, 41, 43, 44, 46, 49, 50, 91, 93, 94, 96, 97
ymonmean, 24, 38, 40, 41, 43, 44, 46, 49, 50, 91, 92, 94, 96, 97
ymonmin, 24, 38, 40, 41, 43, 44, 46, 49, 50, 91, 93, 94, 96, 97
ymonsd, 24, 38, 40, 41, 43, 44, 46, 49, 50, 91, 93, 94, 95, 97
ymonsum, 24, 38, 40, 41, 43, 44, 46, 49, 50, 91, 93, 94, 96, 96

yseasmax, 24, 57–59, 98, 100, 101, 103
yseasmean, 24, 57–59, 98, 99, 101, 103
yseasmin, 24, 57–59, 98, 100, 101, 103
yseassd, 24, 57–59, 98, 100, 101, 102