Package 'cmsafops'

June 28, 2022

Title Tools for CM SAF NetCDF Data

Version 1.2.5

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 (<https://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/safira>).
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), progress, trend, SearchTrees, utils

NeedsCompilation no

Repository CRAN

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

RoxygenNote 7.1.2

Encoding UTF-8

Language en-US

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

Author Steffen Kothe [aut, cre], Danny Parsons [ctb]

Date/Publication 2022-06-28 13:00:05 UTC

R topics documented:

`acsaf_box_mergetime` *Function to combine ACSAF NetCDF files and simultaneously cut a region.*

Description

This function selects a region (and optionally a level) from a bunch of AC SAF NetCDF files that match the same pattern of the filename, and writes the output to a new file.

Usage

```
acsaf_box_mergetime(
  path,
  pattern,
  outfile,
  lon1 = -180,
  lon2 = 180,
  lat1 = -90,
  lat2 = 90,
  nc34 = 3
)
```

Arguments

`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.

Value

A NetCDF file including the merged time series of the selected region is written.

See Also

Other data manipulation functions: add_grid_info(), box_mergetime(), cmsaf.transform.coordinate.system(), levbox_mergetime(), remap()

---

add_grid_info *Add grid info*

Description

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

Usage

```
add_grid_info(infile, auxfile, outfile, overwrite = FALSE, verbose = FALSE)
```

Arguments

infile	Character containing file name or path of input file.
auxfile	Character containing file name or path of auxiliary file.
outfile	Character containing file name or path of output file. If NULL, the input file is directly edited instead.
overwrite	Logical; should existing output file be overwritten? If outfile is NULL, this parameter is ignored.
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).

See Also

Other data manipulation functions: acsaf_box_mergetime(), box_mergetime(), cmsaf.transform.coordinate.system levbox_mergetime(), remap()

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 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

var	Name of NetCDF variable (character).
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).
level	Number of level that should be extracted (integer) or NULL.
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: acsaf_box_mergetime(), add_grid_info(), cmsaf.transform.coordinate.system levbox_mergetime(), remap()

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 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")
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_n2.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)

## Cut a region and merge both example CM SAF NetCDF files into one
## output file.  Get path information of working directory with getwd()
## command.
box_mergetime(var = "SIS", path= tempdir(), pattern = "CMSAF_example_file_n",
```

```
 outfile = file.path(tempdir(),"CMSAF_example_file_box_mergetime.nc"),
 lon1 = 8, lon2 = 12, lat1 = 48, lat2 = 52)

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")))
```

`calc_allDatesNc` *Designed for the CM SAF R Toolbox.*

Description

This function is a helper function called by the CM SAF R Toolbox.

Usage

```
calc_allDatesNc(result.fileslist, ordpath)
```

Arguments

`result.fileslist`	A data frame containing all meta data (data.frame).
`ordpath`	NetCDF file path

`calc_overlapping_time` *Routine to calculate overlapping time periods in two files.*

Description

Designed for CMSAF Toolbox.

Usage

```
calc_overlapping_time(
  var1,
  infile1,
  var2 = NULL,
  infile2,
  nc1 = NULL,
  nc2 = NULL
)
```

Arguments

var1	Name of NetCDF variable of the first data set (character).
infile1	Filename of first input NetCDF file. This may include the directory (character).
var2	Name of NetCDF variable of the second data set (character).
infile2	Filename of second input NetCDF file. This may include the directory (character). Also supported formats for station data are .csv and .RData files.
nc1	Alternatively to infile1 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).
nc2	Alternatively to infile2 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

Start date and end date are the result (list).

---

calc_timestepNc *Designed for the CM SAF R Toolbox.*

Description

This function is a helper function called by the CM SAF R Toolbox.

Usage

```
calc_timestepNc(result.fileslist, ordpath)
```

Arguments

result.fileslist	A data frame containing all meta data (data.frame).
ordpath	NetCDF file path

---

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

```
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

```
## 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(var = "Data1", infile = 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"))
```

`check.coordinate.system`

Designed for the CM SAF R Toolbox.

Description

This function is a helper function called by the CM SAF R Toolbox.

Usage

```
check.coordinate.system(nc_path, nc_temp_path, var, filelist)
```

Arguments

`nc_path`	Path to NetCDF files which should be converted
`nc_temp_path`	Destination NetCDF file path
`var`	Name of NetCDF variable (character)
`filelist`	NetCDF file names (data.frame)

`cmsaf.abs` *Determine absolute values*

Description

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

Usage

```
cmsaf.abs(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other mathematical operators: `cmsaf.addc()`, `cmsaf.add()`, `cmsaf.divc()`, `cmsaf.div()`, `cmsaf.mulc()`, `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")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the absolute values of the example CM SAF NetCDF file and write
## the output to a new file.
cmsaf.abs(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_abs.nc"))

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

`cmsaf.add` *Add the fields of two input NetCDF files.*

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(
  var1,
  var2,
  infile1,
  infile2,
  outfile,
```

```
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc1 = NULL,
  nc2 = NULL
)
```

Arguments

var1	Name of variable in infile1 (character).
var2	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
nc1	Alternatively to infile1 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).
nc2	Alternatively to infile2 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other mathematical operators: cmsaf.abs(), cmsaf.addc(), cmsaf.divc(), cmsaf.div(), cmsaf.mulc(), 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 <- 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)

## Add the fields of both example CM SAF NetCDF files and write the
## result into one output file.
cmsaf.add(var1 = "SIS", var2 = "SIS", infile1 = file.path(tempdir(),
 "CMSAF_example_file_1.nc"), infile2 = file.path(tempdir(),
 "CMSAF_example_file_2.nc"), outfile = 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,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other mathematical operators: cmsaf.abs(), cmsaf.add(), cmsaf.divc(), cmsaf.div(), cmsaf.mulc(), 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")
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(var = "SIS", const = 10, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = 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")))
```

`cmsaf.adjust.two.files` *Routine to adjust the time dimensions and coordinates in two files.*

Description

Designed for CM SAF R Toolbox.

Usage

```
cmsaf.adjust.two.files(
  var1,
  infile1,
  var2,
  infile2,
  outfile1,
  outfile2,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc1 = NULL,
  nc2 = NULL
)
```

Arguments

`var1`	Name of NetCDF variable of the first data set (character).
`infile1`	Filename of first input NetCDF file. This may include the directory (character).

var2	Name of NetCDF variable of the second data set (character).
infile2	Filename of second input NetCDF file. This may include the directory (character).
outfile1	Filename of first output NetCDF file. This may include the directory (character).
outfile2	Filename of second 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
nc1	Alternatively to infile1 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).
nc2	Alternatively to infile2 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

Two NetCDF files with the same time period and coordinate system are the result.

---

cmsaf.cat *Concatenate datasets of several NetCDF input files.*

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

```
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/>

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)
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)

## 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(var = "SIS", infiles = c(file.path(tempdir(),
 "CMSAF_example_file_1.nc"), file.path(tempdir(),"CMSAF_example_file_2.nc")),
 outfile = 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")))
```

`cmsaf.detrend` *Linear detrending of time series*

Description

The function determines detrended values from data of a single NetCDF input file. All time steps should be equidistantly distributed.

Usage

```
cmsaf.detrend(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other temporal operators: `cmsaf.mk.test()`, `cmsaf.regres()`, `num_above()`, `num_below()`, `num_equal()`, `timavg()`, `timmax()`, `timmean()`, `timmin()`, `timpctl()`, `timsd()`, `timsum()`, `trend_advanced()`, `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 detrend values of the example CM SAF NetCDF file and write
## the output to a new file.
cmsaf.detrend(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_detrend.nc"))

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

`cmsaf.div` *Divide the fields of two input NetCDF files.*

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

```
cmsaf.div(
  var1,
  var2,
  infile1,
  infile2,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc1 = NULL,
  nc2 = NULL
)
```

Arguments

var1	Name of variable in infile1 (character).
var2	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
nc1	Alternatively to infile1 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).
nc2	Alternatively to infile2 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other mathematical operators: cmsaf.abs(), cmsaf.addc(), cmsaf.add(), cmsaf.divc(), cmsaf.mulc(), 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(10, 15, 0.5)
lat <- seq(50, 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(11, 11, 1))
data2 <- array(230:320, dim = c(11, 11, 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(var1 = "SIS", var2 = "SIS", infile1 = file.path(tempdir(),
 "CMSAF_example_file_1.nc"), infile2 = file.path(tempdir(),
 "CMSAF_example_file_2.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_div.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")))
```

`cmsaf.divc` *Divide data by a constant.*

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,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other mathematical operators: cmsaf.abs(), cmsaf.addc(), cmsaf.add(), cmsaf.div(), cmsaf.mulc(), 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")
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(var = "SIS", const = 100, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = 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.mk.test` *Apply Mann-Kendall trend test.*

Description

The function determines the trend from data of a single CM SAF NetCDF input file basing on a Mann-Kendall test.

Usage

```
cmsaf.mk.test(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including three data layers is written. One layer contains a measure for the significance of the calculated mann-kendall statistic (S). A very high positive value of S is an indicator of an increasing trend and a very low negative value indicates a decreasing trend. Another layer (Z) contains the calculated normalized test statsitic Z. A positive value of Z is an indicator of an increasing trend and a negative value indicates a decreasing trend.

See Also

Other temporal operators: cmsaf.detrend(), cmsaf.regres(), num_above(), num_below(), num_equal(), timavg(), timmax(), timmean(), timmin(), timpctl(), timsd(), timsum(), trend_advanced(), 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(10, 15, 0.5)
lat <- seq(50, 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(11, 11, 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.
cmsaf.mk.test(var = "SIS", infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_mktrend.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
 file.path(tempdir(),"CMSAF_example_file_mktrend.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(
  var1,
  var2,
  infile1,
  infile2,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc1 = NULL,
  nc2 = NULL
)
```

Arguments

`var1`	Name of variable in infile1 (character).
`var2`	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
nc1	Alternatively to infile1 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).
nc2	Alternatively to infile2 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other mathematical operators: cmsaf.abs(), 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 <- 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)

## Multiply the fields of both example CM SAF NetCDF files and write the
## result into one output file.
cmsaf.mul(var1 = "SIS", var2 = "SIS", infile1 = file.path(tempdir(),
 "CMSAF_example_file_1.nc"), infile2 = file.path(tempdir(),
 "CMSAF_example_file_2.nc"), outfile = 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.mulc` *Multiply data with a constant.*

Description

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

Usage

```
cmsaf.mulc(
  var,
  const = 1,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other mathematical operators: cmsaf.abs(), 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")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(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(var = "SIS", const = 10, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_mulc.nc"))
```

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

`cmsaf.regres` *Estimate regression parameter*

Description

The function estimates the regression parameters b from data of a single NetCDF input file.

Usage

```
cmsaf.regres(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

A NetCDF file including the regression parameters b is written.

See Also

Other temporal operators: `cmsaf.detrend()`, `cmsaf.mk.test()`, `num_above()`, `num_below()`, `num_equal()`, `timavg()`, `timmax()`, `timmean()`, `timmin()`, `timpctl()`, `timsd()`, `timsum()`, `trend_advanced()`, `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, 6, 0.5)
lat <- seq(45, 47, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("2002-12-31"), "month")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- aperm(array(c(1:369), dim = c(3, 5, 36)), c(2, 1, 3))

## 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)

## Estimate the regression parameter b values of the example CM SAF NetCDF file and write
## the output to a new file.
cmsaf.regres(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_regres.nc"))

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

`cmsaf.stats` *Calculates the rmse, mae, bias, correlation in grid space of two NetCDF files. Designed for the CM SAF R Toolbox.*

Description

Calculates the rmse, mae, bias, correlation in grid space of two NetCDF files. Designed for the CM SAF R Toolbox.

Usage

```
cmsaf.stats(
  var1,
  var2,
  infile1,
  infile2,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc1 = NULL,
  nc2 = NULL
)
```

Arguments

var1	Name of NetCDF variable of the first file (character).
var2	Name of NetCDF variable of the second file (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 csv file. This may include the directory (character).
nc34	NetCDF version of input file. Default output is NetCDFv4.
overwrite	logical; should existing output file be overwritten?; Default: FALSE
verbose	logical; if TRUE, progress messages are shown
nc1	Alternatively to infile1 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).
nc2	Alternatively to infile2 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A csv file including the rmse, mae, bias and correlation in grid space is written.

See Also

Other metrics: cmsaf.stats.station.data()

`cmsaf.stats.station.data` *Calculates the rmse, mae, bias, correlation over time of a NetCDF file and a dataframe (station data). Designed for the CM SAF R Toolbox.*

Description

Calculates the rmse, mae, bias, correlation over time of a NetCDF file and a dataframe (station data). Designed for the CM SAF R Toolbox.

Usage

```
cmsaf.stats.station.data(
  var,
  infile,
  data_station,
  outfile,
  overwrite = FALSE,
  nc = NULL
)
```

Arguments

`var`	Name of NetCDF variable of NetCDF file (character).
`infile`	Filename of input NetCDF file. This may include the directory (character).
`data_station`	Dataframe of RData or csv file (station data); Designed for the CM SAF R Toolbox.
`outfile`	Filename of output csv file. This may include the directory (character).
`overwrite`	logical; should existing output file be overwritten?; Default: FALSE
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

A csv file including the rmse, mae, bias and correlation over time is written.

See Also

Other metrics: `cmsaf.stats()`

`cmsaf.sub` *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

```
cmsaf.sub(
  var1,
  var2,
  infile1,
  infile2,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc1 = NULL,
  nc2 = NULL
)
```

Arguments

`var1`	Name of variable in infile1 (character).
`var2`	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
`nc1`	Alternatively to `infile1` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).
`nc2`	Alternatively to `infile2` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other mathematical operators: cmsaf.abs(), 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(var1 = "SIS", var2 = "SIS", infile1 = file.path(tempdir(),
 "CMSAF_example_file_1.nc"), infile2 = file.path(tempdir(),
 "CMSAF_example_file_2.nc"), outfile = 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.sub.rel` *Subtract the fields of two input NetCDF files (relative). Designed for the CM SAF R Toolbox.*

Description

The function subtracts the fields of infile2 from the fields of infile1. Infiles have to have the same spatial and temporal dimension.

Usage

```
cmsaf.sub.rel(
  var1,
  infile1,
  var2,
  infile2,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc1 = NULL,
  nc2 = NULL
)
```

Arguments

`var1`	Name of variable in infile1 (character).
`infile1`	Filename of first input NetCDF file. This may include the directory (character).
`var2`	Name of variable in infile2 (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
`nc1`	Alternatively to `infile1` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).
`nc2`	Alternatively to `infile2` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

`cmsaf.subc` *Subtract a constant from a dataset.*

Description

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

Usage

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

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other mathematical operators: `cmsaf.abs()`, `cmsaf.addc()`, `cmsaf.add()`, `cmsaf.divc()`, `cmsaf.div()`, `cmsaf.mulc()`, `cmsaf.mul()`, `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")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(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(var = "SIS", const = 10, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = 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")))
```

---

cmsaf.transform.coordinate.system

Transform the coordinate system to -180 to 180 longitude of an infile

---

Description

Transform the coordinate system to -180 to 180 longitude of an infile

Usage

```
cmsaf.transform.coordinate.system(infile, var, outfile, nc = NULL)
```

Arguments

infile	Filename of input NetCDF file. This may include the directory (character).
var	Name of NetCDF variable (character).
outfile	Filename of output NetCDF file. This may include the directory (character).
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including the coordinate system (-180 to 180 longitude) is written.

See Also

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

---

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.abs, cmsaf.add, cmsaf.addc, cmsaf.div, cmsaf.divc, cmsaf.mul, cmsaf.mulc, cmsaf.sub, cmsaf.subc, divdpm, muldpm

Hourly statistics

hourmean, hoursum

Daily statistics

dayavg, daymax, daymean, daymin, daypctl, dayrange, daysd, daysum, dayvar, ydaymax, ydaymean, ydaymin, ydayrange, ydaysd, ydaysum

Monthly statistics

mon.anomaly, monavg, mondaymean, monmax, monmean, monmin, monpctl, monsd, monsum, monvar, multimonmean, multimonsum, ymonmax, ymonmean, ymonmin, ymonsd, ymonsum, mon_num_above, mon_num_below, mon_num_equal

Seasonal statistics

seas.anomaly, seasmean, seassd, seassum, seasvar, yseasmax, yseasmean, yseasmin, yseassd

Annual statistics

year.anomaly, yearmax, yearmean, yearmin, yearrange, yearsd, yearsum, yearvar

Zonal statistics

zonmean, zonsum

Meridional statistics

mermean

Running statistics

runmax, runmean, runmin, runrange, runsd, runsum, ydrunmean, ydrunsd, ydrunsum

Grid boxes statistics

gridboxmax, gridboxmean, gridboxmin, gridboxrange, gridboxsd, gridboxsum, gridboxvar

Temporal operators

cmsaf.detrend, cmsaf.mk.test, cmsaf.regres, timmax, timmean, timavg, timmin, timpctl, timsd, timsum, trend_advanced, trend, num_above, num_below, num_equal

Time range statistics

timselmean, timselsum

Spatial operators

fldmax, fldmean, fldmin, fldrange, fldsd, fldsum, wfldmean

Correlation and covariance

fldcor, fldcovar, timcor, timcovar

Selection and removal functions

extract.level, extract.period, sellonlatbox, selmon, selperiod, selpoint.multi, selpoint, seltime, selyear

Data manipulation

box_mergetime, cmsaf.adjust.two.files, cmsaf.transform.coordinate.system, 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 Toolbox—A Tool for the Easy Usage of Satellite-Based Climate Data in NetCDF Format. ISPRS Int. J. Geo-Inf. 2019, 8, 109. doi: 10.3390/ijgi8030109

---

dayavg *Determine daily averages*

Description

The function determines daily averages from data of a single CM SAF NetCDF input file. There is a difference between the operators dayavg and daymean. The mean is regarded as a statistical function, whereas the average is found simply by adding the sample members and dividing the result by the sample size. For example, the mean of 1, 2, miss and 3 is (1 + 2 + 3)/3 = 2, whereas the average is (1 + 2 + miss + 3)/4 = miss/4 = miss. If there are no missing values in the sample, the average and mean are identical.

Usage

```
dayavg(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of daily averages is written.

See Also

Other daily statistics: daymax(), daymean(), daymin(), daypctl(), dayrange(), daysd(), daysum(), dayvar(), ydaymax(), ydaymean(), ydaymin(), ydayrange(), ydaysd(), ydaysum()

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 <- 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 daily averages of the example CM SAF NetCDF file and
## write the output to a new file.
dayavg(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_dayavg.nc"))

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

---

daymax *Determine daily maxima*

Description

The function determines daily maximum from data of a single CM SAF NetCDF input file.

Usage

```
daymax(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other daily statistics: dayavg(), daymean(), daymin(), daypctl(), dayrange(), daysd(), daysum(), dayvar(), ydaymax(), ydaymean(), ydaymin(), ydayrange(), ydaysd(), ydaysum()

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 <- 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 daily maximum of the example CM SAF NetCDF file and
## write the output to a new file.
daymax(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_daymax.nc"))

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

---

daymean *Determine daily means*

Description

The function determines daily means from data of a single CM SAF NetCDF input file. There is a difference between the operators daymean and dayavg. The mean is regarded as a statistical

function, whereas the average is found simply by adding the sample members and dividing the result by the sample size. For example, the mean of 1, 2, miss and 3 is (1 + 2 + 3)/3 = 2, whereas the average is (1 + 2 + miss + 3)/4 = miss/4 = miss. If there are no missing values in the sample, the average and mean are identical.

Usage

```
daymean(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other daily statistics: dayavg(), daymax(), daymin(), daypctl(), dayrange(), daysd(), daysum(), dayvar(), ydaymax(), ydaymean(), ydaymin(), ydayrange(), ydaysd(), ydaysum()

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 <- 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 daily means of the example CM SAF NetCDF file and
## write the output to a new file.
daymean(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_daymean.nc"))

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

---

daymin *Determine daily minima*

Description

The function determines the daily minimum from data of a single CM SAF NetCDF input file.

Usage

```
daymin(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other daily statistics: dayavg(), daymax(), daymean(), daypctl(), dayrange(), daysd(), daysum(), dayvar(), ydaymax(), ydaymean(), ydaymin(), ydayrange(), ydaysd(), ydaysum()

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 <- 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 daily minimum of the example CM SAF NetCDF file and
## write the output to a new file.
daymin(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_daymin.nc"))

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

---

daypctl *Determine daily percentiles*

Description

The function determines daily percentiles from data of a single CM SAF NetCDF input file.

Usage

```
daypctl(
  var,
  p = 0.95,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
p	Percentile number given as probability within [0, 1] (numeric). Default is 0.95.
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of daily percentiles is written.

See Also

Other daily statistics: dayavg(), daymax(), daymean(), daymin(), dayrange(), daysd(), daysum(), dayvar(), ydaymax(), ydaymean(), ydaymin(), ydayrange(), ydaysd(), ydaysum()

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 <- 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% daily percentiles of the example CM SAF NetCDF file and
## write the output to a new file.
daypctl(var = "SIS", p = 0.9, infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_daypctl.nc"))

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

---

dayrange *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,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other daily statistics: dayavg(), daymax(), daymean(), daymin(), daypctl(), daysd(), daysum(), dayvar(), ydaymax(), ydaymean(), ydaymin(), ydayrange(), ydaysd(), ydaysum()

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 <- 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.
dayrange(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_dayrange.nc"))

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

---

daysd *Determine daily standard deviations*

Description

The function determines daily standard deviations from data of a single CM SAF NetCDF input file.

Usage

```
daysd(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other daily statistics: dayavg(), daymax(), daymean(), daymin(), daypctl(), dayrange(), daysum(), dayvar(), ydaymax(), ydaymean(), ydaymin(), ydayrange(), ydaysd(), ydaysum()

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 <- 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 daily standard deviations of the example CM SAF NetCDF file and
## write the output to a new file.
daysd(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_daysd.nc"))
```

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

`daysum` *Determine daily sums*

Description

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

Usage

```
daysum(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other daily statistics: `dayavg()`, `daymax()`, `daymean()`, `daymin()`, `daypctl()`, `dayrange()`, `daysd()`, `dayvar()`, `ydaymax()`, `ydaymean()`, `ydaymin()`, `ydayrange()`, `ydaysd()`, `ydaysum()`

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 <- 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 daily sums of the example CM SAF NetCDF file and
## write the output to a new file.
daysum(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_daysum.nc"))

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

---

dayvar *Determine daily variances*

Description

The function determines daily variances from data of a single CM SAF NetCDF input file.

Usage

```
dayvar(
  var,
```

```
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of daily variances is written.

See Also

Other daily statistics: dayavg(), daymax(), daymean(), daymin(), daypctl(), dayrange(), daysd(), daysum(), ydaymax(), ydaymean(), ydaymin(), ydayrange(), ydaysd(), ydaysum()

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 <- 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 daily variances of the example CM SAF NetCDF file and
## write the output to a new file.
dayvar(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_dayvar.nc"))

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

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

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other mathematical operators: cmsaf.abs(), 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/>

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(var = "SIS", infile= file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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

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

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

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

```
## 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"))
data <- array(250:350, dim = c(21, 21, 11, 132))

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
z <- ncdim_def(name = "height", units = "m", vals = height)
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, z, 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")
ncatt_put(ncnew, "height", "standard_name", "height", prec = "text")
nc_close(ncnew)

## Extract the first level of the example CM SAF NetCDF file and write
## the output to a new file.
extract.level("SIS", file.path(tempdir(),"CMSAF_example_file.nc"),
 file.path(tempdir(),"CMSAF_example_file_extract.level1.nc"))
## Extract all levels of the example CM SAF NetCDF file and write the
## output to a new file.
extract.level(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_extract.level2.nc"),
 level = "all")

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
 file.path(tempdir(),"CMSAF_example_file_extract.level1.nc"),
 file.path(tempdir(),"CMSAF_example_file_extract.level2_level[1-9].nc"),
 file.path(tempdir(),"CMSAF_example_file_extract.level2_level10.nc"),
 file.path(tempdir(),"CMSAF_example_file_extract.level2_level11.nc")))
```

`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,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

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(var = "SIS", start = "2001-01-01", end = "2002-01-01",
 infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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")))
```

`fldcor` *Determine correlations in grid space.*

Description

The function determines correlations in grid space from data of two CM SAF NetCDF input files. This function is applicable to 3-dimensional NetCDF data.

Usage

```
fldcor(
  var1,
  infile1,
  var2,
  infile2,
  outfile,
```

```
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc1 = NULL,
  nc2 = NULL
)
```

Arguments

var1	Name of NetCDF variable of the first data set (character).
infile1	Filename of first input NetCDF file. This may include the directory (character).
var2	Name of NetCDF variable of the second data set (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
nc1	Alternatively to infile1 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).
nc2	Alternatively to infile2 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of correlations in grid space is written.

See Also

Other correlation and covariance: fldcovar(), timcor(), timcovar()

Examples

```
## Create two example NetCDF files with a similar structure as used by CM
## SAF. The files are 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 <- as.Date("2000-05-31")
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 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), -999, prec = "float")
vars <- list(var1)
ncnew_1 <- nc_create(file.path(tempdir(), "CMSAF_example_file_1.nc"), vars)
ncnew_2 <- nc_create(file.path(tempdir(), "CMSAF_example_file_2.nc"), vars)

ncvar_put(ncnew_1, var1, data1)
ncvar_put(ncnew_2, var1, data2)

ncatt_put(ncnew_1, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_1, "lat", "standard_name", "latitude", prec = "text")

ncatt_put(ncnew_2, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_2, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew_1)
nc_close(ncnew_2)

## Determine the correlations in grid space of the example CM SAF NetCDF files and
## write the output to a new file.
fldcor(var1 = "SIS", infile1 = file.path(tempdir(),"CMSAF_example_file_1.nc"),
      var2 = "SIS", infile2 = file.path(tempdir(), "CMSAF_example_file_2.nc"),
      outfile = file.path(tempdir(),"CMSAF_example_file_fldcor.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_fldcor.nc")))
```

`fldcovar` *Determine covariances in grid space.*

Description

The function determines covariances in grid space from data of two CM SAF NetCDF input files. This function is applicable to 3-dimensional NetCDF data.

Usage

```
fldcovar(
  var1,
  infile1,
  var2,
  infile2,
  outfile,
  nc34 = 4,
```

```
  overwrite = FALSE,
  verbose = FALSE,
  nc1 = NULL,
  nc2 = NULL
)
```

Arguments

var1	Name of NetCDF variable of the first data set (character).
infile1	Filename of first input NetCDF file. This may include the directory (character).
var2	Name of NetCDF variable of the second data set (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
nc1	Alternatively to infile1 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).
nc2	Alternatively to infile2 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of covariances in grid space is written.

See Also

Other correlation and covariance: fldcor(), timcor(), timcovar()

Examples

```
## Create two example NetCDF files with a similar structure as used by CM
## SAF. The files are 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 <- as.Date("2000-05-31")
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 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), -999, prec = "float")
vars <- list(var1)
ncnew_1 <- nc_create(file.path(tempdir(), "CMSAF_example_file_1.nc"), vars)
ncnew_2 <- nc_create(file.path(tempdir(), "CMSAF_example_file_2.nc"), vars)

ncvar_put(ncnew_1, var1, data1)
ncvar_put(ncnew_2, var1, data2)

ncatt_put(ncnew_1, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_1, "lat", "standard_name", "latitude", prec = "text")

ncatt_put(ncnew_2, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_2, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew_1)
nc_close(ncnew_2)

## Determine the covariances in grid space of the example CM SAF NetCDF files and
## write the output to a new file.
fldcovar(var1 = "SIS", infile1 = file.path(tempdir(),"CMSAF_example_file_1.nc"),
      var2 = "SIS", infile2 = file.path(tempdir(), "CMSAF_example_file_2.nc"),
      outfile = file.path(tempdir(),"CMSAF_example_file_fldcovar.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_fldcovar.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

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

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other spatial operators: fldmean(), fldmin(), fldrange(), fldsd(), fldsum(), 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/>

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 maximum values of the example CM SAF NetCDF file and
## write the output to a new file.
fldmax(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_fldmax.nc"))

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

```
fldmean(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other spatial operators: fldmax(), fldmin(), fldrange(), fldsd(), fldsum(), 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/>

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.
fldmean(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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

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

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other spatial operators: fldmax(), fldmean(), fldrange(), fldsd(), fldsum(), 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/>

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 minimum values of the example CM SAF NetCDF file and
## write the output to a new file.
fldmin(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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")))
```

`fldrange` *Determine the spatial range*

Description

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

Usage

```
fldrange(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other spatial operators: fldmax(), fldmean(), fldmin(), fldsd(), fldsum(), 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/>

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 range of the example CM SAF NetCDF file and
## write the output to a new file.
fldrange(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_fldrange.nc"))
```

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

`fldsd` *Determine the spatial standard deviation*

Description

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

Usage

```
fldsd(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other spatial operators: `fldmax()`, `fldmean()`, `fldmin()`, `fldrange()`, `fldsum()`, `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/>

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 standard deviation of the example CM SAF NetCDF file and
## write the output to a new file.
fldsd(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_fldsd.nc"))

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

---

fldsum *Determine the spatial sum*

Description

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

Usage

```
fldsum(
```

```
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other spatial operators: fldmax(), fldmean(), fldmin(), fldrange(), fldsd(), 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/>

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 sum of the example CM SAF NetCDF file and
## write the output to a new file.
fldsum(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_fldsum.nc"))

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

`get_basename` *Determine the basename of a NetCDF file*

Description

This function determines the basename of either a file/URL path or an 'nc' object (using nc$filename).

Usage

```
get_basename(infile, nc = NULL)
```

Arguments

`infile`	Filename of input NetCDF file. This may include the directory (character).
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Details

When the origin of the file path is a local .nc file then `get_basename()` is equivalent to `base::basename()`.

`get_basename()` also handles the case of `infile`/nc originating from a URL.

The value of `get_basename()` always ends in ".nc".

If both `infile` and `nc` are specified, `infile` is ignored.

Value

A character string giving the basename.

get_date_time *Get dates and times from NetCDF type date format.*

Description

Get dates and times from NetCDF type date format.

Usage

```
get_date_time(times, unit)
```

Arguments

times	Timesteps from netcdf data (numeric).
unit	Unit from netcdf data (character).

Value

A data frame with the columns years, months, days and times. Careful: The parts of the date are of numeric type, but the times are stored as characters (levels).

Examples

```
date_time <- get_date_time(times = c(159191, 5991820),
 unit = "minutes since 1980-05-07")
date_time
date_time$years
```

get_dimensions *Designed for the CM SAF R Toolbox.*

Description

This function is a helper function called by the CM SAF R Toolbox.

Usage

```
get_dimensions(id, dimnames)
```

Arguments

id	An object of the class NetCDF4
dimnames	Dimension names (data.frame)

get_nc_version *Designed for the CM SAF R Toolbox.*

Description

This function checks the nc version.

Usage

```
get_nc_version(nc34)
```

Arguments

nc34	(numeric)

get_processing_time_string *Get processing time string*

Description

Get processing time string

Usage

```
get_processing_time_string(time_start, time_end)
```

Arguments

time_start	start time of the process (of class "POSIXct" as given by "Sys.time()")
time_end	end time of the process (of class "POSIXct" as given by "Sys.time()")

Value

a specialized string containing the processed time

get_time — *Convert time steps to POSIXct.*

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!

Usage

```
get_time(time.unit, time.step)
```

Arguments

time.unit	Time unit, which is conform to the CF convention (character).
time.step	Time steps in form of a numeric or integer vector.

Value

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

Examples

```
get_time(time.unit = "hours since 1987-01-01", time.step = 249109)
get_time(time.unit = "days since 1987-01-01", time.step = 9109)
```

get_time_info — *Designed for the CM SAF R Toolbox.*

Description

This function is a helper function called by the CM SAF R Toolbox. Not for general use.

Usage

```
get_time_info(id, dimnames, t_name)
```

Arguments

id	id
dimnames	dimnames
t_name	t_name

gridboxmax *Determine maxima of selected grid boxes*

Description

The function determines maxima of selected grid boxes from data of a single CM SAF NetCDF input file.

Usage

```
gridboxmax(
  var,
  lonGrid,
  latGrid,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
lonGrid	INTEGER Number of grid boxes in x direction
latGrid	INTEGER Number of grid boxes in y direction
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of maxima of selected grid boxes is written.

See Also

Other grid boxes statistics: gridboxmean(), gridboxmin(), gridboxrange(), gridboxsd(), gridboxsum(), gridboxvar()

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("2000-03-01"), as.Date("2000-05-31"), "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, 3))

## 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_1 <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)

ncvar_put(ncnew_1, var1, data)

ncatt_put(ncnew_1, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_1, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew_1)

## Determine the maxima of selected grid boxes of the example CM SAF NetCDF file
## and write the output to a new file.
gridboxmax(var = "SIS", lonGrid = 4, latGrid = 4, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_gridboxmax.nc"))

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

---

gridboxmean *Determine means of selected grid boxes*

Description

The function determines means of selected grid boxes from data of a single CM SAF NetCDF input file.

Usage

```
gridboxmean(
  var,
  lonGrid,
  latGrid,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
lonGrid	INTEGER Number of grid boxes in x direction
latGrid	INTEGER Number of grid boxes in y direction
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of means of selected grid boxes is written.

See Also

Other grid boxes statistics: gridboxmax(), gridboxmin(), gridboxrange(), gridboxsd(), gridboxsum(), gridboxvar()

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("2000-03-01"), as.Date("2000-05-31"), "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, 3))

## 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_1 <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)

ncvar_put(ncnew_1, var1, data)

ncatt_put(ncnew_1, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_1, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew_1)

## Determine the means of selected grid boxes of the example CM SAF NetCDF file
## and write the output to a new file.
gridboxmean(var = "SIS", lonGrid = 4, latGrid = 4, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_gridboxmean.nc"))

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

`gridboxmin` *Determine minima of selected grid boxes*

Description

The function determines minima of selected grid boxes from data of a single CM SAF NetCDF input file.

Usage

```
gridboxmin(
  var,
  lonGrid,
  latGrid,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
lonGrid	INTEGER Number of grid boxes in x direction
latGrid	INTEGER Number of grid boxes in y direction
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of minima of selected grid boxes is written.

See Also

Other grid boxes statistics: gridboxmax(), gridboxmean(), gridboxrange(), gridboxsd(), gridboxsum(), gridboxvar()

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("2000-03-01"), as.Date("2000-05-31"), "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, 3))

## 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_1 <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)

ncvar_put(ncnew_1, var1, data)
```

```
ncatt_put(ncnew_1, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_1, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew_1)

## Determine the minima of selected grid boxes of the example CM SAF NetCDF file
## and write the output to a new file.
gridboxmin(var = "SIS", lonGrid = 4, latGrid = 4, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_gridboxmin.nc"))

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

`gridboxrange` *Determine ranges of selected grid boxes*

Description

The function determines ranges of selected grid boxes from data of a single CM SAF NetCDF input file.

Usage

```
gridboxrange(
  var,
  lonGrid,
  latGrid,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

`var`	Name of NetCDF variable (character).
`lonGrid`	INTEGER Number of grid boxes in x direction
`latGrid`	INTEGER Number of grid boxes in y direction
`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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of ranges of selected grid boxes is written.

See Also

Other grid boxes statistics: gridboxmax(), gridboxmean(), gridboxmin(), gridboxsd(), gridboxsum(), gridboxvar()

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("2000-03-01"), as.Date("2000-05-31"), "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, 3))

## 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_1 <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)

ncvar_put(ncnew_1, var1, data)

ncatt_put(ncnew_1, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_1, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew_1)

## Determine the ranges of selected grid boxes of the example CM SAF NetCDF file and write
## the output to a new file.
gridboxrange(var = "SIS", lonGrid = 4, latGrid = 4, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_gridboxrange.nc"))

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

gridboxsd *Determine standard deviations of selected grid boxes*

Description

The function determines standard deviations of selected grid boxes from data of a single CM SAF NetCDF input file.

Usage

```
gridboxsd(
  var,
  lonGrid,
  latGrid,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
lonGrid	INTEGER Number of grid boxes in x direction
latGrid	INTEGER Number of grid boxes in y direction
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of standard deviations of selected grid boxes is written.

See Also

Other grid boxes statistics: gridboxmax(), gridboxmean(), gridboxmin(), gridboxrange(), gridboxsum(), gridboxvar()

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("2000-03-01"), as.Date("2000-05-31"), "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, 3))

## 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_1 <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)

ncvar_put(ncnew_1, var1, data)

ncatt_put(ncnew_1, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_1, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew_1)

## Determine the standard deviations of selected grid boxes of the example CM SAF NetCDF file
## and write the output to a new file.
gridboxsd(var = "SIS", lonGrid = 4, latGrid = 4, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_gridboxsd.nc"))

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

`gridboxsum` *Determine sums of selected grid boxes*

Description

The function determines sums of selected grid boxes from data of a single CM SAF NetCDF input file.

Usage

```
gridboxsum(
  var,
  lonGrid,
  latGrid,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
lonGrid	INTEGER Number of grid boxes in x direction
latGrid	INTEGER Number of grid boxes in y direction
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of sums of selected grid boxes is written.

See Also

Other grid boxes statistics: gridboxmax(), gridboxmean(), gridboxmin(), gridboxrange(), gridboxsd(), gridboxvar()

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("2000-03-01"), as.Date("2000-05-31"), "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, 3))

## 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_1 <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)

ncvar_put(ncnew_1, var1, data)

ncatt_put(ncnew_1, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_1, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew_1)

## Determine the sums of selected grid boxes of the example CM SAF NetCDF file and write
## the output to a new file.
gridboxsum(var = "SIS", lonGrid = 4, latGrid = 4, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_gridboxsum.nc"))

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

---

gridboxvar *Determine variances of selected grid boxes*

Description

The function determines variances of selected grid boxes from data of a single CM SAF NetCDF input file.

Usage

```
gridboxvar(
  var,
  lonGrid,
  latGrid,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
lonGrid	INTEGER Number of grid boxes in x direction
latGrid	INTEGER Number of grid boxes in y direction
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of variances of selected grid boxes is written.

See Also

Other grid boxes statistics: gridboxmax(), gridboxmean(), gridboxmin(), gridboxrange(), gridboxsd(), gridboxsum()

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("2000-03-01"), as.Date("2000-05-31"), "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, 3))

## 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_1 <- nc_create(file.path(tempdir(), "CMSAF_example_file.nc"), vars)

ncvar_put(ncnew_1, var1, data)
```

```
ncatt_put(ncnew_1, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_1, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew_1)

## Determine the variances of selected grid boxes of the example CM SAF NetCDF file and write
## the output to a new file.
gridboxvar(var = "SIS", lonGrid = 4, latGrid = 4, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_gridboxvar.nc"))

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

---

hourmean *Determine hourly means*

Description

The function determines hourly means from data of a single CM SAF NetCDF input file.

Usage

```
hourmean(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other hourly statistics: `hoursum()`

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, 2), "mins")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "min"))
data <- array(250:350, dim = c(21, 21, 1441))
## 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 = "minutes 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 hourly means of the example CM SAF NetCDF file
## and write the output to a new file.
hourmean(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_hourmean.nc"))

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

`hoursum` *Determine hourly sums*

Description

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

Usage

```
hoursum(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other hourly statistics: hourmean()

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, 2), "mins")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "min"))
data <- array(250:350, dim = c(21, 21, 1441))
## 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 = "minutes 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 hourly sums of the example CM SAF NetCDF file
## and write the output to a new file.
hoursum(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_hoursum.nc"))

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

`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

```
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: acsaf_box_mergetime(), add_grid_info(), box_mergetime(), cmsaf.transform.coordinate.system(), remap()

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"))
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)
ncvar_put(ncnew, var1, data2)
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)

## Cut a region and levl, and merge both example CM SAF NetCDF files
## into one output file. First get path information of working
## directory.
levbox_mergetime(var = "SIS", level = 1, path = tempdir(),
 pattern = "CMSAF_example_file_n", outfile = file.path(tempdir(),
 "CMSAF_example_file_levbox_mergetime.nc"), lon1 = 8, lon2 = 12,
 lat1 = 48, lat2 =52)

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

---

mermean *Determine meridional means*

Description

The function determines meridional means from data of a single CM SAF NetCDF input file.

Usage

```
mermean(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

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 meridional means of the example CM SAF NetCDF file and write
## the output to a new file.
mermean(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_mermean.nc"))

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

`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

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

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other monthly statistics: mon_num_above(), mon_num_below(), mon_num_equal(), monavg(), mondaymean(), monmax(), monmean(), monmin(), monpctl(), monsd(), monsum(), monvar(), 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(10, 15, 0.5)
lat <- seq(50, 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(11, 11, 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(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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")))
```

`mon.anomaly.climatology` *Designed for the CM SAF R Toolbox.*

Description

This function is a helper function (warming stripes plot, trend plot, time series plot) called by the CM SAF R Toolbox.

Usage

```
mon.anomaly.climatology(
  var,
  infile,
  outfile,
  climatology_file,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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).
`climatology_file`	Filename of input NetCDF climatology 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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

monavg *Determine monthly averages*

Description

The function determines monthly averages from data of a single CM SAF NetCDF input file. This function is applicable to 3-dimensional NetCDF data. There is a difference between the operators monavg and monmean. The mean is regarded as a statistical function, whereas the average is found simply by adding the sample members and dividing the result by the sample size. For example, the mean of 1, 2, miss and 3 is (1 + 2 + 3)/3 = 2, whereas the average is (1 + 2 + miss + 3)/4 = miss/4 = miss. If there are no missing values in the sample, the average and mean are identical.

Usage

```
monavg(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_below(), mon_num_equal(), mondaymean(), monmax(), monmean(), monmin(), monpctl(), monsd(), monsum(), monvar(), 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("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 averages of the example CM SAF NetCDF file and
## write the output to a new file.
monavg(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_monavg.nc"))

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

---

mondaymean *Determine mean monthly daily variations*

Description

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

Usage

```
mondaymean(
```

```
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of mean monthly daily variations is written.

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_below(), mon_num_equal(), monavg(), monmax(), monmean(), monmin(), monpctl(), monsd(), monsum(), monvar(), 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)

lon <- seq(5, 8, 0.5)
lat <- seq(45, 48, 0.5)
time <- seq(ISOdate(2000, 3, 1), ISOdate(2000, 5, 31), "hours")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:272, dim = c(7, 7, 2185))

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), -999, prec = "short",
                longname = "Surface Incoming Shortwave Radiation")
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")
ncatt_put(ncnew, "SIS", "standard_name", "SIS_standard", prec = "text")
nc_close(ncnew)

## Determine the mean monthly daily variations of the example CM SAF NetCDF file and
## write the output to a new file.
mondaymean(var = "SIS", infile = file.path(tempdir(), "CMSAF_example_file.nc"),
 outfile = file.path(tempdir(), "CMSAF_example_file_mondaymean.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
 file.path(tempdir(),"CMSAF_example_file_mondaymean.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,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_below(), mon_num_equal(), monavg(), mondaymean(), monmean(), monmin(), monpctl(), monsd(), monsum(), monvar(), 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("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
## write the output to a new file.
monmax(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_monmax.nc"))

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

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. There is a difference between the operators monmean and monavg. The mean is regarded as a statistical function, whereas the average is found simply by adding the sample members and dividing the result by the sample size. For example, the mean of 1, 2, miss and 3 is (1 + 2 + 3)/3 = 2, whereas the average is (1 + 2 + miss + 3)/4 = miss/4 = miss. If there are no missing values in the sample, the average and mean are identical.

Usage

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

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_below(), mon_num_equal(), monavg(), mondaymean(), monmax(), monmin(), monpctl(), monsd(), monsum(), monvar(), 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("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(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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

```
monmin(
```

```
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_below(), mon_num_equal(), monavg(), mondaymean(), monmax(), monmean(), monpctl(), monsd(), monsum(), monvar(), 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("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(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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")))
```

`monpctl` *Determine monthly percentiles*

Description

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

Usage

```
monpctl(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  p = 0.95,
  nc = NULL
)
```

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
p	Percentile number given as probability within [0, 1] (numeric). Default is 0.95.
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_below(), mon_num_equal(), monavg(), mondaymean(), monmax(), monmean(), monmin(), monsd(), monsum(), monvar(), 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("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 90% monthly percentiles of the example CM SAF NetCDF
## file and write the output to a new file.
```

```
monpctl(var = "SIS", p = 0.9, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_monpctl.nc"))

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

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

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other monthly statistics: `mon.anomaly()`, `mon_num_above()`, `mon_num_below()`, `mon_num_equal()`, `monavg()`, `mondaymean()`, `monmax()`, `monmean()`, `monmin()`, `monpctl()`, `monsum()`, `monvar()`, `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("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 standard deviation of the example CM SAF NetCDF
## file and write the output to a new file.
monsd(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_monsd.nc"))

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

`monsum` *Determine monthly sums*

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

```
monsum(
```

```
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_below(), mon_num_equal(), monavg(), mondaymean(), monmax(), monmean(), monmin(), monpctl(), monsd(), monvar(), 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("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 sums of the example CM SAF NetCDF file and
## write the output to a new file.
monsum(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_monsum.nc"))

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

`monvar` *Determine monthly variance*

Description

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

Usage

```
monvar(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_below(), mon_num_equal(), monavg(), mondaymean(), monmax(), monmean(), monmin(), monpctl(), 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("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 variance of the example CM SAF NetCDF
## file and write the output to a new file.
monvar(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_monvar.nc"))
```

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

---

mon_num_above *Number of timesteps per month above a threshold.*

Description

This function counts the number of timesteps above a certain threshold for each month and grid point of a dataset (x >= thld). This operator should be applied to data with temporal resolution < monthly (e.g., daily).

Usage

```
mon_num_above(
  var,
  thld = 0,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
thld	Threshold (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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_below(), mon_num_equal(), monavg(), mondaymean(), monmax(), monmean(), monmin(), monpctl(), monsd(), monsum(), monvar(), 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("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 number of timesteps above a threshold of the example
## CM SAF NetCDF file and write the output to a new file.
mon_num_above(var = "SIS", thld = 300, infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_mon_num_above.nc"))

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

---

mon_num_below *Number of timesteps per month below a threshold.*

Description

This function counts the number of timesteps below a certain threshold for each month and grid point of a dataset (x <= thld). This operator should be applied to data with temporal resolution < monthly (e.g., daily).

Usage

```
mon_num_below(
  var,
  thld = 0,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
thld	Threshold (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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_equal(), monavg(), mondaymean(), monmax(), monmean(), monmin(), monpctl(), monsd(), monsum(), monvar(), 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("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 number of timesteps below a threshold of the example
## CM SAF NetCDF file and write the output to a new file.
mon_num_below(var = "SIS", thld = 300, infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_mon_num_below.nc"))

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

---

mon_num_equal *Number of timesteps per month equal a threshold.*

Description

This function counts the number of timesteps equal a certain threshold for each month and grid point of a dataset (x == thld). This operator should be applied to data with temporal resolution < monthly (e.g., daily).

Usage

```
mon_num_equal(
  var,
  thld = 0,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
thld	Threshold (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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_below(), monavg(), mondaymean(), monmax(), monmean(), monmin(), monpctl(), monsd(), monsum(), monvar(), 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("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 number of timesteps equal a threshold of the example
## CM SAF NetCDF file and write the output to a new file.
mon_num_equal(var = "SIS", thld = 300, infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_mon_num_equal.nc"))

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

`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.

Usage

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

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other mathematical operators: cmsaf.abs(), 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 <- 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)

## Multiply each timestep of the example CM SAF NetCDF file with the
## number of days per month and write the output to a new file.
muldpm(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_muldpm.nc"))

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

`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,
  nc = NULL
)
```

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other monthly statistics: `mon.anomaly()`, `mon_num_above()`, `mon_num_below()`, `mon_num_equal()`, `monavg()`, `mondaymean()`, `monmax()`, `monmean()`, `monmin()`, `monpctl()`, `monsd()`, `monsum()`, `monvar()`, `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 = "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(var = "SIS", month = c(6, 7, 8, 9), infile =
 file.path(tempdir(),"CMSAF_example_file.nc"), outfile =
 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

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

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_below(), mon_num_equal(), monavg(), mondaymean(), monmax(), monmean(), monmin(), monpctl(), monsd(), monsum(), monvar(), multimonmean(), 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(10, 15, 0.5)
```

```
lat <- seq(50, 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(11, 11, 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(var = "rain", month = c(6, 7, 8, 9), infile =
 file.path(tempdir(),"CMSAF_example_file.nc"), outfile =
 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

```
ncinfo(infile, info = "s", verbose = FALSE, nc = NULL)
```

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

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)
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)

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

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

---

num_above *Number of timesteps above a threshold.*

Description

This function counts the number of timesteps above a certain threshold for each grid point of a dataset (x >= thld).

Usage

```
num_above(
  var,
  thld = 0,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
thld	Threshold (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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other temporal operators: cmsaf.detrend(), cmsaf.mk.test(), cmsaf.regres(), num_below(), num_equal(), timavg(), timmax(), timmean(), timmin(), timpctl(), timsd(), timsum(), trend_advanced(), 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)

## Count the number of timesteps above a threshold of each grid point
## of the example CM SAF NetCDF file and write the output to a new file.
num_above(var = "SIS", thld = 300, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_num_above.nc"))

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

---

num_below *Number of timesteps below a threshold.*

Description

This function counts the number of timesteps below a certain threshold for each grid point of a dataset (x <= thld).

Usage

```
num_below(
  var,
  thld = 0,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
thld	Threshold (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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other temporal operators: cmsaf.detrend(), cmsaf.mk.test(), cmsaf.regres(), num_above(), num_equal(), timavg(), timmax(), timmean(), timmin(), timpctl(), timsd(), timsum(), trend_advanced(), 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)

## Count the number of timesteps below a threshold of each grid point
## of the example CM SAF NetCDF file and write the output to a new file.
num_below(var = "SIS", thld = 300, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_num_below.nc"))

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

---

num_equal *Number of timesteps equal a threshold.*

Description

This function counts the number of timesteps equal a certain threshold for each grid point of a dataset (x == thld).

Usage

```
num_equal(
  var,
  thld = 0,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
thld	Threshold (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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other temporal operators: cmsaf.detrend(), cmsaf.mk.test(), cmsaf.regres(), num_above(), num_below(), timavg(), timmax(), timmean(), timmin(), timpctl(), timsd(), timsum(), trend_advanced(), 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)

## Count the number of timesteps equal a threshold of each grid point
## of the example CM SAF NetCDF file and write the output to a new file.
num_equal(var = "SIS", thld = 300, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_num_equal.nc"))

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

read_file *Designed for the CM SAF R Toolbox.*

Description

This function is a helper function called by the CM SAF R Toolbox.

Usage

```
read_file(infile, var_name, nc = NULL)
```

Arguments

infile	Filename of input NetCDF file. This may include the directory (character).
var_name	Name of NetCDF variable (character).
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

read_ncvar *Read NetCDF variable.*

Description

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

Usage

```
read_ncvar(var, infile, verbose = FALSE, nc = NULL)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

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

```
## 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)

## 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(var = "SIS", infile = 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

```
remap(
  var,
  infile1,
  infile2,
  outfile,
  method = "nearest",
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc1 = NULL,
  nc2 = NULL
)
```

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".
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
nc1	Alternatively to infile1 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).
nc2	Alternatively to infile2 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other data manipulation functions: acsaf_box_mergetime(), add_grid_info(), box_mergetime(), cmsaf.transform.coordinate.system(), 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)
lat <- seq(45, 55, 0.5)
lon2 <- seq(5, 15, 1)
lat2 <- 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 = "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)

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)
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)

## 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(var = "SIS", infile1 = file.path(tempdir(),"CMSAF_example_file_1.nc"),
 infile2 = file.path(tempdir(),"CMSAF_example_file_2.nc"),
 outfile = 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")))
```

runmax *Determine running maxima*

Description

The function determines running maxima from data of a single CM SAF NetCDF input file.

Usage

```
runmax(
  var,
  nts = 6,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
nts	Number of consecutive timesteps. Computes running statistical values over a selected number of timesteps. Default is 6.
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other running statistics: runmean(), runmin(), runrange(), runsd(), runsum(), ydrunmean(), ydrunsd(), ydrunsum()

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(10, 15, 0.5)
lat <- seq(50, 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(11, 11, 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 running maxima of the example CM SAF NetCDF file and write
## the output to a new file.
runmin(var = "SIS", nts = 10, infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_runmax.nc"))

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

---

runmean *Determine running means*

Description

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

Usage

```
runmean(
  var,
```

```
  nts = 6,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
nts	Number of consecutive timesteps. Computes running statistical values over a selected number of timesteps.
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other running statistics: runmax(), runmin(), runrange(), runsd(), runsum(), ydrunmean(), ydrunsd(), ydrunsum()

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(10, 15, 0.5)
lat <- seq(50, 55, 0.5)
time <- seq(as.Date("2006-01-01"), 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(11, 11, 60))
```

```
## 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("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 running means of the example CM SAF NetCDF file and write
## the output to a new file.
runmean(var = "SIS", nts = 10, infile = "CMSAF_example_file.nc", outfile =
 "CMSAF_example_file_runmean.nc")

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_runmean.nc"))
```

`runmin` *Determine running minima*

Description

The function determines running minima from data of a single CM SAF NetCDF input file.

Usage

```
runmin(
  var,
  nts = 6,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

`var`	Name of NetCDF variable (character).
`nts`	Number of consecutive timesteps. Computes running statistical values over a selected number of timesteps.
`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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other running statistics: runmax(), runmean(), runrange(), runsd(), runsum(), ydrunmean(), ydrunsd(), ydrunsum()

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 running minima of the example CM SAF NetCDF file and write
## the output to a new file.
runmin(var = "SIS", nts = 10, infile = file.path(tempdir(),"CMSAF_example_file.nc"),
```

```
 outfile = file.path(tempdir(),"CMSAF_example_file_runmin.nc"))

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

---

runrange *Determine running range*

Description

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

Usage

```
runrange(
  var,
  nts = 6,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
nts	Number of consecutive timesteps. Computes running statistical values over a selected number of timesteps.
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of running range is written.

See Also

Other running statistics: runmax(), runmean(), runmin(), runsd(), runsum(), ydrunmean(), ydrunsd(), ydrunsum()

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(10, 15, 0.5)
lat <- seq(50, 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(11, 11, 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("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 running range of the example CM SAF NetCDF file and write
## the output to a new file.
runrange(var = "SIS", nts = 10, infile = "CMSAF_example_file.nc",
 outfile = "CMSAF_example_file_runrange.nc")

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_runrange.nc"))
```

---

runsd *Determine running standard deviation*

Description

The function determines running standard deviation from data of a single CM SAF NetCDF input file.

Usage

```
runsd(
  var,
  nts = 6,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
nts	Number of consecutive timesteps. Computes running statistical values over a selected number of timesteps.
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of running standard deviation is written.

See Also

Other running statistics: runmax(), runmean(), runmin(), runrange(), runsum(), ydrunmean(), ydrunsd(), ydrunsum()

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 running standard deviation of the example CM SAF NetCDF
## file and write the output to a new file.
runsd(var = "SIS", nts = 10, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_runsd.nc"))

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

`runsum` *Determine running sums*

Description

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

Usage

```
runsum(
  var,
  nts = 6,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
nts	Number of consecutive timesteps. Computes running statistical values over a selected number of timesteps.
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other running statistics: runmax(), runmean(), runmin(), runrange(), runsd(), ydrunmean(), ydrunsd(), ydrunsum()

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 running sums of the example CM SAF NetCDF file and write
## the output to a new file.
runsum(var = "SIS", nts = 10, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_runsum.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
 file.path(tempdir(),"CMSAF_example_file_runsum.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,
  nc = NULL
)
```

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other seasonal statistics: seasmean(), seassd(), seassum(), seasvar(), 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/>

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 seasonal anomalies of the example CM SAF NetCDF file
## and write the output to a new file.
seas.anomaly(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_seas.anomaly.nc"))

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

`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.

Usage

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

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other seasonal statistics: `seas.anomaly()`, `seassd()`, `seassum()`, `seasvar()`, `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/>

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 seasonal means of the example CM SAF NetCDF file and
## write the output to a new file.
seasmean(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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")))
```

---

seassd *Determine seasonal standard deviations*

Description

The function determines seasonal standard deviations values from data of a single CM SAF NetCDF input file. The seasonal standard deviations 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

```
seassd(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other seasonal statistics: seas.anomaly(), seasmean(), seassum(), seasvar(), 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/>

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 seasonal standard deviations of the example CM SAF NetCDF file and
## write the output to a new file.
seassd(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_seassd.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
 file.path(tempdir(),"CMSAF_example_file_seassd.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,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other seasonal statistics: seas.anomaly(), seasmean(), seassd(), seasvar(), 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/>

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

```
 outfile = 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")))
```

`seasvar` *Determine seasonal variances*

Description

The function determines seasonal variances values from data of a single CM SAF NetCDF input file. The seasonal variances 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

```
seasvar(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other seasonal statistics: `seas.anomaly()`, `seasmean()`, `seassd()`, `seassum()`, `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/>

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 seasonal variances of the example CM SAF NetCDF file and
## write the output to a new file.
seasvar(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_seasvar.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
 file.path(tempdir(),"CMSAF_example_file_seasvar.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 an rectangular grid area.

Usage

```
sellonlatbox(
```

```
  var,
  infile,
  outfile,
  lon1 = -180,
  lon2 = 180,
  lat1 = -90,
  lat2 = 90,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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 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).
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

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

```
## 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(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_sellonlatbox.nc"),
 lon1 = 8, lon2 = 12, lat1 = 48, lat2 = 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

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

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

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

```
## 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
## and write the output to a new file.
selmon(var = "SIS", month = c(3, 6), infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_selmon.nc"))

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

`selperiod` *Extract a list of dates.*

Description

This function selects a time period from a time series.

Usage

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

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

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/>

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 13-months period of the example CM SAF NetCDF file and write
## the output to a new file.
selperiod(var = "SIS", start = "2001-01-01", end = "2002-01-01",
 infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_selperiod.nc"))

unlink(c(file.path(tempdir(),"CMSAF_example_file.nc"),
 file.path(tempdir(),"CMSAF_example_file_selperiod.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,
  nc = NULL
)
```

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

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(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_selpoint.nc"),
 lon1 = 8, lat1 = 48, format = "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 outputfiles 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,
  lat1,
  station_names = NULL,
  format = "nc",
  nc34 = 4,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

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()`, `selperiod()`, `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)

## Select two points of the example CM SAF NetCDF file and write the
## output to a csv-file.
selpoint.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,
  nc = NULL
)
```

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

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 <- 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 12:00 and 21:00 values of the example CM SAF NetCDF file
## and write the output to a new file.
seltime(var = "SIS", hour_min = c("12:00:00", "21:00:00"),
 infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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 *Extract a list of years.*

Description

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

Usage

```
selyear(
```

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

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

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

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 all values of the year 2003 and 2006 of the example CM SAF
## NetCDF file and write the output to a new file.
  selyear(var = "SIS", year = c(2003,2006), infile = file.path(tempdir(),
  'CMSAF_example_file.nc'), outfile = file.path(tempdir(),
  'CMSAF_example_file_selyear.nc'))

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

`timavg` *Determine all-time average.*

Description

The function determines the all-time average 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. There is a difference between the operators timavg and timmean. The mean is regarded as a statistical function, whereas the average is found simply by adding the sample members and dividing the result by the sample size. For example, the mean of 1, 2, miss and 3 is (1 + 2 + 3)/3 = 2, whereas the average is (1 + 2 + miss + 3)/4 = miss/4 = miss. If there are no missing values in the sample, the average and mean are identical.

Usage

```
timavg(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including the temporal average is written.

See Also

Other temporal operators: cmsaf.detrend(), cmsaf.mk.test(), cmsaf.regres(), num_above(), num_below(), num_equal(), timmax(), timmean(), timmin(), timpctl(), timsd(), timsum(), trend_advanced(), 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 climatology of the example CM SAF NetCDF file and write
## the output to a new file.
timavg(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_timavg.nc"))

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

`timcor` *Determine correlations over time.*

Description

The function determines correlations over time from data of two CM SAF NetCDF input files. This function is applicable to 3-dimensional NetCDF data.

Usage

```
timcor(
  var1,
  infile1,
  var2,
  infile2,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc1 = NULL,
  nc2 = NULL
)
```

Arguments

`var1`	Name of NetCDF variable of the first data set (character).
`infile1`	Filename of first input NetCDF file. This may include the directory (character).
`var2`	Name of NetCDF variable of the second data set (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

nc1 Alternatively to infile1 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

nc2 Alternatively to infile2 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including a time series of correlations over time is written.

See Also

Other correlation and covariance: fldcor(), fldcovar(), timcovar()

Examples

```
## Create two example NetCDF files with a similar structure as used by CM
## SAF. The files are 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 <- as.Date("2000-05-31")
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 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), -999, prec = "float")
vars <- list(var1)
ncnew_1 <- nc_create(file.path(tempdir(), "CMSAF_example_file_1.nc"), vars)
ncnew_2 <- nc_create(file.path(tempdir(), "CMSAF_example_file_2.nc"), vars)

ncvar_put(ncnew_1, var1, data1)
ncvar_put(ncnew_2, var1, data2)

ncatt_put(ncnew_1, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_1, "lat", "standard_name", "latitude", prec = "text")

ncatt_put(ncnew_2, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_2, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew_1)
nc_close(ncnew_2)

## Determine the correlations over time of the example CM SAF NetCDF files and
```

```
## write the output to a new file.
timcor(var1 = "SIS", infile1 = file.path(tempdir(),"CMSAF_example_file_1.nc"),
      var2 = "SIS", infile2 = file.path(tempdir(), "CMSAF_example_file_2.nc"),
      outfile = file.path(tempdir(),"CMSAF_example_file_timcor.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_timcor.nc")))
```

`timcovar` *Determine covariances over time.*

Description

The function determines covariances over time from data of two CM SAF NetCDF input files. This function is applicable to 3-dimensional NetCDF data.

Usage

```
timcovar(
  var1,
  infile1,
  var2,
  infile2,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc1 = NULL,
  nc2 = NULL
)
```

Arguments

`var1`	Name of NetCDF variable of the first data set (character).
`infile1`	Filename of first input NetCDF file. This may include the directory (character).
`var2`	Name of NetCDF variable of the second data set (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
`nc1`	Alternatively to `infile1` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).
`nc2`	Alternatively to `infile2` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

A NetCDF file including a time series of covariances over time is written.

See Also

Other correlation and covariance: `fldcor()`, `fldcovar()`, `timcor()`

Examples

```
## Create two example NetCDF files with a similar structure as used by CM
## SAF. The files are 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 <- as.Date("2000-05-31")
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 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), -999, prec = "float")
vars <- list(var1)
ncnew_1 <- nc_create(file.path(tempdir(), "CMSAF_example_file_1.nc"), vars)
ncnew_2 <- nc_create(file.path(tempdir(), "CMSAF_example_file_2.nc"), vars)

ncvar_put(ncnew_1, var1, data1)
ncvar_put(ncnew_2, var1, data2)

ncatt_put(ncnew_1, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_1, "lat", "standard_name", "latitude", prec = "text")

ncatt_put(ncnew_2, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_2, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew_1)
nc_close(ncnew_2)

## Determine the covariances over time of the example CM SAF NetCDF files and
## write the output to a new file.
timcovar(var1 = "SIS", infile1 = file.path(tempdir(),"CMSAF_example_file_1.nc"),
      var2 = "SIS", infile2 = file.path(tempdir(), "CMSAF_example_file_2.nc"),
      outfile = file.path(tempdir(),"CMSAF_example_file_timcovar.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_timcovar.nc")))
```

`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

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

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

timmax *Determine all-time maxima.*

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

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

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other temporal operators: cmsaf.detrend(), cmsaf.mk.test(), cmsaf.regres(), num_above(), num_below(), num_equal(), timavg(), timmean(), timmin(), timpctl(), timsd(), timsum(), trend_advanced(), 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("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(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including the temporal mean is written.

See Also

Other temporal operators: cmsaf.detrend(), cmsaf.mk.test(), cmsaf.regres(), num_above(), num_below(), num_equal(), timavg(), timmax(), timmin(), timpctl(), timsd(), timsum(), trend_advanced(), 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 climatology of the example CM SAF NetCDF file and write
## the output to a new file.
timmean(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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

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

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other temporal operators: cmsaf.detrend(), cmsaf.mk.test(), cmsaf.regres(), num_above(), num_below(), num_equal(), timavg(), timmax(), timmean(), timpctl(), timsd(), timsum(), trend_advanced(), 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("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(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
```

```
 outfile = 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

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

Arguments

`var`	Name of NetCDF variable (character).
`p`	Percentile number given as probability within [0, 1] (numeric). Default is 0.95.
`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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other temporal operators: cmsaf.detrend(), cmsaf.mk.test(), cmsaf.regres(), num_above(), num_below(), num_equal(), timavg(), timmax(), timmean(), timmin(), timsd(), timsum(), trend_advanced(), 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 90% percentile of the example CM SAF NetCDF file and
## write the output to a new file.
timpctl(var = "SIS", p = 0.9, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = 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,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other temporal operators: cmsaf.detrend(), cmsaf.mk.test(), cmsaf.regres(), num_above(), num_below(), num_equal(), timavg(), timmax(), timmean(), timmin(), timpctl(), timsum(), trend_advanced(), 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(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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")))
```

`timselmean` *Determine time selection means*

Description

The function determines the mean values for a pre-selected number of timesteps from data of a single CM SAF NetCDF input file.

Usage

```
timselmean(
  var,
  nts = 6,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
nts	Number of input timesteps for each output timestep
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other time range statistics: timselsum()

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(10, 15, 0.5)
lat <- seq(50, 55, 0.5)
time <- seq(as.Date("2006-01-01"), 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(11, 11, 60))

## 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("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 time selection means of the example CM SAF NetCDF file
## and write the output to a new file.
timselmean(var = "SIS", nts = 10, infile = "CMSAF_example_file.nc",
 outfile = "CMSAF_example_file_timselmean.nc")

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_timselmean.nc"))
```

`timselsum` *Determine time selection sums*

Description

The function determines the sums for a pre-selected number of timesteps from data of a single CM SAF NetCDF input file.

Usage

```
timselsum(
  var,
  nts = 6,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

`var`	Name of NetCDF variable (character).
`nts`	Number of input timesteps for each output timestep
`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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other time range statistics: timselmean()

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 time selection sums of the example CM SAF NetCDF file
## and write the output to a new file.
timselsum(var = "SIS", nts = 10, infile = file.path(tempdir(),
 "CMSAF_example_file.nc"), outfile = file.path(tempdir(),
 "CMSAF_example_file_timselsum.nc"))

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

```
timsum(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including the temporal sum is written.

See Also

Other temporal operators: cmsaf.detrend(), cmsaf.mk.test(), cmsaf.regres(), num_above(), num_below(), num_equal(), timavg(), timmax(), timmean(), timmin(), timpctl(), timsd(), trend_advanced(), 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 sum of the example CM SAF NetCDF file and
## write the output to a new file.
timsum(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

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: cmsaf.detrend(), cmsaf.mk.test(), cmsaf.regres(), num_above(), num_below(), num_equal(), timavg(), timmax(), timmean(), timmin(), timpctl(), timsd(), timsum(), trend_advanced()

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 trend of the example CM SAF NetCDF file and write the
## output to a new file.
trend(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_trend.nc"))

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

`trend_advanced` *Determine multiple linear trends.*

Description

The function determines the trend from data of two CM SAF NetCDF input files basing on a multiple linear model. Learn more <http://www.sthda.com/english/articles/40-regression-analysis/ 168-multiple-linear-regression-in-r/>

Usage

```
trend_advanced(
  var1,
  infile1,
  var2,
  infile2,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc1 = NULL,
  nc2 = NULL
)
```

Arguments

`var1`	Name of NetCDF variable of the first data set (character).
`infile1`	Filename of input NetCDF file. This may include the directory (character).

var2	Name of NetCDF variable of the second data set (character).
infile2	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
nc1	Alternatively to infile1 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).
nc2	Alternatively to infile2 you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

A NetCDF file including four data layers is written. One layer (trend1) contains the linear trend based on the time steps. Another layer (trend2) contains linear trend based on var2. The two other layers contain 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: cmsaf.detrend(), cmsaf.mk.test(), cmsaf.regres(), num_above(), num_below(), num_equal(), timavg(), timmax(), timmean(), timmin(), timpctl(), timsd(), timsum(), trend()

Examples

```
## Create two example NetCDF files with a similar structure as used by CM
## SAF. The files are 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(10, 15, 0.5)
lat <- seq(50, 55, 0.5)
time <- as.Date("2000-05-31")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data1 <- array(250:350, dim = c(11, 11, 1))
data2 <- array(230:320, dim = c(11, 11, 1))

## 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), -999, prec = "float")
vars <- list(var1)
ncnew_1 <- nc_create(file.path(tempdir(), "CMSAF_example_file_1.nc"), vars)
ncnew_2 <- nc_create(file.path(tempdir(), "CMSAF_example_file_2.nc"), vars)

ncvar_put(ncnew_1, var1, data1)
ncvar_put(ncnew_2, var1, data2)

ncatt_put(ncnew_1, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_1, "lat", "standard_name", "latitude", prec = "text")

ncatt_put(ncnew_2, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew_2, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew_1)
nc_close(ncnew_2)

## Determine the multiple linear trend of the example CM SAF NetCDF files and
## write the output to a new file.
trend_advanced(var1 = "SIS", infile1 = file.path(tempdir(),"CMSAF_example_file_1.nc"),
      var2 = "SIS", infile2 = file.path(tempdir(), "CMSAF_example_file_2.nc"),
      outfile = file.path(tempdir(),"CMSAF_example_file_trend_advanced.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_trend_advanced.nc")))
```

`wfldmean` *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

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

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other spatial operators: fldmax(), fldmean(), fldmin(), fldrange(), fldsd(), fldsum()

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, 10, 0.5)
lat <- seq(45, 50, 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(11, 11, 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.
```

```
wfldmean(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_wfldmean.nc"))

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

---

ydaymax *Determine multi-year daily maxima*

Description

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

Usage

```
ydaymax(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other daily statistics: dayavg(), daymax(), daymean(), daymin(), daypctl(), dayrange(), daysd(), daysum(), dayvar(), ydaymean(), ydaymin(), ydayrange(), ydaysd(), ydaysum()

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("2009-01-01"), as.Date("2010-12-31"), "day")
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, 730))

## 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 daily maximum of the example CM SAF NetCDF file
## and write the output to a new file.
ydaymax(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_ydaymax.nc"))

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

---

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,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other daily statistics: dayavg(), daymax(), daymean(), daymin(), daypctl(), dayrange(), daysd(), daysum(), dayvar(), ydaymax(), ydaymin(), ydayrange(), ydaysd(), ydaysum()

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(10, 15, 0.5)
lat <- seq(50, 55, 0.5)
time <- seq(as.Date("2009-01-01"), as.Date("2010-12-31"), "day")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(11, 11, 730))

## 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 daily mean of the example CM SAF NetCDF file
## and write the output to a new file.
ydaymean(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_ydaymean.nc"))

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

---

ydaymin *Determine multi-year daily minima*

Description

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

Usage

```
ydaymin(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other daily statistics: dayavg(), daymax(), daymean(), daymin(), daypctl(), dayrange(), daysd(), daysum(), dayvar(), ydaymax(), ydaymean(), ydayrange(), ydaysd(), ydaysum()

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("2009-01-01"), as.Date("2010-12-31"), "day")
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, 730))

## 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 daily minimum of the example CM SAF NetCDF file
## and write the output to a new file.
ydaymin(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_ydaymin.nc"))

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

ydayrange *Determine multi-year daily range*

Description

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

Usage

```
ydayrange(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other daily statistics: dayavg(), daymax(), daymean(), daymin(), daypctl(), dayrange(), daysd(), daysum(), dayvar(), ydaymax(), ydaymean(), ydaymin(), ydaysd(), ydaysum()

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("2009-01-01"), as.Date("2010-12-31"), "day")
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, 730))

## 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 daily range of the example CM SAF NetCDF file
## and write the output to a new file.
ydayrange(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_ydayrange.nc"))

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

---

ydaysd *Determine multi-year daily standard deviations*

Description

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

Usage

```
ydaysd(
```

```
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other daily statistics: dayavg(), daymax(), daymean(), daymin(), daypctl(), dayrange(), daysd(), daysum(), dayvar(), ydaymax(), ydaymean(), ydaymin(), ydayrange(), ydaysum()

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(10, 15, 0.5)
lat <- seq(50, 55, 0.5)
time <- seq(as.Date("2009-01-01"), as.Date("2010-12-31"), "day")
origin <- as.Date("1983-01-01 00:00:00")
time <- as.numeric(difftime(time, origin, units = "hour"))
data <- array(250:350, dim = c(11, 11, 730))

## 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 daily standard deviations of the example
## CM SAF NetCDF file and write the output to a new file.
ydaysd(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_ydaysd.nc"))

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

---

ydaysum *Determine multi-year daily sums*

Description

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

Usage

```
ydaysum(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other daily statistics: dayavg(), daymax(), daymean(), daymin(), daypctl(), dayrange(), daysd(), daysum(), dayvar(), ydaymax(), ydaymean(), ydaymin(), ydayrange(), ydaysd()

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("2009-01-01"), as.Date("2010-12-31"), "day")
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, 730))

## 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 daily sums of the example CM SAF NetCDF file
## and write the output to a new file.
ydaysum(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_ydaysum.nc"))

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

ydrunmean *Determine multi-year daily running means.*

Description

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

Usage

```
ydrunmean(
  var,
  nts = 6,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
nts	Number of consecutive timesteps. Computes running statistical values over a selected number of timesteps.
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other running statistics: runmax(), runmean(), runmin(), runrange(), runsd(), runsum(), ydrunsd(), ydrunsum()

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 daily running means of the example CM SAF
## NetCDF file and write the output to a new file.
ydrunmean(var = "SIS", nts = 10, infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_ydrunmean.nc"))

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

---

ydrunsd *Determine multi-year daily running standard deviations*

Description

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

Usage

```
ydrunsd(
```

```
  var,
  nts = 6,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).
nts	Number of consecutive timesteps. Computes running statistical values over a selected number of timesteps.
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other running statistics: runmax(), runmean(), runmin(), runrange(), runsd(), runsum(), ydrunmean(), ydrunsum()

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(10, 15, 0.5)
lat <- seq(50, 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(11, 11, 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 daily running standard deviations of the example
## CM SAF NetCDF file and write the output to a new file.
ydrunsd(var = "SIS", nts = 10, infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_ydrunsd.nc"))

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

---

ydrunsum *Determine multi-year daily running sums*

Description

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

Usage

```
ydrunsum(
  var,
  nts = 6,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

Arguments

var	Name of NetCDF variable (character).

nts	Number of consecutive timesteps. Computes running statistical values over a selected number of timesteps.
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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other running statistics: runmax(), runmean(), runmin(), runrange(), runsd(), runsum(), ydrunmean(), ydrunsd()

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 daily running sums of the example CM SAF
## NetCDF file and write the output to a new file.
ydrunsum(var = "SIS", nts = 10, infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_ydrunsum.nc"))

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

`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.

Usage

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

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
`nc`	Alternatively to `infile` you can specify the input as an object of class `ncdf4` (as returned from `ncdf4::nc_open`).

Value

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

See Also

Other annual statistics: yearmax(), yearmean(), yearmin(), yearrange(), yearsd(), yearsum(), yearvar()

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 annual anomalies of the example CM SAF NetCDF file and
## write the output to a new file.
year.anomaly(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_year.anomaly.nc"))

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

---

yearmax *Determine annual maxima*

Description

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

Usage

```
yearmax(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other annual statistics: year.anomaly(), yearmean(), yearmin(), yearrange(), yearsd(), yearsum(), yearvar()

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 annual maxima of the example CM SAF NetCDF file and write
## the output to a new file.
yearmax(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_yearmax.nc"))

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

---

yearmean *Determine annual means*

Description

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,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other annual statistics: year.anomaly(), yearmax(), yearmin(), yearrange(), yearsd(), yearsum(), yearvar()

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 annual means of the example CM SAF NetCDF file and
## write the output to a new file.
yearmean(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_yearmean.nc"))
```

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

---

yearmin *Determine annual minima*

Description

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

Usage

```
yearmin(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other annual statistics: year.anomaly(), yearmax(), yearmean(), yearrange(), yearsd(), yearsum(), yearvar()

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 annual minima of the example CM SAF NetCDF file and write
## the output to a new file.
yearmin(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_yearmin.nc"))

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

---

yearrange *Determine annual range*

Description

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

Usage

```
yearrange(
```

```
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other annual statistics: year.anomaly(), yearmax(), yearmean(), yearmin(), yearsd(), yearsum(), yearvar()

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 annual range of the example CM SAF NetCDF file and write
## the output to a new file.
yearrange(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_yearrange.nc"))

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

`yearsd` *Determine annual standard deviation*

Description

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

Usage

```
yearsd(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other annual statistics: year.anomaly(), yearmax(), yearmean(), yearmin(), yearrange(), yearsum(), yearvar()

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 annual standard deviation of the example CM SAF NetCDF file
## and write the output to a new file.
yearsd(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_yearsd.nc"))

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

yearsum *Determine annual sums*

Description

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

Usage

```
yearsum(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other annual statistics: year.anomaly(), yearmax(), yearmean(), yearmin(), yearrange(), yearsd(), yearvar()

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 annual sums of the example CM SAF NetCDF file and write
## the output to a new file.
yearsum(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_yearsum.nc"))

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

---

yearvar *Determine annual variance*

Description

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

Usage

```
yearvar(
  var,
```

```
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other annual statistics: year.anomaly(), yearmax(), yearmean(), yearmin(), yearrange(), yearsd(), 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)
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 variance of the example CM SAF NetCDF file and write
## the output to a new file.
yearvar(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_yearvar.nc"))

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

---

ymonmax *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,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_below(), mon_num_equal(), monavg(), mondaymean(), monmax(), monmean(), monmin(), monpctl(), monsd(), monsum(), monvar(), multimonmean(), multimonsum(), 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 = "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(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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

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

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_below(), mon_num_equal(), monavg(), mondaymean(), monmax(), monmean(), monmin(), monpctl(), monsd(), monsum(), monvar(), multimonmean(), multimonsum(), ymonmax(), 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 = "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(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_below(), mon_num_equal(), monavg(), mondaymean(), monmax(), monmean(), monmin(), monpctl(), monsd(), monsum(), monvar(), multimonmean(), multimonsum(), ymonmax(), ymonmean(), 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 = "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 minimum of the example CM SAF NetCDF
## file and write the output to a new file.
ymonmin(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_ymonmin.nc"))

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

---

ymonsd *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

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

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_below(), mon_num_equal(), monavg(), mondaymean(), monmax(), monmean(), monmin(), monpctl(), monsd(), monsum(), monvar(), 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(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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.

Usage

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

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other monthly statistics: mon.anomaly(), mon_num_above(), mon_num_below(), mon_num_equal(), monavg(), mondaymean(), monmax(), monmean(), monmin(), monpctl(), monsd(), monsum(), monvar(), 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(0:250, 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("SDU", "h", 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 sum of the example CM SAF NetCDF
## file and write the output to a new file.
ymonsum(var = "SDU", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_ymonsum.nc"))

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

---

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,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other seasonal statistics: seas.anomaly(), seasmean(), seassd(), seassum(), seasvar(), 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/>

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(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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

```
yseasmean(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other seasonal statistics: seas.anomaly(), seasmean(), seassd(), seassum(), seasvar(), 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
## file and write the output to a new file.
yseasmean(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_yseasmean.nc"))
```

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

---

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

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

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other seasonal statistics: seas.anomaly(), seasmean(), seassd(), seassum(), seasvar(), yseasmax(), yseasmean(), 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 minimum of the example CM SAF
## NetCDF file and write the output to a new file.
yseasmin(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other seasonal statistics: seas.anomaly(), seasmean(), seassd(), seassum(), seasvar(), yseasmax(), yseasmean(), 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(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = 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")))
```

---

zonmean *Determine zonal means*

Description

The function determines zonal means from data of a single CM SAF NetCDF input file.

Usage

```
zonmean(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other zonal statistics: zonsum()

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 zonal means of the example CM SAF NetCDF file and write
## the output to a new file.
zonmean(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_zonmean.nc"))

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

zonsum *Determine zonal sums*

Description

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

Usage

```
zonsum(
  var,
  infile,
  outfile,
  nc34 = 4,
  overwrite = FALSE,
  verbose = FALSE,
  nc = NULL
)
```

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
nc	Alternatively to infile you can specify the input as an object of class ncdf4 (as returned from ncdf4::nc_open).

Value

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

See Also

Other zonal statistics: zonmean()

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 zonal sums of the example CM SAF NetCDF file and write
## the output to a new file.
zonsum(var = "SIS", infile = file.path(tempdir(),"CMSAF_example_file.nc"),
 outfile = file.path(tempdir(),"CMSAF_example_file_zonsum.nc"))

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

Index