

# Package ‘cmsaf’

August 26, 2019

**Version** 2.0.1

**Title** Tools for CM SAF NetCDF Data

**Contact** CM SAF Team <contact.cmsaf@dwd.de>

**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 EUMETSAT's Satellite Application Facilities. The CM SAF contributes to the sustainable monitoring of the climate system by providing essential climate variables related to the energy and water cycle of the atmosphere (<<http://www.cmsaf.eu>>). It is a joint cooperation of eight National Meteorological and Hydrological Services. The 'cmsaf' R-package provides a collection of R-operators for the analysis and manipulation of CM SAF NetCDF formatted data. The 'cmsaf' R-package is tested for CM SAF NetCDF data. Other CF conform NetCDF data should be applicable, but there is no guarantee for an error-free application. The 'cmsaf' R-package includes a 'shiny' based interface for an easy application of the 'cmsaf' package operators and the preparation and visualization of CM SAF NetCDF data. CM SAF climate data records are provided for free via (<<https://wui.cmsaf.eu>>). Detailed information and test data are provided on the CM SAF webpage (<[http://www.cmsaf.eu/R\\_toolbox](http://www.cmsaf.eu/R_toolbox)>).

**URL** <https://www.cmsaf.eu>

**License** GPL (>= 3)

**Depends** R (>= 3.5)

**Imports** fields (>= 9.6), FNN (>= 1.1), ncdf4 (>= 1.16), rainfarmr (>= 0.1), raster (>= 2.8), sp (>= 1.3)

**NeedsCompilation** no

**Repository** CRAN

**Suggests** colourpicker (>= 1.0), colorspace (>= 1.4), mapproj (>= 1.2), maps (>= 3.3), maptools (>= 0.9), R.utils (>= 0.9), rworldxtra (>= 1.01), shiny (>= 1.2), shinyjs (>= 1.0), shinythemes (>= 1.1), spelling (>= 2.1), tcltk (>= 3.5), testthat (>= 2.1)

**RoxygenNote** 6.1.1

**Encoding** UTF-8

**Language** en-US

**Author** Steffen Kothe [aut, cre]

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

**Date/Publication** 2019-08-26 14:50:05 UTC

## R topics documented:

add_grid_info . . . . .	3
box_mergetime . . . . .	4
change_att . . . . .	6
cmsaf . . . . .	7
cmsaf.add . . . . .	9
cmsaf.adde . . . . .	10
cmsaf.cat . . . . .	12
cmsaf.div . . . . .	13
cmsaf.divc . . . . .	15
cmsaf.mul . . . . .	17
cmsaf.mulc . . . . .	18
cmsaf.sub . . . . .	20
cmsaf.subc . . . . .	22
dayrange . . . . .	23
divdpm . . . . .	24
extract.level . . . . .	26
extract.period . . . . .	28
fldmax . . . . .	29
fldmean . . . . .	30
fldmin . . . . .	32
get_time . . . . .	33
levbox_mergetime . . . . .	34
mon.anomaly . . . . .	36
monmax . . . . .	37
monmean . . . . .	39
monmin . . . . .	40
monsd . . . . .	41
monsum . . . . .	43
muldpm . . . . .	44
multimonmean . . . . .	46
multimonsum . . . . .	47
ncinfo . . . . .	49
read_ncvar . . . . .	50
remap . . . . .	51
run_toolbox . . . . .	53
seas.anomaly . . . . .	54
seasmean . . . . .	55
seassum . . . . .	56
sellonlatbox . . . . .	58
selmon . . . . .	59

selperiod . . . . .	61
selpoint . . . . .	62
selpoint.multi . . . . .	64
seltime . . . . .	66
selyear . . . . .	67
timmax . . . . .	69
timmean . . . . .	70
timmin . . . . .	72
timpctl . . . . .	73
timsd . . . . .	74
timsun . . . . .	76
trend . . . . .	77
wfldmean . . . . .	79
ydaymean . . . . .	80
year.anomaly . . . . .	82
yearmean . . . . .	83
yearsum . . . . .	85
ymonmax . . . . .	86
ymonmean . . . . .	87
ymonmin . . . . .	89
ymons . . . . .	90
ymonsun . . . . .	92
yseasmax . . . . .	93
yseasmean . . . . .	94
yseasmin . . . . .	96
yseassd . . . . .	97

**Index****100**


---

add_grid_info	<i>Add grid info</i>
---------------	----------------------

---

**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: [box\\_mergetime](#), [levbox\\_mergetime](#), [remap](#)

---

box_mergetime	<i>Function to combine NetCDF files and simultaneously cut a region (and level).</i>
---------------	--

---

### Description

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

### Usage

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

### Arguments

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: [add\\_grid\\_info](#), [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("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("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.
wd <- getwd()
box_mergetime("SIS", wd, "CMSAF_example_file_n",
```

```
"CMSAF_example_file_box_mergetime.nc", 8, 12, 48, 52)

unlink(c("CMSAF_example_file_n1.nc", "CMSAF_example_file_n2.nc",
"CMSAF_example_file_box_mergetime.nc"))
```

---

change\_att

*Change attributes of a NetCDF variable.*


---

### Description

This function can change the name, standard\_name, long\_name, units, \_FillValue and missing\_value of a variable. There is no separate outfile, thus use this function with care. The values for v\_name, s\_name, l\_name, u\_name, F\_val and m\_val are optional and will only be changed if they are given. If an attribute is not defined yet, it is added by the function.

### Usage

```
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 ncd4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

unlink("CMSAF_example_file.nc")
```

---

cmsaf

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

---

## Description

The 'cmsaf' 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. As interface to NetCDF data the [ncdf4-package](#) is used.

**Toolbox**

`run_toolbox` (user-friendly shiny app)

**Mathematical operators**

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

**Daily statistics**

`dayrange`, `ydaymean`

**Monthly statistics**

`mon.anomaly`, `monmax`, `monmean`, `monmin`, `monsd`, `monsum`, `multimonmean`, `multimonsum`, `ymonmax`,  
`ymonmean`, `ymonmin`, `ymonsd`, `ymonsum`

**Seasonal statistics**

`seas.anomaly`, `seasmean`, `seassum`, `yseasmax`, `yseasmean`, `yseasmin`, `yseassd`

**Annual statistics**

`year.anomaly`, `yearmean`, `yearsum`

**Temporal operators**

`timax`, `timmean`, `timmin`, `timpctl`, `timsd`, `tisum`, `trend`

**Spatial operators**

`fldmax`, `fldmean`, `fldmin`, `wfldmean`

**Selection and removal functions**

`extract.level`, `extract.period`, `sellonlatbox`, `selmon`, `selperiod`, `selpoint.multi`, `selpoint`,  
`selttime`, `selyear`

**Data manipulation**

`box_mergetime`, `levbox_mergetime`, `add_grid_info`, `remap`

**Other functions**

`cmsaf.cat`, `get_time`, `ncinfo`, `read_ncvar`

**Author(s)**

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

Contact: CM SAF Team <contact.cmsaf@dwd.de>



## References

<http://www.cmsaf.eu>

Kothe, S.; Hollmann, R.; Pfeifroth, U.; Träger-Chatterjee, C.; Trentmann, J. The CM SAF R Tool-box—A Tool for the Easy Usage of Satellite-Based Climate Data in NetCDF Format. ISPRS Int. J. Geo-Inf. 2019, 8, 109. <https://doi.org/10.3390/ijgi8030109>

---

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(vari1, vari2, infile1, infile2, outfile, nc34 = 4,
          overwrite = FALSE, verbose = FALSE)
```

## Arguments

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

## Value

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

## See Also

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

## Examples

```

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("CMSAF_example_file_2.nc", vars)
ncvar_put(ncnew, var1, data2)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

unlink(c("CMSAF_example_file_1.nc", "CMSAF_example_file_2.nc",
  "CMSAF_example_file_add.nc"))

```

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

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

**See Also**

Other mathematical operators: [cmsaf.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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_addc.nc"))

```

---

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 ncd4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("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("SIS", c("CMSAF_example_file_1.nc", "CMSAF_example_file_2.nc"),
  "CMSAF_example_file_cat.nc")

unlink(c("CMSAF_example_file_1.nc", "CMSAF_example_file_2.nc",
  "CMSAF_example_file_cat.nc"))

```

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

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

**See Also**

Other mathematical operators: [cmsaf.addc](#), [cmsaf.add](#), [cmsaf.divc](#), [cmsaf.mulc](#), [cmsaf.mul](#), [cmsaf.subc](#), [cmsaf.sub](#), [divdpm](#), [muldpm](#)

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("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.divc("SIS", "SIS", "CMSAF_example_file_1.nc", "CMSAF_example_file_2.nc",
  "CMSAF_example_file_div.nc")

unlink(c("CMSAF_example_file_1.nc", "CMSAF_example_file_2.nc",
  "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)

```

## Arguments

var	Name of NetCDF variable (character).
const	Constant number (numeric).
infile	Filename of input NetCDF file. This may include the directory (character).

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

### Value

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

### See Also

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

### Examples

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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



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

---

 cmsaf.mul

*Multiply the fields of two input NetCDF files.*


---

### Description

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

### Usage

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

### Arguments

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

### Value

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

### See Also

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

## Examples

```

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("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("SIS", "SIS", "CMSAF_example_file_1.nc", "CMSAF_example_file_2.nc",
  "CMSAF_example_file_mul.nc")

unlink(c("CMSAF_example_file_1.nc", "CMSAF_example_file_2.nc",
  "CMSAF_example_file_mul.nc"))

```

## 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)
```

## Arguments

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

## Value

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

## See Also

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

## Examples

```
## Create an example NetCDF file with a similar structure as used by CM  
## SAF. The file is created with the ncdf4 package. Alternatively  
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>  
  
library(ncdf4)  
  
## create some (non-realistic) example data  
  
lon <- seq(5, 15, 0.5)  
lat <- seq(45, 55, 0.5)  
time <- seq(as.Date("2000-01-01"), as.Date("2010-12-31"), "month")  
origin <- as.Date("1983-01-01 00:00:00")  
time <- as.numeric(difftime(time, origin, units = "hour"))  
data <- array(250:350, dim = c(21, 21, 132))  
  
## create example NetCDF  
  
x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
```

```

y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
  vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create("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("SIS", 10, "CMSAF_example_file.nc",
  "CMSAF_example_file_mulc.nc")

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_mulc.nc"))

```

---

cmsaf.sub

*Subtract the fields of two input NetCDF files.*

---

### Description

The function subtracts the fields of infile1 from 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.sub(vari1, vari2, infile1, infile2, outfile, nc34 = 4,
  overwrite = FALSE, verbose = FALSE)

```

### Arguments

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

**Value**

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

**See Also**

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

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("CMSAF_example_file_2.nc", vars)
ncvar_put(ncnew, var1, data2)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

```
unlink(c("CMSAF_example_file_1.nc", "CMSAF_example_file_2.nc",
" CMSAF_example_file_sub.nc"))
```

---

cmsaf.subc

*Subtract a constant from a dataset.*

---

## Description

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

## Usage

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

## Arguments

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

## Value

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

## See Also

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

## Examples

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

library(ncdf4)

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

```

lon <- seq(5, 15, 0.5)
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("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("SIS", 10, "CMSAF_example_file.nc",
  "CMSAF_example_file_subc.nc")

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_subc.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)

```

## Arguments

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

**Value**

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

**See Also**

Other daily statistics: [ydaymean](#)

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_dayrange.nc")

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



**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)
```

**Arguments**

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

**Value**

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

**See Also**

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

**Examples**

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

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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Divide each timestep of the example CM SAF NetCDF file by the number
## of days per month and write the output to a new file.
divdpm("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_divdpm.nc")

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

```

### Arguments

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

### Value

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

**See Also**

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

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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", "CMSAF_example_file.nc",
  "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("SIS", "CMSAF_example_file.nc",
  "CMSAF_example_file_extract.level2.nc", "all")

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_extract.level*.nc"))
```

---

extract.period	<i>Remove a time period.</i>
----------------	------------------------------

---

### 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)
```

### Arguments

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

### Value

A NetCDF file excluding the selected time period is written.

### See Also

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

### Examples

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
```

```

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

## create example NetCDF

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

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

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_extract.period.nc"))

```

---

fldmax

*Determine the field 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)

```

## Arguments

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

**Value**

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

**See Also**

Other spatial operators: [fldmean](#), [fldmin](#), [wfldmean](#)

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_fldmax.nc")

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

---

fldmean

*Determine the field 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)
```

**Arguments**

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

**Value**

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

**See Also**

Other spatial operators: [fldmax](#), [fldmin](#), [wfldmean](#)

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_fldmean.nc")

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_fldmean.nc"))

```

---

fldmin

*Determine the field 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)

```

## Arguments

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

## Value

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

## See Also

Other spatial operators: [fldmax](#), [fldmean](#), [wfldmean](#)



**Examples**

```

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the minimum values of the example CM SAF NetCDF file and
## write the output to a new file.
fldmin("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_fldmin.nc")

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_fldmin.nc"))

```

---

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

---

levbox_mergetime	<i>Function to combine NetCDF files and simultaneously cut a region and level.</i>
------------------	--

---

**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: [add\\_grid\\_info](#), [box\\_mergetime](#), [remap](#)

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("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 lev1, and merge both example CM SAF NetCDF files
## into one output file. First get path information of working
## directory.
wd <- getwd()
levbox_mergetime("SIS", 1, wd, "CMSAF_example_file_n",
  "CMSAF_example_file_levbox_mergetime.nc", 8, 12, 48, 52)

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

### Arguments

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

### Value

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

### See Also

Other monthly statistics: [monmax](#), [monmean](#), [monmin](#), [monsd](#), [monsum](#), [multimonmean](#), [multimonsum](#), [ymonmax](#), [ymonmean](#), [ymonmin](#), [ymonsd](#), [ymonsum](#)

## Examples

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the monthly anomalies of the example CM SAF NetCDF file and
## write the output to a new file.
mon.anomaly("SIS", "CMSAF_example_file.nc",
  "CMSAF_example_file_mon.anomaly.nc")

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_mon.anomaly.nc"))
```

---

monmax

*Determine monthly maxima.*

---

## Description

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

## Usage

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

**Arguments**

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

**Value**

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

**See Also**

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

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_monmax.nc")

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

---

monmean	<i>Determine monthly means</i>
---------	--------------------------------

---

## Description

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

## Usage

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

## Arguments

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

## Value

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

## See Also

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

## Examples

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

library(ncdf4)

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

```

lon <- seq(5, 15, 0.5)
lat <- seq(45, 55, 0.5)
time <- seq(as.Date("2000-01-01"), as.Date("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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

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

```

## Arguments

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



**Value**

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

**See Also**

Other monthly statistics: [mon.anomaly](#), [monmax](#), [monmean](#), [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 ncd4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

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

**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)
```

**Arguments**

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

**Value**

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

**See Also**

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

**Examples**

```
## 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("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("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_monsd.nc")

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

```

## Arguments

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

## Value

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

## See Also

Other monthly statistics: [mon.anomaly](#), [monmax](#), [monmean](#), [monmin](#), [monsd](#), [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 ncd4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_monsum.nc")

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

**Arguments**

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

**Value**

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

**See Also**

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

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_muldpm.nc")

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

---

multimonmean	<i>Determine multi-monthly means</i>
--------------	--------------------------------------

---

### Description

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

### Usage

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

### Arguments

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

### Value

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

### See Also

Other monthly statistics: [mon.anomaly](#), [monmax](#), [monmean](#), [monmin](#), [monsd](#), [monsum](#), [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 ncd4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

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

**Arguments**

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

**Value**

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

**See Also**

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

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

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

## create example NetCDF

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



```
nc_close(ncnew)

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

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

## Arguments

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

## Value

prints the content of the infile NetCDF.

## Examples

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("CMSAF_example_file.nc", "m")

unlink("CMSAF_example_file.nc")

```

---

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)
```

## Arguments

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

## Value

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

## Examples

```

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

library(ncdf4)

```

```

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("SIS", "CMSAF_example_file.nc")

unlink("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)
```

## Arguments

<code>var</code>	Name of NetCDF variable (character).
<code>infile1</code>	Filename of first input NetCDF file. This may include the directory (character). The data of <code>infile1</code> 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-neighbour 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

### Value

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

### See Also

Other data manipulation functions: [add\\_grid\\_info](#), [box\\_mergetime](#), [levbox\\_mergetime](#)

### Examples

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("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("SIS", "CMSAF_example_file_1.nc", "CMSAF_example_file_2.nc",
  "CMSAF_example_file_remap.nc")

unlink(c("CMSAF_example_file_1.nc", "CMSAF_example_file_2.nc",
  "CMSAF_example_file_remap.nc"))
```

---

run\_toolbox

*Run the CMSAF toolbox.*

---

## Description

Run the interactive shiny-based toolbox based on the cmsaf package.

## Usage

```
run_toolbox(...)
```

## Arguments

... Arguments to be passed to [runApp](#).

## Examples

```
## Not run: run_toolbox(launch.browser = TRUE)
```

---

seas.anomaly                      *Determine seasonal anomalies.*

---

### Description

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

### Usage

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

### Arguments

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

### Value

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

### See Also

Other seasonal statistics: [seasmean](#), [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 ncd4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("SIS", "CMSAF_example_file.nc",
  "CMSAF_example_file_seas.anomaly.nc")

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

## Arguments

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

**Value**

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

**See Also**

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

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

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



**Description**

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

**Usage**

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

**Arguments**

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

**Value**

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

**See Also**

Other seasonal statistics: [seas.anomaly](#), [seasmean](#), [yseasmax](#), [yseasmean](#), [yseasmin](#), [yseassd](#)

**Examples**

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

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("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("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_seassum.nc")

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_seassum.nc"))

```

---

sellonlatbox

*Select a region by longitude and latitude.*


---

## Description

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

## Usage

```

sellonlatbox(var, infile, outfile, lon1 = -180, lon2 = 180,
  lat1 = -90, lat2 = 90, nc34 = 4, overwrite = FALSE,
  verbose = FALSE)

```

## Arguments

var	Name of NetCDF variable (character).
infile	Filename of input NetCDF file. This may include the directory (character).
outfile	Filename of output NetCDF file. This may include the directory (character).
lon1	Longitude of 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 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 ncd4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

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

**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)
```

**Arguments**

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

**Value**

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

**See Also**

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

**Examples**

```
## 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("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("SIS", c(3, 6), "CMSAF_example_file.nc",
  "CMSAF_example_file_selmon.nc")

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

```

## Arguments

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

## Value

A NetCDF file including the selected time period is written.

**See Also**

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

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("SIS", "2001-01-01", "2002-01-01", "CMSAF_example_file.nc",
  "CMSAF_example_file_selperiod.nc")

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

**Arguments**

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

**Value**

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

**See Also**

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

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

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

## create example NetCDF

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

```

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

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

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_selpoint.nc.csv"))

```

---

selpoint.multi

*Extract data at a multiple points.*


---

## Description

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

## Usage

```
selpoint.multi(var, infile, path, pattern, outpath, lon1, lat1,
  station_names, format = "nc", nc34 = 4, verbose = FALSE)
```

## Arguments

var	Name of NetCDF variable (character).
infile	Filename of input NetCDF file. This may include the directory (character). Infile is not needed if path and pattern are given.
path	Directory of input files (character). Will not be used if infile is given.
pattern	Pattern that all desired files in the 'path' directory have in common (character).
outpath	Directory where output files will be stored (character).
lon1	Longitude vector of desired points (numeric vector). Must have the same length as lat1.
lat1	Latitude vector of desired points (numeric vector). Must have the same length as lon1.
station_names	Optional vector of names, which will be used for the output files (character vector). Must have the same length as lon1 and lat1.



format	Intended output format. Options are nc or csv. Default is nc (character).
nc34	NetCDF version of output file. If nc34 = 3 the output file will be in NetCDFv3 format (numeric). Default output is NetCDFv4.
verbose	logical; if TRUE, progress messages are shown

### Value

For each pair of longitude and latitude coordinates one separate NetCDF or csv file including the selected data is written. The csv files are tested for use in Excel and include four columns (Time ; Data ; Longitude ; Latitude), which are separated by ';'. If station\_names are defined, the output files will be named according to this vector. Otherwise, the output files will be named as selpoint\_longitude\_latitude.format. Already existing files will be overwritten in case that station\_names are given or renamed (e.g., selpoint1\_longitude\_latitude.nc) in case that no station\_names are given.

### See Also

Other selection and removal functions: [extract.level](#), [extract.period](#), [sellonlatbox](#), [selmon](#), [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("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 = "CMSAF_example_file.nc",
  outpath = getwd(), lon1 = c(8, 9), lat1 = c(48, 49),
  station_names = c("A", "B"), format = "csv")

unlink(c("CMSAF_example_file.nc", "A.csv", "B.csv"))
```

---

seltime

*Extract specific timestep.*


---

## Description

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

## Usage

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

## Arguments

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

## Value

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

## See Also

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

**Examples**

```

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

library(ncdf4)

## create some (non-realistic) example data

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

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
  vals = time, unlim = TRUE)
var1 <- 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)

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

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

```

**Arguments**

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

**Value**

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

**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 ncd4 package.
## Alternatively example data can be freely downloaded here:
## <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5,15,0.5)
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 <- ncvdef('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)

## 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('SIS',c(2003,2006),'CMSAF_example_file.nc',
        'CMSAF_example_file_selyear.nc')

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_selyear.nc"))

```

---

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)

```

## Arguments

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

## Value

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

## See Also

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

## Examples

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

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

---

timmean

*Determine all-time mean.*

---

## Description

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

## Usage

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

**Arguments**

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

**Value**

A NetCDF file including the temporal mean is written.

**See Also**

Other temporal operators: [timmax](#), [timmin](#), [timpctl](#), [timsd](#), [timsum](#), [trend](#)

**Examples**

```
## 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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

```
timmean("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_timmean.nc")
unlink(c("CMSAF_example_file.nc", "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)
```

## Arguments

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

## Value

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

## See Also

Other temporal operators: [timmax](#), [timmean](#), [timpctl](#), [timsd](#), [timsum](#), [trend](#)

## Examples

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

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

### 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

**Value**

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

**See Also**

Other temporal operators: `timmax`, `timmean`, `timmin`, `timsd`, `timsdsum`, `trend`

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

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

---

timsd

*Determine all-time standard deviations.*

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

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

**See Also**

Other temporal operators: [timmax](#), [timmean](#), [timmin](#), [timpctl](#), [timsum](#), [trend](#)

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
```

```

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

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

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_timsd.nc"))

```

---

timsum

*Determine all-time sum.*


---

## Description

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

## Usage

```

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

```

## Arguments

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

## Value

A NetCDF file including the temporal sum is written.

## See Also

Other temporal operators: [timmax](#), [timmean](#), [timmin](#), [timpctl](#), [timsd](#), [trend](#)

## Examples

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_timsum.nc")

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

---

trend

*Determine linear trends.*

---

## Description

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

**Usage**

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

**Arguments**

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

**Value**

A NetCDF file including three data layers is written. One layer (trend1) contains the linear trend multiplied by the number of time steps. In older versions of the package ( $\leq 1.7$ ) 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 The significance is calculated from the lower and upper value of the 95 lower or upper value  $< 0$ : sig = 0 (not significant); lower and upper value  $< 0$ : sig = -1 (negative significant); lower and upper value  $> 0$ : sig = 1 (positive significant)

**See Also**

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

**Examples**

```
## 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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the trend of the example CM SAF NetCDF file and write the
## output to a new file.
trend("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_trend.nc")

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_trend.nc"))

```

---

wfldmean

*Determine the weighted field 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)
```

## Arguments

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

## Value

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

**See Also**

Other spatial operators: [fldmax](#), [fldmean](#), [fldmin](#)

**Examples**

```
## 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("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("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_wfldmean.nc")

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

**Arguments**

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

**Value**

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

**See Also**

Other daily statistics: [dayrange](#)

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_ydaymean.nc")

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_ydaymean.nc"))

```

---

year.anomaly	<i>Determine annual anomalies.</i>
--------------	------------------------------------

---

## 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)
```

## Arguments

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

## Value

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

## See Also

Other annual statistics: [yearmean](#), [yearsum](#)

**Examples**

```

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

library(ncdf4)

## create some (non-realistic) example data

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

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
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 annual anomalies of the example CM SAF NetCDF file and
## write the output to a new file.
year.anomaly("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_year.anomaly.nc")

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_year.anomaly.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)

```

**Arguments**

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

**Value**

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

**See Also**

Other annual statistics: [year.anomaly](#), [yearsum](#)

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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 <- ncvdef("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 annual means of the example CM SAF NetCDF file and
## write the output to a new file.
```

```
yearmean("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_yearmean.nc")
unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_yearmean.nc"))
```

---

yearsум	<i>Determine annual sums</i>
---------	------------------------------

---

## Description

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

## Usage

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

## Arguments

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

## Value

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

## See Also

Other annual statistics: [year.anomaly](#), [yearmean](#)

## Examples

```
## 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("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("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_yearsum.nc")

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

## Arguments

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

**Value**

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

**See Also**

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

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

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

**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)
```

**Arguments**

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

**Value**

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

**See Also**

Other monthly statistics: [mon.anomaly](#), [monmax](#), [monmean](#), [monmin](#), [monsd](#), [monsum](#), [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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

unlink(c("CMSAF_example_file.nc", "CMSAF_example_file_ymonmean.nc"))

```

ymonmin

*Determine multi-year monthly minima.***Description**

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

**Usage**

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

**Arguments**

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

**Value**

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

**See Also**

Other monthly statistics: [mon.anomaly](#), [monmax](#), [monmean](#), [monmin](#), [monsd](#), [monsum](#), [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 ncd4 package. Alternatively
## example data can be freely downloaded here: <https://wui.cmsaf.eu/>

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("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("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_ymonmin.nc")

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

**Arguments**

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

**Value**

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

**See Also**

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

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

## Determine the multi-year monthly standard deviation of the example CM
```

```
## SAF NetCDF file and write the output to a new file.
ymonsd("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_ymonsd.nc")

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

## Arguments

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

## Value

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

## See Also

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

## Examples

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

library(ncdf4)

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

```

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

## create example NetCDF

x <- ncdim_def(name = "lon", units = "degrees_east", vals = lon)
y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
  vals = time, unlim = TRUE)
var1 <- ncvar_def("rain", "mm", 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 multi-year monthly sum of the example CM SAF NetCDF
## file and write the output to a new file.
ymonsum("rain", "CMSAF_example_file.nc", "CMSAF_example_file_ymonsum.nc")

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

**Arguments**

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

**Value**

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

**See Also**

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

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

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

**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)
```

**Arguments**

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

**Value**

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

**See Also**

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

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

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

## create example NetCDF

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

```

y <- ncdim_def(name = "lat", units = "degrees_north", vals = lat)
t <- ncdim_def(name = "time", units = "hours since 1983-01-01 00:00:00",
  vals = time, unlim = TRUE)
var1 <- ncvar_def("SIS", "W m-2", list(x, y, t), -1, prec = "short")
vars <- list(var1)
ncnew <- nc_create("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("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_yseasmean.nc")

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

```

## Arguments

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

## Value

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

## See Also

Other seasonal statistics: [seas.anomaly](#), [seasmean](#), [seassum](#), [yseasmax](#), [yseasmean](#), [yseassd](#)



## Examples

```
## 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("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("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_yseasmin.nc")

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

---

yseassd

*Determine multi-year seasonal standard deviations.*

---

## Description

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

## Usage

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

**Arguments**

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

**Value**

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

**See Also**

Other seasonal statistics: [seas.anomaly](#), [seasmean](#), [seassum](#), [yseasmax](#), [yseasmean](#), [yseasmin](#)

**Examples**

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

library(ncdf4)

## create some (non-realistic) example data

lon <- seq(5, 15, 0.5)
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("CMSAF_example_file.nc", vars)
ncvar_put(ncnew, var1, data)
ncatt_put(ncnew, "lon", "standard_name", "longitude", prec = "text")
ncatt_put(ncnew, "lat", "standard_name", "latitude", prec = "text")
nc_close(ncnew)

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

```
yseassd("SIS", "CMSAF_example_file.nc", "CMSAF_example_file_yseassd.nc")
```

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

# Index

- \*Topic **datagen**
    - cmsaf, 7
  - \*Topic **manip**
    - cmsaf, 7
  - \*Topic **package**
    - cmsaf, 7
  - \*Topic **spatial**
    - cmsaf, 7
  - \*Topic **ts**
    - cmsaf, 7
  - \*Topic **univar**
    - cmsaf, 7
- add (cmsaf.add), 9
- add\_grid\_info, 3, 5, 8, 35, 52
- addc (cmsaf.addc), 10
- box\_mergetime, 4, 4, 8, 35, 52
- cat (cmsaf.cat), 12
- change\_att, 6
- cmsaf, 7
- cmsaf-package (cmsaf), 7
- cmsaf.add, 8, 9, 11, 14, 16, 17, 19, 21, 22, 25, 45
- cmsaf.addc, 8, 9, 10, 14, 16, 17, 19, 21, 22, 25, 45
- cmsaf.cat, 8, 12
- cmsaf.div, 8, 9, 11, 13, 16, 17, 19, 21, 22, 25, 45
- cmsaf.divc, 8, 9, 11, 14, 15, 17, 19, 21, 22, 25, 45
- cmsaf.mul, 8, 9, 11, 14, 16, 17, 19, 21, 22, 25, 45
- cmsaf.mulc, 8, 9, 11, 14, 16, 17, 18, 21, 22, 25, 45
- cmsaf.sub, 8, 9, 11, 14, 16, 17, 19, 20, 22, 25, 45
- cmsaf.subc, 8, 9, 11, 14, 16, 17, 19, 21, 22, 25, 45
- dayrange, 8, 23, 81
- div (cmsaf.div), 13
- divc (cmsaf.divc), 15
- divdpm, 8, 9, 11, 14, 16, 17, 19, 21, 22, 24, 45
- extract.level, 8, 26, 28, 59, 60, 62, 63, 65, 66, 68
- extract.period, 8, 27, 28, 59, 60, 62, 63, 65, 66, 68
- fldmax, 8, 29, 31, 32, 80
- fldmean, 8, 30, 30, 32, 80
- fldmin, 8, 30, 31, 32, 80
- get.knnx, 51
- get\_time, 8, 33
- interp.surface.grid, 51
- levbox\_mergetime, 4, 5, 8, 34, 52
- mon.anomaly, 8, 36, 38, 39, 41–43, 46, 48, 87–89, 91, 92
- monmax, 8, 36, 37, 39, 41–43, 46, 48, 87–89, 91, 92
- monmean, 8, 36, 38, 39, 41–43, 46, 48, 87–89, 91, 92
- monmin, 8, 36, 38, 39, 40, 42, 43, 46, 48, 87–89, 91, 92
- monsd, 8, 36, 38, 39, 41, 41, 43, 46, 48, 87–89, 91, 92
- monsum, 8, 36, 38, 39, 41, 42, 43, 46, 48, 87–89, 91, 92
- mul (cmsaf.mul), 17
- mulc (cmsaf.mulc), 18
- muldpm, 8, 9, 11, 14, 16, 17, 19, 21, 22, 25, 44
- multimonmean, 8, 36, 38, 39, 41–43, 46, 48, 87–89, 91, 92
- multimonsum, 8, 36, 38, 39, 41–43, 46, 47, 87–89, 91, 92

- ncdf4-package, 7
- ncinfo, 8, 49
  
- read\_ncvar, 8, 50
- remap, 4, 5, 8, 35, 51
- remapcon, 51
- run\_toolbox, 8, 53
- runApp, 53
  
- seas.anomaly, 8, 54, 56, 57, 94–96, 98
- seasmean, 8, 54, 55, 57, 94–96, 98
- seassum, 8, 54, 56, 56, 94–96, 98
- sellonlatbox, 8, 27, 28, 58, 60, 62, 63, 65, 66, 68
- selmon, 8, 27, 28, 59, 59, 62, 63, 65, 66, 68
- selperiod, 8, 27, 28, 59, 60, 61, 63, 65, 66, 68
- selpoint, 8, 27, 28, 59, 60, 62, 62, 65, 66, 68
- selpoint.multi, 8, 27, 28, 59, 60, 62, 63, 64, 66, 68
- selttime, 8, 27, 28, 59, 60, 62, 63, 65, 66, 68
- selyear, 8, 27, 28, 59, 60, 62, 63, 65, 66, 67
- sub (cmsaf.sub), 20
- subc (cmsaf.subc), 22
  
- timmax, 8, 69, 71, 72, 74–76, 78
- timmean, 8, 69, 70, 72, 74–76, 78
- timmin, 8, 69, 71, 72, 74–76, 78
- timpctl, 8, 69, 71, 72, 73, 75, 76, 78
- timsd, 8, 69, 71, 72, 74, 74, 76, 78
- tisum, 8, 69, 71, 72, 74, 75, 76, 78
- trend, 8, 69, 71, 72, 74–76, 77
  
- wfldmean, 8, 30–32, 79
  
- ydaymean, 8, 24, 80
- year.anomaly, 8, 82, 84, 85
- yearmean, 8, 82, 83, 85
- yearsum, 8, 82, 84, 85
- ymonmax, 8, 36, 38, 39, 41–43, 46, 48, 86, 88, 89, 91, 92
- ymonmean, 8, 36, 38, 39, 41–43, 46, 48, 87, 87, 89, 91, 92
- ymonmin, 8, 36, 38, 39, 41–43, 46, 48, 87, 88, 89, 91, 92
- ymonsd, 8, 36, 38, 39, 41–43, 46, 48, 87–89, 90, 92
- ymonsum, 8, 36, 38, 39, 41–43, 46, 48, 87–89, 91, 92
- yseasmax, 8, 54, 56, 57, 93, 95, 96, 98
- yseasmean, 8, 54, 56, 57, 94, 94, 96, 98
- yseasmin, 8, 54, 56, 57, 94, 95, 96, 98
- yseassd, 8, 54, 56, 57, 94–96, 97