Package 'INQC'

January 20, 2025

Title Quality Control of Climatological Daily Time Series

Version 2.0.5

Date 2021-05-19

Description Collection of functions for quality control (QC) of climatological daily time series (e.g. the ECA&D station data).

License GPL (>= 3)

URL https://github.com/INDECIS-Project/INQC

BugReports https://github.com/INDECIS-Project/INQC/issues

Encoding UTF-8

RoxygenNote 7.1.1

Imports evd, gdata, suncalc, stats, utils

Suggests readr

NeedsCompilation no

Author Enric Aguilar [aut, cre] (<https://orcid.org/0000-0002-8384-377X>), Oleg Skrynyk [aut] (<https://orcid.org/0000-0001-8827-0280>)

Maintainer Enric Aguilar <enric.aguilar@urv.cat>

Repository CRAN

Date/Publication 2021-05-24 14:00:02 UTC

Contents

`climdex2ecad` *Converter from the ClimDex format into the ECA&D format (blended version)*

Description

This function will convert station and data files in ClimDex format into corresponding station and data files in the ECA&D format (blended version)

Usage

```
climdex2ecad(
  homefolder = "./",
  stationlist = "stations.csv",
  countrycode = "DE"
)
```

Arguments

homefolder	path to the home directory which should contain the subdirectory 'raw_ClimDex' with files in the ClimDex format
stationlist	list (as 'csv'-file) of climatological stations to be considered. Each line should be in the format: lat (as dec. degree), lon (as dec. degree), height, staname
countrycode	two character country code

Value

station and data files in the ECA&D format stored in the subdirectory 'raw'

Examples

```
#Set a temporal working directory:
wd <- tempdir(); wd0 <- setwd(wd)
#Create subdirectory where raw data files in the ClimDex format have to be located
dir.create(file.path(wd, "raw_ClimDex"))
#Extract the ClimDex data and station files from the example data folder
path2stalist<-system.file("extdata", "stations.csv", package = "INQC")
stalist<-readr::read_lines_raw(path2stalist)
readr::write_lines(stalist,file=paste0(wd,"/raw_ClimDex/stations.csv"))
path2data1<-system.file("extdata", "Deuselbach.txt", package = "INQC")
data1<-readr::read_lines_raw(path2data1)
readr::write_lines(data1, file=paste0(wd,"/raw_ClimDex/Deuselbach.txt"))
path2data2<-system.file("extdata", "Staname.txt", package = "INQC")
data2<-readr::read_lines_raw(path2data2)
readr::write_lines(data2, file=paste0(wd,"/raw_ClimDex/Staname.txt"))
#Call the converter
climdex2ecad(homefolder = "./",stationlist = "stations.csv",countrycode = "DE")
#The results can be found in the directory:
print(wd)
#Return to user's working directory:
setwd(wd0)
```

---

clocov *QC for Cloud Cover (CC)*

Description

This function will centralize temperature-like QC routines. It will create a file in the folder QC with an additional 0/1 column, where "1" means test failed.

Usage

```
clocov(
  element = "CC",
  maxseq = 8,
  blocksizeround = 20,
  blockmanymonth = 20,
  blockmanyyear = 200,
  inisia = FALSE
)
```

Arguments

element	two-letters ECA&D code for the element (CC for cloud cover)
maxseq	maximum number of consecutive repeated values, FUNCTION: flat (8,8,8 would be 3 consecutive values).
blocksizeround	maximum number of values in a month with the same decimal, FUNCTION: rounding
blockmanymonth	maximum number of equal values in a month, FUNCTION: toomany
blockmanyyear	maximum number of equal values in a year, FUNCTION: toomany
inisia	a logical flag. If it is TRUE inithome() will be called

Value

QC results for CC

Examples

```
#Set a temporal working directory:
wd <- tempdir()
wd0 <- setwd(wd)
#Create subdirectory where raw data files have to be located
dir.create(file.path(wd, 'raw'))
options("homefolder"='./'); options("blend"=FALSE)
#Extract the ECA&D data and station files from the example data folder
path2cclist<-system.file("extdata", "ECA_blend_source_cc.txt", package = "INQC")
cclist<-readr::read_lines_raw(path2cclist)
readr::write_lines(cclist,'ECA_blend_source_cc.txt')
path2ccdata<-system.file("extdata", "CC_SOUID132727.txt", package = "INQC")
ccdata<-readr::read_lines_raw(path2ccdata)
readr::write_lines(ccdata, file=paste(wd,'/raw/CC_SOUID132727.txt',sep=''))
#Perform QC of Cloud Cover data
clocov(inisia=TRUE)
#Remove some temporary files
list = list.files(pattern = "Rfwf")
file.remove(list)
#Return to user's working directory:
setwd(wd0)
#The QC results can be found in the directory:
print(wd)
```

computecal *Prepares a calendar frame*

Description

This function prepares a calendar frame and returns it as year,month,day (i.e., the 3 first columns of the RClimdex format)

Usage

```
computecal(fy, ly)
```

Arguments

fy	first year to work with (past)
ly	last year to work with (present)

Value

3 columns containing year,month,day

Examples

```
fy<-1981
ly<-2020
clndr<-computecal(fy,ly)
```

consolidator *Consolidates QC files*

Description

This function is not intended to be called as a stand-alone function. It is automatically called each time a file ends its QC. It will write the quality control files. One file will be placed in a subfolder of the homefolder named QCConslidated. It will use the exact ECA&D format (date, value, QC flag). The QC flags include:

0: Passed QC; 1: ERROR; 2: ALMOST CERTAIN, ERROR; 3: OUTLIER, SUSPECT; 4: COLLECTIVELY SUSPECT; 9: Missing value.

A second file is placed in the subfolder QC and includes all date, value and a column for each QC test ran over this file. Values passing/not passing QC are labelled with 0/1. A third file summarizes the number of values falling on each category (0,1,2,3,4,9) and the number of values failing each test

Usage

```
consolidator(filename, x)
```

Arguments

filename	ECA&D file name, expressed as VV_SOUIDXXXXXX.txt, where "VV" is the two-letters variable code, "SOUID" is literal, XXXXXX is the ECA&D SOUID code and ".txt" is literal
x	QCed series, formatted as date, value, QC flag

Value

It does not return any value. Each time when called, it will create three files: Summary file, placed at ./QCSumamry/SummaryVV_SOUIDXXXXXX.txt; QC consolidated file, placed at ./QCConsolidated/VV_SOUIDXXXXXX.txt; Verbose QC file, placed at ./QC/qc_VV_SOUIDXXXXXX.txt.

---

cost2ecad *Converter from the COST Home format into the ECA&D format (blended version)*

Description

This function will convert station and data files in COST Home format into corresponding station and data files in the ECA&D format (blended version)

Usage

```
cost2ecad(homefolder = "./")
```

Arguments

homefolder	path to the home directory which should contain the subdirectory 'raw_COST' with files in the COST Home format. Files of all variables must be stored in 'raw_COST'

Value

station and data files in the ECA&D format stored in the subdirectory 'raw'

Examples

```
#Set a temporal working directory:
wd <- tempdir(); wd0 <- setwd(wd)
#Create subdirectory where raw data files in the COST format have to be located
dir.create(file.path(wd, "raw_COST"))
#TG: Extract the COST data and station files from the example data folder
path2tglist<-system.file("extdata", "000001stations.txt", package = "INQC")
tglist<-readr::read_lines_raw(path2tglist)
readr::write_lines(tglist,file=paste0(wd,"/raw_COST/000001stations.txt"))
path2tgdata1<-system.file("extdata", "ratmd00000001d.txt", package = "INQC")
tgdata1<-readr::read_lines_raw(path2tgdata1)
readr::write_lines(tgdata1, file=paste0(wd,"/raw_COST/ratmd00000001d.txt"))
```

```
path2tgdata2<-system.file("extdata", "ratmd00000005d.txt", package = "INQC")
tgdata2<-readr::read_lines_raw(path2tgdata2)
readr::write_lines(tgdata2, file=paste0(wd,"/raw_COST/ratmd00000005d.txt"))
#PP: Extract the COST data and station files from the example data folder
path2pplist<-system.file("extdata", "000002stations.txt", package = "INQC")
pplist<-readr::read_lines_raw(path2pplist)
readr::write_lines(pplist,file=paste0(wd,"/raw_COST/000002stations.txt"))
path2ppdata1<-system.file("extdata", "rappd00000001d.txt", package = "INQC")
ppdata1<-readr::read_lines_raw(path2ppdata1)
readr::write_lines(ppdata1, file=paste0(wd,"/raw_COST/rappd00000001d.txt"))
path2ppdata2<-system.file("extdata", "rappd00000012d.txt", package = "INQC")
ppdata2<-readr::read_lines_raw(path2ppdata2)
readr::write_lines(ppdata2, file=paste0(wd,"/raw_COST/rappd00000012d.txt"))
#Call the converter
cost2ecad(homefolder = "./")
#The results can be found in the directory:
print(wd)
#Return to user's working directory:
setwd(wd0)
```

`decimaldegrees` *Converter for geographical coordinates from the ECA&D format into decimal degrees*

Description

This function takes sexagesimal degrees in the ECA&D format and converts them into decimal degrees. Initial idea was taken from: https://modtools.wordpress.com/2013/09/25/dms2dec/

Usage

```
decimaldegrees(dms, sep = ":")
```

Arguments

`dms`	ONE ELEMENT from the LAT or LON field in ECA&D listings
`sep`	the separator between elements, in ECA&D ":"

Value

geographical coordinates (latitude or longitude) in decimal degrees

Examples

```
dms<-'+48:03:00'
dec<-decimaldegrees(dms)

dms<-'-015:03:00'
dec<-decimaldegrees(dms)
```

`dostats` *Create QC statistical summary*

Description

This function creates two report files (Mystats.txt and CasesSummary.txt) with a statistical summary of QCs performed over the whole data set

Usage

```
dostats()
```

Value

files with QC summary

`downloadator` *Downloads the latest version of blended data from the ECA&D website*

Description

This function will use the default or specified links to download one or several files from ECA&D and place them for their use with INQC. For each variable a data file and a station file will/should be specified.

Usage

```
downloadator(
  homefolder = "../ecad_updated",
 tx = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_nonblend_tx.zip",
 tx2 = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_blend_source_tx.txt",
 tn = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_nonblend_tn.zip",
 tn2 = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_blend_source_tn.txt",
 tg = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_nonblend_tg.zip",
 tg2 = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_blend_source_tg.txt",
 sd = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_nonblend_sd.zip",
 sd2 = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_blend_source_sd.txt",
 ss = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_nonblend_ss.zip",
 ss2 = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_blend_source_ss.txt",
 rr = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_nonblend_rr.zip",
 rr2 = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_blend_source_rr.txt",
 pp = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_nonblend_pp.zip",
 pp2 = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_blend_source_rr.txt",
 cc = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_nonblend_cc.zip",
```

```
 cc2 = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_blend_source_cc.txt",
 hu = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_nonblend_hu.zip",
 hu2 = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_blend_source_hu.txt",
 fg = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_nonblend_fg.zip",
 fg2 = "http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_blend_source_fg.txt"
)
```

Arguments

homefolder	full path to local folder in the form './homefolder'. The function will store there the station files and create ./homefolder/raw and will store there the data
tx	link to download daily maximum temperature or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
tx2	link to download daily maximum temperatures station list or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
tn	link to download daily minimum temperature or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
tn2	link to download daily minimum temperature station list or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
tg	link to download daily average temperature or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
tg2	link to download daily average temperature station list or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
sd	link to download daily snow depth or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
sd2	link to download daily snow depth station list or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
ss	link to download daily sunshine duration or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
ss2	link to download daily sunshine duration station list or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
rr	link to download daily rainfall or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
rr2	link to download daily rainfall station list or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
pp	link to download daily sea level pressure or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
pp2	link to download daily sea level pressure station list or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
cc	link to download daily cloud coverage or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
cc2	link to download daily cloud coverage station list or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.

hu	link to download daily relative humidity or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
hu2	link to download daily relative humidity station list or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
fg	link to download daily wind speed or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.
fg2	link to download daily wind speed station list or NULL. Default set to working ECA&D link, as of 22/12/2020. Provided link MUST exist.

Value

For each valid link, the corresponding file will be downloaded. Data files will be unzipped to the ./raw folder (as requested by INQC) and station files will be stored at the specified homefolder

Examples

```
## Not run:
#Set a temporal working directory:
wd <- tempdir()
wd0 <- setwd(wd)
#Please note, the command below might take a while and will download the ECA&D data
#with a size more than 0.5GB
downloadator('./data',
             tx=NULL,
             tx2=NULL,
             tn=NULL,
             tn2=NULL,
             tg=NULL,
             tg2=NULL,
             sd=NULL,
             sd2=NULL,
         ss='http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_nonblend_ss.zip',
         ss2="http://knmi-ecad-assets-prd.s3.amazonaws.com/download/ECA_blend_source_ss.txt",
             rr=NULL,
             rr2=NULL,
             pp=NULL,
             pp2=NULL,
             cc=NULL,
             cc2=NULL,
             hu=NULL,
             hu2=NULL,
             fg=NULL,
             fg2=NULL)
#Delete the downloaded archive (the zip-file)
file.remove(paste(wd,"/data/raw/","ss.zip",sep=""))
#Return to user's working directory:
setwd(wd0)
#The downloaded files can be found in directory:
print(wd)

## End(Not run)
```

drywetlong *Detects wet/dry long periods*

Description

This function detects episodes of too many consecutive wet or dry days

Usage

```
drywetlong(x, ret = 300, sueco = 9.9, dry = TRUE, wet = TRUE)
```

Arguments

x	vector with values
ret	pseudo-return period (pareto-based) to compute the maximum tolerable spell
sueco	threshold for dividing dry and wet. This is useful to label other binary sequences, e.g. for 0 radiation. Now it is <= and >, instead of < and >=
dry	if set to TRUE, dry sequences are sent to result; if FALSE, omitted
wet	same as previous, for wet sequences

Value

list of positions in the input data time series which do not pass QC test

Examples

```
#Extract the ECA&D data file from the example data folder
path2inptfl<-system.file("extdata", "RR_SOUID132730.txt", package = "INQC")
#Read the data file
x<-readecad(input=path2inptfl,missing= -9999)[,4:4]
#Find all suspicious positions in the precipitation time series
drywetlong(x,ret=300,sueco=9.9,dry=TRUE,wet=TRUE)

#Introduce the long wet period
x[1:600]<-10
#Find all suspicious positions in the precipitation time series
drywetlong(x,ret=300,sueco=9.9,dry=TRUE,wet=TRUE)
```

duplas *Detects duplicated dates*

Description

This function detects duplicated dates in the input time series

Usage

```
duplas(x)
```

Arguments

x	vector of dates in the ECA&D format (YYYYMMDD)

Value

vector with the list of positions which do not pass this QC test. If all positions pass the test, returns NULL

Examples

```
#Extract the ECA&D data file from the example data folder
path2inptfl<-system.file("extdata", "TX_SOUID132734.txt", package = "INQC")
#Read the data file
x<-readecad(input=path2inptfl,missing= -9999)[,3]
#Find all duplicated dates in the time series
duplas(x)

#Introduce the duplicated dates
x[31]<-'19610130'
#Find all duplicated dates in the time series
duplas(x)
```

flat *Flat sequences*

Description

This function detects consecutive equal values (e.g., 15.1, 15.1, 15.1, 15.1...) in a data time series. Also can be used to detect consecutive equal decimal part of the values (e.g., 15.1, 12.1, 13.1, 10.1 ...)

Usage

```
flat(y, maxseq, exclude = NULL)
```

Arguments

y	data vector
maxseq	the maximum number of contiguous repetitions of a value (e.g., if 3, sequences of 4 will be flagged)
exclude	values to be excluded. This is useful for variables where a single value is expected to repeat many times, e.g. 0.0 in precipitation.

Value

list of positions which do not pass this QC test. If all positions pass the test, returns NULL

Examples

```
y<-rnorm(100)
y[10:20]<-10
flat(y,5)

#Extract the ECA&D data file from the example data folder
path2inptfl<-system.file("extdata", "TX_SOUID132734.txt", package = "INQC")
#Read the data file
y<-readecad(input=path2inptfl,missing= -9999)[,4]
#Find all consecutive (with a length > 5 elements) equal values in the time series
flat(y,5)

#Introduce the duplicated dates
y[6:12]<-10
#Find all consecutive (with a length > 5 elements) equal values in the time series
flat(y,5)
```

---

flatsun *Flat sequences for sunshine duration (only for "non-blended" ECA&D data)*

Description

This function uses flat() and modifies it with "smart" comparison with clouds. If close to 8 and close to 0 clouds, allowed; if close to maxsundur and clouds near 0, allowed

Usage

```
flatsun(x, maxseq, id, modonube = FALSE)
```

Arguments

x	data.frame date/value (need dates in this implementation of flat)
maxseq	maximum number of contiguous repetitions of a value (e.g., if 3, sequences of 4 will be flagged)

id	name of a file ("SS_SOUIDxxxxxx.txt", non-blended) with sunshine data (see parameter x) to be checked
modonube	logical flag. If FALSE (the default), the "sun" mode of the function is used. If TRUE, the "cloud" mode is used

Value

list of positions which do not pass this QC test

Examples

```
#Set a temporal working directory:
wd <- tempdir(); wd0 <- setwd(wd)
#Create subdirectory where raw data files have to be located
dir.create(file.path(wd, 'raw'))
#Extract the non-blended ECA&D data and station files from the example data folder
path2cclist<-system.file("extdata", "ECA_blend_source_cc.txt", package = "INQC")
cclist<-readr::read_lines_raw(path2cclist)
readr::write_lines(cclist,'ECA_blend_source_cc.txt')
path2ccdata<-system.file("extdata", "CC_SOUID132727.txt", package = "INQC")
ccdata<-readr::read_lines_raw(path2ccdata)
readr::write_lines(ccdata, file=paste(wd,'/raw/CC_SOUID132727.txt',sep=''))
path2sslist<-system.file("extdata", "ECA_blend_source_ss.txt", package = "INQC")
sslist<-readr::read_lines_raw(path2sslist)
readr::write_lines(sslist,'ECA_blend_source_ss.txt')
path2ssdata<-system.file("extdata", "SS_SOUID132728.txt", package = "INQC")
ssdata<-readr::read_lines_raw(path2ssdata)
readr::write_lines(ssdata, file=paste(wd,'/raw/SS_SOUID132728.txt',sep=''))
#Read the sunshine data
x<-readecad(input=path2ssdata,missing= -9999)[,3:4]
options("homefolder"='./'); options("blend"=FALSE)
listonator(check=TRUE)
#Call flatsun()
flatsun(x,5,"SS_SOUID132728.txt",modonube=FALSE)
#Introduce error values in the sunshine data
x[1:10,2]<-10
#Call flatsun()
flatsun(x,5,"SS_SOUID132728.txt",modonube=FALSE)
#Return to user's working directory:
setwd(wd0)
```

---

inithome *Creates necessary folders (if not exist)*

Description

This function will checks if all necessary folders ('QCSummary, QC and QCConsolidated) exist and if not, creates them. Not intended as a stand-alone function. Called from other routines.

Usage

```
inithome()
```

Value

it does not return any values, just creates the described folders if they do not exist

---

inqc *Wrapper for QC'ing all variables*

Description

This function calls functions which perform QC for all climate variables

Usage

```
inqc(homefolder = "./", blend = TRUE)
```

Arguments

homefolder	path to the homefolder, as string
blend	logical flag which means performing (if TRUE) QC on blended time series

Value

QC results, in both formats (verbose and workable file in exact ECA&D format)

Examples

```
#Set a temporal working directory:
wd <- tempdir(); wd0 <- setwd(wd)
#Create subdirectory where raw data files have to be located
dir.create(file.path(wd, 'raw'))
#NON-BLENDED ECA&D SERIES
#Extract the non-blended ECA&D data and station files from the example data folder
#Only TX (maximum air temperature) and CC (cloud cover) data are used in the example
path2txlist<-system.file("extdata", "ECA_blend_source_tx.txt", package = "INQC")
txlist<-readr::read_lines_raw(path2txlist)
readr::write_lines(txlist,'ECA_blend_source_tx.txt')
path2txdata<-system.file("extdata", "TX_SOUID132734.txt", package = "INQC")
txdata<-readr::read_lines_raw(path2txdata)
readr::write_lines(txdata, file=paste(wd,'/raw/TX_SOUID132734.txt',sep=''))
path2cclist<-system.file("extdata", "ECA_blend_source_cc.txt", package = "INQC")
cclist<-readr::read_lines_raw(path2cclist)
readr::write_lines(cclist,'ECA_blend_source_cc.txt')
path2ccdata<-system.file("extdata", "CC_SOUID132727.txt", package = "INQC")
ccdata<-readr::read_lines_raw(path2ccdata)
readr::write_lines(ccdata, file=paste(wd,'/raw/CC_SOUID132727.txt',sep=''))
#This is the MAIN starting point of the INQC software calculation:
```

```
inqc(homefolder='./',blend=FALSE) #Work with non-blended ECA&D data
#Remove some temporary files
list = list.files(pattern = "Rfwf")
file.remove(list)
#The QC results can be found in the directory:
print(wd)
#BLENDED ECA&D SERIES
#Extract the blended ECA&D data and station files from the example data folder
#Only TX (maximum air temperature) and CC (cloud cover) data are used in the example
path2list<-system.file("extdata", "stations.txt", package = "INQC")
list<-readr::read_lines_raw(path2list)
readr::write_lines(list,file=paste(wd,'/raw/stations.txt',sep=''))
path2txdata<-system.file("extdata", "TX_STAID000002.txt", package = "INQC")
txdata<-readr::read_lines_raw(path2txdata)
readr::write_lines(txdata, file=paste(wd,'/raw/TX_STAID000002.txt',sep=''))
path2ccdata<-system.file("extdata", "CC_STAID000001.txt", package = "INQC")
ccdata<-readr::read_lines_raw(path2ccdata)
readr::write_lines(ccdata, file=paste(wd,'/raw/CC_STAID000001.txt',sep=''))
#This is the MAIN starting point of the INQC software calculation:
inqc(homefolder='./',blend=TRUE) #work with blended ECA&D data
#Remove some temporary files
list = list.files(pattern = "Rfwf")
file.remove(list)
#The QC results can be found in the directory:
print(wd)
#Return to user's working directory:
setwd(wd0)
```

`IQRoutliers` *Computes outliers*

Description

This function computes outliers centralized around a day, using a number of days around it

Usage

```
IQRoutliers(date, value, level = 3, window = 11, exclude = NULL)
```

Arguments

`date`	vector with dates
`value`	vector with data values
`level`	number of IQRs to be added to percentile 75 and subtracted to percentile 25 to determinate the tolerance interval. Values outside this interval, will be declared as outliers.
`window`	number of days to be considered (including the target)
`exclude`	if it is not null, the code will exclude this value from the analysis (i.e., good to exclude 0 for precipitation)

Value

positions which do not pass this QC test

Examples

```
#Extract the ECA&D data file from the example data folder
path2inptfl<-system.file("extdata", "TX_SOUID132734.txt", package = "INQC")
#Read the data file
date<-readecad(input=path2inptfl,missing= -9999)[,3]
value<-readecad(input=path2inptfl,missing= -9999)[,4]
#Find all suspicious positions in the time series
IQRoutliers(date,value,level=3,window=11,exclude=NULL)
```

---

jumps2 *Labels interdiurnal large differences*

Description

This function looks for interdiurnal differences considered too large (larger than a threshold value). The threshold can be defined by two different ways: (1) as an absolute value, the same for all differences. It is specified directly through the parameter 'force' (see below); (2) as a quantile in a probability distribution of the interdiurnal differences (built for each month separately). In this case, the threshold is specified indirectly through the parameter 'quanty' (see below). The calculated threshold quantiles can be also modified (increased/decreased) by means of the parameter 'times' (see below).

Consequently, jumps2() can be used in two different modes: 'absolute' and 'quantile'

Usage

```
jumps2(date, value, quanty = 0.999, times = 1, force = NULL)
```

Arguments

date	vector of dates
value	vector of values
quanty	threshold quantile rank (cumulative probability) to define corresponding quantiles in the distributions of the interdiurnal differences (for each month separately)
times	factor to modify (increase/decrease) the threshold quantile values
force	value of threshold for daily value differences to be forced

Value

list of positions which do not pass this QC test

Examples

```
#Extract the ECA&D data file from the example data folder
path2inptfl<-system.file("extdata", "TX_SOUID132734.txt", package = "INQC")
#Read the data file
date<-readecad(input=path2inptfl,missing= -9999)[,3]
value<-readecad(input=path2inptfl,missing= -9999)[,4]
#Find all suspicious positions in the time series (in 'quantile' mode)
jumps2(date,value,quanty=0.999,times=1)
#Find all suspicious positions in the time series (in 'absolute' mode)
jumps2(date,value,force=100)
```

---

listas *Creates listings for stations ('non-blended' case) linking STAID and SOUID*

Description

This function takes all the elements and rbinds them into a single list to process

Usage

```
listas(country = "all", name = "allstations.txt")
```

Arguments

country	country for which the list is created. If 'all', no country filter.
name	output file name, do not touch, default is always good.

Value

data frame and the list file containing all stations for all elements, linking STAID and SOUID and metadata

Examples

```
#Set a temporal working directory:
wd <- tempdir(); wd0 <- setwd(wd)
#Extract the non-blended ECA&D station files from the example data folder
#Only TX (maximum air temperature) and CC (cloud cover) variables are used in the example
path2txlist<-system.file("extdata", "ECA_blend_source_tx.txt", package = "INQC")
txlist<-readr::read_lines_raw(path2txlist)
readr::write_lines(txlist,'ECA_blend_source_tx.txt')
path2cclist<-system.file("extdata", "ECA_blend_source_cc.txt", package = "INQC")
cclist<-readr::read_lines_raw(path2cclist)
readr::write_lines(cclist,'ECA_blend_source_cc.txt')
options("homefolder"='./')
liston.nb<-listas(country='all',name='allstations.txt')
#The created list file can be found in the directory:
print(wd)
```

```
#Return to user's working directory:
setwd(wd0)
```

`lister` *Creates a list of blended/non-bladed files for some climate variable*

Description

This function creates a list of blended or non-bladed files containing data of a specified element to be QCed.

Usage

```
lister(element)
```

Arguments

`element`	climatological element (defined by means of two letters, i.e. 'TX')

Value

list of blended or non-bladed files to be QCed

Examples

```
#Set a temporal working directory:
wd <- tempdir(); wd0 <- setwd(wd)
#Create subdirectory where a station file has to be located
dir.create(file.path(wd, 'raw'))
#NON-BLENDED ECA&D SERIES
#Extract the non-blended ECA&D data and station files from the example data folder
#Only TX (maximum air temperature) and CC (cloud cover) data are used in the example
path2txdata<-system.file("extdata", "TX_SOUID132734.txt", package = "INQC")
txdata<-readr::read_lines_raw(path2txdata)
readr::write_lines(txdata, file=paste(wd,'/raw/TX_SOUID132734.txt',sep=''))
path2ccdata<-system.file("extdata", "CC_SOUID132727.txt", package = "INQC")
ccdata<-readr::read_lines_raw(path2ccdata)
readr::write_lines(ccdata, file=paste(wd,'/raw/CC_SOUID132727.txt',sep=''))
options("homefolder"='./'); options("blend"=FALSE)
list.nb<-lister('TX')
#BLENDED ECA&D SERIES
#Extract the blended ECA&D data and station files from the example data folder
#Only TX (maximum air temperature) and CC (cloud cover) data are used in the example
path2txdata<-system.file("extdata", "TX_STAID000002.txt", package = "INQC")
txdata<-readr::read_lines_raw(path2txdata)
readr::write_lines(txdata, file=paste(wd,'/raw/TX_STAID000002.txt',sep=''))
path2ccdata<-system.file("extdata", "CC_STAID000001.txt", package = "INQC")
ccdata<-readr::read_lines_raw(path2ccdata)
readr::write_lines(ccdata, file=paste(wd,'/raw/CC_STAID000001.txt',sep=''))
options("blend"=TRUE)
```

```
list.b<-lister('TX')
#Return to user's working directory:
setwd(wd0)
```

`listonator` *Creates a list (as 'Global' variable) of stations to be QCed.*

Description

This function creates a list (and makes it 'Global' variable) of stations to be QCed. It can be 'blended' or 'non-blended' stations. Geographical coordinates are converted into decimal degrees

Usage

```
listonator(check = TRUE)
```

Arguments

check	logical parameter TRUE/FALSE. If check=TRUE a list of stations is created.

Value

list of stations to be QCed

Examples

```
#Set a temporal working directory:
wd <- tempdir(); wd0 <- setwd(wd)
#NON-BLENDED ECA&D SERIES
#Extract the non-blended ECA&D station files from the example data folder
#Only TX (maximum air temperature) and CC (cloud cover) variables are used in the example
path2txlist<-system.file("extdata", "ECA_blend_source_tx.txt", package = "INQC")
txlist<-readr::read_lines_raw(path2txlist)
readr::write_lines(txlist,'ECA_blend_source_tx.txt')
path2cclist<-system.file("extdata", "ECA_blend_source_cc.txt", package = "INQC")
cclist<-readr::read_lines_raw(path2cclist)
readr::write_lines(cclist,'ECA_blend_source_cc.txt')
options("homefolder"='./'); options("blend"=FALSE)
listonator(check=TRUE)
liston.nb<-getOption("liston")
#BLENDED ECA&D SERIES
#Create subdirectory where a station file has to be located
dir.create(file.path(wd, 'raw'))
#Extract the blended ECA&D station file from the example data folder
path2list<-system.file("extdata", "stations.txt", package = "INQC")
list<-readr::read_lines_raw(path2list)
readr::write_lines(list,file=paste(wd,'/raw/stations.txt',sep=''))
options("blend"=TRUE)
listonator(check=TRUE)
liston.b<-getOption("liston")
```

```
#Return to user's working directory:
setwd(wd0)
```

`newfriki` *Isolates values which are not continuous in the distribution*

Description

The function isolates extreme values which are not continuous in the distribution. If the gap is larger (or smaller) than a pre-set big margin, the values above (or below) are flagged

Usage

```
newfriki(date, value, margina = 0.999, times = 2)
```

Arguments

`date`	vector of dates with the ECA&D format yyyymmdd
`value`	vector of data values
`margina`	tolerance margin, expressed as quantile of the differences
`times`	multiplier for the tolerance margin. Intended usage is to run this twice. Once with times = 1 and flag values as suspect; once with times = 2 and flag as error

Value

positions which do not pass this QC test

Examples

```
#Extract the ECA&D data file from the example data folder
path2inptfl<-system.file("extdata", "TX_SOUID132734.txt", package = "INQC")
#Read the data file
date<-readecad(input=path2inptfl,missing= -9999)[,3]
value<-readecad(input=path2inptfl,missing= -9999)[,4]
#Find all suspicious positions in the time series
newfriki(date,value,margina=0.999,times=1)
```

paretogadget *Finds outliers*

Description

This function finds outliers for variables which can be described/evaluated by means of the Pareto distribution (e.g. atmospheric precipitation or wind speed)

Usage

```
paretogadget(x, ret)
```

Arguments

x	vector of values (a series) to be analyzed
ret	pseudo-return period for the pot-pareto distribution approach

Value

list of positions which do not pass this QC test (which can be considered as outliers)

Examples

```
#Extract the ECA&D precipitation data file from the example data folder
path2inptfl<-system.file("extdata", "RR_SOUID132730.txt", package = "INQC")
#Read the data file
x<-readecad(input=path2inptfl,missing= -9999)[,4]
#Find all suspicious positions in the time series corresponding to the requested return period
paretogadget(x,25)
#Suspicious values
x[paretogadget(x,25)]
```

physics *Isolates anomalous values*

Description

Given a data vector, the function will compare the values to the specified threshold

Usage

```
physics(x, nyu = 0, compare = 1)
```

Arguments

x	data vector
nyu	threshold, numeric
compare	logical operation to apply over the threshold. 1: larger; 2: larger or equal; 3: smaller; 4: smaller or equal; 5 equal

Value

list of positions which do not pass this QC test. If all positions pass the test, returns NULL

Examples

```
x<-rnorm(100)
x[10]<-100
physics(x,5,1)
```

---

potpareto *Peaks over threshold modelling*

Description

This function fits the Generalized Pareto distribution for exeedances over a threshold

Usage

```
potpareto(y, thres = 0.99)
```

Arguments

y	vector of values (a series) to be analyzed
thres	threshold value of probability to define a corresponding threshold percentile

Value

list containing results of modelling/fitting the generalized Pareto distribution

Examples

```
#Extract the ECA&D precipitation data file from the example data folder
path2inptfl<-system.file("extdata", "RR_SOUID132730.txt", package = "INQC")
#Read the data file
y<-readecad(input=path2inptfl,missing= -9999)[,4]
#Fit the Generalized Pareto distribution
pato<-potpareto(y)
#The parameters of the fitted distribution:
location<-pato$threshold
shape<-pato$estimate[2]
scale<-pato$estimate[1]
print(c(location,shape,scale))
```

precip *QC for Atmospheric Precipitation (RR)*

Description

This function will centralize precipitation-like QC routines. It will create a file in the folder QC with an additional 0/1 column, where "1" means test failed.

Usage

```
precip(
  element = "RR",
  large = 5000,
  small = 0,
  ret = 500,
  retornoracha = 500,
  margin = 20,
  friki = 150,
  blocksizeround = 20,
  blockmanymonth = 15,
  blockmanyyear = 180,
  limit = 1500,
  tolerance = 8,
  maxseq = 3,
  roundmax = 10,
  level = 15,
  window = 30,
  margina = 0.999,
  inisia = FALSE
)
```

Arguments

element	two-letters ECA&D code for the element (RR for precipitation)
large	value above which the observation is considered physically impossible for the region
small	value below which the observation is considered physically impossible for the region
ret	pseudo-return period for the pareto outliers
retornoracha	return period for the calculation of the maximum dry and wet spell
margin	frequency difference between consecutive values for repeatedvalue()
friki	minimum value to be considered by repeatedvalue()
blocksizeround	maximum number of repeated values with the same decimal, FUNCTION: roundprecip()
blockmanymonth	maximum number of equal values in a month, FUNCTION: toomany()

blockmanyyear	maximum number of equal values in a year, FUNCTION: toomany()
limit	cut threshold for FUNCTION suspectacumprec()
tolerance	number of NA or 0s before allowed before the limit, FUNCTION: suspectacumprec()
maxseq	maximum number of consecutive repeated values, FUNCTION: flat() (11.1,11.1,11.1 would be 3 consecutive values)
roundmax	maximum number of consecutive decimal part values, FUNCTION: flat() (10.0, 11.0, 12.0 would be 3 consecutive values)
level	number of IQRs, FUNCTION: IQRoutliers()
window	number of days to be considered (including the target), FUNCTION: IQRoutliers()
margina	a tolerance margin, expressed as quantile of the differences, FUNCTION: newfriki()
inisia	a logical flag. If it is TRUE inithome() will be called

Value

results of QC for RR

Examples

```
#Set a temporal working directory:
wd <- tempdir()
wd0 <- setwd(wd)
#Create subdirectory where raw data files have to be located
dir.create(file.path(wd, 'raw'))
options("homefolder"='./'); options("blend"=FALSE)
#Extract the ECA&D data and station files from the example data folder
path2rrlist<-system.file("extdata", "ECA_blend_source_rr.txt", package = "INQC")
rrlist<-readr::read_lines_raw(path2rrlist)
readr::write_lines(rrlist,'ECA_blend_source_rr.txt')
path2rrdata<-system.file("extdata", "RR_SOUID132730.txt", package = "INQC")
rrdata<-readr::read_lines_raw(path2rrdata)
readr::write_lines(rrdata, file=paste(wd,'/raw/RR_SOUID132730.txt',sep=''))
#Perform QC of Atmospheric Precipitation data
precip(inisia=TRUE)
#Remove some temporary files
list = list.files(pattern = "Rfwf")
file.remove(list)
#Return to user's working directory:
setwd(wd0)
#The QC results can be found in the directory:
print(wd)
```

putjulian *Merges julian days*

Description

This function merges julian days to a yyyy,mm,dd and data

Usage

```
putjulian(x)
```

Arguments

x	data frame with year, month, day and data columns

Value

the same data frame with added 1 column: year, month, day, julian and data

Examples

```
date<-c('20201230','20201231','20210101')
value<-c(-10,-12,-9)
df<-data.frame(date,value)
year<-as.numeric(substring(date,1,4))
month<-as.numeric(substring(date,5,6))
day<-as.numeric(substring(date,7,8))
x<-data.frame(year,month,day,date,value)
y<-putjulian(x)
```

readecad *Reads an ECA&D data/sources/stations file*

Description

This function reads one ECA&D file and puts it in yyyy/mm/dd/value. Data is NOT divided by 10, to transform it into true units

Usage

```
readecad(input = "SS_STAID000143.txt", missing = -9999)
```

Arguments

input	ECA&D filename
missing	missing value code, set to the default ECA&D mvc

Value

data frame containing data (time series) from the ECA&D file. An introductory part of the ECA&D file with meta data information is skipped

Examples

```
#Extract the ECA&D data file from the example data folder
path2inptfl<-system.file("extdata", "CC_SOUID132727.txt", package = "INQC")
#Read the data file
df<-readecad(input=path2inptfl,missing= -9999)
```

`readheader` *Reads the header of an ECA&D file*

Description

This function reads one ECA&D file and returns the header (an introductory part of the ECA&D file), so it can be written in the same way

Usage

```
readheader(input = "SS_STAID000143.txt")
```

Arguments

`input`	ECA&D filename

Value

header of an ECA&D file

Examples

```
#Extract the ECA&D data file from the example data folder
path2inptfl<-system.file("extdata", "CC_SOUID132727.txt", package = "INQC")
#Read the data file
head<-readheader(input=path2inptfl)
```

relhum *QC for Relative Humidity (HU)*

Description

This function will centralize temperature-like QC routines. Will create a file in the folder QC with an additional 0/1 column, where "1" means test failed.

Usage

```
relhum(
  element = "HU",
  maxseq = 3,
  blocksizeround = 20,
  blockmanymonth = 15,
  blockmanyyear = 180,
  roundmax = 10,
  inisia = FALSE
)
```

Arguments

element	two-letters ECA&D code for the element (HU for relative humidity)
maxseq	maximum number of consecutive repeated values, for flat function (11.1,11.1,11.1 would be 3 consecutive values). Passed on to flat(). See ?flat for details
blocksizeround	maximum number of values in a month with the same decimal, FUNCTION: rounding()
blockmanymonth	maximum number of equal values in a month, FUNCTION: toomany()
blockmanyyear	maximum number of equal values in a year, FUNCTION: toomany()
roundmax	maximum number of consecutive decimal part value, for flat function (10.0, 11.0, 12.0 would be 3 consecutive values)
inisia	a logical flag. If it is TRUE inithome() will be called

Value

results of QC for HU

Examples

```
#Set a temporal working directory:
wd <- tempdir()
wd0 <- setwd(wd)
#Create subdirectory where raw data files have to be located
dir.create(file.path(wd, 'raw'))
options("homefolder"='./'); options("blend"=FALSE)
#Extract the ECA&D data and station files from the example data folder
```

```
path2hulist<-system.file("extdata", "ECA_blend_source_hu.txt", package = "INQC")
hulist<-readr::read_lines_raw(path2hulist)
readr::write_lines(hulist,'ECA_blend_source_hu.txt')
path2hudata<-system.file("extdata", "HU_SOUID132735.txt", package = "INQC")
hudata<-readr::read_lines_raw(path2hudata)
readr::write_lines(hudata, file=paste(wd,'/raw/HU_SOUID132735.txt',sep=''))
#Perform QC of Relative Humidity data
relhum(inisia=TRUE)
#Remove some temporary files
list = list.files(pattern = "Rfwf")
file.remove(list)
#Return to user's working directory:
setwd(wd0)
#The QC results can be found in the directory:
print(wd)
```

`repeatedvalue` *Finds repeated values*

Description

This function looks for a value which repeats too many times and, given the decaying shape of empirical distribution of precipitation data, is considered too large to happen that many times

Usage

```
repeatedvalue(x, margin = 20, friki = 150)
```

Arguments

`x`	precipitation time series
`margin`	threshold for differences in frequency of the nearest value
`friki`	minimum of precipitation values to be considered

Value

list of positions which do not pass this QC test. If all positions pass the test, returns NULL

Examples

```
#Extract the ECA&D data file from the example data folder
path2inptfl<-system.file("extdata", "RR_SOUID132730.txt", package = "INQC")
#Read the data file
x<-readecad(input=path2inptfl,missing= -9999)[,4]
#Find all suspicious positions in the time series
repeatedvalue(x,margin=10,friki=10)
```

returnpotpareto *Threshold percentile for the Pareto outliers*

Description

This function returns a value of a threshold percentile for the Pareto outliers

Usage

```
returnpotpareto(pato, ret, w = 1.65)
```

Arguments

pato	list with results of modelling/fitting the generalized Pareto distribution
ret	pseudo-return period (in yr)
w	average sampling frequency (in 1/yr), a parameter to equate to return period to a temporal interval (recall the approach is not block maxima but peak over threshold. Typical value of w to equate the return period to years is 1.65 (See Wilks, 2011. Statistical Analysis for the Atmospheric Sciences)

Value

for a given Pareto distribution, returns the value (the quantile) representing a requested return period

Examples

```
#Extract the ECA&D precipitation data file from the example data folder
path2inptfl<-system.file("extdata", "RR_SOUID132730.txt", package = "INQC")
#Read the data file
y<-readecad(input=path2inptfl,missing= -9999)[,4]
#Fit the Generalized Pareto distribution
pato<-potpareto(y)
#Define the quantile corresponding to the requested return period 25 years (ret=25)
returnpotpareto(pato,25)
#Define the quantile assuming the existence of 2 precipitation peaks/extreme values
#every year (on average)
returnpotpareto(pato,25,w=2)
```

rounding *Detects rounded sections*

Description

This function splits data by month and looks if a decimal value is repeated too many times

Usage

```
rounding(y, blocksize = 20)
```

Arguments

y	two columns with date in the ECA&D format (yyyymmdd) and data
blocksize	maximum number of repeated values with the same decimal allowed on each block (blocks = months)

Value

list of positions which do not pass this QC test. If all positions pass the test, returns NULL

Examples

```
#Extract the ECA&D data file from the example data folder
path2inptfl<-system.file("extdata", "TX_SOUID132734.txt", package = "INQC")
#Read the data file
y<-readecad(input=path2inptfl,missing= -9999)[,3:4]
#Introduce the rounding errors in first 50 data values
y[1:50,2]<-round((y[1:50,2])/10)*10
#Find all suspicious positions in the time series
rounding(y,blocksize=20)
```

roundprecip *Rounding in precipitation data*

Description

This function splits data by month and looks if a decimal value is repeated too many times

Usage

```
roundprecip(y, blocksize = 20, exclude = 0)
```

Arguments

y	two columns with date and data
blocksize	maximum number of repeated values with the same decimal
exclude	value to be excluded (zero for precipitation)

Value

list of positions which do not pass this QC test. If all positions pass the test, returns NULL

Examples

```
#Extract the ECA&D data file from the example data folder
path2inptfl<-system.file("extdata", "RR_SOUID132730.txt", package = "INQC")
#Read the data file
y<-readecad(input=path2inptfl,missing= -9999)[,3:4]
#Find all suspicious positions in the precipitation time series
roundprecip(y,blocksize=20,exclude=0)
```

---

selepe *QC for Atmospheric Pressure (PP)*

Description

This function will centralize temperature-like QC routines. It will create a file in the folder QC with an additional 0/1 column where "1" means test failed.

Usage

```
selepe(
  element = "PP",
  large = 15000,
  small = 8000,
  maxjump = 2000,
  maxseq = 3,
  margina = 0.999,
  level = 5,
  window = 30,
  roundmax = 10,
  blockmanymonth = 15,
  blockmanyyear = 180,
  blocksizeround = 20,
  qjump = 0.999,
  tjump = 1.5,
  inisia = FALSE
)
```

Arguments

element	two-letters ECA&D code for the element (PP for sea level pressure)
large	value above which the observation is considered physically impossible for the region
small	value below which the observation is considered physically impossible for the region
maxjump	forcing for jumps2() in absolute mode (in the same units of the variable). Passed on to jumps2(). See ?jumps2 for further details.
maxseq	maximum number of consecutive repeated values, for flat function (11.1,11.1,11.1 would be 3 consecutive values)
margina	tolerance margin, expressed as quantile of the differences, FUNCTION: newfriki(). Passed on to newfriki(). See ?newfriki for details
level	number of IQRs for IQR outliers
window	window, in days, for IQR outliers
roundmax	maximum number of consecutive decimal part value, for flat function (10.0, 11.0, 12.0 would be 3 consecutive values)
blockmanymonth	maximum number of equal values in a month, FUNCTION: toomany()
blockmanyyear	maximum number of equal values in a year, FUNCTION: toomany()
blocksizeround	maximum number of values in a month with the same decimal, for rounding function
qjump	quantile for jumps2() in quantile mode. Passed on to jumps2(). See ?jumps2 for further details
tjump	factor to multiply the quantile value for jumps2(). Passed on to jumps2(). See ?jumps2 for further details
inisia	a logical flag. If it is TRUE inithome() will be called

Value

results of QC for PP

Examples

```
#Set a temporal working directory:
wd <- tempdir()
wd0 <- setwd(wd)
#Create subdirectory where raw data files have to be located
dir.create(file.path(wd, 'raw'))
options("homefolder"='./'); options("blend"=FALSE)
#Extract the ECA&D data and station files from the example data folder
path2pplist<-system.file("extdata", "ECA_blend_source_pp.txt", package = "INQC")
pplist<-readr::read_lines_raw(path2pplist)
readr::write_lines(pplist,'ECA_blend_source_pp.txt')
path2ppdata<-system.file("extdata", "PP_SOUID132729.txt", package = "INQC")
ppdata<-readr::read_lines_raw(path2ppdata)
readr::write_lines(ppdata, file=paste(wd,'/raw/PP_SOUID132729.txt',sep=''))
```

```
#Perform QC of Atmospheric Pressure data
selepe(inisia=TRUE)
#Remove some temporary files
list = list.files(pattern = "Rfwf")
file.remove(list)
#Return to user's working directory:
setwd(wd0)
#The QC results can be found in the directory:
print(wd)
```

---

snowdepth *QC for Snow Depth (SD)*

Description

This function will centralize temperature-like QC routines. It will create a file in the folder QC with an additional 0/1 column, where "1" means test failed.

Usage

```
snowdepth(
  element = "SD",
  maxseq = 20,
  blocksizeround = 20,
  blockmanymonth = 20,
  blockmanyyear = 200,
  large = 5000,
  exclude = 0,
  inisia = FALSE
)
```

Arguments

element	two-letters ECA&D code for the element (SD for snow depth)
maxseq	maximum number of consecutive repeated values, FUNCTION: flat() (11.1,11.1,11.1 would be 3 consecutive values)
blocksizeround	maximum number of values in a month with the same decimal, FUNCTION: rounding()
blockmanymonth	maximum number of equal values in a month, FUNCTION: toomany()
blockmanyyear	maximum number of equal values in a year, FUNCTION: toomany()
large	value above which the observation is considered physically impossible for the region, FUNCTION: physics()
exclude	value to be excluded from a function (in this case, 0 for flats)
inisia	logical flag. If it is TRUE inithome() will be called

Value

results of QC for SD

Examples

```
#Set a temporal working directory:
wd <- tempdir()
wd0 <- setwd(wd)
#Create subdirectory where raw data files have to be located
dir.create(file.path(wd, 'raw'))
options("homefolder"='./'); options("blend"=FALSE)
#Extract the ECA&D data and station files from the example data folder
path2sdlist<-system.file("extdata", "ECA_blend_source_sd.txt", package = "INQC")
sdlist<-readr::read_lines_raw(path2sdlist)
readr::write_lines(sdlist,'ECA_blend_source_sd.txt')
path2sddata<-system.file("extdata", "SD_SOUID132731.txt", package = "INQC")
sddata<-readr::read_lines_raw(path2sddata)
readr::write_lines(sddata, file=paste(wd,'/raw/SD_SOUID132731.txt',sep=''))
#Perform QC of Snow Depth data
snowdepth(inisia=TRUE)
#Remove some temporary files
list = list.files(pattern = "Rfwf")
file.remove(list)
#Return to user's working directory:
setwd(wd0)
#The QC results can be found in the directory:
print(wd)
```

---

sunafterdark *Maximum sunshine hours (only for "non-blended" ECA&D data)*

Description

This function compares sunshine data to the maximum theoretical sunshine at an ECA&D station, according the day, lat and lon. Maximum sunshine hours are computed from the "suncalc" package, using "night" and "dawn" parameters. This contrasts quite a lot with other functions computing "daylength". This formulation is more conservative

Usage

```
sunafterdark(y, code = "991274")
```

Arguments

y	ECA&D style two columns with date (yyyymmdd) and values (expressed in 0.1 hours)
code	"numeric" part of the ECA&D SOUID, expressed as character, to avoid trouble with leading zeroes

Details

depends on either a previous execution of listas() or on a proper execution of listas() to run properly

Value

vector with the list of positions which do not pass this test. If all positions pass the test, returns NULL

See Also

listas()

Examples

```
#Set a temporal working directory:
wd <- tempdir(); wd0 <- setwd(wd)
#Extract the non-blended ECA&D data and a station file from the example data folder
path2sslist<-system.file("extdata", "ECA_blend_source_ss.txt", package = "INQC")
sslist<-readr::read_lines_raw(path2sslist)
readr::write_lines(sslist,'ECA_blend_source_ss.txt')
path2ssdata<-system.file("extdata", "SS_SOUID132728.txt", package = "INQC")
#Read the sunshine data
y<-readecad(input=path2ssdata,missing= -9999)[,3:4]
options("homefolder"='./'); options("blend"=FALSE)
listonator(check=TRUE)
#Call sunafterdark()
sunafterdark(y,code='132728')
#Introduce error values in the sunshine data
y[1:10,2]<-200
#Call sunafterdark()
sunafterdark(y,code='132728')
#Return to user's working directory:
setwd(wd0)
```

---

sundur *QC for Sunshine Duration (SS)*

Description

This function will centralize temperature-like QC routines. Will create a file in the folder QC with an additional 0/1 column, where "1" means test failed

Usage

```
sundur(
  element = "SS",
  maxseq = 3,
  blocksizeround = 20,
  blockmanymonth = 15,
```

```
  blockmanyyear = 180,
  roundmax = 10,
  inisia = FALSE
)
```

Arguments

element	two-letters ECA&D code for the element (SS for sunshine duration)
maxseq	maximum number of consecutive repeated values, for flat function (11.1,11.1,11.1 would be 3 consecutive values). Passed on to flat(). See ?flat for details
blocksizeround	maximum number of values in a month with the same decimal, FUNCTION: rounding()
blockmanymonth	maximum number of equal values in a month, FUNCTION: toomany()
blockmanyyear	maximum number of equal values in a year, FUNCTION: toomany()
roundmax	maximum number of consecutive decimal part value, for flat() function (10.0, 11.0, 12.0 would be 3 consecutive values)
inisia	logical flag. If it is TRUE inithome() will be called

Value

results of QC for SS

Examples

```
#Set a temporal working directory:
wd <- tempdir()
wd0 <- setwd(wd)
#Create subdirectory where raw data files have to be located
dir.create(file.path(wd, 'raw'))
options("homefolder"='./'); options("blend"=FALSE)
#Extract the ECA&D data and station files from the example data folder
path2sslist<-system.file("extdata", "ECA_blend_source_ss.txt", package = "INQC")
sslist<-readr::read_lines_raw(path2sslist)
readr::write_lines(sslist,'ECA_blend_source_ss.txt')
path2ssdata<-system.file("extdata", "SS_SOUID132728.txt", package = "INQC")
ssdata<-readr::read_lines_raw(path2ssdata)
readr::write_lines(ssdata, file=paste(wd,'/raw/SS_SOUID132728.txt',sep=''))
#Perform QC of Sunshine Duration data
sundur(inisia=TRUE)
#Remove some temporary files
list = list.files(pattern = "Rfwf")
file.remove(list)
#Return to user's working directory:
setwd(wd0)
#The QC results can be found in the directory:
print(wd)
```

suspectacumprec *Detects precipitation values above limit*

Description

This function detects values above limit preceded by a number of "non precipitation days", given by tolerance

Usage

```
suspectacumprec(datos, limit = 2000, tolerance = 10)
```

Arguments

datos	two columns vector, date and data, in the ECA&D format
limit	threshold/limit value for atmospheric precipitation
tolerance	how many consecutive days with 0 or NA you need to jump

Value

list of positions which do not pass this QC test

Examples

```
#Extract the ECA&D data file from the example data folder
path2inptfl<-system.file("extdata", "RR_SOUID132730.txt", package = "INQC")
#Read the data file
datos<-readecad(input=path2inptfl,missing= -9999)[,3:4]
#Find all suspicious positions in the precipitation time series
suspectacumprec(datos,limit=2000,tolerance=10)
```

temperature *QC for Air Temperature (TX/TN/TG)*

Description

This function will centralize temperature-like QC routines. It will QC files for temperature. Reads all the temperature data in the ./raw folder (TX, TN or TG) and quality controls each of them. Notice that ECA&D stores temperature in 1/10th of Celsius degrees when entering new parameter values

Usage

```
temperature(
  element = "TX",
  large = 500,
  small = -500,
  maxjump = 200,
  maxseq = 3,
  margina = 0.999,
  level = 4,
  window = 11,
  roundmax = 10,
  blockmanymonth = 15,
  blockmanyyear = 180,
  blocksizeround = 20,
  qjump = 0.999,
  tjump = 1.5,
  inisia = FALSE
)
```

Arguments

element	two-letters ECA&D code for the element ('TX' for daily maximum temperature, 'TN' for daily minimum temperature, 'TG' for daily mean temperature) passed as character string
large	value above which the observation is considered physically impossible for the region. Defaulted to 500. Passed on to physics(). See ?physics for details
small	value below which the observation is considered physically impossible for the region. Defaulted to -500. Passed on to physics(). See ?physics for details
maxjump	forcing for jumps2() in absolute mode (in the same units of the variable). Passed on to jumps2(). See ?jumps2 for further details
maxseq	maximum number of consecutive repeated values, for flat function (11.1,11.1,11.1 would be 3 consecutive values). Passed on to flat(). See ?flat for details
margina	tolerance margin, expressed as quantile of the differences, FUNCTION: newfriki(). Passed on to newfriki(). See ?newfriki for details
level	number of IQRs for IQRoutliers()
window	number of days to be considered (including the target), FUNCTION: IQRoutliers()
roundmax	maximum number of consecutive decimal part value, for flat function (10.0, 11.0, 12.0 would be 3 consecutive value). Passed on to flat()
blockmanymonth	maximum number of equal values in a month, FUNCTION: toomany()
blockmanyyear	maximum number of equal values in a year, FUNCTION: toomany()
blocksizeround	the maximum number of repeated values with the same decimal, FUNCTION: roundprecip()
qjump	quantile for jumps2() in quantile mode. Passed on to jumps2(). See ?jumps2 for further details

tjump	factor to multiply the quantile value for jumps2(). Passed on to jumps2(). See ?jumps2 for further details
inisia	logical flag. If it is TRUE inithome() will be called

Value

results of QC for TX/TN/TG

See Also

consolidator(), duplas(), flat(), IQRoutliers(), jumps2(), newfriki(), physics(), toomany(), rounding(), txtn(), weirdate()

Examples

```
#Set a temporal working directory:
wd <- tempdir()
wd0 <- setwd(wd)
#Create subdirectory where raw data files have to be located
dir.create(file.path(wd, 'raw'))
options("homefolder"='./'); options("blend"=FALSE)
#Extract the ECA&D data and station files from the example data folder
path2tnlist<-system.file("extdata", "ECA_blend_source_tn.txt", package = "INQC")
tnlist<-readr::read_lines_raw(path2tnlist)
readr::write_lines(tnlist,'ECA_blend_source_tn.txt')
path2tndata<-system.file("extdata", "TN_SOUID132733.txt", package = "INQC")
tndata<-readr::read_lines_raw(path2tndata)
readr::write_lines(tndata, file=paste(wd,'/raw/TN_SOUID132733.txt',sep=''))
#Perform QC of Air Temperature data
temperature(element='TN',inisia=TRUE)
#Remove some temporary files
list = list.files(pattern = "Rfwf")
file.remove(list)
#Return to user's working directory:
setwd(wd0)
#The QC results can be found in the directory:
print(wd)
```

---

toomany *Looks if a value is repeated too many times*

Description

This function splits data by month and looks if a value is repeated too many times

Usage

```
toomany(y, blockmany = 15, scope = 1, exclude = NULL)
```

Arguments

y	two columns with date and data
blockmany	maximum number of repeated values in a month, year, or season
scope	monthly (1), annual (2)
exclude	values to exclude, e.g. if precipitation, 0 must be excluded

Value

list of positions which do not pass this QC test. If all positions pass the test, returns NULL

Examples

```
#Extract the ECA&D data file (maximum air temperature) from the example data folder
path2inptfl<-system.file("extdata", "TX_SOUID132734.txt", package = "INQC")
#Read the data file
y<-readecad(input=path2inptfl,missing= -9999)[,3:4]
#Introduce the errors in first 20 data values
y[1:20,2]<-30
#Find all suspicious positions in the time series
toomany(y,blockmany=15,scope=1,exclude=NULL)

#Extract the ECA&D data file (atmospheric precipitation) from the example data folder
path2inptfl<-system.file("extdata", "RR_SOUID132730.txt", package = "INQC")
#Read the data file
y<-readecad(input=path2inptfl,missing= -9999)[,3:4]
#Introduce the errors in first 20 data values
y[1:20,2]<-10
#Find all suspicious positions in the time series
toomany(y,blockmany=15,scope=1,exclude=0)
```

---

txtn *Comparison of tx an tn data (for "non-blended" ECA&D data)*

Description

This function compares tx an tn data. First it looks for the closest station and then merges both data frames. If one value is flagged, looks at the ecdfs of tx and tn. If the target variable (e.g tx) is central (between quantiles 0.2 and 0.8) and the other variable (e.g. tn) is outside this range, the value is not flagged, assuming the other variable is the culprit

Usage

```
txtn(y, id)
```

Arguments

y	two columns with date and data
id	name of a file ("xx_SOUIDxxxxxx.txt", non-blended) we are working with

Value

list of positions which do not pass this QC test. If all positions pass the test, returns NULL

Examples

```
#Set a temporal working directory:
wd <- tempdir(); wd0 <- setwd(wd)
#Create subdirectory where raw data files have to be located
dir.create(file.path(wd, "raw"))
#Extract the non-blended ECA&D data and station files from the example data folder
path2txlist<-system.file("extdata", "ECA_blend_source_tx.txt", package = "INQC")
txlist<-readr::read_lines_raw(path2txlist)
readr::write_lines(txlist,"ECA_blend_source_tx.txt")
path2txdata<-system.file("extdata", "TX_SOUID132734.txt", package = "INQC")
txdata<-readr::read_lines_raw(path2txdata)
readr::write_lines(txdata, file=paste(wd,"/raw/TX_SOUID132734.txt",sep=""))
path2tnlist<-system.file("extdata", "ECA_blend_source_tn.txt", package = "INQC")
tnlist<-readr::read_lines_raw(path2tnlist)
readr::write_lines(tnlist,"ECA_blend_source_tn.txt")
path2tndata<-system.file("extdata", "TN_SOUID132733.txt", package = "INQC")
tndata<-readr::read_lines_raw(path2tndata)
readr::write_lines(tndata, file=paste(wd,"/raw/TN_SOUID132733.txt",sep=""))
#Read the tn data
y<-readecad(input=path2tndata,missing= -9999)[,3:4]
options("homefolder"="./"); options("blend"=FALSE)
listonator(check=TRUE)
#Call txtn()
txtn(y,"TN_SOUID132733.txt")
#Introduce error values in the tn data
y[c(1,3),2]<-100
#Call txtn()
txtn(y,"TN_SOUID132733.txt")
#Return to user's working directory:
setwd(wd0)
```

---

txtnblend *Comparison of tx an tn data (for "blended" ECA&D data)*

Description

This function first looks for the closest station and then merges both data frames. If one value is flagged, looks at the ecdfs of tx and tn. If the target variable (e.g tx) is central (between quantiles 0.2 and 0.8) and the other variable (e.g. tn) is outside this range, the value is not flagged, assuming the other variable is the culprit

Usage

```
txtnblend(y, id)
```

Arguments

y	two columns with date and data
id	name of a file ("xx_STAIDxxxxxx.txt", blended) we are working with

Value

list of positions which do not pass this QC test. If all positions pass the test, returns NULL

Examples

```
#Set a temporal working directory:
wd <- tempdir(); wd0 <- setwd(wd)
#Create subdirectory where raw data files have to be located
dir.create(file.path(wd, "raw"))
#Extract the blended ECA&D data and station files from the example data folder
path2list<-system.file("extdata", "stations.txt", package = "INQC")
list<-readr::read_lines_raw(path2list)
readr::write_lines(list,file=paste(wd,"/raw/stations.txt",sep=""))
path2txdata<-system.file("extdata", "TX_STAID000002.txt", package = "INQC")
txdata<-readr::read_lines_raw(path2txdata)
readr::write_lines(txdata, file=paste(wd,"/raw/TX_STAID000002.txt",sep=""))
path2tndata<-system.file("extdata", "TN_STAID000002.txt", package = "INQC")
tndata<-readr::read_lines_raw(path2tndata)
readr::write_lines(tndata, file=paste(wd,"/raw/TN_STAID000002.txt",sep=""))
#Read the tn data
y<-readecad(input=path2tndata,missing= -9999)[,3:4]
options("homefolder"="./")
#Call txtnblend()
txtnblend(y,"TN_STAID000002.txt")
#Introduce error values in the tn data
y[c(1,3),2]<-100
#Call txtnblend()
txtnblend(y,"TN_STAID000002.txt")
#Return to user's working directory:
setwd(wd0)
```

---

weirddate *Locate impossible dates*

Description

This function is intended to flag impossible dates (e.g., 19990230 or 29990112, etc)

Usage

```
weirddate(x)
```

Arguments

x	two-columns dataframe. First column is date in the ECA&D format (yyyymmdd), second columns is value

Value

list of positions which do not pass this QC test. If all positions pass the test, returns NULL

Examples

```
#Extract the ECA&D data file from the example data folder
path2inptfl<-system.file("extdata", "TX_SOUID132734.txt", package = "INQC")
#Read the data file
x<-readecad(input=path2inptfl,missing= -9999)[,3:4]
#Find all suspicious positions in the time series
weirddate(x)

#Introduce the weird dates
x[31,1]<-'19610132'
#Find all suspicious positions in the time series
weirddate(x)
```

---

windspeed *QC for Wind Speed (FG)*

Description

This function will centralize temperature-like QC routines. It will create a file in the folder QC with an additional 0/1 column, where "1" means test failed.

Usage

```
windspeed(
  element = "FG",
  maxseq = 3,
  blocksizeround = 20,
  blockmanymonth = 20,
  blockmanyyear = 200,
  large = 3000,
  roundmax = 10,
  level = 5,
  window = 30,
  ret = 500,
  margina = 0.999,
  inisia = FALSE
)
```

Arguments

element	two-letters ECA&D code for the element (e.g., FG for wind speed)
maxseq	maximum number of consecutive repeated values, FUNCTION: flat (11.1,11.1,11.1 would be 3 consecutive values)
blocksizeround	maximum number of values in a month with the same decimal, FUNCTION: rounding()
blockmanymonth	maximum number of equal values in a month, FUNCTION: toomany()
blockmanyyear	maximum number of equal values in a year, FUNCTION: toomany()
large	value above which the observation is considered physically impossible for the region, FUNCTION: physics()
roundmax	maximum number of consecutive decimal part value, for flat function (10.0, 11.0, 12.0 would be 3 consecutive values)
level	number of IQRs for IQR outliers
window	window, in days, for IQR outliers
ret	pseudo-return period for the Pareto outliers
margina	quantile for newfriki function
inisia	a logical flag. If it is TRUE inithome() will be called

Value

results of QC for FG

Examples

```
#Set a temporal working directory:
wd <- tempdir()
wd0 <- setwd(wd)
#Create subdirectory where raw data files have to be located
dir.create(file.path(wd, 'raw'))
options("homefolder"='./'); options("blend"=FALSE)
#Extract the ECA&D data and station files from the example data folder
path2fglist<-system.file("extdata", "ECA_blend_source_fg.txt", package = "INQC")
fglist<-readr::read_lines_raw(path2fglist)
readr::write_lines(fglist,'ECA_blend_source_fg.txt')
path2fgdata<-system.file("extdata", "FG_SOUID132736.txt", package = "INQC")
fgdata<-readr::read_lines_raw(path2fgdata)
readr::write_lines(fgdata, file=paste(wd,'/raw/FG_SOUID132736.txt',sep=''))
#Perform QC of Wind Speed data
windspeed(inisia=TRUE)
#Remove some temporary files
list = list.files(pattern = "Rfwf")
file.remove(list)
#Return to user's working directory:
setwd(wd0)
#The QC results can be found in the directory:
print(wd)
```

Index