Package 'dataresqc'

February 4, 2020

Type Package

Title C3S Quality Control Tools for Historical Climate Data

Version 1.0.2

Date 2020-02-04

Description
Quality control functions developed for the Copernicus Data Rescue Service (see <https://data-rescue.copernicus-climate.eu/>).

Depends R (>= 3.2)

License Apache License 2.0

Suggests ggplot2, grid

LazyData true

RoxygenNote 7.0.2

Encoding UTF-8

NeedsCompilation no

Author Yuri Brugnara [aut, cre] (<https://orcid.org/0000-0001-8427-0064>),
Alba Gilabert [aut],
Clara Ventura [aut],
Stefan Hunziker [aut]

Maintainer Yuri Brugnara <yuri.brugnara@giub.unibe.ch>

R topics documented:

`Bern` *Sub-daily meteorological observations for Bern*

Description

Observations of pressure and temperature for the city of Bern (Switzerland) for the period 1800-1827.

Usage

```
Bern
```

Format

A list of data frames (one data frame per variable). The format of the data frames is that required by the QC functions.

Source

Institute of Geography - University of Bern

check_sef *Check compliance with SEF guidelines*

Description

Check compliance with SEF guidelines

Usage

```
check_sef(file = file.choose())
```

Arguments

file	Character string giving the path of the SEF file.

Value

TRUE if no errors are found, FALSE otherwise.

Note

For more information on error/warning messages produced by this function see the SEF documentation.

Author(s)

Yuri Brugnara

climatic_outliers *Climatic outliers test*

Description

Considers as outliers all values falling outside a range between, for example, p25 - 3 interquartile ranges and p75 + 3 interquartile The number of interquantile ranges can be modified through the parameter IQR.

Usage

```
climatic_outliers(
  Data,
  meta = NULL,
  outpath,
  IQR = NA,
  bplot = FALSE,
  outfile = NA,
  ...
)
```

Arguments

Data	A character string giving the path of the input file, or a matrix with 5 (7) columns for daily (sub-daily) data: variable code, year, month, day, (hour), (minute), value.
meta	A character vector with 6 elements: station ID, latitude, longitude, altitude, variable code, units. If Data is a path, meta is ignored.
outpath	Character string giving the path for the QC results.
IQR	Interquantile range used to define outliers. By default it is 5 for precipitation, 3 for air temperature, and 4 for any other variable.
bplot	If TRUE, create a boxplot and print it into a PDF.
outfile	Filename for the plot. Ignored if bplot is FALSE.
...	Graphical parameters passed to the function boxplot.

Details

The input file must follow the Copernicus Station Exchange Format (SEF). This function works with any numerical variable.

Zeroes are automatically excluded in bounded variables such as precipitation.

Author(s)

Alba Gilabert, Yuri Brugnara

Examples

```
climatic_outliers(Rosario$Tn, Meta$Tn, outpath = tempdir(), IQR = 4)
```

---

climexp_to_sef *Download a GHCN-Daily data file from the Climate Explorer and convert it into the Station Exchange Format*

Description

Download a GHCN-Daily data file from the Climate Explorer and convert it into the Station Exchange Format

Usage

```
climexp_to_sef(url, outpath)
```

Arguments

url	Character string giving the url of the data file.
outpath	Character string giving the path where to save the file.

Author(s)

Yuri Brugnara

`convert_pressure` *Convert pressure data to hPa*

Description

Converts pressure observations made with a mercury barometer to SI units. If geographical coordinates are given, a gravity correction is applied. If attached temperature is given, a temperature correction is applied.

Usage

```
convert_pressure(p, f = 1, lat = NA, alt = NA, atb = NULL)
```

Arguments

`p`	A numerical vector of barometer observations in any unit of length.
`f`	Conversion factor to mm (e.g., 25.4 for English inches).
`lat`	Station latitude (degrees North in decimal).
`alt`	Station altitude (metres). Assumed zero if not given. Ignored if `lat` is NA.
`atb`	A vector of the attached temperature observations in Celsius.

Value

A numerical vector of pressure values in hPa.

Author(s)

Yuri Brugnara

References

WMO, 2008: Guide to meteorological instruments and methods of observation, WMO-No. 8, World Meteorological Organization, Geneva.

Examples

```
convert_pressure(760) # Gives a standard pressure of 1013.25 hPa

convert_pressure(760, lat=70, alt=100) # Gives a higher pressure because of higher g

convert_pressure(760, lat=70, alt=100, atb=20) # Gives a lower pressure because the
                                               # temperature correction is larger than
                                               # the gravity correction.
```

daily_out_of_range *Daily big errors test*

Description

Find the daily maximum, minimum, precipitation, mean wind direction, mean wind speed, snow cover and snow depth that exceed thresholds selected by the user. The output is a list with the days in which Tx, Tn, rr, dd, w, sc, sd or fs exceeds some threshold.

Usage

```
daily_out_of_range(
  dailydata,
  meta = NULL,
  outpath,
  tmax_upper = 45,
  tmax_lower = -30,
  tmin_upper = 30,
  tmin_lower = -40,
  rr_upper = 200,
  rr_lower = 0,
  w_upper = 30,
  w_lower = 0,
  dd_upper = 360,
  dd_lower = 0,
  sc_upper = 100,
  sc_lower = 0,
  sd_upper = 200,
  sd_lower = 0,
  fs_upper = 100,
  fs_lower = 0
)
```

Arguments

dailydata	A character string giving the path of the input file, or a 5-column matrix with following columns: variable code, year, month, day, and the daily value.
meta	A character vector with 6 elements: station ID, latitude, longitude, altitude, variable code, units. If dailydata is a path, meta is ignored.
outpath	Character string giving the path for the QC results.
tmax_upper	is the tx maximum threshold in degrees Celsius. By default, tmax_upper = 45 C.
tmax_lower	is the tx minimum threshold in degrees Celsius. By default, tmax_lower = -30 C.
tmin_upper	is the tn maximum threshold in degrees Celsius. By default, tmin_upper = 30 C.
tmin_lower	is the tn minimum threshold in degrees Celsius. By default, tmin_lower = -40 C.
rr_upper	is the rr maximum threshold in millimetres. By default, rr_upper = 200 mm.
rr_lower	is the rr minimum threshold in millimetres. By default, rr_lower = 0 mm.

w_upper	is the w maximum threshold in metres per second. By default, w_upper = 30 m/s.
w_lower	is the w mimumum threshold in metres per second. By default, w_lower = 0 m/s.
dd_upper	is the dd maximum threshold in degrees North. By default, dd_upper = 360.
dd_lower	is the dd minimum threshold in degrees North. By default, dd_lower = 0.
sc_upper	is the sc maximum threshold in percent. By default, sc_upper = 100%.
sc_lower	is the sc minimum threshold in percent. By default, sc_lower = 0%.
sd_upper	is the sd maximum threshold in centimetres. By default, sd_upper = 200 cm.
sd_lower	is the sd minimum threshold in centimetres. By default, sd_lower = 0 cm.
fs_upper	is the fs maximum threshold in centimetres. By default, fs_upper = 100 cm.
fs_lower	is the fs minimum threshold in centimetres. By default, fs_lower = 0 cm.

Details

The input file must follow the Copernicus Station Exchange Format (SEF).

Author(s)

Alba Gilabert, Yuri Brugnara

Examples

```
daily_out_of_range(Rosario$Tn, Meta$Tn, outpath = tempdir(), tmin_upper = 25)
```

---

daily_repetition *Daily repetition test*

Description

Report occurrences of equal consecutive values in daily data.

Usage

```
daily_repetition(dailydata, meta = NULL, outpath, n = 4)
```

Arguments

dailydata	A character string giving the path of the input file, or a 5-column matrix with following columns: variable code, year, month, day, and the daily value.
meta	A character vector with 6 elements: station ID, latitude, longitude, altitude, variable code, units. If dailydata is a path, meta is ignored.
outpath	Character string giving the path for the QC results.
n	Number of minimum equal consecutive values required for a flag. The default is 4.

Details

The input file must follow the Copernicus Station Exchange Format (SEF).

Zeroes are automatically excluded in bounded variables such as precipitation.

Author(s)

Alba Gilabert, Yuri Brugnara

Examples

```
daily_repetition(Rosario$Tx, Meta$Tx, outpath = tempdir(), n = 3)
```

---

duplicate_columns *Duplicate columns test*

Description

Looks for data that have been digitized twice by mistake. For sud-daily data, this is done by looking for series of zero differences between adjacent observation times. For daily data, by looking for series of zero differences between the same days of adjacent months.

Usage

```
duplicate_columns(Data, meta = NULL, outpath, ndays = 5)
```

Arguments

`Data`	A character string giving the path of the input file, or a matrix with 5 (7) columns for daily (sub-daily) data: variable code, year, month, day, (hour), (minute), value.
`meta`	A character vector with 6 elements: station ID, latitude, longitude, altitude, variable code, units. If `Data` is a path, `meta` is ignored.
`outpath`	Character string giving the path for the QC results.
`ndays`	Number of consecutive days with zero difference required to flag the data. The default is 5.

Details

The input file must follow the Copernicus Station Exchange Format (SEF). This function works with any numerical variable.

Zeroes are automatically excluded in bounded variables such as precipitation.

Author(s)

Yuri Brugnara

Examples

```
climatic_outliers(Rosario$Tn, Meta$Tn, outpath = tempdir(), ndays = 3)
```

duplicate_dates *Duplicate dates test*

Description

Flag dates that appear more than once in daily data.

Usage

```
duplicate_dates(dailydata, meta = NULL, outpath)
```

Arguments

dailydata	A character string giving the path of the input file, or a 5-column matrix with following columns: variable code, year, month, day, and the daily value.
meta	A character vector with 6 elements: station ID, latitude, longitude, altitude, variable code, units. If dailydata is a path, meta is ignored.
outpath	Character string giving the path for the QC results.

Details

The input file must follow the Copernicus Station Exchange Format (SEF).

Author(s)

Alba Gilabert, Yuri Brugnara

Examples

```
duplicate_dates(Rosario$Tx, Meta$Tx, outpath = tempdir())
```

duplicate_times *Duplicate times test*

Description

Flag times that appear more than once.

Usage

```
duplicate_times(subdailydata, meta = NULL, outpath)
```

Arguments

subdailydata	A character string giving the path of the input file, or a 7-column matrix with following columns: variable code, year, month, day, hour, minute, value.
meta	A character vector with 6 elements: station ID, latitude, longitude, altitude, variable code, units. If subdailydata is a path, meta is ignored.
outpath	Character string giving the path for the QC results.

Details

The input file must follow the Copernicus Station Exchange Format (SEF).

Author(s)

Alba Gilabert, Yuri Brugnara

Examples

```
duplicate_times(Bern$p, Meta$p[which(Meta$p$id=="Bern"),], outpath = tempdir())
```

`impossible_values` *Gross Errors Test for Cloud Cover and Relative Humidity.*

Description

Applicable to a series (daily or sub-daily) of relative humidity (rh) in percent or to a series of cloud cover (n) in percent or oktas.

Usage

```
impossible_values(series, meta = NULL, outpath)
```

Arguments

`series`	A character string giving the path of the SEF file, or a five or seven-column (daily or subdaily) data frame with the series.
`meta`	A character vector with 6 elements: station ID, latitude, longitude, altitude, variable code, units. If `series` is a path, `meta` is ignored.
`outpath`	Character string giving the path for the QC results.

Details

Input:

- A SEF file or a data frame and metadata. The observations data frame must have five or seven columns: variable code, year (YYYY), month (MM), day (DD), (hour (HH), minute (MM)), observation.

Output:

- A text file of flagged observations with six or eight columns: variable code, year (YYYY), month (MM), day (DD), (hour (HH), minute (MM)), observation, test. The test column has the description "gross_errors".
- The flagged observations correspond to values that don't belong to the integer interval (0, 100) if the unit is percent or that don't belong to the integer interval (0, 9) if the unit is oktas.

Author(s)

Clara Ventura, Yuri Brugnara

Examples

```
impossible_values(series = Rosario$n, meta = Meta$n, outpath = tempdir())
impossible_values(series = Rosario$rh, meta = Meta$rh, outpath = tempdir())
```

---

internal_consistency *Daily internal consistency test*

Description

Determines the coherence between daily maximum temperature (Tx) values and daily minimum temperature (Tn) values; daily wind speed (w) and wind direction (dd); daily snow cover (sc) and snow depth (sd); daily fresh snow (fs) and snow depth (sd); daily fresh snow (fs) and minimum temperature (Tn); daily snow depth (sd) and minimum temperature (Tn).

Usage

```
internal_consistency(dailydata, meta = NULL, outpath)
```

Arguments

`dailydata`	A character vector giving the paths of two input files, or a 5-column matrix with following columns: variable code, year, month, day, and the daily value.
`meta`	A data frame with 2 rows and 6 columns: station ID, latitude, longitude, altitude, variable code, units. If `dailydata` is a vector, `meta` is ignored.
`outpath`	Character string giving the path for the QC results.

Details

The input file must follow the Copernicus Station Exchange Format (SEF).

The daily minimum temperature is assumed to be observed at the same time of the snow depth / fresh snow, and to refer to the same 24-hour period. Snow accumulation is flagged if the minimum temperature is higher than 3 degrees Celsius.

Author(s)

Alba Gilabert, Yuri Brugnara

Examples

```
internal_consistency(rbind(Rosario$Tx, Rosario$Tn),
                     rbind(Meta$Tx, Meta$Tn),
                     outpath = tempdir())
```

Meta *Metadata for the stations of Bern and Rosario de Santa Fe*

Description

Metadata for the stations of Bern and Rosario de Santa Fe

Usage

```
Meta
```

Format

A list of data frames (one data frame per variable)

Source

Institute of Geography - University of Bern

plot_daily *Plot daily data points*

Description

Plot daily data points for custom intervals.

Usage

```
plot_daily(
  dailydata,
  len = 1,
  outfile,
  startyear = NA,
  endyear = NA,
  miss = TRUE,
  units = NA,
  ...
)
```

Arguments

dailydata	A character string giving the path of the input file, or a 5-column matrix with following columns: variable code, year, month, day, and the daily value.
len	Integer indicating the number of years shown in each panel.
outfile	Character string giving the path of the output pdf file.
startyear	First year to plot. If not indicated, all available years until endyear will be plotted.
endyear	Last year to plot. If not indicated, all available years since startyear will be plotted.

miss	If TRUE (the default), missing data are plotted as red crosses at the bottom of the plot.
units	Character string giving the units (will be printed in the y-axis). If dailydata is a path to a file, then the units are read from the SEF header.
...	Graphical parameters passed to the function plot, such as cex, lwd, pch, col, etc. (see par).

Details

The input file must follow the C3S Station Exchange Format (SEF).

Missing data are shown as red dots at the bottom of the plot.

Author(s)

Stefan Hunziker, Yuri Brugnara

References

Hunziker et al., 2017: Identifying, attributing, and overcoming common data quality issues of manned station observations. Int. J. Climatol, 37: 4131-4145.

Hunziker et al., 2018: Effects of undetected data quality issues on climatological analyses. Clim. Past, 14: 1-20.

Examples

```
plot_daily(Rosario$Tx, len = 2, outfile = paste0(tempdir(),"/test.pdf"))
```

---

plot_decimals *Plot decimals*

Description

Plot year-by-year distribution of the decimals in order to investigate the actual reporting resolution.

Usage

```
plot_decimals(Data, outfile, startyear = NA, endyear = NA)
```

Arguments

Data	A character string giving the path of the input file, or a 5 or 7-column matrix (depending on data type) with following columns: variable code, year, month, day, (hour), (minute), value.
outfile	Character string giving the path of the output pdf file.
startyear	First year to plot. If not indicated, all available years until endyear will be plotted.
endyear	Last year to plot. If not indicated, all available years since startyear will be plotted.

Details

The input file must follow the C3S Station Exchange Format (SEF).

Only the first digit after the decimal point is analysed. If there is more than one digit, the data will be rounded to the first decimal place.

Note

For precipitation and other bounded variables one needs to remove the values at the boundaries from the input data (e.g., zeros for precipitation).

Author(s)

Stefan Hunziker, Yuri Brugnara

References

Hunziker et al., 2017: Identifying, attributing, and overcoming common data quality issues of manned station observations. Int. J. Climatol, 37: 4131-4145.

Hunziker et al., 2018: Effects of undetected data quality issues on climatological analyses. Clim. Past, 14: 1-20.

Examples

```
plot_decimals(Rosario$Tx, outfile = paste0(tempdir(),"/test.pdf"))
```

---

plot_subdaily *Plot sub-daily data points*

Description

Plot sub-daily data points divided by month.

Usage

```
plot_subdaily(subdailydata, year = NA, outfile, fixed = TRUE, units = NA, ...)
```

Arguments

subdailydata	A character string giving the path of the input file, or a 7-column matrix with following columns: variable code, year, month, day, hour, minute, value.
year	Integer vector giving the year(s) to plot. If not specified (NA), all available years will be plotted. One pdf per year will be created.
outfile	Character string giving the path of the output pdf file. If year has more than one element or is NA, then this is a root to the filenames to which the year will be added ('root.year.pdf').
fixed	If TRUE (default), use the same y axis for all months. If FALSE, the axis limits are set based on the data range of each month.
units	Character string giving the units (will be printed in the y-axis). If subdailydata is a path to a file, then the units are read from the SEF header.
...	Graphical parameters passed to the function plot, such as cex, lwd, pch, col, etc. (see par).

Details

The input file must follow the C3S Station Exchange Format (SEF).

Creates one pdf for each year plotted.

Author(s)

Stefan Hunziker, Yuri Brugnara

References

Hunziker et al., 2017: Identifying, attributing, and overcoming common data quality issues of manned station observations. Int. J. Climatol, 37: 4131-4145.

Hunziker et al., 2018: Effects of undetected data quality issues on climatological analyses. Clim. Past, 14: 1-20.

Examples

```
plot_subdaily(Bern$p, year = 1803:1804, outfile = paste0(tempdir(),"/test"))
```

---

plot_weekly_cycle *Plot weekly cycle*

Description

Check if there is a significant weekly cycle in daily precipitation data by means of a binomial test.

Usage

```
plot_weekly_cycle(dailypcp, outpath, p = 0.95)
```

Arguments

dailypcp	A character vector giving the paths of the input files, or a list of 5-column matrices with following columns: variable code (must be 'rr'), year, month, day, value. The names of the list elements are assumed to be the station IDs.
outpath	Character string giving the path for the output files.
p	Probability threshold for the binomial test (default is 0.95).

Details

The input files must follow the C3S Station Exchange Format (SEF).

Creates one pdf for each station ('weekly.ID.pdf') plus one pdf with an overview of the entire dataset ('weekly.pdf').

Author(s)

Stefan Hunziker, Yuri Brugnara

References

Hunziker et al., 2017: Identifying, attributing, and overcoming common data quality issues of manned station observations. Int. J. Climatol, 37: 4131-4145.

Hunziker et al., 2018: Effects of undetected data quality issues on climatological analyses. Clim. Past, 14: 1-20.

Examples

```
plot_weekly_cycle(list(Rosario = Rosario$rr), outpath = tempdir())
```

---

qc *Apply all tests*

Description

Perform all quality tests at once on multiple stations and multiple variables.

Usage

```
qc(Data, Metadata = NULL, outpath, time_offset = 0)
```

Arguments

`Data`	Either a character vector of paths to SEF files, a data frame or a list of data frames with 7 columns (one data frame for each station): variable code, year, month, day, hour, minute, value. Each data frame can contain more than one variable code.
`Metadata`	A data frame with 7 columns: station ID, latitude, longitude, altitude, variable code, units, resolution. If `Data` is a list, the station IDs must appear in the same order of the respective elements in the list. 'resolution' can be either 'daily' (or 'd') or 'subdaily' (or 's'). If `Data` is a vector, `Metadata` is ignored and all the required information is read from the SEF files.
`outpath`	Character string giving the path where the output is saved.
`time_offset`	Numerical vector (of length 1 or equal to the number of analysed stations) of the number of hours to add to the time to obtain local time. This is used for tests on day and night temperature. Recycled for all stations if only one value is given. Data not stored in SEF files (i.e., not in UTC) are typically already expressed in local time: in this case the offset is zero (the default).

Details

This is a wrapper of all functions that can be applied to the variables given in `Data` (except the plotting functions).

`Data` can include any supported variable (see Variables) from different stations. The algorithm will select the tests that can be applied to each variable. Note that some tests require more than one variable from the same station.

This function produces flag files (one for each variable at each station). The filenames follow the standard 'qc_<stationID>_<varcode>_<resolution>.txt'. Each files contains a table of flagged values, with the last column indicating the tests failed by each flagged observation.

The flag files can be edited by hand to remove or add flags. The flags can then be added to the 'Meta' column of SEF files by using the function write_flags.

Note

The tests will use their default parameters (e.g. thresholds). To use custum parameters run the tests one by one.

Author(s)

Yuri Brugnara

Examples

```
# Testing all variables for Rosario de Santa Fe

# Create a data frame with all data from list Rosario
# For daily data we need to add the hour and minute columns (NAs)
Ros <- Rosario
Ros$Tx[,c("Hour","Minute")] <- NA
Ros$Tn[,c("Hour","Minute")] <- NA
Ros$rr[,c("Hour","Minute")] <- NA
Ros <- do.call("rbind", Ros)
Ros <- Ros[, c("Var","Year","Month","Day","Hour","Minute","Value")]

# Create a data frame with metadata including data resolution
df_meta <- do.call("rbind", Meta)
df_meta <- df_meta[which(df_meta$id=="Rosario"), ]
df_meta$res <- c("s", "s", "d", "d", "s", "s", "d", rep("s",4))

# Run all qc tests at once
# Time for Rosario is in UTC, therefore an offset is needed to get local time
qc(Ros, df_meta, outpath = tempdir(), time_offset=-4.28)


# Testing one variable at one station
qc(Bern$ta, cbind(Meta$ta[which(Meta$ta$id=="Bern"),],"s"),
   outpath = tempdir(), time_offset=0)
```

`read_meta` *Read metadata from the Station Exchange Format version 1.0.0*

Description

Read metadata from the Station Exchange Format version 1.0.0

Usage

```
read_meta(file = file.choose(), parameter = NULL)
```

Arguments

file	Character string giving the path of the data file.
parameter	Character vector of required parameters. Accepted values are "version", "id", "name", "lat", "lon", "alt", "source", "link", "var", "stat", "units", "meta". By default all parameters are read at once.

Value

A character vector with the required parameters.

Author(s)

Yuri Brugnara

---

read_sef *Read data files in Station Exchange Format version 1.0.0*

Description

Read data files in Station Exchange Format version 1.0.0

Usage

```
read_sef(file = file.choose(), all = FALSE)
```

Arguments

file	Character string giving the path of the SEF file.
all	If FALSE (the default), omit the columns 'Period' and 'Meta' (also 'Hour' and 'Minute' for non-instantaneous data)

Value

A data frame with up to 9 variables, depending on whether all is set to TRUE. The variables are: variable code, year, month, day, hour, minute, value, period, metadata.

Author(s)

Yuri Brugnara

Rosario — *Daily and sub-daily meteorological observations for Rosario de Santa Fe (Argentina)*

Description

Observations of minimum and maximum temperature, wind direction, cloud cover, pressure, precipitation, air temperature, wet bulb temperature, relative humidity, dew point, and wind speed for the city of Rosario de Santa Fe (Argentina) for the period 1886-1900.

Usage

```
Rosario
```

Format

A list of data frames (one data frame per variable). The format of the data frames is that required by the QC functions.

Source

ACRE

subdaily_out_of_range — *Sub-daily big errors test*

Description

Find the subdaily temperature (ta), wind speed (w), wind direction (dd), snow cover (sc), snow depth (sd) and fresh snow (fs) values that exceed thresholds selected by the user. The output is a list with the days in which ta, rr, dd, w, sc, sd or fs exceeds some threshold.

Usage

```
subdaily_out_of_range(
  subdailydata,
  meta = NULL,
  outpath,
  time_offset = 0,
  ta_day_upper = 45,
  ta_day_lower = -35,
  ta_night_upper = 40,
  ta_night_lower = -40,
  rr_upper = 100,
  rr_lower = 0,
  w_upper = 50,
  w_lower = 0,
  dd_upper = 360,
  dd_lower = 0,
  sc_upper = 100,
```

```
  sc_lower = 0,
  sd_upper = 200,
  sd_lower = 0,
  fs_upper = 100,
  fs_lower = 0
)
```

Arguments

subdailydata	A character string giving the path of the input file, or a 7-column matrix with following columns: variable code, year, month, day, hour, minute, value.
meta	A character vector with 6 elements: station ID, latitude, longitude, altitude, variable code, units. If subdailydata is a path, meta is ignored.
outpath	Character string giving the path for the QC results.
time_offset	Offset in hours to add to the time to obtain local time. By default, time_offset = 0.
ta_day_upper	is the ta maximum day threshold in degrees Celsius. By default, ta_day_upper = 45 C.
ta_day_lower	is the ta minimum day threshold in degrees Celsius. By default, ta_day_lower = -35 C.
ta_night_upper	is the ta maximum night threshold in degrees Celsius. By default, ta_night_upper = 40 C.
ta_night_lower	is the ta minimum night threshold in degrees Celsius. By default, ta_night_lower = -40 C.
rr_upper	is the rr maximum threshold in millimetres. By default, rr_upper = 100 mm.
rr_lower	is the rr minimum threshold in millimetres. By default, rr_lower = 0 mm.
w_upper	is the w maximum threshold in metres per second. By default, w_upper = 50 m/s.
w_lower	is the w mimumum threshold in metres per second. By default, w_lower = 0 m/s.
dd_upper	is the dd maximum threshold in degrees North. By default, dd_upper = 360.
dd_lower	is the dd minimum threshold in degrees North. By default, dd_lower = 0.
sc_upper	is the sc maximum threshold in percent. By default, sc_upper = 100%.
sc_lower	is the sc minimum threshold in percent. By default, sc_lower = 0%.
sd_upper	is the sd maximum threshold in centimetres. By default, sd_upper = 200 cm.
sd_lower	is the sd minimum threshold in centimetres. By default, sd_lower = 0 cm.
fs_upper	is the fs maximum threshold in centimetres. By default, fs_upper = 100 cm.
fs_lower	is the fs minimum threshold in centimetres. By default, fs_lower = 0 cm.

Details

The input file must follow the Copernicus Station Exchange Format (SEF).

Author(s)

Alba Gilabert, Yuri Brugnara

Examples

```
subdaily_out_of_range(Rosario$ta, Meta$ta[which(Meta$ta$id=="Rosario"),],
                      outpath = tempdir(), time_offset = -4.28,
                      ta_day_upper = 35)
```

---

subdaily_repetition *Sub-daily repetition test*

Description

Report occurrences of equal consecutive values in subdaily data.

Usage

```
subdaily_repetition(subdailydata = file.choose(), meta = NULL, outpath, n = 6)
```

Arguments

subdailydata	A character string giving the path of the input file, or a 7-column matrix with following columns: variable code, year, month, day, hour, minute, value.
meta	A character vector with 6 elements: station ID, latitude, longitude, altitude, variable code, units. If subdailydata is a path, meta is ignored.
outpath	Character string giving the path for the QC results.
n	Number of minimum equal consecutive values required for a flag. The default is 6.

Details

The input file must follow the Copernicus Station Exchange Format (SEF).

Zeroes are automatically excluded in bounded variables such as precipitation.

Author(s)

Alba Gilabert, Yuri Brugnara

Examples

```
subdaily_repetition(Rosario$ta, Meta$ta[which(Meta$ta$id=="Rosario"),],
                    outpath = tempdir(), n = 3)
```

`temporal_coherence` *Daily temporal coherence test*

Description

Find those records where daily maximum or minimum temperature, mean wind speed, snow depth, snow cover, or fresh snow differences with previous day are too large.

Usage

```
temporal_coherence(
  dailydata,
  meta = NULL,
  outpath,
  temp_jumps = 20,
  windspeed_jumps = 15,
  snowdepth_jumps = 50
)
```

Arguments

`dailydata`	A character string giving the path of the input file, or a 5-column matrix with following columns: variable code, year, month, day, and the daily value.
`meta`	A character vector with 6 elements: station ID, latitude, longitude, altitude, variable code, units. If `dailydata` is a path, `meta` is ignored.
`outpath`	Character string giving the path for the QC results.
`temp_jumps`	given a daily maximum or minimum temperature values of two consecutive days, maximum difference in degrees Celsius. By default, temp_jumps = 20 C.
`windspeed_jumps`	given a daily mean wind speed value of two consecutive days, maximum difference in metres per second. By default, wind_jumps = 15 m/s.
`snowdepth_jumps`	given a daily snow depth of two consecutive days, maximum difference in centimetres. By default, snowdepth_jumps = 50 cm.

Details

The input file must follow the Copernicus Station Exchange Format (SEF).

Author(s)

Alba Gilabert, Yuri Brugnara

Examples

```
temporal_coherence(Rosario$Tx, Meta$Tx, outpath = tempdir(), temp_jumps = 10)
```

Tests — *Table of available qc tests*

Description

Table of available qc tests

Usage

```
Tests
```

Format

Data frame

Variables — *Table of supported variable codes*

Description

Table of supported variable codes

Usage

```
Variables
```

Format

Data frame

wmo_gross_errors — *WMO Gross Errors Tests for Pressure, Temperature, Dew Point, and Wind Speed.*

Description

Applicable to a series (daily or sub-daily) of air pressure, air temperature (ta), dew point temperature (td), wind speed (w). The pressure series can be at mean sea level (mslp) or at station level (p). Flags the records where the observations values exceed the limit values given by WMO (1993).

Usage

```
wmo_gross_errors(series, meta = NULL, outpath)
```

Arguments

`series`	A character string giving the path of the SEF file, or a five or seven-column (daily or subdaily) data frame with the series.
`meta`	A character vector with 6 elements: station ID, latitude, longitude, altitude, variable code, units. If `series` is a path, `meta` is ignored.
`outpath`	Character string giving the path for the QC results.

Details

Input:

- A SEF file or a data frame and metadata. The observations data frame must have five or seven columns: variable code, year (YYYY), month (MM), day (DD), (hour (HH), minute (MM)), observation. The required metadata are the station identifier, the station latitude and variable units.

The WMO gross error limits for air pressure, air temperature, dew point temperature, and wind speed:

- For station level pressure the gross error limits are latitude and meteorological season independent (WMO, 1993: VI.7). According to the same reference, for mean sea level pressure, temperature, dew point, and wind speed, the WMO establishes the gross error limits as function of the station latitude and the meteorological season in which the observations were collected.
- The tests divide the meteorological seasons in Winter and Summer. So, based on the meteorological calendar for the Northern Hemisphere, which defines seasons as Spring (March, April, May), Summer (June, July, August), Autumn (September, October, November) and Winter (December, January, February), it was here considered:
  - Northern Hemisphere Winter / Southern Hemisphere Summer - January, February, March, October, November, December;
  - Northern Hemisphere Summer / Southern Hemisphere Winter - April, May, June, July, August, September.
- The gross error limits for each variable divide the flagged values in suspect and erroneous (WMO, 1993: VI.6 - VI.8).
- **Latitude independent**
  - **Meteorological season independent**
    - **Station Level Pressure (p):**
      - Suspect: *300 <= p < 400 hPa or 1080 < p <= 1100 hPa*
      - Erroneous: *p < 300 or p > 1100 hPa*
- **Latitudes belonging to the interval [-45, +45]**
  - **Winter**
    - **Mean Sea Level Pressure (mslp)**
      - Suspect: *870 <= mslp < 910 hPa or 1080 < mslp <= 1100 hPa*
      - Erroneous: *mslp < 870 hPa or mslp > 1100 hPa*
    - **Air Temperature (ta)**
      - Suspect: *-40 <= ta < -30 ºC or 50 < ta <= 55 ºC*
      - Erroneous: *ta < -40 ºC or ta > 55 ºC*
    - **Dew Point Temperature (td)**
      - Suspect: *-45 <= td < -35 ºC or 35 < td <= 40 ºC*

        - Erroneous: *td < -45 ºC or td > 40 ºC*
      - **Wind Speed (w)**
        - Suspect: *w > 60 m/s and w <= 125 m/s*
        - Erroneous: *w > 125 m/s*
    - **Summer**
      - **Mean Sea Level Pressure (mslp)**
        - Suspect: *850 <= mslp < 900 hPa or 1080 < mslp <= 1100 hPa*
        - Erroneous: *mslp < 850 hPa or mslp > 1100 hPa*
      - **Air Temperature (ta)**
        - Suspect: *-30 <= ta < -20 ºC or 50 < ta <= 60 ºC*
        - Erroneous: *ta < -30 ºC or ta > 60 ºC*
      - **Dew Point Temperature (td)**
        - Suspect: *-35 <= td < -25 ºC or 35 < td <= 40 ºC*
        - Erroneous: *td < -35 ºC or td > 40 ºC*
      - **Wind Speed (w)**
        - Suspect: *w > 90 m/s and w <= 150 m/s*
        - Erroneous: *w > 150 m/s*
- **Latitudes belonging to the interval [-90, -45[U]+45, +90]**
  - **Winter**
    - **Mean Sea Level Pressure (mslp)**
      - Suspect: *910 <= mslp < 940 hPa or 1080 < mslp <= 1100 hPa*
      - Erroneous: *mslp < 910 hPa or mslp > 1100 hPa*
    - **Air Temperature (ta)**
      - Suspect: *-90 <= ta < -80 ºC or 35 < ta <= 40 ºC*
      - Erroneous: *ta < -90 ºC or ta > 40 ºC*
    - **Dew Point Temperature (td)**
      - Suspect: *-99 <= td < -85 ºC or 30 < td <= 35 ºC*
      - Erroneous: *td < -99 ºC or td > 35 ºC*
    - **Wind Speed (w)**
      - Suspect: *w > 50 m/s and w <= 100 m/s*
      - Erroneous: *w > 100 m/s*
  - **Summer**
    - **Mean Sea Level Pressure (mslp)**
      - Suspect: *920 <= mslp < 950 hPa or 1080 < mslp <= 1100 hPa*
      - Erroneous: *mslp < 920 hPa or mslp > 1100 hPa*
    - **Air Temperature (ta)**
      - Suspect: *-40 <= ta < -30 ºC or 40 < ta <= 50 ºC*
      - Erroneous: *ta < -40 ºC or ta > 50 ºC*
    - **Dew Point Temperature (td)**
      - Suspect: *-45 <= td < -35 ºC or 35 < td <= 40 ºC*
      - Erroneous: *td < -45 ºC or td > 40 ºC*
    - **Wind Speed (w)**
      - Suspect values: *w > 40 m/s and w <= 75 m/s*
      - Erroneous values: *w > 75 m/s*

Output:

- A text file of flagged observations with six or eight columns: variable code, year (YYYY), month (MM), day (DD), (hour (HH), minute (MM)), observation, test. The test column has the description "gross_errors".

Author(s)

Clara Ventura, Yuri Brugnara

References

WMO, 1993: Chapter 6 - Quality Control Procedures. Guide on the Global Data-processing System, World Meteorological Organization, Geneva, No. 305, VI.1-VI.27, ISBN 92-63-13305-0.

Examples

```
wmo_gross_errors(series = Rosario$p, meta = Meta$p[which(Meta$p$id=="Rosario"),],
                 outpath = tempdir())
wmo_gross_errors(series = Rosario$ta, meta = Meta$ta[which(Meta$p$id=="Rosario"),],
                 outpath = tempdir())
wmo_gross_errors(series = Rosario$td, meta = Meta$td, outpath = tempdir())
```

---

wmo_time_consistency *WMO Time Consistency Test for Pressure, Temperature and Dew Point.*

Description

Applicable to a series of sub-daily air pressure (p, mslp), air temperature (ta) or dew point temperature (td) observations with at least some time intervals between observations less or equal to twelve hours. Flags the records where the observations exceed the WMO suggested tolerances for the temperatures and pressure tendency as function of time period between consecutive reports.

Usage

```
wmo_time_consistency(series, meta = NULL, outpath)
```

Arguments

`series`	A character string giving the path of the input file, or a 7-column matrix with following columns: variable code, year, month, day, hour, minute, value.
`meta`	A character vector with 6 elements: station ID, latitude, longitude, altitude, variable code, units. If `series` is a path, `meta` is ignored.
`outpath`	Character string giving the path for the QC results.

Details

Input:

- A SEF file or a data frame and metadata. The observations data frame must have seven columns: variable code, year (YYYY), month (MM), day (DD), hour (HH), minute (MM), observation.

The WMO time consistency test:

- WMO suggested tolerances for the temperatures and pressure tendency as function of time period between consecutive reports (WMO, 1993: VI.21):

Parameter	dt = 1 hour	dt = 2 hours	dt = 3 hours	dt = 6 hours	dt = 12 hours
ta_tol	4 ºC	7 ºC	9 ºC	15 ºC	25 ºC
td_tol	4 ºC	6 ºC	8 ºC	12 ºC	20 ºC
pp_tol	3 hPa	6 hPa	9 hPa	18 hPa	36 hPa

- The temperatures tolerance - *ta_tol* and *td_tol* - considered for 1, 2, 3, 6 and 12 hours is given by the table above.
- The pressure tolerance *p_tol* is determined for time intervals belonging to [1, 12] hours, assuming that there is a linear variation of 3 hPa per hour, based in the table above.
- Time consistency test (WMO, 1993: VI.21):
  $|obs(t) - obs(t - dt)| > tol => flag_obs(t) = suspect and flag_obs(t - dt) = suspect$
  $obs - observation(ta, td or p), t - hour in decimal, dt - hour difference in decimal, tol - tolerance$
- The flag, correspondent to suspect values, is always associated with two consecutive observations within twelve hours.

Output:

- A text file of flagged observations with eight columns: variable code, year, month, day, hour, minute, value, test. The test column has the description "wmo_time_consistency".

Author(s)

Clara Ventura, Yuri Brugnara

References

WMO, 1993: Chapter 6 - Quality Control Procedures. Guide on the Global Data-processing System, World Meteorological Organization, Geneva, No. 305, VI.1-VI.27, ISBN 92-63-13305-0.

Examples

```
wmo_time_consistency(series = Bern$p, meta = Meta$p[which(Meta$p$id=="Bern"),],
                     outpath = tempdir())
```

`write_flags` *Add quality flags to a data file in Station Exchange Format version 1.0.0*

Description

Add quality flags to a data file in Station Exchange Format version 1.0.0

Usage

```
write_flags(infile, qcfile, outpath, note = "")
```

Arguments

infile	Character string giving the path of the SEF file.
qcfile	Character string giving the path of the file with the quality flags as produced with the QC tests. This file must have 6 (8) tab-separated columns for daily (sub-daily) data: variable code, year, month, day, (hour), (minute), value, semicolon(';')-separated failed tests.
outpath	Character string giving the output path.
note	Character string to be added to the end of the name of the input file to form the output filename. It will be separated from the rest of the name by an underscore. Blanks will be also replaced by underscores. If not specified, input and output filenames will be identical.

Note

The data will be converted to the standard units adopted by the qc (see Variables). An exception is made for cloud cover (oktas will not be converted).

Author(s)

Yuri Brugnara

---

write_sef *Write data in Station Exchange Format version 1.0.0*

Description

Write data in Station Exchange Format version 1.0.0

Usage

```
write_sef(
  Data,
  outpath,
  variable,
  cod,
  nam = "",
  lat = "",
  lon = "",
  alt = "",
  sou = "",
  link = "",
  units,
  stat,
  metaHead = "",
  meta = "",
  period = "",
  time_offset = 0,
  note = "",
  keep_na = FALSE,
  outfile = NA
)
```

Arguments

Data	A data frame with 6 variables in this order: year, month, day, hour, minute, value.
outpath	Character string giving the output path (note that the filename is generated from the source identifier, station code, start and end dates, and variable code).
variable	Variable code. This is a required field.
cod	Station code. This is a required field.
nam	Station name.
lat	Station latitude (degrees North in decimal).
lon	Station longitude (degreees East in decimal).
alt	Station altitude (metres).
sou	Character string giving the source identifier.
link	Character string giving an url for metadata (e.g., link to the C3S Data Rescue registry).
units	Character string giving the units. This is a required field.
stat	Character string giving the statistic code. This is a required field.
metaHead	Character string giving metadata entries for the header (pipe separated).
meta	Character vector with length equal to the number of rows of Data, giving metadata entries for the single observations (pipe separated).
period	Observation time period code. Must be a character vector with length equal to the number of rows of Data unless all observations have the same period code.
time_offset	Numerical vector of offsets from UTC in hours. This value will be subtracted from the observation times to obtain UTC times, so for instance the offset of Central European Time is +1 hour. Recycled for all observations if only one value is given.
note	Character string to be added to the end of the standard output filename. It will be separated from the rest of the name by an underscore. Blanks will be also replaced by underscores.
keep_na	If FALSE (the default), lines where observations are NA are removed.
outfile	Output filename. If specified, ignores note.

Note

Times in SEF files must be expressed in UTC.

If outfile is not specified, the output filename is generated automatically as sou_cod_startdate_enddate_variable.tsv

Author(s)

Yuri Brugnara

Examples

```
# Create a basic SEF file for air temperature in Bern
# (assuming the observation times are in local solar time)
# The file will be written in the working directory
meta_bern <- Meta$ta[which(Meta$ta$id == "Bern"), ]
write_sef(Bern$ta[, 2:7], outpath = tempdir(), variable = "ta",
```

```
cod = meta_bern$id, nam = "Bern", lat = meta_bern$lat,
lon = meta_bern$lon, alt = meta_bern$alt,
units = meta_bern$units, stat = "point", period = "0",
time_offset = meta_bern$lon * 24 / 360)
```

Index