Package ‘swisswatertemp’

January 9, 2020

Type Package

Title Swiss water temperature analysis

Version 1.1.1

Author Adrien Michel

Maintainer Adrien Michel <adrien.michel@epfl.ch>

Description This package allow to reproduce the analysis and results presented in: 'Stream temperature and discharge evolution in Switzerland over the last 50 years: annual and seasonal behaviour, Adrien Michel, Tristan Brauchli, Michael Lehning, Bettina Schaefli, and Hendrik Huwald, Hydrol. Earth Syst. Sci., 2020'.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports data.table, rgdal, sp, raster, zoo, gridExtra, GISTools, RColorBrewer, graphics, Hmisc, lubridate, Partiallyoverlapping, MASS

R topics documented:

`add_means` *Return a SMET_OBJECT with daily, monthly, seasonnal, and yerly means added*

Description

Return a SMET_OBJECT with daily, monthly, seasonnal and yearly means added. The`data.frame` added are `$daily` (daily means), `$monthly` (monthly means), `$yearly` (yearly means), `$DJF` (winter means), `$MAM` (spring means), `$JJA` (summer means), and `$SON` (fall means)

Usage

```
add_means(data)
```

Arguments

`data`	SMET_OBJECT

Value

SMET_OBJECT

add_meteo *Add meteo data to SMET_OBJECT*

Description

Read SMET_OBJECT , returns a SMET_OBJECT with meteo stations indicated in `$meteo` added as a SMET_OBJECT object list in `$meteo_ts`.

Usage

```
add_meteo(data, meteo_data)
```

Arguments

`data`	SMET_OBJECT containing river data
`meteo_data`	SMET_OBJECT continning meteo data

Value

SMET_OBJECT with meteo added as a SMET_OBJECT object list `$meteo_ts`

add_meteo_and_trim *Wrapper function*

Description

Wrapper for `cut_all_data` and `add_meteo` functions

Usage

```
add_meteo_and_trim(data, meteo_data, start, end)
```

Arguments

`data`	A SMET_OBJECT with river data
`meteo_data`	A SMET_OBJECT with meteo data
`start`	Starting year to keep
`end`	Ending year to keep

Value

SMET_OBJECT cut to the given dates with meteo data added

check_and_cut_variables

Check that variable and cut them to the same length

Description

Check that both water temperature (T) and discharge (Q) are provided and cut the daily means of the two variables to have the same start and end

Usage

```
check_and_cut_variables(data)
```

Arguments

data	A SMET_OBJECT obtained through the function get_file_data containing the data of one river station

Value

A SMET_OBJECT with only the `$data` part filled with the cut daily means T and Q time series.

compute_daily_means_over_period

Compute the mean of Julian days value over a period

Description

Compute the day-of-the-yaer mean over a period of a given length in year. The day-of-the-yaer mean is the mean between each first of jannuary, each second of jannuary, and-so-on. The periods are tkaen from the end of the available data and as many periods as possible are taken.

Usage

```
compute_daily_means_over_period(data, period)
```

Arguments

data	A SMET_OBJECT obtained through the function get_file_data containing the data of one river station
period	An integer representing the number of year over whihch the average should be computed

Value

A list with the period as key. Periods have the format "AAAA-BBBB" where AAAA is the starting year and BBBB the ending year. Each list element contains a `data.frame` with columns T and Q, which are the day-of-the-yaer mean over the period for the water temperature and the discharge.

`cut_all_data` *Cut timeseries in a SMET_OBJECT*

Description

Read a SMET_OBJECT, returns a SMET cut between the years indicated by start and end for the inner `data.frame` `$data`,`$daily`, `$monthly`, and `$yearly`

Usage

```
cut_all_data(data, start, end)
```

Arguments

`data`	SMET_OBJECT
`start`	Starting year to keep
`end`	Ending year to keep

Value

SMET_OBJECT

`cut_data` *Read a SMET_OBJECT object, returns a SMET cut between the years indicated by start and end*

Description

Read a SMET_OBJECT object, returns a SMET cut between the years indicated by start and end

Usage

```
cut_data(data, start, end)
```

Arguments

`data`	A SMET_OBJECT
`start`	Starting year to keep
`end`	Ending year to keep

Value

A SMET_OBJECT where timeseries are cut between start and end

cut_full_year *Cut SMET_OBJECT to only keep complete years*

Description

Cut SMET_OBJECT to only keep complete years (partial years at the beggining or at the end of the time series are removed)

Usage

```
cut_full_year(data)
```

Arguments

data	SMET_OBJECT

Value

SMET_OBJECT

daily_mean_for_year *Compute the daily mean over a year*

Description

Compute the daily mean over a year for all raw variables in SMET_OBJECT obtained through the function `get_file_data` (raw variables are stored in the `$data` part of theSMET_OBJECT).

Usage

```
daily_mean_for_year(data, yr)
```

Arguments

data	A SMET_OBJECT for a given station
yr	The year over which the data should be daily averaged

Value

A `data.frame` the daily mean of each variables (the timestamp is removed).

genreal_analysis_plots *Plot variable distributions and compute t-tests*

Description

This function produces the plots variable distributions as shown in Figures 5, 6, and 10, and in Figures S14, and S16 to S21 in supplementary. It also computes wilcox test shown in Table S3 in supplementary and print the results to the console.

Usage

```
genreal_analysis_plots(period, rivers_data)
```

Arguments

`period`	A string, either "1999-2018", "1979-1998", "1979-2018", or "1970-2018" defining the period over which the plots and analysis should be produced
`rivers_data`	Either `"NONE"` (default), `"PDF"` or `"PNG"`.

get_data *Read raw of river or meteo data*

Description

Read raw of river or meteo data. Data can be read from files or from RDS data. All files to be read must be SMET files (see `https://models.slf.ch/docserver/meteoio/SMET_specifications.pdf`). Returns a list of SMET_OBJECT (see `get_file_data`), each entry corresponding to one station.

Usage

```
get_data(list, path, type, RData = FALSE)
```

Arguments

`list`	A list of lists. Each inner list contains the name of the file where the temperature and discharge data, or the meteoSwiss data, are located. If data to load are temperature and discharge data, a vector containing the three capital letters apreviation of the associated meteoSwiss stations must also be in the inner lists for each station.
`path`	The relative path the prepend to the files to be read.
`type`	Either `"WATER"` or `"METEO"` to specify if the data to be loaded are meteoSwiss data
`RData`	A `RDS` objec tto be loaded. If provided data won't be loaded from SMET files

Value

A list of SMET object continning the data. The keys are the input files names (without extension).

`get_day` *Returns day from a PosixCT (or vector of PosixCT)*

Description

Returns day from a PosixCT (or vector of PosixCT)

Usage

```
get_day(timestamp)
```

Arguments

`timestamp`	A PosixCT or vector of PosixCT

Value

The corresponding day (as numeric)

`get_file_data` *Read data from a SMET file, returns a SMET_OBJECT*

Description

Read data from a SMET file, returns a SMET_OBJECT with only full years, bisssextile days removed, daily, monthly, seasonal, and yearly means computed optionally cut between indicated years, and optoptionally associated with a list of meteo stations.

Usage

```
get_file_data(file, start = NULL, end = NULL, meteo = NULL)
```

Arguments

`file`	Path of the SMET file to be loaded
`start`	Starting year to keep (default `NULL`)
`end`	Ending year to keep(default `NULL`)
`meteo`	Three capital letters apreviation of the associated meteoSwiss stations (default `NULL`)

Details

The inner data are `$header` (SMET header),`$data` (raw data), `$daily` (daily means), `$monthly` (monthly means), `$yearly` (yearly means), `$DJF` (winter means), `$MAM` (spring means), `$JJA` (summer means), and`$SON` (fall means). The meteo station list is accissible through `$meteo`.

Value

SMET_OBJECT

get_file_data_only *Read data from a SMET file, returns a SMET_OBJECT without means computed*

Description

Same as get_file_data but does not add the mean values.

Usage

```
get_file_data_only(file, start = NULL, end = NULL, meteo = NULL)
```

Arguments

file	Path of the SMET file to be loaded
start	Starting year to keep (default NULL)
end	Ending year to keep (default NULL)
meteo	Three capital letters apreviation of the associated meteoSwiss stations (default NULL)

Value

A list of SMET object continning the data. The keys are the input files names (without extension).

get_hysteresis_data *Produce hysteresis data*

Description

Read data from a list of SMET_OBJECT, which are obtained through the function get_file_data and contain the data of one river station. It keeps only subsequent years and compute necessary values for hysteresis plots (daily means over years and smoothed daily means over years)

Usage

```
get_hysteresis_data(rivers_data, period, smoothing)
```

Arguments

rivers_data	A list of SMET_OBJECT, which are obtained through the function get_file_data and contain the data of one river station.
period	The lenght (in year) of the periods over which the hystheresis data should be computed
smoothing	The length (in day) of the moving average window to be applied

Value

The input SMET_OBJECT whih a new list entry, "hysteresis", containing the hysteresis data. The hysteresyis data are discharge and temperature values, averaged for each day of the year separately over various periods and smoothed with a circular moving average window. The new "hysteresis" antry of the SMET_OBJECT contains a list where the keys are the periods (in the format "AAAA-BBBB" where AAAA is the starting year and BBBB the ending year) and the associated data are a `data.frame` containing discharge and temperature values.

`get_lm_summary` *Return the summary of `lm` model in a list of numeric values*

Description

The values retuned are obtained by using `summary` on the `lm` object

Usage

```
get_lm_summary(lm)
```

Arguments

`lm`	An object generated by `lm`

Value

A list with the folowing entries, all as numeric:

`intercept`	The intercept value
`trend`	The trend value
`intercept_std`	The std error of the intercept value
`trend_std`	The std error of the trend value
`intercept_p`	The p-value of the intercept value
`trend_p`	The p-value of the trend value
`r_squared`	The R^2 value
`adj_r_squared`	Teh adjusted R R^2 value

get_lm_summary_printable

Return the summary of `lm` model in a list of strings

Description

The values retuned are obtained by using `summary` on the `lm` object

Usage

```
get_lm_summary_printable(lm)
```

Arguments

`lm`	An object generated by `lm`

Value

A list with the folowing entries, all as strings and rounded to signifigant digits:

`intercept`	The intercept value
`trend`	The trend value
`intercept_std`	The std error of the intercept value
`trend_std`	The std error of the trend value
`intercept_p`	The p-value of the intercept value
`trend_p`	The p-value of the trend value
`r_squared`	The R^2 value
`adj_r_squared`	Teh adjusted R R^2 value

get_month

Returns month from a PosixCT (or vector of PosixCT)

Description

Returns month from a PosixCT (or vector of PosixCT)

Usage

```
get_month(timestamp)
```

Arguments

`timestamp`	A PosixCT or vector of PosixCT

Value

The corresponding month (as numeric)

get_remainder_analysis *Perform analysis of the remainder of the STL*

Description

Get the ACF and PACF for the remainder of the STL analysis. For meterological stations, the CCF between the hydrological variable and the meteorological variable is also computed.

Usage

```
get_remainder_analysis(STL_output, s_windows, version)
```

Arguments

`STL_output`	List of output from the STL analysis, where keys are `s.window` values.
`s_windows`	Vector of `s.window` values for the stl analysis (see `link[stats]{stl}`)
`version`	"raw" or "post_seasons_loess" depending if a "post_seasons_loess" has be applied (whihc is the case when `get_STL_analysis`) is run at daily time scale.

Value

aa

get_STL_analysis *Genenral call to perform STL analysis.*

Description

Genenral call to perform STL analysis.

Usage

```
get_STL_analysis(rivers_data, meteo_data, variable, s_win, frequency,
  start = NULL, end = NULL)
```

Arguments

`rivers_data`	A SMET_OBJECT with river data
`meteo_data`	A SMET_OBJECT with meteo data
`variable`	Variable on which the STL analysis should be performed
`s_win`	Vector of `s.window` values for the stl analysis (see `link[stats]{stl}`)
`frequency`	Frequency of data do be used. "monthly" or "monthly".
`start`	Year to start the STL analysis (years before are cut). Default `NULL`.
`end`	Year to end the STL analysis (years after are cut). Default `NULL`.

Value

A list where each key the station name and each value is a list. In this list each key is a value of `s.window`. The content in each list entry is the output of the STL analysis. In addiditon, the entry `$meteo` gives access to a list where keys are the meteo station three capital letters abbreviation and the content is a list with keys `s.window` containing the STL analysis for the meteorological variable.

`get_STL_output` *Perform STL analysis*

Description

Perform STL analysis for the given variable and the associated mete variable (air temperature if the variable is water temperature and precipitation if the river variable is discharge)

Usage

```
get_STL_output(station_data, meteo_data, variable, s_win, frequency)
```

Arguments

`station_data`	A SMET_OBJECT with river data
`meteo_data`	A SMET_OBJECT with meteo data
`variable`	Variable on which the STL analysis should be performed
`s_win`	vector of `s.window` values for the stl analysis (see `link[stats]{stl}`)
`frequency`	Frequency of data do be used. "monthly" or "monthly".

Value

A list where each key is a value of `s.window`. The content in each list entry is the output of the STL analysis. In addiditon, the entry `$meteo` gives access to a list where keys are the meteo station three capital letters abbreviation and the content is a list with keys `s.window` containing the STL analysis for the meteorological variable.

`get_timestep` *Returns the timestep of the timeseries in a SMET_OBJECT*

Description

Returns the timestep of the timeseries in a SMET_OBJECT. An error is thrown if the timestep is not constant

Usage

```
get_timestep(data)
```

Arguments

`data`	SMET_OBJECT

Value

Timestep of the timeseries

get_time_series *Convert data from SMET_OBJECT to ts objects*

Description

Read a SMET_OBJECT, returns a list of `data.frame` containing data as `ts` objects. Input data.frames are split into list of times series, each list entry being one variable.

Usage

```
get_time_series(data)
```

Arguments

`data`	A SMET_OBJECT

Value

A SMET_OBJECT with all inner `data.frame` transformed into lists of `ts` objects

get_year *Returns year from a PosixCT (or vector of PosixCT)*

Description

Returns year from a PosixCT (or vector of PosixCT)

Usage

```
get_year(timestamp)
```

Arguments

`timestamp`	A PosixCT or vector of PosixCT

Value

The corresponding year (as numeric)

`keep_subs_years_preprocessing` *Filter tiem series for complete years*

Description

Remove years starting or ending with NaN, kepps only subsequent years, starting from the end of the timeseries.

Usage

```
keep_subs_years_preprocessing(data)
```

Arguments

`data`	A SMET_OBJECT

Value

A SMET_OBJECT

`plot_acf` *Plot acf and pacf of the residuals of the STL analysis*

Description

This function plots the acf and pacf of the residuals of the STL analysis for the four variables T, Q, TA and P for the water station passed in parameters and the associated meteoSwiss stations. This produces the plots shown in Figures S7 and S8 in supplementary.

Usage

```
plot_acf(station, output_type = "NONE")
```

Arguments

`station`	A SMET object containing the data for one station
`output_type`	Either `"NONE"` (default), `"PDF"` or `"PNG"`. `output_type = "NONE"` creates the plot in a normal plot window (default), `output_type = "PDF"` saves the plot as pdf in plots/analysis/'station_name'/, `output_type = "PNG"` saves the plot as png in plots/analysis/'station_name'/,

`plot_alpine` *Produces plot for alpine catchemts*

Description

This function produces plot for alpine catchments (Figure S35). The plots are saved in plots/alpine.pdf

Usage

```
plot_alpine(rivers_data)
```

Arguments

`rivers_data`	The dataset of rivers data

`plot_correlations` *Print correlation matrices*

Description

This function prints to the console the correlations matrices shown in Tables 3 and in Tables S6 and S7 in supplementary.

Usage

```
plot_correlations(rivers_data)
```

Arguments

`rivers_data`	The dataset of rivers data

Details

Some additional plots nor present in the paper are also produced and saveud under plots/correlations_plots.pdf

plot_general *Produce general T and Q plot and variance plot*

Description

This function produces general T and Q plot (all the catchments, Figures 2 and 3). The plot for T also contains a lower pannel showing the decadan anomalies (Figure 2). This function also produces the plot of the evolution of the infra-annual variability (Figure S34 in supplementary). Plots are written in the 'plots' directory. The plots are saved in plots/general_plot.pdf, plots/general_plot_Q.pdf, and plots/annual_var.pdf

Usage

```
plot_general(rivers_data)
```

Arguments

`rivers_data` The dataset of rivers data

plot_general_situation *Plot a map with the location of the river station and meteostation used*

Description

Plot a map with the location of the river station and meteostation used

Usage

```
plot_general_situation(rivers_data, output_type = "NONE")
```

Arguments

`rivers_data` A list of SMET_OBJECT obtained through the function `get_file_data` containing the data on the rivers stations

`output_type` Either "NONE" (default), "PDF" or "PNG".

`output_type = "NONE"` creates the plot in a normal plot window,

`output_type = "PDF"` saves the plot as pdf under plots/General_situation.pdf,

`output_type = "PNG"` saves the plot as png under plots/General_situation.png

Requirements

This functions needs the following files to be abailable: maps/processed_maps/swiss_map.tif, maps/processed_maps/lake
maps/processed_maps/rivers.shp, maps/processed_maps/borders.shp, meteo/MeteoSwiss_StationList.txt.
In addition, the plot directory must exist.

plot_hysteresis *Plot hysteresis*

Description

This function plots the day-of-the -year decadal mean of Q and T for the given station along wiht the Q-T hysteresis plot. This function produces the plot shown in Figure 15

Usage

```
plot_hysteresis(station, output_type = "NONE")
```

Arguments

`station`	A SMET object containing the data for one station
`output_type`	Either `"NONE"` (default), `"PDF"` or `"PNG"`. `output_type = "NONE"` creates the plot in a normal plot window (default), `output_type = "PDF"` saves the plot as pdf in plots/analysis/'station_name'/, `output_type = "PNG"` saves the plot as png in plots/analysis/'station_name'/,

plot_lakes *Produces plots for lakes*

Description

This function produces the plots for the trends before and after lakes, shown in Figure 7 and in Figures S22 to S25 in supplementary. Figures are saved under plots/lakes_plots.pdf

Usage

```
plot_lakes(rivers_data)
```

Arguments

`rivers_data`	The dataset of rivers data

plot_long_term *Produce long term anomaly plots*

Description

This function produces the decadal animalies plots (Figure 4 and Figure S13 in suplpementary), seasonnal decadal anomalies pots (Figure 8 and 9, and Figures S28 and S27 in suplementary) and hysteresys plots (Figure 12 and Figures S36 and S37 in supplementary). These Figures are saved in plots/long_term_plots.pdf. This function also print to the console the partially overlapping samples two-sided t-test (see section 4.1) and the figure showing discharge and precipitation decadal anomalies along with the NAO and AMO (Figure S15 in suplpementary).

Usage

```
plot_long_term(rivers_data, meteo_data)
```

Arguments

`rivers_data`	The dataset of rivers data
`meteo_data`	The dataset of homegenous MeteoSwiss data

Details

Note that plots of meteorological data use meteo stations related to water station except the long term precipitation decadal anomalies plot (Figure 4 and Figure S15 in suplpementary), which uses all available homegenous MeteoSwiss data not necessarly linked to catchments (as stated in the paper).

plot_remainder_analysis *Title*

Description

Title

Usage

```
plot_remainder_analysis(STL_output, s_windows, version, name)
```

Arguments

`STL_output`	a
`s_windows`	a
`version`	a
`name`	a

Value

a

plot_snow *Produce snow cover and glacier mass balance plots*

Description

This function produces the monthly snow cover plots (Figures S31 and S32 in suplpementary) and the glaciers mass balance plot (Figure S33 in supplementary). The plots are saved in plots/snow_plots.pdf

Usage

```
plot_snow(rivers_data)
```

Arguments

`rivers_data`	The dataset of rivers data

plot_stl *Plot component of STL for one station*

Description

This function plots the component of the STL for the four variables T, Q, TA and P for the water station passed in parameters and the associated meteoSwiss stations. This produces the plots shown in Figures S1 to S4 in supplementary.

Usage

```
plot_stl(station, output_type = "NONE")
```

Arguments

`station`	A SMET object containing the data for one station
`output_type`	Either `"NONE"` (default), `"PDF"` or `"PNG"`. `output_type = "NONE"` creates the plot in a normal plot window, `output_type = "PDF"` saves the plot as pdf under plots/General_situation.pdf, `output_type = "PNG"` saves the plot as png under plots/General_situation.png

`plot_thresholds` *Returns plots for the 15C and 25C thresholds analysis*

Description

This function return plots for the 15C and 25C thresholds analysis shown in Figures 13 and 14 and in Figure S38. PLots are saved under plots/25_degs.pdf and plots/15_degs.pdf

Usage

```
plot_thresholds(rivers_data, rivers_data_1h)
```

Arguments

`rivers_data`	The dataset of rivers data
`rivers_data_1h`	The dataset of rivers data at 1 hour resolution

`plot_time_series` *Plot time series and means*

Description

This function produces plots for time series and monthly and annual means for the given station and for the variables T, Q, TA, P. These plots are not used in the paper.

Usage

```
plot_time_series(station, output_type = "NONE")
```

Arguments

`station`	A SMET object containing the data for one station
`output_type`	Either "NONE" (default), "PDF" or "PNG". output_type = "NONE" creates the plot in a normal plot window (default), output_type = "PDF" saves the plot as pdf in plots/analysis/’station_name’/, output_type = "PNG" saves the plot as png in plots/analysis/’station_name’/,

plot_trends *Plot time series and trends*

Description

This function produces plots for time series and trends for the given station and for the variables T, Q, TA, P. These plots are not used in the paper.

Usage

```
plot_trends(station, output_type = "NONE")
```

Arguments

`station`	A SMET object containing the data for one station
`output_type`	Either "NONE" (default), "PDF" or "PNG". output_type = "NONE" creates the plot in a normal plot window (default), output_type = "PDF" saves the plot as pdf in plots/analysis/'station_name'/, output_type = "PNG" saves the plot as png in plots/analysis/'station_name'/,

plot_trends_robustness *Produce tend robustness plots for T, Q, TA and P.*

Description

This function produces the trend robustness plots by using the robust liner model function (rlm) from the MASS package (Figures S9 and S10 in supplementary) and by using trends when removing one year at the begining or end of the time series (Figures S11 and S12 in supplementary). Figures are saved in plots/trend_robustness.pdf

Usage

```
plot_trends_robustness(rivers_data)
```

Arguments

`rivers_data`	The dataset of rivers data

`plot_yearly_anomalies` *Plot yearyl seasonnal anomlaies*

Description

This function plots the yearly seasonnal anomalies for T, Q, TA and P shown in Figures 11 and Figures S28 to S30 in supplementary. The figure are saved under plots/summer_anomalies.pdf, plots/winter_anomalies.pdf, plots/spring_anomalies.pdf, and plots/fall_anomalies.pdf

Usage

```
plot_yearly_anomalies(rivers_data)
```

Arguments

`rivers_data`	The dataset of rivers data

`post_season_loess` *Do an additional Loess fitting on the STL analysis*

Description

Do an additional Loess fitting on the STL analysis seasonnal signal as suggested in R. B. Cleveland, W. S. Cleveland, J.E. McRae, and I. Terpenning (1990) STL: A Seasonal-Trend Decomposition Procedure Based on Loess. Journal of Official Statistics, 6, 3–73

Usage

```
post_season_loess(stl_input, s_win)
```

Arguments

`stl_input`	List of stl analysis output, keys being the used `s.window` values
`s_win`	Vector of `s.window` values for the stl analysis (see `link[stats]{stl}`)

Value

List of stl analysis output, keys being the used `s.window` values

print_all_trends *Print trends table in Latex format*

Description

Tihs function prints to the console the trends table shown in Tables A1 and A2 in appendix and in Tables S3 and S4 in supplementary. The table are printed in latex format

Usage

```
print_all_trends(rivers_data, meteo_data)
```

Arguments

`rivers_data`	The dataset of rivers data
`meteo_data`	The dataset of homegenous MeteoSwiss data

read_smet *Read SMET file, returns a SMET object (a list)*

Description

Read SMET file, returns a SMET object (a list). The SMET_OBJECT has the following structure:

`$header` The header information, based on smet header value

`$data` A `data.frame` containing read data, column names from SMET header, timestamp as PosixCT

Usage

```
read_smet(file_name)
```

Arguments

`file_name`	String containing the path th file to read

Value

A SMET_OBJECT containing the data

`remove_bissextile` *Remove 29th of february from a SMET_OBJECT*

Description

Remove 29th of february from a SMET_OBJECT

Usage

```
remove_bissextile(data)
```

Arguments

`data`	SMET_OBJECT

Value

SMET_OBJECT with 29th of February removed

`smooth_circular` *Compute a circular moving window average over all columns of a `data.frame` containing daily data over a year (365 data)*

Description

Compute a circular moving window average over all columns of a `data.frame` containing daily data over a year (365 data)

Usage

```
smooth_circular(year_daily_means, smooth_time)
```

Arguments

`year_daily_means`	A `data.frame` with numeric values containing daily data over a year (365 data)
`smooth_time`	The window to be used for the moving average

Value

The `data.frame` smoothed

smooth_daily_means_over_period

Function wrapper for smooth_circular

Description

Wrapper for smooth_circular to call it over all the periods defined in compute_daily_means_over_period

Usage

```
smooth_daily_means_over_period(daily_means, smooth_time)
```

Arguments

daily_means	A list of data.frame. Each data.frame is made of numeric values and contains daily data over a year (365 data)
smooth_time	The window to be used for the moving average

Value

A list where each key is one time period. List emntries are data.frame of daily mean data circularly smoothed over the period used as key. Periods have the format "AAAA-BBBB" where AAAA is the starting year and BBBB the ending year

swisswatertemp

swisswatertemp: A package to produce results presented in 'Stream temperature evolution in Switzerland over the last 50 years, Adrien Michel, Tristan Brauchli, Michael Lehning, Bettina Schaefli, and Hendrik Huwald, 2019'

Description

The swisswatertemp package is divided in two main parts: one is responsible to generate the dataset, and one to perform the analysis and produce plots.

Produce the data sets

The data set can be produced from raw data in a SMET fromat. Raw data are not provided here. The details about how to get the raw data and the scripts to transform them in the SMET format are given in the directory 1_Obtain_raw_data. Once the raw data are in the correct SMET format, the dataset can be generated by running Preprocessing.R in the 3_Produce_data directory. These steps are not mandatory, the datasets are indeed already available in 4_Run_analysis/data/rds_data. Metedata can be found in the excel table 3_Produce_data/data/discharge_gauging_station.xlsx

Description of data sets

Produced data sets have the general structure described below. Some data sets produced have only part of it. Structure of the data set:

```
["station name"]
|--header
  |--station_id = station number
  |--station_name = station name
  |--latitude: WG94 latitude
  |--longitude: WG94 longitude
  |--easting: CH1903 easting
  |--northing: CH1903 nothing
  |--altitude: altitude of the station
  |--operator: source of the data
  |--river: name of the river
  |--area: area of the catchment at the station
  |--mean_elevation: mean elevation of the catchment
  |--glacier_percent: percentage of the catchment glacier covered
  |--regime1: hydrological regime (classical)
  |--regime2: hydrological regime with regards to location
  |--regime3:Hydrological regime (following Aschwanden 1985, different from HADES 5.2)
  |--nodata: no data value used
  |--tz: timezone
  |--fields: variables in the [data] table
|--data: raw data
  |--timestamp: timestamp of the measurement as R date
  |--T: measured temperature (°C)
  |--Q: measured discharge (m3/s)
|--[monthly, yearly, DJF, MAM, JJA or SON]: data averaged over the given period
  |--[T Q]
    |--timestamp: timestamp in decimal year
    |--values: raw data averaged over the indicated period
    |--lm: output from linear model applied to trend + remainder
      |--["1999-2018" "1979-1998" "1979-2018" "1970-2018"] Periods over which
         trend is calculated, not necessarily all available
        |--timestamp: timestamp over the used period
        |--values: raw data over the given period
        |--trend: slope from linear model
        |--trend_std: std error of the trend value
        |--trend_p: p_value of the trend value
        |--intercept: intercept value from linear model
        |--intercept_std: std error of the intercept value
        |--intercept_p: p_value the of intercept value
        |--r_squared: r^2
        |--adj_r_squared: adjusted r^2
        |--printable:
          |--[trend, trend_std, trend_p, intercept, intercept_std, intercept_p,
        r_squared, adj_r_squared]: Same value as above but as string in "e" notation for display
|--hysteresis:
  |--[daily_mean or daily_mean_smoothed]: daily decadal mean, with or without
     smoothing (smoothed data is used in QT plots)
    |--["from_to" in years, e.g. "2009_1018"]
```

```
      |--T: temperature values(°C), 365 values
      |--Q: discharge values(m3/s), 365 values
|--meteo: attached meteo data
  |--[[station name]]
    |--header:
      |--station_id = station ID
      |--station_name = station name, same as ID
      |--latitude: WG94 latitude
      |--longitude: WG94 longitude
      |--easting: CH1903 easting
      |--northing: CH1903 nothing
      |--altitude: altitude of the station
      |--nodata: no data value used
      |--source: source of the meteodata
      |--tz: timezone
      |--fields: variables in the [data] table
    |--data: raw meteo data
      |--timestamp: timestamp of the measurement as R date
    |--[TA, P, TA_HOM, P_HOM, HS6, HS18, HSAUTO6, HSAUTO18]: available meteo variables
   |--[monthly, yearly, DJF, MAM, JJA or SON]: data averaged over the given period
      |--[TA,P]
        |--timestamp: timestamp in decimal year
        |--values: raw data averaged over the indicated period
        |--lm: output from linear model applied to trend + remainder
         |--["1999-2018" "1979-1998" "1979-2018" "1970-2018"] Periods over which
             trend is calculated, not necessarily all available
            |--timestamp: timestamps over the used period
            |--values: raw data over the given period
            |--trend: slope from linear model
            |--trend_std: std error of the trend value
            |--trend_p: p_value of the trend value
            |--intercept: intercept value from linear model
            |--intercept_std: std error of the intercept value
            |--intercept_p: p_value the of intercept value
            |--r_squared: r^2
            |--adj_r_squared: adjusted r^2
            |--printable:
           |--[trend, trend_std, trend_p, intercept, intercept_std, intercept_p,
           r_squared, adj_r_squared]: Same value as above but as string in "e" notation for display
|--STL
  |--[T or Q]
    |--timestamp: date, in decimal years
    |--seasonal: seasonal component from STL
    |--trend: trend from STL
    |--remainder: remainders from STL
    |--raw: raw data used for STL
    |--acf: acf analysis as R acf object
    |--pacf: pacf analysis as R pacf object
    |--lm: output from linear model applied to trend + remainder
      |--["1999-2018" "1979-1998" "1979-2018" "1970-2018"] Periods over which
         trend is calculated, not necessarily all available
        |--timestamp: timestamp over the used period
        |--values: raw data over the given period
        |--trend: slope from linear model
        |--trend_std: std error of the trend value
        |--trend_p: p_value of the trend value
        |--intercept: intercept value from linear model
        |--intercept_std: std error of the intercept value
        |--intercept_p: p_value the of intercept value
        |--r_squared: r^2
        |--adj_r_squared: adjusted r^2
        |--printable:
          |--[trend, trend_std, trend_p, intercept, intercept_std, intercept_p,
         r_squared, adj_r_squared]: Same value as above but as string in "e" notation for display
  |--meteo
    |--[station name]
      |--[TA or P]
        |--timestamp: date, in decimal years
        |--seasonal: seasonal component from STL
        |--trend: trend from STL
        |--remainder: remainders from STL
        |--raw: raw data used for STL
        |--acf: acf analysis as R acf object
        |--pacf: pacf analysis as R pacf object
      |--ccf: ccf analysis (between meteo and river data T-TA and Q-P) as R ccf object
        |--lm: output from linear model applied to trend + remainder
          |--["1999-2018" "1979-1998" "1979-2018" "1970-2018"] Periods over
             which trend is calculated, not necessarily all available
            |--timestamp: timestamp over the used period
            |--values: raw data over the given period
            |--trend: slope from linear model
            |--trend_std: std error of the trend value
            |--trend_p: p_value of the trend value
            |--intercept: intercept value from linear model
            |--intercept_std: std error of the intercept value
            |--intercept_p: p_value the of intercept value
            |--r_squared: r^2
            |--adj_r_squared: adjusted r^2
            |--printable:
              |--[trend, trend_std, trend_p, intercept, intercept_std,
                 intercept_p, r_squared, adj_r_squared]: Same value as above
                 but as string in "e" notation for display
```

Usage of the data sets

```
Entries can be accessed following the structure describes above
and with double brackets [["entry name here"]] (the name shoulb be
between quote marks), or with the "$" signe (in this case no quote
mark is needed except in the names contains special character).

If a variable containing the name of the entry to be accesses is
used, double brakets shoudl be used [[var]], note that $var will
not work (text after $ is taken as string, i.e. variable will not
be accessed).
```

```
Examples:
1) rivers_data[["Aare-Brienzwiler"]][["STL"]][["meteo"]][["GRH"]][["TA"]][["trend"]]
   or
   rivers_data$"Aare-Brienzwiler"$STL$meteo$GRH$TA$trend
   are quivalent and return the trend component of the meteo station GRH linked
   to the Aare-Brienzwiler water station.
2) Note that the function "names" is useful to retrieve the next entries at a
   given entry level. E.g. names(rivers_data$"Aare-Brienzwiler"$meteo) returns
   a list of the names of meteo station attached to the Aare-Brienzwiler river
   station.
3) for (river_station in names(rivers_data))
   will loop over all stations names which are stored in river_station. Data can
   be thus accessed through: rivers_data[[river_station]]$...
  For example rivers_data[[river_station]]$header$mean_elevation, if in the above
   loop, will return the mean elevation for each catchment.
```

trim *Trim spaces in string*

Description

Trim spaces in string

Usage

```
trim(x)
```

Arguments

x	A string

Value

Input string with leading or trailing spaces removed

Source

Function taken from https://stackoverflow.com/questions/2261079/how-to-trim-leading-and-trailing-wh

Index