Package 'mediterraneancalculations'

November 25, 2024

Type Package

Version 0.2-3

Date 2024-11-25

Title Mediterranean Calculations

Author Sergio M. Vicente-Serrano [aut], Fergus Reig-Gracia [aut, cre]

Maintainer Fergus Reig-Gracia <fergus@ipe.csic.es>

Depends R (>= 2.10)

Imports chron, sf, SpatialTools, lmom, SPEI, RobustLinearReg, hydroGOF, snowfall, Kendall, stats

Description .

License GPL (>= 3)

URL https://github.com/lcsc/mediterraneancalculations, https://lcsc.csic.es

LazyLoad no

Encoding UTF-8

Suggests MASS,
rmarkdown,
knitr,
testthat (>= 3.0.0)

NeedsCompilation no

RoxygenNote 7.3.2

VignetteBuilder knitr

Collate 'functions-mediterranean-calculations.r'
'mediterranean-calculations.r'
'mediterraneancalculations.R'
'mediterraneancalculationsNews.R'

R topics documented:

mediterraneancalculations-package

mediterraneancalculations: Mediterranean Calculations

Description

A suite of functions designed for quality control, reconstruction, and homogenization of monthly precipitation series, enabling the generation of series for specific periods and the analysis of precipitation trends.

Details

Function `main_mediterranean_calculations` are the workhorse of the mediterraneancalculations library. Other functions such as `calculate_statistics_data`, `alexanderson_homogenize` is an auxiliary low-level function and will not be used directly by the typical user.

Author(s)

Sergio M. Vicente-Serrano and Fergus Reig-Gracia

References

https://journals.ametsoc.org/view/journals/clim/32/22/jcli-d-19-0244.1.xml https://rmets.onlinelibrary.wiley.com/doi/10.1002/joc.2115

See Also

Useful links:

- https://github.com/lcsc/mediterraneancalculations
- https://lcsc.csic.es

alexanderson_homogenize

Alexanderson test for all available files (which have successfully passed the second fill)

Description

Alexanderson test for all available files (which have successfully passed the second fill)

Usage

```
alexanderson_homogenize(data, folder)
```

Arguments

`data`	data and coordinates
`folder`	directory to save the output files

Value

data and coordinates

`alexanderson_homogenize_data`

*Homogenization - Alexanderson test Reference series, compares and corrects For each dataset (1870, 1900...) The 5 most correlated series are chosen using the difference series A weighted average is made with the 5 series (correlation * data1 + ...) / sum(correlations), which will be the reference series Alexanderson will give a breakpoint and a ratio value to multiply the older part by... iterate while breakpoints are given Save statistics on inhomogeneities. Basically, the number of data points changed in each series and the time of the inhomogeneity. CSV with the number of data points changed and CSV with inhomogeneity point - all x 12 months*

Description

Homogenization - Alexanderson test Reference series, compares and corrects For each dataset (1870, 1900...) The 5 most correlated series are chosen using the difference series A weighted average is made with the 5 series (correlation * data1 + ...) / sum(correlations), which will be the reference series Alexanderson will give a breakpoint and a ratio value to multiply the older part by... iterate while breakpoints are given Save statistics on inhomogeneities. Basically, the number of data points changed in each series and the time of the inhomogeneity. CSV with the number of data points changed and CSV with inhomogeneity point - all x 12 months

Usage

```
alexanderson_homogenize_data(file_data, no_use_series = c())
```

Arguments

`file_data`	path to the data file
`no_use_series`	series that will not be homogenized

Value

None

apply_ecdf *Monthly data anomalies*

Description

Monthly data anomalies

Usage

```
apply_ecdf(data)
```

Arguments

data	monthly data

Value

data anomalies

apply_ecdf_month *Data anomalies*

Description

Data anomalies

Usage

```
apply_ecdf_month(data)
```

Arguments

data	monthly data

Value

data anomalies

calculate_reconstruction_statistics

Calculate reconstruction statistics - hydroGOF – statistic per station - D / MAE / PBIAS / RMSE - per station and month

Description

Calculate reconstruction statistics - hydroGOF – statistic per station - D / MAE / PBIAS / RMSE - per station and month

Usage

```
calculate_reconstruction_statistics(sim, obs)
```

Arguments

sim	filled data
obs	original data

Value

deleted data and input data per station

calculate_statistics

Final output with all statistics, regional average series, trends, SPI...

Description

Final output with all statistics, regional average series, trends, SPI...

Usage

```
calculate_statistics(data, data_ori)
```

Arguments

data	data and coordinates
data_ori	original data

Value

data and coordinates

`calculate_statistics_data` *Calculates statistics for the data Monthly, seasonal, and annual trends, Trend package, sens.slope function Add 1 to everything to avoid zeros Significance, modifiedmk package, bbsmk function National average series SPI at scales 3, 12, and 24 for each series, important to ensure operations cannot be reversed*

Description

Calculates statistics for the data Monthly, seasonal, and annual trends, Trend package, sens.slope function Add 1 to everything to avoid zeros Significance, modifiedmk package, bbsmk function National average series SPI at scales 3, 12, and 24 for each series, important to ensure operations cannot be reversed

Usage

```
calculate_statistics_data(file_data, data_ori)
```

Arguments

`file_data`	data and coordinates
`data_ori`	original data

Value

None

`calc_data_year` *Sums the data for each year to return a single annual value*

Description

Sums the data for each year to return a single annual value

Usage

```
calc_data_year(data)
```

Arguments

`data`	data matrix

Value

one value per year

calc_data_year_month_station *Returns the slope z for years and stations*

Description

Returns the slope z for years and stations

Usage

```
calc_data_year_month_station(data, calc_function)
```

Arguments

data	station data
calc_function	function to use

Value

list of results

calc_mkTrend_pval *Returns the p-value calculated by mkTrend or pval0 if pval was NA*

Description

Returns the p-value calculated by mkTrend or pval0 if pval was NA

Usage

```
calc_mkTrend_pval(data)
```

Arguments

data	data matrix

Value

pval

calc_mkTrend_slp *Linear regression of the data against years.*

Description

Linear regression of the data against years.

Usage

```
calc_mkTrend_slp(data)
```

Arguments

data	index

Value

lm

calc_percentage *Percentage difference*

Description

Percentage difference

Usage

```
calc_percentage(datos, years = NA)
```

Arguments

datos	data
years	years

Value

percentage

`ClimIndNews` *mediterraneancalculationsNews*

Description

Show the NEWS file of the **mediterraneancalculations** package.

Usage

```
ClimIndNews()
```

Details

(See description)

`coef_var` *Coefficients of variation, standard deviation https://fhernanb.github.io/Manual-de-R/varia.html*

Description

Coefficients of variation, standard deviation https://fhernanb.github.io/Manual-de-R/varia.html

Usage

```
coef_var(x, na.rm = FALSE)
```

Arguments

`x`	data
`na.rm`	Ignore NAs

Value

percentage

`delete_zero` *Removes data if there are 8 or more consecutive months of zeros, and if one of the involved months has less than 70% zeros, its data is removed. Also removes data if there are 5 or more consecutive months of zeros, and if all the involved months have less than 70% zeros, all are removed.*

Description

Removes data if there are 8 or more consecutive months of zeros, and if one of the involved months has less than 70% zeros, its data is removed. Also removes data if there are 5 or more consecutive months of zeros, and if all the involved months have less than 70% zeros, all are removed.

Usage

```
delete_zero(data)
```

Arguments

`data`	data

Value

data with zero groups removed

`delete_zones` *Below 28 degrees north, remove stations*

Description

Below 28 degrees north, remove stations

Usage

```
delete_zones(data)
```

Arguments

`data`	data and coordinates

Value

data and coordinates

`dry_spell_trend` *Calculates the trend. A 'years' object must be defined with the corresponding year for each case.*

Description

Calculates the trend. A 'years' object must be defined with the corresponding year for each case.

Usage

```
dry_spell_trend(index, threshold)
```

Arguments

`index`	index
`threshold`	threshold

Value

output

`fill_one_series` *Fills the received series using others in the order they appear in other_series*

Description

Fills the received series using others in the order they appear in other_series

Usage

```
fill_one_series(series, other_series)
```

Arguments

`series`	data series to complete
`other_series`	data series to use for completing in the order they must be used

Value

filled data series

`fill_series` *Monthly series filling We use stations less than 200km away with correlations above 0.7. For June, July, and August, we fill with the closest station. Use the method that correlates best with the original series.*

Description

Monthly series filling We use stations less than 200km away with correlations above 0.7. For June, July, and August, we fill with the closest station. Use the method that correlates best with the original series.

Usage

```
fill_series(control_data, min_correlation, max_dist)
```

Arguments

`control_data`	data from the stations and their coordinates
`min_correlation`	minimum correlation to use the data in filling
`max_dist`	maximum distance between series to use

Value

data and coordinates with data that did not pass the control removed

`fill_unfillable_station`

In countries where no series are found, we allow up to three years of data to be filled with the average. For example, if for a specific period 1900-2020 no series appear, but there are a maximum of three years of data (i.e., 36 months), we fill these data with the average of the 15 closest data points in time. This applies as long as these five years are not between 2015 and 2020 or within the first five years of the series, as this could affect trends. If the series are from 1981-2020, the same rule applies, but only two years of lost data are allowed.

Description

In countries where no series are found, we allow up to three years of data to be filled with the average. For example, if for a specific period 1900-2020 no series appear, but there are a maximum of three years of data (i.e., 36 months), we fill these data with the average of the 15 closest data points in time. This applies as long as these five years are not between 2015 and 2020 or within the first five years of the series, as this could affect trends. If the series are from 1981-2020, the same rule applies, but only two years of lost data are allowed.

Usage

```
fill_unfillable_station(data, fillable_years)
```

Arguments

`data`	station data to be filled
`fillable_years`	years that can be filled with the station's monthly average

Value

None

`main_mediterranean_calculations`

Reads precipitation files, calculates statistics, and saves the results The input files are 2 CSVs: one with coordinates in degrees (stations in rows and lat/lon in columns) and the other with monthly data (dates in rows and stations in columns)

Description

Reads precipitation files, calculates statistics, and saves the results The input files are 2 CSVs: one with coordinates in degrees (stations in rows and lat/lon in columns) and the other with monthly data (dates in rows and stations in columns)

Usage

```
main_mediterranean_calculations(file_data, file_coor)
```

Arguments

`file_data`	path to the data file
`file_coor`	path to the coordinates file

Value

None

`main_mediterranean_calculations_` *Calculates statistics for a country*

Description

Calculates statistics for a country

Usage

```
main_mediterranean_calculations_(read_all_data, folder, pb = NULL)
```

Arguments

`read_all_data`	input data
`folder`	directory where files are saved
`pb`	progress bar

Value

None

`mediterranean_calculations` *Performs quality control Quality control: Stations with less than 20 years of data are removed, and using the 10 most correlated stations within 200 km, data with a percentile difference greater than 0.6 are discarded.*

Description

Performs quality control Quality control: Stations with less than 20 years of data are removed, and using the 10 most correlated stations within 200 km, data with a percentile difference greater than 0.6 are discarded.

Usage

```
mediterranean_calculations(data, max_dist_eval)
```

Arguments

`data`	path to the data file
`max_dist_eval`	maximum distance between two stations to use one to evaluate or complete the other

Value

data and coordinates with data that do not pass the control removed

---

`mkTrend` *Calculates p-value (sometimes it doesn't result due to iteration issues, so take pval0).*

Description

Calculates p-value (sometimes it doesn't result due to iteration issues, so take pval0).

Usage

```
mkTrend(x, ci = 0.95)
```

Arguments

`x`	x
`ci`	ci

Value

list

`mobile_trends` *Calculates the moving trends of a series.*

Description

Calculates the moving trends of a series.

Usage

```
mobile_trends(datos)
```

Arguments

`datos`	datos

Value

list

near_correlations — *Returns the correlation between stations Ignoring those that are more than 200 km away (NA in those cases)*

Description

Returns the correlation between stations Ignoring those that are more than 200 km away (NA in those cases)

Usage

```
near_correlations(data, coor, max_dist)
```

Arguments

data	monthly data
coor	coordinates of the stations corresponding to the data
max_dist	maximum distance between the series to be used

Value

correlation between stations

near_estations — *Returns stations in order of proximity Ignoring those that are more than 200 km away (NA in those cases)*

Description

Returns stations in order of proximity Ignoring those that are more than 200 km away (NA in those cases)

Usage

```
near_estations(data, coor, max_dist)
```

Arguments

data	monthly data
coor	coordinates of the stations corresponding to the data
max_dist	maximum distance between the series to be used

Value

correlation between stations

`order_data` *Sorts the data and returns a list with the order*

Description

Sorts the data and returns a list with the order

Usage

```
order_data(data)
```

Arguments

`data`	data

Value

list

`overlap_station` *Calculates the overlap time between each pair of series*

Description

Calculates the overlap time between each pair of series

Usage

```
overlap_station(control_data)
```

Arguments

`control_data`	data from the stations and their coordinates

Value

matrix with the months where the stations overlap with each other

`overlap_station_no_0` *Calculates the overlap time between each pair of series without counting zeros*

Description

Calculates the overlap time between each pair of series without counting zeros

Usage

```
overlap_station_no_0(control_data)
```

Arguments

`control_data`	data from the stations and their coordinates

Value

matrix with the months where the stations overlap with each other

`percentage_of_zeros` *Returns the percentage of valid data that are zeros*

Description

Returns the percentage of valid data that are zeros

Usage

```
percentage_of_zeros(data)
```

Arguments

`data`	data

Value

percentage

`quality_control` *Quality control Stations with less than 20 years of data are removed Using the 10 closest within 200 km, discard if the average percentile differs by more than 0.6 or more than 0.5 for data points with a value of 0*

Description

Quality control Stations with less than 20 years of data are removed Using the 10 closest within 200 km, discard if the average percentile differs by more than 0.6 or more than 0.5 for data points with a value of 0

Usage

```
quality_control(data, coor, max_dist, max_diff_anomaly, max_diff_anomaly_0)
```

Arguments

`data`	data
`coor`	coordinates
`max_dist`	maximum distance between two stations to use one to evaluate or complete the other
`max_diff_anomaly`	maximum anomaly difference to keep data in quality control
`max_diff_anomaly_0`	maximum anomaly difference to keep data in quality control, if the data point is 0

Value

data and coor with data points that do not pass the control removed

`read_data` *Reads data from CSVs in the agreed format The input files are 2 CSVs: one with coordinates in degrees (stations in rows and columns lat and lon) and another with monthly data (dates in rows and stations in columns)*

Description

Reads data from CSVs in the agreed format The input files are 2 CSVs: one with coordinates in degrees (stations in rows and columns lat and lon) and another with monthly data (dates in rows and stations in columns)

Usage

```
read_data(file_data, file_coor)
```

Arguments

file_data	path to the data file
file_coor	path to the coordinates file

Value

original data, data of interest, and coordinates of the stations read

---

read_years *Reads the years from text strings that end with years*

Description

Reads the years from text strings that end with years

Usage

```
read_years(txt)
```

Arguments

txt	text or vector of texts

Value

list

---

save_csvs *Saves the data into CSVs*

Description

Saves the data into CSVs

Usage

```
save_csvs(i_ini, folder_name, data_save, coor_save)
```

Arguments

i_ini	identifier for the files
folder_name	folder where the file will be saved
data_save	data from the stations to save
coor_save	coordinates data to save

Value

None

---

save_data *Saves the output in 5 files with the data 5 files indicating whether each data point is original or filled (1 for unaltered data, 0 for altered data) and 5 coordinate files for the stations in each data file, which are: - 1870 to 2020 with more than 0.75 of years of original data - 1900 to 2020 with more than 0.75 of years of original data - 1930 to 2020 with more than 0.75 of years of original data - 1950 to 2020 with more than 0.75 of years of original data - 1990 to 2020 with more than 0.75 of years of original data*

Description

Saves the output in 5 files with the data 5 files indicating whether each data point is original or filled (1 for unaltered data, 0 for altered data) and 5 coordinate files for the stations in each data file, which are: - 1870 to 2020 with more than 0.75 of years of original data - 1900 to 2020 with more than 0.75 of years of original data - 1930 to 2020 with more than 0.75 of years of original data - 1950 to 2020 with more than 0.75 of years of original data - 1990 to 2020 with more than 0.75 of years of original data

Usage

```
save_data(data_ori, control_data)
```

Arguments

data_ori	original data read from CSV files
control_data	data from the stations and their coordinates

Value

data and coordinates with data that did not pass the control removed

save_delete_data *For each station, save the number of input data and deleted data*

Description

For each station, save the number of input data and deleted data

Usage

```
save_delete_data(ori_data, process_data, folder)
```

Arguments

ori_data	original data
process_data	processed data
folder	folder where the resulting file is saved

Value

deleted data and input data per station

second_data_fill *Performs a second fill For each dataset (1870, 1900...) Stations with more than 90 or 95 Stations are sorted by correlation (minimum 0.5), and filling is done using the 10 methods... Stations without total fill are discarded*

Description

Performs a second fill For each dataset (1870, 1900...) Stations with more than 90 or 95 Stations are sorted by correlation (minimum 0.5), and filling is done using the 10 methods... Stations without total fill are discarded

Usage

```
second_data_fill(data, max_dist_eval = NA)
```

Arguments

data	data and coordinates
max_dist_eval	maximum distance for filling

Value

data and coordinates

second_data_fill_data *Performs a second fill For each dataset (1870, 1900...) Stations with more than 90 or 95 Sort the stations by correlation (minimum 0.5) and fill using the 10 methods... Stations without total fill are discarded*

Description

Performs a second fill For each dataset (1870, 1900...) Stations with more than 90 or 95 Sort the stations by correlation (minimum 0.5) and fill using the 10 methods... Stations without total fill are discarded

Usage

```
second_data_fill_data(file_data, fillable_years = 36, max_dist = NA)
```

Arguments

file_data	path to the data file
fillable_years	years that can be filled with the station's monthly average
max_dist	maximum allowed distance for filling

Value

None

select_data *First valid data points (non-NAs)*

Description

First valid data points (non-NAs)

Usage

```
select_data(data, n_reference_stations)
```

Arguments

data	data
n_reference_stations	number of data points to return

Value

first non-NA data points

sum_no_nas *Number of non-NA data points*

Description

Number of non-NA data points

Usage

```
sum_no_nas(data)
```

Arguments

data	data

Value

number of non-NA data points

Index