Package 'synoptReg'

April 22, 2021

Type Package

Title Synoptic Climate Classification and Spatial Regionalization of Environmental Data

Version 1.2.1

Depends R (>= 3.5)

Description Set of functions to compute different types of synoptic classification methods and for analysing their effect on environmental variables. More information about the methods used in Lemus-Canovas et al. 2019 <DOI:10.1016/j.atmosres.2019.01.018>, Martin-Vide et al. 2008 <DOI:10.5194/asr-2-99-2008>, Jenkinson and Collison 1977.

License GPL (>= 3)

Maintainer Marc Lemus-Canovas <mlemus@ub.edu>

URL <https://lemuscanovas.github.io/synoptreg/>

BugReports https://github.com/lemuscanovas/synoptReg/issues

Encoding UTF-8

LazyData true

Imports dplyr, ggplot2, lubridate, magrittr, sf, rnaturalearth, rnaturalearthdata, metR, raster, RNCEP, stringr, tidyr, tibble, kohonen

Suggests maptools, ncdf4, rgeos, udunits2, gridExtra

NeedsCompilation no

RoxygenNote 7.1.1

Author Marc Lemus-Canovas [aut, cre] (<https://orcid.org/0000-0002-0925-3827>), Dominic Roye [ctb] (<https://orcid.org/0000-0002-5516-6396>)

Repository CRAN

Date/Publication 2021-04-21 23:20:02 UTC

R topics documented:

`ct2env` *Establishing the relationship between CT and a environmental variable*

Description

This function applies the approach: "circulation types to environment".

Usage

```
ct2env(x, clas, fun = mean, out = "data.frame")
```

Arguments

`x`	data.frame. A data.frame containing the environmental data (i.e. precipitation, temperature, PM10, etc.) with the following variables: `lon`,`lat`,`time`,`value`,`anom_value`. See `tidy_nc`.
`clas`	data.frame. A data.frame of the synoptic classification (time and WT) obtained from the `synoptclas` function.
`fun`	function. A function to be applied to the environmental variable for each WT.
`out`	character. Choose between `"data.frame"` (default) or `"raster"` A function to be applied to the environmental variable for each WT.

Value

a data.frame or a Raster Stack containing the environmental grids based on the weather types.

Examples

```
# Load data (mslp or precp_grid)
data(mslp)
data(z500)
# Tidying our atmospheric variables (500 hPa geopotential height
# and mean sea level pressure) together.

# Time subset between two dates
atm_data1 <- tidy_nc(x = list(mslp,z500),
             name_vars = c("mslp","z500"))

# S-mode classification
smode_clas <- synoptclas(atm_data1, ncomp = 6)

# ct2env (precipitation example)
ct2env(x = pcp, clas = smode_clas$clas, fun = mean, out = "data.frame")
```

`download_ncep` *Download NCEP/NCAR data*

Description

Weather Data from NCEP/NCAR Reanalysis via RNCEP package

Usage

```
download_ncep(
  var = "slp",
  level = "surface",
  month_range = c(1, 12),
  year_range = c(2010, 2017),
  lat_range = c(30, 60),
  lon_range = c(-30, 10),
  dailymean = TRUE,
  hour = NULL,
  reanalysis2 = TRUE,
  save_download = TRUE,
  file_name = NULL
)
```

Arguments

`var`	slp 'sea level pressure' (default) for more variables see help of ?NCEP.gather
`level`	surface (default)
`month_range`	min,max month c(1,12) (default)

year_range	min,max year c(2010,2017) (default)
lat_range	min,max latitude c(30, 60) (default)
lon_range	min,max longitud c(-30, 10) (default)
dailymean	daily avarage of the variable retrived. Default TRUE.
hour	One hour of the following: 0,6,12 or 18.
reanalysis2	Logical. Default TRUE. variables are downloaded from the NCEP-DOE Reanalysis 2. If FALSE, data downloaded from NCEP/NCAR Reanalysis 1
save_download	Logical. Default TRUE. Do yoy want to save the downloaded data into an RDS file?
file_name	character. Provide a name for the file downloaded.

Value

a data.frame with the following variables: lon,lat,time,value

Examples

```
## Not run:
#Daily mean air temperature 2m for 2017
#ta_data <- download_ncep(year_range=2017)

#Air temperature 2m at 06:00 for 2017
#ta_data_h6 <- download_ncep(year_range=2017,dailymean = FALSE,hour=6)

## End(Not run)
```

---

get_lamb_points *Determine the 16 grid points for the Lamb classification*

Description

Compute the 16 pair of coordinates necessary for using the objective version of the Lamb method

Usage

```
get_lamb_points(x, y)
```

Arguments

x	longitude coordinate of the central point of the scheme.
y	latitude coordinate of the central point of the scheme.

Value

a data.frame with the 16 points of coordinates.

Examples

```
points <- get_lamb_points(x = -5, y = 40)
points
```

`lamb_clas` *Objective Lamb Weather Type Classification*

Description

Calculates the classification of the main weather types for the 16-points defined in `get_lamb_points`. Wind-flow characteristics are computed for the daily pressure field according to the rules proposed by the original Jenkinson and Collison classification (see Jenkinson and Collison, 1977; Jones et al., 2013) (1), and to the rules proposed by Trigo and DaCamara, 2000 (2).

Usage

```
lamb_clas(points, mslp, U = FALSE, thr = c(6, 6))
```

Arguments

`points`	16 point pair of coordinates obtained from `get_lamb_points`.
`mslp`	Mean Sea Level pressure gridded data.
`U`	Logical. If T, Jones et al. 2013 approach is applied, maintaining the U-type in the classification. If F, U is removed as detailed in Trigo and DaCamara, 2000.
`thr`	threshold used for Unclassified days (total shear vorticity and total flow, respectively). Default c(6,6).

Value

A list with:

- A data.frame containing the dates and the weather types.
- A data frame containing the gridded data grouped by circulation types.

References

Jenkinson, A.F., Collison F.P (1977) *An initial climatology of gales over the North Sea* Synoptic Climatology Branch Memorandum, No. 62.Meteorological Office: Bracknell, England.

Jones, P. D., Hulme M., Briffa K. R. (1993) *A comparison of Lamb circulation types with an objective classification scheme* Int. J. Climatol. 13: 655–663.

Jones, P. D., Harpham C, Briffa K. R. (2013) *Lamb weather types derived from Reanalysis products* Int. J. Climatol. 33: 1129–1139.

Trigo, R., DaCamara C. (2000) *Circulation weather types and their impact on the precipitation regime in Portugal* Int. J. Climatol. 20: 1559-1581.

See Also

get_lamb_points

Examples

```
data(mslp)

points <- get_lamb_points(x = 5,y = 40)
lamb_clas(points = points, mslp = mslp)
```

---

mslp *Mean Sea Level pressure data*

Description

Data from the NCEP/NCAR Reanalysis 1 (https://psl.noaa.gov/data/gridded/data.ncep.reanalysis.html). This data corresponds to daily values of mean sea level pressure with 2.5 x 2.5º of spatial resolution from January 2000 to december 2002.

Usage

```
data(mslp)
```

Format

A data.frame with the following variables: `lon`,`lat`,`time`,`value`.

geographical area: -10,30,30,60

time period: 2000-01-01 to 2002-12-31

units: Pascals

References

Kalnay et al. (1996) *The NCEP/NCAR 40-year reanalysis project, Bull. Amer. Meteor. Soc., 77, 437-470, 1996*

Examples

```
data(mslp)
```

pca_decision *PCA decision*

Description

pca_decision plots the explained variances against the number of the principal component. In addition, it returns all the information about the PCA performance.

Usage

```
pca_decision(x, ncomp = 30, norm = T, matrix_mode = "S-mode")
```

Arguments

x	data.frame. A data.frame with the following variables: lon,lat,time,value,anom_value. See tidy_nc.
ncomp	integer. Number of principal components to show/retain
norm	logical. Default TRUE. norm = TRUE is recommended for classify two ore more variables.
matrix_mode	character. The mode of matrix to use. Choose between S-mode and T-mode

Value

a list with:

- A list with class princomp containing all the results of the PCA
- A data frame containing the main results of the ncomp selected (standard deviation, proportion of variance and cumulative variance).
- A ggplot2 object to visualize the scree test

Note

To perform the PCA the x must contain more rows than columns. In addition, x cannot contain NA values.

See Also

tidy_nc

Examples

```
# Load data (mslp or precp_grid)
data(mslp)
data(z500)
# Tidying our atmospheric variables (500 hPa geopotential height
# and mean sea level pressure) together.
```

```
# Time subset between two dates
atm_data1 <- tidy_nc(x = list(mslp,z500))

# Deciding on the number of PC to retain
info <- pca_decision(atm_data1)
```

---

pcp *Daily precipitation grid of Balearic Islands (Spain)*

Description

Data from the SPREAD data set downloaded from the Spanish National Research Council (CSIC). (http://spread.csic.es/info.html). This data corresponds to daily values of precipitation with a spatial resolution of 5 x 5 km from January 2000 to december 2010

Usage

```
data(pcp)
```

Format

A data.frame with the following variables: `lon`,`lat`,`time`,`value`.

geographical area: Balearic Islands

time period: 2000-01-01 to 2010-12-31

units: mm*10

coordinates reference system: +proj=utm +zone=30 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs

References

Serrano-Notivoli et al. (2017) *SPREAD: a high-resolution daily gridded precipitation dataset for Spain, an extreme events frequency and intensity overview. Earth Syst. Sci. Data, 9, 721-738, 2017, https://doi.org/10.5194/essd-9-721-2017*

Examples

```
data(pcp)
```

plot_lamb_scheme *Plot Lamb Scheme*

Description

Visualize the Lamb Scheme

Usage

```
plot_lamb_scheme(points)
```

Arguments

points	points obtained from the get_lamb_points function.

Value

a ggplot map.

Examples

```
points <- get_lamb_points(x = -5, y = 40)

plot_lamb_scheme(points)
```

raster_pca *Raster PCA*

Description

Perform a Principal Component Analysis on a RasterStack

Usage

```
raster_pca(raststack, aggregate = 0, focal = 0)
```

Arguments

raststack	Raster Stack.
aggregate	Integer. Aggregation factor based on function aggregate of **raster** package.
focal	Integer. smooth filter based on function focal of **raster** package.

Value

a list with:

- A raster stack containing the results of the PCA
- A data frame containing the main results of the PCA (standard deviation, proportion of variance and cumulative variance

---

`regionalization` *Environmental regionalization*

Description

Perform an unspervised clustering of the Raster Stack

Usage

```
regionalization(raststack, centers, iter.max = 100, nstart = 100)
```

Arguments

`raststack`	Raster Stack.
`centers`	Integer. Number of clusters.
`iter.max`	Integer. The maximum number of iterations allowed. Default 100.
`nstart`	Integer. How many random sets should be chosen? Default 100.

Value

a list with:

- A raster with the final regionalization
- A list with the results of the K-means performance
- A raster displaying a pseudo-MAE error based on the difference between each pixel value and its respective centroide
- A numeric pseudo-MAE mean value for the entire map

som_clas *Self-Organizing Maps classification*

Description

som_clas allows to perform a SOM synoptic classification

Usage

```
som_clas(
  x,
  xdim,
  ydim,
  iter = 2000,
  alpha = c(0.05, 0.01),
  dist.fcts = "euclidean",
  mode = "online",
  cores = 1,
  norm = T
)
```

Arguments

x	data.frame. A data.frame with the following variables: lon,lat,time,value,anom_value. See tidy_nc.
xdim	Integer. X dimension of the grid. See somgrid from kohonen package.
ydim	Integer. Y dimension of the grid. See somgrid from kohonen package.
iter	integer. Number of iterations.
alpha	vector. learning rate. See som from kohonen package for details.
dist.fcts	character. vector of distance functions to be used for the individual data layers. See som from kohonen package for details.
mode	carachter. type of learning algorithm. Default "on-line". See kohonen package for details.
cores	Integer. Parallel processing only available for "pbatch" algorithm.
norm	logical. Default TRUE. norm = TRUE is recommended for classifying two ore more variables.

Value

A list with:

- A data.frame containing the dates and the weather types.
- A data frame containing the gridded data grouped by circulation types.
- An object of class kohonen with all the components returned by the function som

References

Wehrens, R. and BuydenL. (2007) *Self- and Super-organizing Maps in R: The kohonen Package* Journal of Statistical Software, 21(5), 1 - 19.

See Also

tidy_nc

Examples

```
# Load data
data(z500)
# Tidying our atmospheric variables (500 hPa geopotential height).
z500_tidy <- tidy_nc(x = list(z500),
             name_vars = c("z500"))

# SOM classification
som_cl <- som_clas(z500_tidy, xdim = 4, ydim = 4, iter = 200)
```

---

synoptclas *PCA Synoptic classification*

Description

synoptclas allows to perform several types of synoptic classification approaches based on one or several atmospheric variables (i.e. mean sea level pressure, geoptential height at 500 hPa, etc.)

Usage

```
synoptclas(x, ncomp, norm = T, matrix_mode = "S-mode", extreme_scores = 2)
```

Arguments

x	data.frame. A data.frame with the following variables: lon,lat,time,value,anom_value. See tidy_nc.
ncomp	Integer. Number of components to be retained.
norm	logical. Default TRUE. norm = TRUE is recommended for classifying two ore more variables.
matrix_mode	character. The mode of matrix to use. Choose between S-mode and T-mode
extreme_scores	Integer. Definition of extreme score threshold (Esteban et al., 2005). Default is 2. Only applicable for a matrix_mode = "S-mode"

Details

The `matrix_mode` argument allows to conduct different types of synoptic classifications depending on the user's objective. If the user wants to perform a synoptic classification of a long and continuous series, he must set the `matrix_mode = "S-mode"`. When we apply the PCA to a matrix in S-mode, the variables are the grid points (lon,lat) and the observations are the days (time series), so the linear relationships that the PCA establishes are between the time series of the grid points. One of the results we obtain from the PCA are the "scores", which indicate the degree of representativeness of each day for each of the principal components. However, the scores do not allow us to directly obtain the weather types (WT) classification, since one day can be represented by several principal components. For this reason, a clustering method is required to group each day to an specific WT based on the multivariate coordinates provided by the "scores". Before using a clustering method, a VARIMAX rotation is performed on the principal Components retained, with the aim of redistributing the variance of such components. With the rotated components, the scores are used to apply the extreme scores method (Esteban et al., 2005). The scores show the degree of representativeness associated with the variation modes of each principal component, i.e., the classification of each day to its more representative centroid. Thus, the extreme scores method uses the scores > 2 and < -2, establishing a positive and negative phase for each principal component. The extreme scores procedure establishes the number of groups and their centroids in order to apply the K-means method without iterations. Conversely, if the user wants to perform a synoptic classification of specific events (i.e. flood events, extreme temperatures events,etc.), he must set the `matrix_mode = "T-mode"`. In this case, the variables are the days (time series) and the observations are the grid points. The relationships established in this case are between each daily gridded map. For this reason, the eigenvalues (correlations) allow to allow us to associate each day to a WT without using a clustering method as in the case of the S-mode matrix.

Value

A list with:

- A data.frame containing the dates and the weather types. If "T-mode" is selected, two classifications are returned (absolute and positive/negative classification).
- A data frame containing the gridded data grouped by circulation types.If "T-mode" is selected, 3 classifications are returned (absolute correlation,maximum positive correlation, and positive/negative classification). In addition, p-values of a t-test computed to the anomalies, comparing them to 0 with a conf.level = 0.95, are returned

References

Esteban, P. , Jones, P. D., Martin.Vide, J. *Atmospheric circulation patterns related to heavy snowfall days in Andorra, Pyrenees* Int. J. Climatol. 25: 319-329. doi:10.1002/joc.1103

See Also

`pca_decision`

Examples

```
# Load data (mslp or precp_grid)
data(mslp)
```

```
data(z500)
# Tidying our atmospheric variables (500 hPa geopotential height
# and mean sea level pressure) together.
atm_data1 <- tidy_nc(x = list(mslp,z500),
             name_vars = c("mslp","z500"))

# S-mode classification
smode_cl <- synoptclas(atm_data1, ncomp = 6)

# Time subset using a vector of dates of interest
dates_int <- c("2000-01-25","2000-04-01","2000-07-14","2001-05-08","2002-12-20")
atm_data2 <- tidy_nc(x = list(mslp,z500),
                     time_subset = dates_int,
                     name_vars = c("mslp","z500"))

# T-mode classification
tmode_cl <- synoptclas(atm_data2, ncomp = 2, matrix_mode = "T-mode")
```

`tidy_nc` *Set the time period and the geogprahical extension, as well as computes the anomaly of the atmospheric variable/s*

Description

This function allows to subset the time series and geogprahical area of your atmospheric variable. In addition, even if no argument is given, the anomaly of the atmospheric variable/s will be computed. The anomaly value is provided in order to facilitate the visualization of the results after use the `synoptclas` function. It is mandatory to pass the `tidy_nc` even if you do not want to change the time period or the geographical extension.

Usage

```
tidy_nc(
  x,
  time_subset = NULL,
  geo_subset = NULL,
  monthly_subset = NULL,
  name_vars = NULL
)
```

Arguments

`x`	data.frame. A data.frame with the following variables: `lon,lat,time,value`. The same structure returned when using `download_ncep`.
`time_subset`	vector. Starting and ending date, or a vector of dates of interest.
`geo_subset`	vector. A vector providing the `xmin,xmax,ymin,ymax`.

monthly_subset	an integer or a vector of integers. Number of the month/s desired.
name_vars	character or a vector of characters. Name of the atmospheric variable/s. If name is not specified, then will be coded as integers.

Value

A data.frame with the following variables: lon,lat,time,value,anom_value

See Also

download_ncep

Examples

```
# Load data (mslp or precp_grid)
data(mslp)
data(z500)
# Tidying our atmospheric variables (500 hPa geopotential height
# and mean sea level pressure) together.

# Time subset between two dates
atm_data1 <- tidy_nc(x = list(mslp,z500), time_subset = c("2000-05-01","2001-04-30"))

# Time subset using a vector of dates of interest. Including a geographical crop
dates_int <- c("2000-01-25","2000-04-01","2000-07-14","2001-05-08","2002-12-20")
atm_data1 <- tidy_nc(x = list(mslp,z500),
                     time_subset = dates_int,
                     geo_subset = c(-20,10,30,50),
                     name_vars = c("mslp","z500")) # following the list sequence
```

---

z500 *500 hPa Geopotential Height*

Description

Data from the NCEP/NCAR Reanalysis 1 (https://psl.noaa.gov/data/gridded/data.ncep.reanalysis.html). This data corresponds to global daily values of 500 hPa geopotential height with 2.5 x 2.5?? of spatial resolution from January 2000 to december 2002.

Usage

```
data(z500)
```

Format

A data.frame with the following variables: lon,lat,time,value.

geographical area: -10,30,30,60

time period: 2000-01-01 to 2002-12-31

units: meters

References

Poli et al. (2016) *Kalnay et al., The NCEP/NCAR 40-year reanalysis project, Bull. Amer. Meteor. Soc., 77, 437-470, 1996*

Examples

```
data(z500)
```

Index