Package 'esviz'

September 4, 2026

Title Plotting Functions for Climate Science and Services

Version 0.0.4

Description A plotting package for climate science and services. Provides a set of functions for visualizing climate data, including maps, time series, scorecards and other diagnostics. Some functions are adapted and extended from the 's2dv' and 'CSTools' packages (Manubens et al. (2018) <doi:10.1016/j.envsoft.2018.01.018>; Pérez-Zanón et al. (2022) <doi:10.5194/gmd-15-6115-2022>), with more consistent and integrated functionalities.

Depends R (>= 3.6.0)

Imports graphics, grDevices, maps, ClimProjDiags, sf, ggplot2, rnaturalearth, cowplot, s2dv, RColorBrewer, CSTools, easyNCDF (>= 0.1.4), dplyr, plyr, data.table, reshape2, scales, stats, utils, foreach, rlang, kableExtra, gridExtra, gtable, doParallel, mapproj, webshot2, jsonlite, sp, methods, magrittr

Suggests testthat, rnaturalearthdata

Maintainer Ariadna Batalla <ariadna.batalla@bsc.es>

License GPL-3

URL https://gitlab.earth.bsc.es/es/esviz/

BugReports https://gitlab.earth.bsc.es/es/esviz/-/work_items

SystemRequirements GDAL (>= 2.0.1), GEOS (>= 3.4.0), PROJ (>= 4.8.0)

Encoding UTF-8

RoxygenNote 7.3.1

Config/testthat/edition 3

LazyData true

NeedsCompilation no

Author BSC-CNS [aut, cph], An-Chi Ho [aut], Eva Rifà [ctb], Victòria Agudetse [ctb],

Ariadna Batalla [ctb, cre],
Núria Pérez-Zanón [ctb],
Nadia Milders [ctb],
Verónica Torralba [ctb],
Raúl Capellán [ctb],
Anindita Patra [ctb]

Repository CRAN

Date/Publication 2026-09-04 13:20:02 UTC

Contents

BarPlotCat . . . 3
ClimPalette . . . 6
ColorBarContinuous . . . 7
ColorBarDiscrete . . . 10
Hatching . . . 13
map_temp . . . 15
ShapeToMask . . . 17
ts_temp . . . 20
Viz2VarsVsLTime . . . 20
VizACC . . . 23
VizAnimateMap . . . 25
VizAno . . . 28
VizBoxWhisker . . . 30
VizClim . . . 32
VizCombinedMap . . . 34
VizEquiMap . . . 37
VizForecastPDF . . . 45
VizLayout . . . 48
VizMatrix . . . 53
VizMostLikelyQuantileMap . . . 55
VizPDFsOLE . . . 58
VizPolygonsMap . . . 59
VizRobinson . . . 64
VizScorecard . . . 67
VizSection . . . 70
VizStereoMap . . . 72
VizTriangles4Categories . . . 77
VizVsLTime . . . 80
VizWeeklyClim . . . 82

Index **85**

BarPlotCat *Plot Bar Charts of Forecast Category Probabilities*

Description

This function creates bar plots for probabilistic forecasts split into categories (e.g., below-normal, normal, above-normal), optionally including extreme categories (e.g., below P10, above P90). Probabilities are displayed on the y-axis. The function supports multi-panel plotting for different time steps, the addition of a shared legend and titles, subtitles, bar transparency, a text label and output to file.

Usage

```
BarPlotCat(
  probs,
  lims,
  extreme_probs = NULL,
  extreme_lims = NULL,
  toptitle = "",
  legend_title = "",
  cat_dim = "cat",
  panel_dim = "ftime",
  color.set = "s2s4e",
  transparency = NULL,
  category_names = c("bn", "norn", "an"),
  panel_title = "",
  panel_subtitle = "",
  panel_bottom_name = "",
  panel_label = "",
  lims_pos = -4,
  legend_width = 3.5,
  extreme_bars_width = 0.4,
  xaxis_title = "",
  extreme_cat_names = c("p10", "p90"),
  toptitle_size = 16,
  label_size = 12,
  axis_title_size = 14,
  axis_text_size = 12,
  legend_title_size = 14,
  legend_text_size = 12,
  bar_text_size = 12,
  toptitle_pos = "center",
  label_height = 15,
  fileout = NULL,
  width = 8,
  height = 6,
  size_units = NULL,
```

```
  res = 100
)
```

Arguments

probs	A named 2D array with dimensions 'cat_dim' × 'panel_dim', containing probabilities for each category (in relative units, summing to 1). Can also be a named vector (recycled to 2D internally).
lims	A named 2D array with category threshold values (e.g., tercile cutoffs), or a vector with length one less than the category dimension of 'probs'. Should match the shape and names of 'probs'.
extreme_probs	(optional) A 2D array (or vector) with probabilities for extreme categories (e.g., below P10, above P90), with dimensions matching 'probs'.
extreme_lims	(optional) A 2D array (or vector) with limits corresponding to the extreme categories.
toptitle	A character string with the main plot title.
legend_title	A string with the title of the legend that shows the categories.
cat_dim	A string indicating the name of the category dimension. Default is "cat".
panel_dim	A string indicating the name of the panel dimension (e.g., time). Default is "ftime".
color.set	A string selecting the color palette to use. One of `"s2s4e"`, `"ggplot"`, `"hydro"`, `"vitigeoss"`. Default is "s2s4e".
transparency	(optional) A numeric value or vector with values between 0 and 1 controlling the transparency applied to the bars. If a single value is provided, it is applied to all panels. If a vector is provided, its length must match the number of panels. A value of 0 means fully opaque bars, while 1 makes them fully transparent. This feature can be used, for example, to indicate negative skill.
category_names	A character vector with names for each forecast category. Default is `c("bn", "norn", "an")`.
panel_title	A string or vector of strings with titles for each panel. If a single value is provided, it is repeated across panels. Use an empty string to omit the title in a given panel.
panel_subtitle	A string or vector with subtitles for each panel. If a single value is provided, it is repeated across panels. Use an empty string to omit the subtitle in a given panel.
panel_bottom_name	A string or vector for the bottom axis label of each panel.
panel_label	(optional) A character string or vector of strings to annotate on top of each panel. If a single value is provided, it is repeated across panels. Use an empty string to omit the label in a given panel.
lims_pos	A numeric value indicating the vertical position of threshold annotations. Default is -1.5.
legend_width	A numeric value (in cm) for the width of the legend area. Default is 3.
extreme_bars_width	A numeric value between 0 and 1 defining the width of the bars for extreme categories. Default is 0.4.

`xaxis_title` A string for the shared x-axis title.

`extreme_cat_names`
A character vector with names for the extreme categories (left and right). Default is `c("p10", "p90")`.

`toptitle_size` A numeric value for the font size of the top title. Default is 16.

`label_size` A numeric value indicating the font size of the panel label added with 'panel_label'. Default is 12.

`axis_title_size`
A numeric value for the font size of the x- and y-axis titles. Default is 14.

`axis_text_size` A numeric value for the font size of the x- and y-axis tick labels and threshold annotations. Default is 12.

`legend_title_size`
A numeric value for the font size of the legend title. Default is 14.

`legend_text_size`
A numeric value for the font size of the legend labels. Default is 12.

`bar_text_size` A numeric value for the font size of the percentage labels displayed above the bars. Default is 12.

`toptitle_pos` A string for justification of the title. One of `"left"`, `"center"`, `"right"`. Default is "center".

`label_height` A numeric value indicating the vertical position of the panel label added with 'panel_label'. Default is 15.

`fileout` (optional) Path to save the resulting plot (e.g., `"plot.png"`). If NULL, the plot is returned.

`width` (optional) Width of the output figure, passed to `ggsave()`. Units are defined by 'size_units'.

`height` (optional) Height of the output figure, passed to `ggsave()`. Units are defined by 'size_units'.

`size_units` Units for `width` and `height`. Default is inches. See `ggsave()`.

`res` Plot resolution in dpi when saving to file. Default is NULL.

Value

A grob object representing the composed bar plot, invisibly if written to file.

Examples

```
# Basic example
probs <- array(rep(c(0.3, 0.4, 0.3), 4), c(cat = 3, ftime = 4))
lims <- array(rep(c(2, 4), 4), c(cat = 2, ftime = 4))
BarPlotCat(probs, lims, toptitle = "Example Forecast")

# Example with transparency and panel labels
probs <- array(c(0.2, 0.5, 0.3, 0.6, 0.2, 0.2), c(cat = 3, ftime = 2))
lims <- array(c(2, 4, 2, 4), c(cat = 2, ftime = 2))
BarPlotCat(probs, lims, transparency = c(0, 0.5),
  panel_label = c("", "Skill < 0"), panel_title = c("Panel 1", "Panel 2"),
```

```
  toptitle = "Example with custom transparency and labels"
)
```

`ClimPalette` *Generate Climate Color Palettes*

Description

Generates a colorblind friendly color palette with color ranges useful in climate temperature variable plotting.

Usage

```
ClimPalette(palette = "bluered")

ClimColors(n, palette = "bluered")
```

Arguments

`palette`	A character string of palette. The current choices: • 'bluered': from blue through white to red. • 'redblue': from red through white to blue. • 'yellowred': from yellow through orange to red. • 'redyellow': from red through orange to yellow. • 'purpleorange': from purple through white to orange. • 'orangepurple': from orange through white to purple.
`n`	A number indicating how many colors to generate.

Value

ClimPalette() returns the function that generates the color palette and the attribute 'na_color'. ClimColors() returns a vector of the colors.

Examples

```
lims <- seq(-1, 1, length.out = 21)
cb <- ColorBarContinuous(lims, color_fun = ClimPalette('redyellow'), plot = FALSE)

cols <- ClimColors(20)
cb <- ColorBarContinuous(lims, cols, plot = FALSE)
```

ColorBarContinuous *Draws a Continuous Color Bar*

Description

Generates a color bar to use as colouring function for map plots and optionally draws it (horizontally or vertically) to be added to map multipanels or plots. It is possible to draw triangles at the ends of the colour bar to represent values that go beyond the range of interest. A number of options is provided to adjust the colours and the position and size of the components. The drawn colour bar spans a whole figure region and is compatible with figure layouts.

The generated colour bar consists of a set of breaks that define the length(brks) - 1 intervals to classify each of the values in each of the grid cells of a two-dimensional field. The corresponding grid cell of a given value of the field will be coloured in function of the interval it belongs to.

The only mandatory parameters are 'var_limits' or 'brks' (in its second format, see below).

Usage

```
ColorBarContinuous(
  brks = NULL,
  cols = NULL,
  vertical = TRUE,
  subsampleg = NULL,
  bar_limits = NULL,
  var_limits = NULL,
  triangle_ends = NULL,
  col_inf = NULL,
  col_sup = NULL,
  color_fun = ClimPalette(),
  plot = TRUE,
  draw_bar_ticks = TRUE,
  draw_separators = FALSE,
  triangle_ends_scale = 1,
  bar_extra_labels = NULL,
  extra_labels = NULL,
  title = NULL,
  title_scale = 1,
  bar_label_scale = 1,
  label_scale = NULL,
  bar_tick_scale = 1,
  tick_scale = NULL,
  bar_extra_margin = rep(0, 4),
  extra_margin = NULL,
  bar_label_digits = 4,
  label_digits = NULL,
  ...
```

)

Arguments

brks	Can be provided in two formats: • A single value with the number of breaks to be generated automatically, between the minimum and maximum specified in 'var_limits' (both inclusive). Hence the parameter 'var_limits' is mandatory if 'brks' is provided with this format. If 'bar_limits' is additionally provided, values only between 'bar_limits' will be generated. The higher the value of 'brks', the smoother the plot will look. • A vector with the actual values of the desired breaks. Values will be reordered by force to ascending order. If provided in this format, no other parameters are required to generate/plot the colour bar. This parameter is optional if 'var_limits' is specified. If 'brks' not specified but 'cols' is specified, it will take as value length(cols) + 1. If 'cols' is not specified either, 'brks' will take 21 as value.
cols	Vector of length(brks) - 1 valid colour identifiers, for each interval defined by the breaks. This parameter is optional and will be filled in with a vector of length(brks) - 1 colours generated with the function provided in 'color_fun' (`clim.colors` by default). 'cols' can have one additional colour at the beginning and/or at the end with the aim to colour field values beyond the range of interest represented in the colour bar. If any of these extra colours is provided, parameter 'triangle_ends' becomes mandatory in order to disambiguate which of the ends the colours have been provided for.
vertical	TRUE/FALSE for vertical/horizontal colour bar (disregarded if plot = FALSE).
subsampleg	The first of each subsampleg breaks will be ticked on the colorbar. Takes by default an approximation of a value that yields a readable tick arrangement (extreme breaks always ticked). If set to 0 or lower, no labels are drawn. See the code of the function for details or use 'bar_extra_labels' for customized tick arrangements.
bar_limits	Vector of two numeric values with the extremes of the range of values represented in the colour bar. If 'var_limits' go beyond this interval, the drawing of triangle extremes is triggered at the corresponding sides, painted in 'col_inf' and 'col_sup'. Either of them can be set as NA and will then take as value the corresponding extreme in 'var_limits' (hence a triangle end won't be triggered for these sides). Takes as default the extremes of 'brks' if available, else the same values as 'var_limits'.
var_limits	Vector of two numeric values with the minimum and maximum values of the field to represent. These are used to know whether to draw triangle ends at the extremes of the colour bar and what colour to fill them in with. If not specified, take the same value as the extremes of 'brks'. Hence the parameter 'brks' is mandatory if 'var_limits' is not specified.
triangle_ends	Vector of two logical elements, indicating whether to force the drawing of triangle ends at each of the extremes of the colour bar. This choice is automatically made from the provided 'brks', 'bar_limits', 'var_limits', 'col_inf' and

'col_sup', but the behaviour can be manually forced to draw or not to draw the triangle ends with this parameter. If 'cols' is provided, 'col_inf' and 'col_sup' will take priority over 'triangle_ends' when deciding whether to draw the triangle ends or not.

`col_inf` Colour to fill the inferior triangle end with. Useful if specifying colours manually with parameter 'cols', to specify the colour and to trigger the drawing of the lower extreme triangle, or if 'cols' is not specified, to replace the colour automatically generated by ColorBar().

`col_sup` Colour to fill the superior triangle end with. Useful if specifying colours manually with parameter 'cols', to specify the colour and to trigger the drawing of the upper extreme triangle, or if 'cols' is not specified, to replace the colour automatically generated by ColorBar().

`color_fun` Function to generate the colours of the color bar. Must take an integer and must return as many colours. The returned colour vector can have the attribute 'na_color', with a colour to draw NA values. This parameter is set by default to ClimPalette().

`plot` Logical value indicating whether to only compute its breaks and colours (FALSE) or to also draw it on the current device (TRUE).

`draw_bar_ticks` Whether to draw ticks for the labels along the colour bar (TRUE) or not (FALSE). TRUE by default. Disregarded if 'plot = FALSE'.

`draw_separators`
Whether to draw black lines in the borders of each of the colour rectancles of the colour bar (TRUE) or not (FALSE). FALSE by default. Disregarded if 'plot = FALSE'.

`triangle_ends_scale`
Scale factor for the drawn triangle ends of the colour bar, if drawn at all. Takes 1 by default (rectangle triangle proportional to the thickness of the colour bar). Disregarded if 'plot = FALSE'.

`bar_extra_labels`
Numeric vector of extra labels to draw along axis of the colour bar. The number of provided decimals will be conserved. Disregarded if 'plot = FALSE'.

`extra_labels` Deprecated. Use 'bar_extra_labels' instead.

`title` Title to draw on top of the colour bar, most commonly with the units of the represented field in the neighbour figures. Empty by default.

`title_scale` Scale factor for the 'title' of the colour bar. Takes 1 by default.

`bar_label_scale`
Scale factor for the labels of the colour bar. Takes 1 by default.

`label_scale` Deprecated. Use 'bar_label_scale' instead.

`bar_tick_scale` Scale factor for the length of the ticks of the labels along the colour bar. Takes 1 by default.

`tick_scale` Deprecated. Use 'bar_tick_scale' instead.

`bar_extra_margin`
Extra margins to be added around the colour bar, in the format c(y1, x1, y2, x2). The units are margin lines. Takes rep(0, 4) by default.

extra_margin	Deprecated. Use 'bar_extra_margin' instead.
bar_label_digits	Number of significant digits to be displayed in the labels of the colour bar, usually to avoid too many decimal digits overflowing the figure region. This does not have effect over the labels provided in 'bar_extra_labels'. Takes 4 by default.
label_digits	Deprecated. Use 'bar_label_digits' instead.
...	Arguments to be passed to the method. Only accepts the following graphical parameters: adj ann ask bg bty cex.lab cex.main cex.sub cin col.axis col.lab col.main col.sub cra crt csi cxy err family fg fig fin font font.axis font.lab font.main font.sub lend lheight ljoin lmitre lty lwd mai mex mfcol mfrow mfg mkh oma omd omi page pch pin plt pty smo srt tck tcl usr xaxp xaxs xaxt xlog xpd yaxp yaxs yaxt ylbias ylog. For more information about the parameters see 'par'.

Value

brks	Breaks used for splitting the range in intervals.
cols	Colours generated for each of the length(brks) - 1 intervals. Always of length length(brks) - 1.
col_inf	Colour used to draw the lower triangle end in the colour bar (NULL if not drawn at all).
col_sup	Colour used to draw the upper triangle end in the colour bar (NULL if not drawn at all).

Examples

```
cols <- c("dodgerblue4", "dodgerblue1", "forestgreen", "yellowgreen", "white",
          "white", "yellow", "orange", "red", "saddlebrown")
lims <- seq(-1, 1, 0.2)
cb <- ColorBarContinuous(lims, cols, plot = FALSE)
```

ColorBarDiscrete *Draw a Discrete Color Bar*

Description

Generates a color bar to use as colouring function for map plots and optionally draws it (horizontally or vertically) to be added to map multipanels or plots. A number of options is provided to adjust the colours and the position and size of the components. The drawn colour bar spans a whole figure region and is compatible with figure layouts.

The generated colour bar consists of a set of breaks that define the length(brks) - 1 intervals to classify each of the values in each of the grid cells of a two-dimensional field. The corresponding grid cell of a given value of the field will be coloured in function of the interval it belongs to.

Usage

```
ColorBarDiscrete(
  brks = NULL,
  cols = NULL,
  vertical = TRUE,
  subsampleg = NULL,
  bar_limits = NULL,
  var_limits = NULL,
  color_fun = ClimPalette(),
  plot = TRUE,
  draw_bar_ticks = FALSE,
  draw_separators = TRUE,
  labels = NULL,
  bar_extra_labels = NULL,
  extra_labels = NULL,
  title = NULL,
  title_scale = 1,
  bar_label_scale = 1,
  label_scale = NULL,
  bar_tick_scale = 1,
  tick_scale = NULL,
  bar_extra_margin = rep(0, 4),
  extra_margin = NULL,
  bar_label_digits = 4,
  label_digits = NULL,
  ...
)
```

Arguments

`brks` Can be provided in two formats:

- A single value with the number of breaks to be generated automatically, between the minimum and maximum specified in 'var_limits' (both inclusive). Hence the parameter 'var_limits' is mandatory if 'brks' is provided with this format. If 'bar_limits' is additionally provided, values only between 'bar_limits' will be generated. The higher the value of 'brks', the smoother the plot will look.
- A vector with the actual values of the desired breaks. Values will be reordered by force to ascending order. If provided in this format, no other parameters are required to generate/plot the colour bar.

This parameter is optional if 'var_limits' is specified. If 'brks' not specified but 'cols' is specified, it will take as value length(cols) + 1. If 'cols' is not specified either, 'brks' will take 21 as value.

`cols` Vector of length(brks) - 1 valid colour identifiers, for each interval defined by the breaks. This parameter is optional and will be filled in with a vector of length(brks) - 1 colours generated with the function provided in 'color_fun' (`clim.colors` by default).

'cols' can have one additional colour at the beginning and/or at the end with the aim to colour field values beyond the range of interest represented in the colour bar. If any of these extra colours is provided, parameter 'triangle_ends' becomes mandatory in order to disambiguate which of the ends the colours have been provided for.

`vertical` TRUE/FALSE for vertical/horizontal colour bar (disregarded if plot = FALSE).

`subsampleg` The first of each subsampleg breaks will be ticked on the colorbar. Takes by default an approximation of a value that yields a readable tick arrangement (extreme breaks always ticked). If set to 0 or lower, no labels are drawn. See the code of the function for details or use 'bar_extra_labels' for customized tick arrangements.

`bar_limits` Vector of two numeric values with the extremes of the range of values represented in the colour bar. If 'var_limits' go beyond this interval, the drawing of triangle extremes is triggered at the corresponding sides, painted in 'col_inf' and 'col_sup'. Either of them can be set as NA and will then take as value the corresponding extreme in 'var_limits' (hence a triangle end won't be triggered for these sides). Takes as default the extremes of 'brks' if available, else the same values as 'var_limits'.

`var_limits` Vector of two numeric values with the minimum and maximum values of the field to represent. These are used to know whether to draw triangle ends at the extremes of the colour bar and what colour to fill them in with. If not specified, take the same value as the extremes of 'brks'. Hence the parameter 'brks' is mandatory if 'var_limits' is not specified.

`color_fun` Function to generate the colours of the color bar. Must take an integer and must return as many colours. The returned colour vector can have the attribute 'na_color', with a colour to draw NA values. This parameter is set by default to ClimPalette().

`plot` Logical value indicating whether to only compute its breaks and colours (FALSE) or to also draw it on the current device (TRUE).

`draw_bar_ticks` Whether to draw ticks for the labels along the colour bar (TRUE) or not (FALSE). TRUE by default. Disregarded if 'plot = FALSE'.

`draw_separators` Whether to draw black lines in the borders of each of the colour rectancles of the colour bar (TRUE) or not (FALSE). FALSE by default. Disregarded if 'plot = FALSE'.

`labels` A charater string vector of the names of colors. Must be the same length as 'cols'.

`bar_extra_labels` Numeric vector of extra labels to draw along axis of the colour bar. The number of provided decimals will be conserved. Disregarded if 'plot = FALSE'.

`extra_labels` Deprecated. Use 'bar_extra_labels' instead.

`title` Title to draw on top of the colour bar, most commonly with the units of the represented field in the neighbour figures. Empty by default.

`title_scale` Scale factor for the 'title' of the colour bar. Takes 1 by default.

`bar_label_scale`
: Scale factor for the labels of the colour bar. Takes 1 by default.

`label_scale`
: Deprecated. Use 'bar_label_scale' instead.

`bar_tick_scale`
: Scale factor for the length of the ticks of the labels along the colour bar. Takes 1 by default.

`tick_scale`
: Deprecated. Use 'bar_tick_scale' instead.

`bar_extra_margin`
: Extra margins to be added around the colour bar, in the format c(y1, x1, y2, x2). The units are margin lines. Takes rep(0, 4) by default.

`extra_margin`
: Deprecated. Use 'bar_extra_margin' instead.

`bar_label_digits`
: Number of significant digits to be displayed in the labels of the colour bar, usually to avoid too many decimal digits overflowing the figure region. This does not have effect over the labels provided in 'bar_extra_labels'. Takes 4 by default.

`label_digits`
: Deprecated. Use 'bar_label_digits' instead.

`...`
: Arguments to be passed to the method. Only accepts the following graphical parameters:
adj ann ask bg bty cex.lab cex.main cex.sub cin col.axis col.lab col.main col.sub cra crt csi cxy err family fg fig fin font font.axis font.lab font.main font.sub lend lheight ljoin lmitre lty lwd mai mex mfcol mfrow mfg mkh oma omd omi page pch pin plt pty smo srt tck tcl usr xaxp xaxs xaxt xlog xpd yaxp yaxs yaxt ylbias ylog.
For more information about the parameters see 'par'.

Value

`brks`
: Breaks used for splitting the range in intervals.

`cols`
: Colours generated for each of the length(brks) - 1 intervals. Always of length length(brks) - 1.

Examples

```
cb <- ColorBarDiscrete(
  brks = 0:4, cols = c("green1", "green2", "green3", "green4"),
  vertical = FALSE, labels = paste0('lev ', 1:4), bar_label_scale = 1.5,
  bar_extra_margin = c(0.5, 2, 0.5, 2), plot = FALSE)
```

Hatching *Draws hatching or crosshatching over a mask on a map*

Description

Adds hatching or crosshatching lines over grid cells of a map based on a logical mask. The hatching is applied only to grid cells where the mask is TRUE (or 1). A number of graphical options are available to customize the appearance of the hatching, such as line density, angle, line width, and color. Optionally, crosshatching can be drawn using a second set of lines in the opposite direction. The function assumes that the input mask is a 2D array with dimensions corresponding to latitude (rows) and longitude (columns), and it is designed to be used as a layer over existing map plots (e.g., within a call to `VizEquiMap`) and does not initiate a plot by itself.

Usage

```
Hatching(
  hatching_mask,
  lat,
  lon,
  data = NULL,
  hatching_density = 10,
  hatching_angle = 45,
  hatching_color = "#252525",
  hatching_lwd = 0.5,
  hatching_cross = FALSE
)
```

Arguments

`hatching_mask`	Logical or binary (0/1) array with two named dimensions: c(latitude, longitude).Hatching is applied to grid cells where 'hatching_mask' is TRUE (or 1). Arrays with dimensions c(longitude, latitude) are also accepted, but the resulting hatching may appear transposed. To ensure correct alignment with the map, provide 'data'. The function will compare the dimension order of 'hatching_mask' and 'data', and automatically transpose 'hatching_mask' if the latitude and longitude dimensions appear to be reversed.
`lat`	Numeric vector of latitude locations of the cell centers of the grid.
`lon`	Numeric vector of longitude locations of the cell centers of the grid.
`data`	Array of the data that the hatching will be drawn over. The array should have named latidude and longitude dimensions. If the dimension order is reversed relative to 'hatching_mask', the mask is automatically transposed. Default is NULL.
`hatching_density`	The density of shading lines, in lines per inch. A zero value of density means no shading nor filling, whereas negative values and NA suppress shading (and so allow color filling). NULL means that no shading lines are drawn. Default is 10.
`hatching_angle`	The slope of shading lines, given as an angle in degrees (counter-clockwise). Default is 45.
`hatching_color`	Color of the hatching lines. Default is `"#252525"`.

hatching_lwd The line width, a positive number. The interpretation is device-specific, and some devices do not implement line widths less than one. Default is 0.5.

hatching_cross A logical value indicating crosshatching. If TRUE, adds a second set of lines in the opposite angle. Default is FALSE.

Value

Invisibly returns NULL. The function is intended to be used as an overlay layer (e.g., when called within VizEquiMap()). It draws hatching or crosshatching onto an existing graphics device.

Examples

```
mask_small <- array(c(TRUE, FALSE, TRUE, FALSE), dim = c(lat = 2, lon = 2))
lat_small <- 1:2
lon_small <- 1:2
image(lon_small, lat_small, matrix(1:4, nrow = 2))
Hatching(hatching_mask = mask_small, lat = lat_small, lon = lon_small,
         hatching_lwd = 2)
```

---

map_temp *Sample Of Experimental And Observational Climate Spatial Data*

Description

This sample data contain gridded seasonal forecast and corresponding observational data from ECMWF-System 5 forecast system and ERA-5 reconstruction. Specifically, for the monthly mean 2-meter temperature ("tas") variable, the first 3 ensemble members, the first 5 forecast time steps from November initial month, year 2000 to 2005, the Iberian Peninsula region (35N-45N, 10W-5E).

Details

The package "startR" is used to load the data from the data esarchive in the Earth Sciences Department of Barcelona Supercomputing Center.

```
# 1. Load libraries and define common variables
library(startR)
library(CSTools)

sdates <- sapply(2000:2005, function(x) paste0(x, '1101'))
lonmax <- 5
lonmin <- -10
latmax <- 45
latmin <- 35

# 2. Load exp
```

```
repos_exp <- paste0('/esarchive/exp/ecmwf/system5c3s/monthly_mean/',
                    '$var$_f6h/$var$_$sdate$.nc')

exp <- Start(dat = repos_exp,
             var = 'tas',
             member = indices(1:3),
             sdate = sdates,
             time = indices(1:5),
             lat = values(list(latmin, latmax)),
             lat_reorder = Sort(decreasing = FALSE),
             lon = values(list(lonmin, lonmax)),
             lon_reorder = CircularSort(-180, 180),
             synonims = list(lon = c('lon', 'longitude'),
                             lat = c('lat', 'latitude'),
                             member = c('member', 'ensemble')),
             return_vars = list(lat = NULL, lon = NULL, time = 'sdate'),
             retrieve = TRUE)


# 3. Load obs

exp_time <- attr(exp, "Variables")$common$time
obs_date <- array(format(exp_time, "

path_obs <- '/esarchive/recon/ecmwf/era5/monthly_mean/$var$_f1h-r1440x721cds/$var$_$date$.nc'
obs <- Start(dat = path_obs,
             var = 'tas',
             date = obs_date,
             split_multiselected_dims = TRUE,
             lat = values(list(latmin, latmax)),
             lat_reorder = Sort(decreasing = FALSE),
             lon = values(list(lonmin, lonmax)),
             lon_reorder = CircularSort(-180, 180),
             synonims = list(lon = c('lon', 'longitude'),
                             lat = c('lat', 'latitude')),
             transform = CDORemapper,
             transform_extra_cells = 2,
             transform_params = list(grid = 'r360x181',
                                     method = 'conservative'),
             transform_vars = c('lat', 'lon'),
             return_vars = list(lon = NULL, lat = NULL, time = 'date'),
             retrieve = TRUE)


# 4. Check data

obs_time <- attr(obs,"Variables")$common$time
```

```
 identical(format(obs_time, "
#[1] TRUE
 exp_lat <- attr(exp, "Variables")$common$lat
 exp_lon <- attr(exp, "Variables")$common$lon
 obs_lat <- attr(obs, "Variables")$common$lat
 obs_lon <- attr(obs, "Variables")$common$lon

 all.equal(exp_lat, obs_lat, check.attributes = F)
#[1] TRUE
 all.equal(exp_lon, obs_lon, check.attributes = F)
#[1] TRUE


# 5. Combine into one object

 map_temp <- list()
 map_temp$exp <- exp
 map_temp$obs <- obs
```

ShapeToMask *Convert Shapefile to Mask Array*

Description

This function reads a shapefile (.shp) containing information about polygonal regions. It then transfers the shapefile data into an array and subsets the output based on requested region names or IDs. The accepted shapefile databases are 'NUTS', 'LAU', and 'GADM', each with its own unique format. However, the function can use other shapefiles databases with specifying the categories names with the parameter 'id_shape_col'.

Usage

```
ShapeToMask(
  shp_file,
  ref_grid,
  compute_area_coverage = FALSE,
  shp_system = "NUTS",
  reg_names = NULL,
  reg_ids = NULL,
  reg_level = 3,
  lat_dim = NULL,
  lon_dim = NULL,
  region = FALSE,
  check_valid = FALSE,
  find_min_dist = FALSE,
  max_dist = 50,
```

```
  ncores = NULL,
  fileout = NULL,
  units = "degrees",
  id_shape_col = NULL,
  name_shape_col = NULL,
  ...
)
```

Arguments

shp_file	A character string indicating the shp file path.
ref_grid	A character string indicating the path to the reference data. Either (1) a netCDF file or (2) a list of lon and lat to provide the reference grid points. It is NULL by default.
compute_area_coverage	A logical value indicating the method to find the intersection of the reference grid and the shapefile. When it is TRUE, the method used is the calculation of the area coverage fraction of intersection. If it is FALSE, the method used is searching if the centroid of the grid cell falls inside the shapefile or not. It is FALSE by default.
shp_system	A character string containing the Shapefile System Database Name used to subset the shapefile into regions by using parameters 'reg_ids' or 'reg_names'. The accepted systems are: 'NUTS', 'LAU', and 'GADM'. When it is used, you must specify either 'reg_ids' or 'reg_names'; if you don't need to subset different regions, set it to NULL. It is set to "NUTS" by default (optional).
reg_names	A named list of character string vectors indicating the country and the region name. The name of the list stands for the country name code and the vector character strings indicate the region name for each country. It is NULL by default (optional).
reg_ids	A character string indicating the unique ID in shapefile. It is NULL by default (optional).
reg_level	An integer number from 1 to 3 indicating the 'NUTS' dataset level. For other datasets this parameter is not used. One mask can only have a unique level. It is set to 3 by default (optional).
lat_dim	A character string indicating the latitudinal dimension. If it is NULL, the latitudinal name will be searched using an internal function with the following possible names: 'lat', 'latitude', 'y', 'j' and 'nav_lat'. It is set to NULL by default.
lon_dim	A character string indicating the longitudinal dimension. If it is NULL, the longitudinal name will be searched using an internal function with the following possible names: 'lon', 'longitude', 'x', 'i' and 'nav_lon'. It is set to NULL by default.
region	A logical value indicating if we want a dimension for the regions in the resulting mask array. It is FALSE by default.
check_valid	A logical value that when it is TRUE it uses the function 'sf::st_make_valid' applied to the shapefile and to the coordinates.

find_min_dist	A logical value indicating if we want to look for the nearest coordinate between the shapefile region and the reference grid when there is no intersection between the shapefile and the reference grid. It is FALSE by default.
max_dist	A numeric value indicating the maximum distance is accepted to the closest gridpoint when there is no intersection between the shapefile and the reference grid.
ncores	The number of parallel processes to spawn for the use for parallel computation in multiple cores.
fileout	A character string of the path to save the NetCDF mask. If not specified (default), only the mask array will be returned.
units	A character string indicating if your GIS files has a grid in degrees or meters. If it is NULL, the units will be set as "meters" with the following possible names: 'degrees', 'meters'
id_shape_col	A character string indicating in the shape file which is the name of the column with the values of the IDs of the different polygons. It is NULL by default.
name_shape_col	A character string indicating in the shape file which is the name of the column with the values of the names of the different polygons. It is NULL by default.
...	Arguments passed on to 's2_options' in function 'st_intersection'. See 's2 package'.

Details

To ensure accurate comparison with the shapefile, the function loads a reference dataset that provides longitude and latitude information. By intersecting each subset of the shapefile with the reference coordinates, the function selects only the desired regions. The final step involves creating a mask array. Depending on the chosen option, the mask array is either returned as the function's output or saved into a NetCDF format in the specified directory.

Note: Modules GDAL, PROJ and GEOS are required.

Value

A multidimensional array containing a mask array with longitude and latitude dimensions. If 'region' is TRUE, there will be a dimension for the region.

Examples

```
## Not run:
# Example using an external shapefile not distributed with the package
shp_file <- paste0('/esarchive/shapefiles/NUTS3/NUTS_RG_60M_2021_4326.shp/',
                   'NUTS_RG_60M_2021_4326.shp')
ref_grid <- list(lon = seq(10, 40, 0.5), lat = seq(40, 85, 0.5))
NUTS_name <- list(FI = c('Lappi', 'Kainuu'), SI = c('Pomurska', 'Podravska'))
mask <- ShapeToMask(shp_file = shp_file, ref_grid = ref_grid,
                    reg_names = NUTS_name)

## End(Not run)
```

ts_temp *Sample Of Experimental And Observational Climate Time Series Data*

Description

This sample data contain gridded seasonal forecast and corresponding observational data from ECMWF-System 5 forecast system and ERA-5 reconstruction. Specifically, for the monthly mean 2-meter temperature ("tas") variable, the first 3 ensemble members, the first 5 forecast time steps from November initial month, year 2000 to 2005, spatial average of the Iberian Peninsula region (35N-45N, 10W-5E).

Details

The package "startR" is used to load the data from the data esarchive in the Earth Sciences Department of Barcelona Supercomputing Center (check `map_temp`). Caclulate the weighted area-mean by function `ClimProjDiags::WeightedMean`.

```
# 1. Load data "map_temp" first and load libraries

library(ClimProjDiags)

map_temp <- esviz::map_temp
attrs_exp <- attributes(map_temp$exp)[-1]
attrs_obs <- attributes(map_temp$obs)[-1]

# 2. Calculate weighted area mean

lats <- attr(map_temp$exp, "Variables")$common$lat
lons <- attr(map_temp$exp, "Variables")$common$lon

ts_temp <- list()
ts_temp$exp <- WeightedMean(map_temp$exp, lon = lons, lat = lats)
ts_temp$obs <- WeightedMean(map_temp$obs, lon = lons, lat = lats)

attributes(ts_temp$exp) <- c(attributes(ts_temp$exp), attrs_exp)
attributes(ts_temp$obs) <- c(attributes(ts_temp$obs), attrs_obs)
```

Viz2VarsVsLTime *Plot two scores with confidence intervals in a common plot*

Description

Plot two input variables that have the same dimensions in a common plot. One plot for all experiments. The input variables should have dimensions (nexp/nmod, nltime).

Usage

```
Viz2VarsVsLTime(
  var1,
  var2,
  toptitle = "",
  ytitle = "",
  monini = 1,
  freq = 12,
  nticks = NULL,
  limits = NULL,
  listexp = c("exp1", "exp2", "exp3"),
  listvars = c("var1", "var2"),
  biglab = FALSE,
  hlines = NULL,
  drawleg = TRUE,
  leg = NULL,
  siglev = FALSE,
  title_scale = 1,
  sizetit = NULL,
  show_conf = TRUE,
  fileout = NULL,
  width = 8,
  height = 5,
  size_units = "in",
  res = 100,
  ...
)
```

Arguments

var1	Matrix of dimensions (nexp/nmod, nltime).
var2	Matrix of dimensions (nexp/nmod, nltime).
toptitle	Main title, optional.
ytitle	Title of Y-axis, optional.
monini	Starting month between 1 and 12. Default = 1.
freq	1 = yearly, 12 = monthly, 4 = seasonal, ... Default = 12.
nticks	Number of ticks and labels on the x-axis, optional.
limits	c(lower limit, upper limit): limits of the Y-axis, optional.
listexp	List of experiment names, up to three, optional.
listvars	List of names of input variables, optional.
biglab	TRUE/FALSE for presentation/paper plot. Default = FALSE.
hlines	c(a, b, ...) Add horizontal black lines at Y-positions a, b, ... The default value is NULL.
drawleg	TRUE/FALSE if legend should be added or not to the plot. Default = TRUE.

leg	Deprecated. Use 'drawleg' instead.
siglev	TRUE/FALSE if significance level should replace confidence interval. Default = FALSE.
title_scale	Multiplicative factor to change title size, optional.
sizetit	Deprecated. Use 'title_scale' instead.
show_conf	TRUE/FALSE to show/not confidence intervals for input variables.
fileout	Name of output file. Extensions allowed: eps/ps, jpeg, png, pdf, bmp and tiff. The default value is NULL.
width	File width, in the units specified in the parameter size_units (inches by default). Takes 8 by default.
height	File height, in the units specified in the parameter size_units (inches by default). Takes 5 by default.
size_units	Units of the size of the device (file or window) to plot in. Inches ('in') by default. See ?Devices and the creator function of the corresponding device.
res	Resolution of the device (file or window) to plot in. See ?Devices and the creator function of the corresponding device.
...	Arguments to be passed to the method. Only accepts the following graphical parameters: adj ann ask bg bty cex.sub cin col.axis col.lab col.main col.sub cra crt csi cxy err family fg fig font font.axis font.lab font.main font.sub lend lheight ljoin lmitre mar mex mfcol mfrow mfg mkh oma omd omi page pch plt smo srt tck tcl usr xaxp xaxs xaxt xlog xpd yaxp yaxs yaxt ylbias ylog For more information about the parameters see 'par'.

Value

A figure in popup window by default, or saved to the specified path via `fileout`.

Examples

```
clim <- s2dv::Clim(ts_temp$exp, ts_temp$obs, time_dim = "sdate",
                   dat_dim = c("dat", "member"))
ano_exp <- s2dv::Ano(ts_temp$exp, clim$clim_exp)
ano_obs <- s2dv::Ano(ts_temp$obs, clim$clim_obs)
corr_ano <- s2dv::Corr(s2dv::MeanDims(ano_exp, 'member'), ano_obs,
                       time_dim = 'sdate', dat_dim = 'dat')
input_cor <- array(dim = c(dat = 1, 3, time = 5))
input_cor[, 1, ] <- corr_ano$conf.lower[, 1, 1, ]
input_cor[, 2, ] <- corr_ano$corr[, 1, 1, ]
input_cor[, 3, ] <- corr_ano$conf.upper[, 1, 1, ]

rms_ano <- s2dv::RMS(s2dv::MeanDims(ano_exp, 'member'), ano_obs,
                     time_dim = 'sdate', dat_dim = 'dat')

input_rms <- array(dim = c(dat = 1, 3, time = 5))
input_rms[, 1, ] <- rms_ano$conf.lower[, 1, 1, ]
input_rms[, 2, ] <- rms_ano$rms[, 1, 1, ]
```

```
input_rms[, 3, ] <- rms_ano$conf.upper[, 1, 1, ]
Viz2VarsVsLTime(input_cor, input_rms,
                toptitle = "Time correlation and RMSE with ERA5",
                ytitle = "K", title_scale = 0.7,
                monini = 11, freq = 1, limits = c(-1, 5),
                listexp = c('SEAS5'), listvars = c('Corr', 'RMSE'),
                fileout = NULL)
```

VizACC *Plot Plumes/Timeseries Of Anomaly Correlation Coefficients*

Description

Plots plumes/timeseries of ACC from an array with dimensions (output from ACC()):
c(nexp, nobs, nsdates, nltime, 4)
where the fourth dimension is of length 4 and contains the lower limit of the 95% confidence interval, the ACC, the upper limit of the 95% confidence interval and the 95% significance level given by a one-sided T-test.

Usage

```
VizACC(
  ACC,
  sdates,
  toptitle = "",
  title_scale = 1,
  sizetit = NULL,
  ytitle = "",
  limits = NULL,
  legends = NULL,
  freq = 12,
  biglab = FALSE,
  fill = FALSE,
  linezero = FALSE,
  points = TRUE,
  vlines = NULL,
  fileout = NULL,
  width = 8,
  height = 5,
  size_units = "in",
  res = 100,
  ...
)
```

Arguments

ACC	An ACC array with with dimensions: c(nexp, nobs, nsdates, nltime, 4) with the fourth dimension of length 4 containing the lower limit of the 95% confidence interval, the ACC, the upper limit of the 95% confidence interval and the 95% significance level.
sdates	A character vector of startdates: c('YYYYMMDD','YYYYMMDD').
toptitle	A character string of the main title, optional.
title_scale	A multiplicative factor to scale title size, optional.
sizetit	Deprecated. Use 'title_scale' instead.
ytitle	A character string of the title of Y-axis for each experiment: c('', ''), optional.
limits	A numeric vector c(lower limit, upper limit): limits of the Y-axis, optional.
legends	A character vector of flags to be written in the legend, optional.
freq	A integer: 1 = yearly, 12 = monthly, 4 = seasonal, ... Default: 12.
biglab	A logical value for presentation/paper plot, Default = FALSE.
fill	A logical value if filled confidence interval. Default = FALSE.
linezero	A logical value if a line at y=0 should be added. Default = FALSE.
points	A logical value if points instead of lines. Default = TRUE. Must be TRUE if only 1 leadtime.
vlines	A vector of x location where to add vertical black lines, optional.
fileout	A character string of the output file name. Extensions allowed: eps/ps, jpeg, png, pdf, bmp and tiff. Default is NULL.
width	A numeric of the file width, in the units specified in the parameter size_units (inches by default). Takes 8 by default.
height	A numeric of the file height, in the units specified in the parameter size_units (inches by default). Takes 5 by default.
size_units	A character string of the units of the size of the device (file or window) to plot in. Inches ('in') by default. See ?Devices and the creator function of the corresponding device.
res	Resolution of the device (file or window) to plot in. See ?Devices and the creator function of the corresponding device.
...	Arguments to be passed to the method. Only accepts the following graphical parameters: adj ann ask bg bty cex.sub cin col.axis col.lab col.main col.sub cra crt csi cxy err family fg fig fin font font.axis font.lab font.main font.sub lend lheight ljoin lmitre mar mex mfcol mfrow mfg mkh oma omd omi page plt smo srt tck tcl usr xaxp xaxs xaxt xlog xpd yaxp yaxs yaxt ylbias ylog For more information about the parameters see 'par'.

Value

A figure in popup window by default, or saved to the specified path via fileout.

Examples

```
ano <- s2dv::Ano_CrossValid(map_temp$exp, map_temp$obs, memb = FALSE,
                            dat_dim = c('dat', 'member'), memb_dim = 'member')
lats <- attr(map_temp$exp, "Variables")$common$lat
lons <- attr(map_temp$exp, "Variables")$common$lon
acc <- s2dv::ACC(ano$exp, ano$obs, lat = lats, lon = lons,
                 dat_dim = 'dat', memb_dim = 'member')
input_acc <- array(dim = c(dim(acc$acc)[c('nexp', 'nobs', 'sdate', 'time')], 4))
input_acc[, , , , 1] <- acc$conf.lower
input_acc[, , , , 2] <- acc$acc
input_acc[, , , , 3] <- acc$conf.upper
input_acc[, , , , 4] <- acc$p.val
sdates <- paste0(2000:2005, '1101')
VizACC(input_acc, sdates,
       toptitle = "Spatial anomaly corr. coeff. with ERA5",
       ytitle = "K", title_scale = 0.7, freq = 12,
       legends = 'SEAS5', fileout = NULL)
```

`VizAnimateMap` *Animate Maps of Forecast/Observed Values or Scores Over Forecast Time*

Description

Create animations of maps in an equi-rectangular or stereographic projection, showing the anomalies, the climatologies, the mean InterQuartile Range, Maximum-Mininum, Standard Deviation, Median Absolute Deviation, the trends, the RMSE, the correlation or the RMSSS, between modelled and observed data along the forecast time (lead-time) for all input experiments and input observational datasets.

Usage

```
VizAnimateMap(
  data,
  lon,
  lat,
  toptitle = rep("", 11),
  title_scale = 1,
  sizetit = NULL,
  units = "",
  monini = 1,
  freq = 12,
  msk95lev = FALSE,
  brks = NULL,
  cols = NULL,
  filled.continents = FALSE,
  lonmin = 0,
```

```
  lonmax = 360,
  latmin = -90,
  latmax = 90,
  intlon = 20,
  intlat = 30,
  drawleg = TRUE,
  subsampleg = 1,
  colNA = "white",
  equi = TRUE,
  fileout = c("output1_animvsltime.gif", "output2_animvsltime.gif",
    "output3_animvsltime.gif"),
  var = NULL,
  ...
)
```

Arguments

data	Matrix of dimensions (nltime, nlat, nlon) or (nexp/nmod, nltime, nlat, nlon) or (nexp/nmod, 3/4, nltime, nlat, nlon) or (nexp/nmod, nobs, 3/4, nltime, nlat, nlon).
lon	Vector containing longtitudes (degrees).
lat	Vector containing latitudes (degrees).
toptitle	c(",", ...) array of main title for each animation, optional. If RMS, RMSSS, correlations: first exp with successive obs, then second exp with successive obs, etc ...
title_scale	Scale factor for the figure top title. Defaults to 1.
sizetit	Deprecated. Use 'title_scale' instead.
units	Units, optional.
monini	Starting month between 1 and 12. Default = 1.
freq	1 = yearly, 12 = monthly, 4 = seasonal ...
msk95lev	TRUE/FALSE grid points with dots if 95% significance level reached. Default = FALSE.
brks	Limits of colour levels, optional. For example: seq(min(data), max(data), (max(data) - min(data)) / 10).
cols	Vector of colours of length(brks) - 1, optional.
filled.continents	Continents filled in grey (TRUE) or represented by a black line (FALSE). Default = TRUE. Filling unavailable if crossing Greenwich and equi = TRUE. Filling unavailable if square = FALSE and equi = TRUE.
lonmin	Westward limit of the domain to plot (> 0 or < 0). Default : 0 degrees.
lonmax	Eastward limit of the domain to plot (> 0 or < 0). lonmax > lonmin. Default : 360 degrees.
latmin	Southward limit of the domain to plot. Default : -90 degrees.
latmax	Northward limit of the domain to plot. Default : 90 degrees.

intlon	Interval between longitude ticks on x-axis. Default = 20 degrees.
intlat	Interval between latitude ticks on y-axis for equi = TRUE or between latitude circles for equi = FALSE. Default = 30 degrees.
drawleg	Draw a colorbar. Can be FALSE only if square = FALSE or equi = FALSE. Default = TRUE.
subsampleg	Supsampling factor of the interval between ticks on colorbar. Default = 1 = every colour level.
colNA	Color used to represent NA. Default = 'white'.
equi	TRUE/FALSE == cylindrical equidistant/stereographic projection. Default: TRUE.
fileout	c(", ", ...) array of output file name for each animation. If RMS, RMSSS, correlations : first exp with successive obs, then second exp with successive obs, etc ...
var	Deprecated. Use 'data' instead.
...	Arguments to be passed to the method. Only accepts the following graphical parameters: adj ann ask bty cex cex.axis cex.lab cex.main cex.sub cin col.axis col.lab col.main col.sub cra crt csi cxy err family fg fig font font.axis font.lab font.main font.sub las lheight ljoin lmitre lty lwd mai mar mex mfcol mfrow mfg mgp mkh oma omd omi page pch plt pty smo srt tck tcl usr xaxp xaxs xaxt xlog xpd yaxp yaxs yaxt ylbias ylog. For more information about the parameters see 'par'.

Details

Examples of input:

1. Outputs from clim (exp, obs, memb = FALSE): (nmod, nltime, nlat, nlon) or (nobs, nltime, nlat, nlon)
2. Model output from load/ano/smoothing: (nmod, nmemb, sdate, nltime, nlat, nlon) then passed through spread(data, posdim = 2, narm = TRUE) & mean1dim(data, posdim = 3, narm = TRUE) or through trend(mean1dim(data, 2), posTR = 2): (nmod, 3, nltime, nlat, nlon) animates average along start dates of IQR/MaxMin/SD/MAD across members or trends of the ensemble-mean computed accross the start dates.
3. model and observed output from load/ano/smoothing: (nmod, nmemb, sdate, nltime, nlat, nlon) & (nobs, nmemb, sdate, nltime, nlat, nlon) then averaged along members mean1dim(var_exp/var_obs, posdim = 2): (nmod, sdate, nltime, nlat, nlon) (nobs, sdate, nltime, nlat, nlon) then passed through corr(exp, obs, posloop = 1, poscor = 2) or RMS(exp, obs, posloop = 1, posRMS = 2): (nmod, nobs, 3, nltime, nlat, nlon) animates correlations or RMS between each exp & each obs against leadtime.

Value

A figure in popup window by default, or saved to the specified path via fileout.

Examples

```
clim <- s2dv::Clim(map_temp$exp, map_temp$obs, memb = FALSE,
                   dat_dim = c('dat', 'member'), memb_dim = 'member')
lats <- attr(map_temp$exp, "Variables")$common$lat
lons <- attr(map_temp$exp, "Variables")$common$lon


VizAnimateMap(clim$clim_exp[1, 1, , , ], lon = lons, lat = lats,
  toptitle = "climatology of decadal prediction", title_scale = 1,
  units = "K", brks = seq(270, 300, 3), monini = 11, freq = 12,
  msk95lev = FALSE, filled.continents = FALSE, intlon = 10, intlat = 10,
  drawleg = FALSE, fileout = "VizAnimateMap.gif")
unlink("VizAnimateMap.gif")
```

VizAno *Plot Anomaly or Raw time series*

Description

Plots time series of raw or smoothed anomalies of any variable output from Load() or Ano() or or Ano_CrossValid() or Smoothing().

Usage

```
VizAno(
  exp_ano,
  obs_ano = NULL,
  sdates,
  toptitle = rep("", 15),
  ytitle = rep("", 15),
  limits = NULL,
  legends = NULL,
  freq = 12,
  biglab = FALSE,
  fill = TRUE,
  memb = TRUE,
  ensmean = TRUE,
  linezero = FALSE,
  points = FALSE,
  vlines = NULL,
  title_scale = 1,
  sizetit = NULL,
  fileout = NULL,
  width = 8,
  height = 5,
  size_units = "in",
```

```
  res = 100,
  ...
)
```

Arguments

exp_ano	A numerical array containing the experimental data: c(nmod/nexp, nmemb/nparam, nsdates, nltime).
obs_ano	A numerical array containing the observational data: c(nobs, nmemb, nsdates, nltime)
sdates	A character vector of start dates in the format of c('YYYYMMDD','YYYYMMDD').
toptitle	Main title for each experiment: c("","), optional.
ytitle	Title of Y-axis for each experiment: c("","), optional.
limits	c(lower limit, upper limit): limits of the Y-axis, optional.
legends	List of observational dataset names, optional.
freq	1 = yearly, 12 = monthly, 4 = seasonal, ... Default: 12.
biglab	TRUE/FALSE for presentation/paper plot. Default = FALSE.
fill	TRUE/FALSE if the spread between members should be filled. Default = TRUE.
memb	TRUE/FALSE if all members/only the ensemble-mean should be plotted. Default = TRUE.
ensmean	TRUE/FALSE if the ensemble-mean should be plotted. Default = TRUE.
linezero	TRUE/FALSE if a line at y=0 should be added. Default = FALSE.
points	TRUE/FALSE if points instead of lines should be shown. Default = FALSE.
vlines	List of x location where to add vertical black lines, optional.
title_scale	Multiplicative factor to scale title size, optional.
sizetit	Deprecated. Use 'title_scale' instead.
fileout	Name of the output file for each experiment: c("","). Extensions allowed: eps/ps, jpeg, png, pdf, bmp and tiff. If filenames with different extensions are passed, it will be considered only the first one and it will be extended to the rest. The default value is NULL, which the pop-up window shows.
width	File width, in the units specified in the parameter size_units (inches by default). Takes 8 by default.
height	File height, in the units specified in the parameter size_units (inches by default). Takes 5 by default.
size_units	Units of the size of the device (file or window) to plot in. Inches ('in') by default. See ?Devices and the creator function of the corresponding device.
res	Resolution of the device (file or window) to plot in. See ?Devices and the creator function of the corresponding device.
...	Arguments to be passed to the method. Only accepts the following graphical parameters: adj ann ask bg bty cex.sub cin col.axis col.lab col.main col.sub cra crt csi cxy err family fg fig font font.axis font.lab font.main font.sub lend lheight ljoin lmitre mar mex mfcol mfrow mfg mkh oma omd omi page plt smo srt tck tcl usr xaxp xaxs xaxt xlog xpd yaxp yaxs yaxt ylbias ylog For more information about the parameters see 'par'.

Value

A figure in popup window by default, or saved to the specified path via `fileout`.

Examples

```
dim(ts_temp$exp) <- c(dat = 1, member = 3, sdate = 6, time = 5)
dim(ts_temp$obs) <- c(dat = 1, member = 1, sdate = 6, time = 5)
sdates <- paste0(2000:2005, '1101')
VizAno(ts_temp$exp, ts_temp$obs, sdates,
       toptitle = "Raw 'tas'", ytitle = 'K', legends = 'ERA5', biglab = FALSE)
```

VizBoxWhisker *Box-And-Whisker Plot of Time Series with Ensemble Distribution*

Description

Produce time series of box-and-whisker plot showing the distribution of the members of a forecast vs. the observed evolution. The correlation between forecast and observational data is calculated and displayed. Only works for n-monthly to n-yearly time series.

Usage

```
VizBoxWhisker(
  exp,
  obs,
  toptitle = "",
  ytitle = "",
  monini = 1,
  yearini = 0,
  freq = 1,
  expname = "exp 1",
  obsname = "obs 1",
  drawleg = TRUE,
  fileout = NULL,
  width = 8,
  height = 5,
  size_units = "in",
  res = 100,
  ...
)
```

Arguments

`exp`	Forecast array of multi-member time series, e.g., the NAO index of one experiment. The expected dimensions are c(members, start dates/forecast horizons). A vector with only the time dimension can also be provided. Only monthly or lower frequency time series are supported. See parameter freq.

obs	Observational vector or array of time series, e.g., the NAO index of the observations that correspond the forecast data in exp. The expected dimensions are c(start dates/forecast horizons) or c(1, start dates/forecast horizons). Only monthly or lower frequency time series are supported. See parameter freq.
toptitle	Character string to be drawn as figure title.
ytitle	Character string to be drawn as y-axis title.
monini	Number of the month of the first time step, from 1 to 12.
yearini	Year of the first time step.
freq	Frequency of the provided time series: 1 = yearly, 12 = monthly,
expname	Experimental dataset name.
obsname	Name of the observational reference dataset.
drawleg	TRUE/FALSE: whether to draw the legend or not.
fileout	Name of output file. Extensions allowed: eps/ps, jpeg, png, pdf, bmp and tiff. Default = 'output_PlotBox.ps'.
width	File width, in the units specified in the parameter size_units (inches by default). Takes 8 by default.
height	File height, in the units specified in the parameter size_units (inches by default). Takes 5 by default.
size_units	Units of the size of the device (file or window) to plot in. Inches ('in') by default. See ?Devices and the creator function of the corresponding device.
res	Resolution of the device (file or window) to plot in. See ?Devices and the creator function of the corresponding device.
...	Arguments to be passed to the method. Only accepts the following graphical parameters: ann ask bg cex.lab cex.sub cin col.axis col.lab col.main col.sub cra crt csi cxy err family fg fig font font.axis font.lab font.main font.sub lend lheight ljoin lmitre mex mfcol mfrow mfg mkh oma omd omi page pin plt pty smo srt tck tcl usr xaxp xaxs xaxt xlog xpd yaxp yaxs yaxt ylbias ylog For more information about the parameters see 'par'.

Value

A figure in popup window by default, or saved to the specified path via fileout.

See Also

EOF, ProjectField, NAO

Examples

```
# No example data is available over NAO region, so in this example we will
# tweak the longitude and latitude.
ano <- s2dv::Ano_CrossValid(map_temp$exp, map_temp$obs, memb = FALSE,
                            dat_dim = c('dat', 'member'), memb_dim = 'member')
nao <- s2dv::NAO(ano$exp, ano$obs, lat = seq(20, 80, length.out = 11),
```

```
                lon = seq(-80, 40, length.out = 16), memb_dim = "member",
                ftime_dim = "time")
nao$exp <- drop(aperm(nao$exp, c(2, 1, 3, 4)))
nao$obs <- drop(nao$obs)
VizBoxWhisker(nao$exp, nao$obs, toptitle = "NAO index",
              ytitle = "NAO index (PC1) TOS", monini = 11, freq = 1,
              yearini = 2000, expname = "SEAS5", obsname = "ERA5")
```

`VizClim` *Plots Climatologies*

Description

Plots climatologies as a function of the forecast time for any index output from `Clim()` and organized in matrix with dimensions:
c(nmod/nexp, nmemb/nparam, nltime) or c(nmod/nexp, nltime) for the experiment data
c(nobs, nmemb, nltime) or c(nobs, nltime) for the observational data

Usage

```
VizClim(
  exp_clim,
  obs_clim = NULL,
  toptitle = "",
  ytitle = "",
  monini = 1,
  freq = 12,
  limits = NULL,
  listexp = c("exp1", "exp2", "exp3"),
  listobs = c("obs1", "obs2", "obs3"),
  biglab = FALSE,
  drawleg = TRUE,
  leg = NULL,
  title_scale = 1,
  sizetit = NULL,
  fileout = NULL,
  width = 8,
  height = 5,
  size_units = "in",
  res = 100,
  ...
)
```

Arguments

`exp_clim`	Matrix containing the experimental data with dimensions: c(nmod/nexp, nmemb/nparam, nltime) or c(nmod/nexp, nltime)

obs_clim	Matrix containing the observational data (optional) with dimensions: c(nobs, nmemb, nltime) or c(nobs, nltime)
toptitle	Main title, optional.
ytitle	Title of Y-axis, optional.
monini	Starting month between 1 and 12. Default = 1.
freq	1 = yearly, 12 = monthly, 4 = seasonal, ... Default = 12.
limits	c(lower limit, upper limit): limits of the Y-axis, optional.
listexp	List of experiment names, optional.
listobs	List of observational dataset names, optional.
biglab	TRUE/FALSE for presentation/paper plot. Default = FALSE.
drawleg	TRUE/FALSE to plot the legend or not. Default = TRUE.
leg	Deprecated. Use 'drawleg' instead.
title_scale	Multiplicative factor to scale title size, optional.
sizetit	Deprecated. Use 'title_scale' instead.
fileout	Name of output file. Extensions allowed: eps/ps, jpeg, png, pdf, bmp and tiff. The default value is NULL, which the figure is shown in a pop-up window.
width	File width, in the units specified in the parameter size_units (inches by default). Takes 8 by default.
height	File height, in the units specified in the parameter size_units (inches by default). Takes 5 by default.
size_units	Units of the size of the device (file or window) to plot in. Inches ('in') by default. See ?Devices and the creator function of the corresponding device.
res	Resolution of the device (file or window) to plot in. See ?Devices and the creator function of the corresponding device.
...	Arguments to be passed to the method. Only accepts the following graphical parameters: adj ann ask bg bty cex.sub cin col.axis col.lab col.main col.sub cra crt csi cxy err family fg fig font font.axis font.lab font.main font.sub lend lheight ljoin lmitre mar mex mfcol mfrow mfg mkh oma omd omi page pch plt smo srt tck usr xaxp xaxs xaxt xlog xpd yaxp yaxs yaxt ylbias ylog For more information about the parameters see 'par'.

Value

A figure in popup window by default, or saved to the specified path. via `fileout`.

Examples

```
clim <- s2dv::Clim(ts_temp$exp, ts_temp$obs, time_dim = "sdate",
                   dat_dim = c("dat", "member"))
dim(clim$clim_exp) <- dim(clim$clim_exp)[-3]
dim(clim$clim_obs) <- dim(clim$clim_obs)[-2]
VizClim(clim$clim_exp, clim$clim_obs, toptitle = 'Climatologies',
        ytitle = 'K', monini = 11, listexp = 'SEAS5',
        listobs = 'ERA5', biglab = FALSE, fileout = NULL)
```

VizCombinedMap *Plot Multiple Lon-Lat Variables In a Single Map According to a Decision Function*

Description

Plot a number a two dimensional matrices with (longitude, latitude) dimensions on a single map with the cylindrical equidistant latitude and longitude projection.

Usage

```
VizCombinedMap(
  maps,
  lon,
  lat,
  map_select_fun,
  display_range,
  map_dim = "map",
  brks = NULL,
  cols = NULL,
  bar_limits = NULL,
  triangle_ends = c(FALSE, FALSE),
  col_inf = NULL,
  col_sup = NULL,
  col_unknown_map = "white",
  mask = NULL,
  mask_color = "grey",
  col_mask = NULL,
  dots = NULL,
  bar_titles = NULL,
  bar_label_scale = 1,
  legend_scale = NULL,
  cex_bar_titles = 1.5,
  margin_scale = NULL,
  plot_margin = NULL,
  bar_extra_margin = c(2, 0, 2, 0),
  fileout = NULL,
  width = 8,
  height = 5,
  size_units = "in",
  res = 100,
  drawleg = TRUE,
  return_leg = FALSE,
  ...
)
```

Arguments

maps	List of matrices to plot, each with (longitude, latitude) dimensions, or 3-dimensional array with the dimensions (longitude, latitude, map). Dimension names are required.
lon	Vector of longitudes. Must match the length of the corresponding dimension in 'maps'.
lat	Vector of latitudes. Must match the length of the corresponding dimension in 'maps'.
map_select_fun	Function that selects, for each grid point, which value to take among all the provided maps. This function receives as input a vector of values for a same grid point for all the provided maps, and must return a single selected value (not its index!) or NA. For example, the min and max functions are accepted.
display_range	Range of values to be displayed for all the maps. This must be a numeric vector c(range min, range max). The values in the parameter 'maps' can go beyond the limits specified in this range. If the selected value for a given grid point (according to 'map_select_fun') falls outside the range, it will be coloured with 'col_unknown_map'.
map_dim	Optional name for the dimension of 'maps' along which the multiple maps are arranged. Only applies when 'maps' is provided as a 3-dimensional array. Takes the value 'map' by default.
brks	Colour levels to be sent to VizEquiMap. This parameter is optional and adjusted automatically by the function.
cols	List of vectors of colours to be sent to VizEquiMap for the colour bar of each map. This parameter is optional and adjusted automatically by the function (up to 5 maps). The colours provided for each colour bar will be automatically interpolated to match the number of breaks. Each item in this list can be named, and the name will be used as title for the corresponding colour bar (equivalent to the parameter 'bar_titles').
bar_limits	A numeric vector of 2 indicating the range of color bar. The default is NULL, and the function will decide the range automatically.
triangle_ends	A logical vector of two indicating if the lower and upper triangles of the color bar should be plotted. The default is c(FALSE, FALSE).
col_inf	A character string of recognized color name or code indicating the color of the lower triangle of the color bar. The default is NULL.
col_sup	A character string of recognized color name or code indicating the color of the upper triangle of the color bar. The default is NULL.
col_unknown_map	Colour to use to paint the grid cells for which a map is not possible to be chosen according to 'map_select_fun' or for those values that go beyond 'display_range'. Takes the value 'white' by default.
mask	Optional numeric array with dimensions (latitude, longitude), with values in the range [0, 1], indicating the opacity of the mask over each grid point. Cells with a 0 will result in a totally opaque superimposed pixel coloured in 'mask_color', whereas cells with a 1 will have no mask and remain totally visible.

mask_color	Colour to be used for the superimposed mask (if specified in 'mask'). Takes the value 'grey' by default.
col_mask	Deprecated. Use 'mask_color' instead.
dots	Array of same dimensions as 'var' or with dimensions c(n, dim(var)), where n is the number of dot/symbol layers to add to the plot. A value of TRUE at a grid cell will draw a dot/symbol on the corresponding square of the plot. By default all layers provided in 'dots' are plotted with dots, but a symbol can be specified for each of the layers via the parameter 'dot_symbol'.
bar_titles	Optional vector of character strings providing the titles to be shown on top of each of the colour bars.
bar_label_scale	Scale factor for the size of the colour bar labels. Takes 1 by default.
legend_scale	Deprecated. Use 'bar_label_scale' instead.
cex_bar_titles	Scale factor for the sizes of the bar titles. Takes 1.5 by default.
margin_scale	Numeric vector of length 4 for the margin sizes in the following order: bottom, left, top, and right. If not specified (NULL), the default of par("mar"), c(5.1, 4.1, 4.1, 2.1), is used. Default is NULL.
plot_margin	Deprecated. Use 'margin_scale' instead.
bar_extra_margin	A numeric vector of 4 indicating the extra margins to be added around the color bar, in the format c(y1, x1, y2, x2). The units are margin lines. The default values are c(2, 0, 2, 0).
fileout	File where to save the plot. If not specified (default) a graphics device will pop up. Extensions allowed: eps/ps, jpeg, png, pdf, bmp and tiff
width	File width, in the units specified in the parameter 'size_units' (inches by default). Takes 8 by default.
height	File height, in the units specified in the parameter 'size_units' (inches by default). Takes 5 by default.
size_units	Units of the size of the device (file or window) to plot in. Inches ('in') by default. See ?Devices and the creator function of the corresponding device.
res	Resolution of the device (file or window) to plot in. See ?Devices and the creator function of the corresponding device.
drawleg	Where to draw the common colour bar. Can take values TRUE, FALSE or: 'up', 'u', 'U', 'top', 't', 'T', 'north', 'n', 'N' 'down', 'd', 'D', 'bottom', 'b', 'B', 'south', 's', 'S' (default) 'right', 'r', 'R', 'east', 'e', 'E' 'left', 'l', 'L', 'west', 'w', 'W'
return_leg	A logical value indicating if the color bars information should be returned by the function. If TRUE, the function doesn't plot the color bars but still creates the layout with color bar areas, and the arguments for GradientCatsColorBar() or ColorBarContinuous() will be returned. It is convenient for users to adjust the color bars manually. The default is FALSE, the color bars will be plotted directly.
...	Additional parameters to be passed on to VizEquiMap.

Value

Invisibly returns NULL when return_leg = FALSE (default). When return_leg = TRUE, returns a list with color bar configuration information.

Author(s)

Nicolau Manubens, <nicolau.manubens@bsc.es>

Veronica Torralba, <veronica.torralba@bsc.es>

See Also

VizCombinedMap and VizEquiMap

Examples

```
# Simple example
x <- array(1:(20 * 10), dim = c(lat = 10, lon = 20)) / 200
a <- x * 0.6
b <- (1 - x) * 0.6
c <- 1 - (a + b)
lons <- seq(0, 359.5, length = 20)
lats <- seq(-89.5, 89.5, length = 10)
VizCombinedMap(list(a, b, c), lons, lats,
               toptitle = 'Maximum map',
               map_select_fun = max,
               display_range = c(0, 1),
               bar_titles = paste('% of belonging to', c('a', 'b', 'c')),
               drawleg = FALSE, brks = 20, width = 8, height = 5)

# Mask example
lons2 <- c(0:40)
lats2 <- 51:26
data <- rnorm(41 * 26 * 3)
dim(data) <- c(map = 3, lon = 41, lat = 26)
mask <-  sample(c(0,1), replace = TRUE, size = 41 * 26)
dim(mask) <- c(lat = 26, lon = 41)
VizCombinedMap(data, lon = lons2, lat = lats2, map_select_fun = max,
               display_range = range(data), mask = mask,
               drawleg = FALSE, width = 14, height = 10)
```

VizEquiMap *Maps A Two-Dimensional Variable On A Cylindrical Equidistant Projection*

Description

Map longitude-latitude array (on a regular rectangular or gaussian grid) on a cylindrical equidistant latitude and longitude projection with coloured grid cells. Only the region for which data has been provided is displayed. A colour bar (legend) can be plotted and adjusted. It is possible to draw superimposed arrows, dots, symbols, contour lines and boxes. A number of options is provided to adjust the position, size and colour of the components. Some parameters are provided to add and adjust the masks that include continents, oceans, and lakes. This plot function is compatible with figure layouts if colour bar is disabled.

Usage

```
VizEquiMap(
  data,
  lon,
  lat,
  varu = NULL,
  varv = NULL,
  toptitle = NULL,
  sizetit = NULL,
  caption = NULL,
  units = NULL,
  brks = NULL,
  cols = NULL,
  bar_limits = NULL,
  triangle_ends = NULL,
  col_inf = NULL,
  col_sup = NULL,
  colNA = NULL,
  color_fun = ClimPalette(),
  square = TRUE,
  filled.continents = FALSE,
  filled.oceans = FALSE,
  country.borders = FALSE,
  coast_color = NULL,
  coast_width = 1,
  lake_color = NULL,
  shapefile = NULL,
  shapefile_color = NULL,
  shapefile_lwd = 1,
  contours = NULL,
  brks2 = NULL,
  contour_lwd = 0.5,
  contour_color = "black",
  contour_lty = 1,
  contour_draw_label = TRUE,
  contour_label_scale = 1,
  dots = NULL,
  dot_symbol = 4,
```

```
  dot_size = 1,
  mask = NULL,
  mask_color = "white",
  arr_subsamp = floor(length(lon)/30),
  arr_scale = 1,
  arr_ref_len = 15,
  arr_units = "m/s",
  arr_scale_shaft = 1,
  arr_scale_shaft_angle = 1,
  axelab = TRUE,
  labW = FALSE,
  lab_dist_x = NULL,
  lab_dist_y = NULL,
  degree_sym = FALSE,
  intylat = 20,
  intxlon = 20,
  xlonshft = 0,
  ylatshft = 0,
  xlabels = NULL,
  ylabels = NULL,
  axes_tick_scale = 1,
  axes_label_scale = 1,
  drawleg = TRUE,
  vertical = FALSE,
  subsampleg = NULL,
  bar_extra_labels = NULL,
  draw_bar_ticks = TRUE,
  draw_separators = FALSE,
  triangle_ends_scale = 1,
  bar_label_digits = 4,
  bar_label_scale = 1,
  units_scale = 1,
  bar_tick_scale = 1,
  bar_extra_margin = rep(0, 4),
  include_lower_boundary = TRUE,
  include_upper_boundary = TRUE,
  hatching_mask = NULL,
  hatching_density = 10,
  hatching_angle = 45,
  hatching_color = "#252525",
  hatching_lwd = 0.5,
  hatching_cross = FALSE,
  boxlim = NULL,
  boxcol = "purple2",
  boxlwd = 5,
  margin_scale = rep(1, 4),
  title_scale = 1,
  caption_size = 0.8,
```

```
  caption_height = 1,
  numbfig = NULL,
  fileout = NULL,
  width = 8,
  height = 5,
  size_units = "in",
  res = 100,
  var = NULL,
  ...
)
```

Arguments

data	Array with the values at each cell of a grid on a regular rectangular or gaussian grid. The array is expected to have two dimensions: c(latitude, longitude). Longitudes can be in ascending or descending order and latitudes in any order. It can contain NA values (coloured with 'colNA'). Arrays with dimensions c(longitude, latitude) will also be accepted but 'lon' and 'lat' will be used to disambiguate so this alternative is not appropriate for square arrays. It is allowed that the positions of the longitudinal and latitudinal coordinate dimensions are interchanged.
lon	Numeric vector of longitude locations of the cell centers of the grid of 'data', in ascending or descending order (same as 'data'). Expected to be regularly spaced, within either of the ranges [-180, 180] or [0, 360]. Data for two adjacent regions split by the limits of the longitude range can also be provided, e.g. `lon = c(0:50, 300:360)` ('data' must be provided consitently).
lat	Numeric vector of latitude locations of the cell centers of the grid of 'data', in any order (same as 'data'). Expected to be from a regular rectangular or gaussian grid, within the range [-90, 90].
varu	Array of the zonal component of wind/current/other field with the same dimensions as 'data'. It is allowed that the positions of the longitudinal and latitudinal coordinate dimensions are interchanged.
varv	Array of the meridional component of wind/current/other field with the same dimensions as 'data'. It is allowed that the positions of the longitudinal and latitudinal coordinate dimensions are interchanged.
toptitle	Top title of the figure, scalable with parameter 'title_scale'.
sizetit	Scale factor for the figure top title provided in parameter 'toptitle'. Deprecated. Use 'title_scale' instead.
caption	A character string of the caption located at the left-bottom of the plot. Captions with multiple lines can be constructed using string manipulation functions like `paste()` or `paste0()`, using `"\n"` to indicate line breaks.
units	Title at the top of the colour bar, most commonly the units of the variable provided in parameter 'data'.
brks, cols, bar_limits, triangle_ends	Usually only providing 'brks' is enough to generate the desired colour bar. These parameters allow to define n breaks that define n - 1 intervals to classify each of the values in 'data'. The corresponding grid cell of a given value in

'data' will be coloured in function of the interval it belongs to. These parameters are sent to `ColorBar()` to generate the breaks and colours. Additional colours for values beyond the limits of the colour bar are also generated and applied to the plot if 'bar_limits' or 'brks' and 'triangle_ends' are properly provided to do so. See ?ColorBar for a full explanation.

`col_inf`, `col_sup`, `colNA`
: Colour identifiers to colour the values in 'data' that go beyond the extremes of the colour bar and to colour NA values, respectively. 'colNA' takes attr(cols, 'na_color') if available by default, where cols is the parameter 'cols' if provided or the vector of colors returned by 'color_fun'. If not available, it takes 'pink' by default. 'col_inf' and 'col_sup' will take the value of 'colNA' if not specified. See ?ColorBar for a full explanation on 'col_inf' and 'col_sup'.

`color_fun`, `subsampleg`, `bar_extra_labels`, `draw_bar_ticks`
: Set of parameters to control the visual aspect of the drawn colour bar (1/3). See ?ColorBar for a full explanation.

`square`
: Logical value to choose either to draw a coloured square for each grid cell in 'data' (TRUE; default) or to draw contour lines and fill the spaces in between with colours (FALSE). In the latter case, 'filled.continents' will take the value FALSE if not specified.

`filled.continents`
: Colour to fill in drawn projected continents. If 'square = FALSE', it is set as FALSE. If set to FALSE (default), the continents are not filled.

`filled.oceans`
: A logical value or the color name to fill in drawn projected oceans. The default value is FALSE. If it is TRUE, the default colour is "light blue".

`country.borders`
: A logical value indicating if the country borders should be plotted (TRUE) or not (FALSE). It only works when 'filled.continents' is FALSE. The default value is FALSE.

`coast_color`
: Colour of the coast line of the drawn projected continents. Takes the value gray(0.5) by default.

`coast_width`
: Line width of the coast line of the drawn projected continents. Takes the value 1 by default.

`lake_color`
: Colour of the lake or other water body inside continents. The default value is NULL.

`shapefile`
: A character string of the path to a .rds file or a list object containinig shape file data. If it is a .rds file, it should contain a list. The list should contains 'x' and 'y' at least, which indicate the location of the shape. The default value is NULL.

`shapefile_color`
: Line color of the shapefile.

`shapefile_lwd`
: Line width of the shapefile. The default value is 1.

`contours`
: Array of same dimensions as 'data' to be added to the plot and displayed with contours. Parameter 'brks2' is required to define the magnitude breaks for each contour curve. Disregarded if 'square = FALSE'. It is allowed that the positions of the longitudinal and latitudinal coordinate dimensions are interchanged.

`brks2` Vector of magnitude breaks where to draw contour curves for the array provided in 'contours' or if 'square = FALSE'.

`contour_lwd` Line width of the contour curves provided via 'contours' and 'brks2', or if 'square = FALSE'.

`contour_color` Line color of the contour curves provided via 'contours' and 'brks2', or if 'square = FALSE'.

`contour_lty` Line type of the contour curves. Takes 1 (solid) by default. See help on 'lty' in par() for other accepted values.

`contour_draw_label`
A logical value indicating whether to draw the contour labels or not. The default value is TRUE.

`contour_label_scale`
Scale factor for the superimposed labels when drawing contour levels.

`dots` Array of same dimensions as 'data' or with dimensions c(n, dim(data)), where n is the number of dot/symbol layers to add to the plot. A value of TRUE at a grid cell will draw a dot/symbol on the corresponding square of the plot. By default all layers provided in 'dots' are plotted with dots, but a symbol can be specified for each of the layers via the parameter 'dot_symbol'. It is allowed that the positions of the longitudinal and latitudinal coordinate dimensions are interchanged.

`dot_symbol` Single character/number or vector of characters/numbers that correspond to each of the symbol layers specified in parameter 'dots'. If a single value is specified, it will be applied to all the layers in 'dots'. Takes 4 (cross) by default. See 'pch' in par() for additional accepted options.

`dot_size` Scale factor for the dots/symbols to be plotted, specified in 'dots'. If a single value is specified, it will be applied to all layers in 'dots'. Takes 1 by default.

`mask` An array with the same dimensions as 'data' with values in the range of [0, 1] or logical, indicating the opacity of the mask over each grid point. Cells with a 0 or FALSE will result in a totally opaque superimposed pixel coloured in 'mask_color', whereas cells with a 1 or TRUE will have no mask and remain totally visible. If the array is numeric, values between 0 and 1 will have shades of transparency.

`mask_color` Color of the mask. The default value is 'white'.

`arr_subsamp` Subsampling factor to select a subset of arrows in 'varu' and 'varv' to be drawn. Only one out of arr_subsamp arrows will be drawn. Takes 1 by default.

`arr_scale` Scale factor for drawn arrows from 'varu' and 'varv'. Takes 1 by default.

`arr_ref_len` Length of the refence arrow to be drawn as legend at the bottom of the figure (in same units as 'varu' and 'varv', only affects the legend for the wind or variable in these arrays). Defaults to 15.

`arr_units` Units of 'varu' and 'varv', to be drawn in the legend. Takes 'm/s' by default.

`arr_scale_shaft`
Parameter for the scale of the shaft of the arrows (which also depend on the number of figures and the arr_scale parameter). Defaults to 1.

arr_scale_shaft_angle
: Parameter for the scale of the angle of the shaft of the arrows (which also depend on the number of figure and the arr_scale parameter). Defaults to 1.

axelab
: Whether to draw longitude and latitude axes or not. TRUE by default.

labW
: Whether to label the longitude axis with a 'W' instead of minus for negative values. Defaults to FALSE.

lab_dist_x
: A numeric of the distance of the longitude labels to the box borders. The default value is NULL and is automatically adjusted by the function.

lab_dist_y
: A numeric of the distance of the latitude labels to the box borders. The default value is NULL and is automatically adjusted by the function.

degree_sym
: A logical indicating whether to include degree symbol (30° N) or not (30N; default).

intylat
: Interval between latitude ticks on y-axis, in degrees. Defaults to 20.

intxlon
: Interval between latitude ticks on x-axis, in degrees. Defaults to 20.

xlonshft
: A numeric of the degrees to shift the latitude ticks. The default value is 0.

ylatshft
: A numeric of the degrees to shift the longitude ticks. The default value is 0.

xlabels
: A vector of character string of the custumized x-axis labels. The values should correspond to each tick, which is decided by the longitude and parameter 'intxlon'. The default value is NULL and the labels will be automatically generated.

ylabels
: A vector of character string of the custumized y-axis labels. The values should correspond to each tick, which is decided by the latitude and parameter 'intylat'. The default value is NULL and the labels will be automatically generated.

axes_tick_scale
: Scale factor for the tick lines along the longitude and latitude axes.

axes_label_scale
: Scale factor for the labels along the longitude and latitude axes.

drawleg
: Whether to plot a color bar (legend, key) or not. Defaults to TRUE. It is not possible to plot the colour bar if 'add = TRUE'. Use ColorBar() and the return values of PlotEquiMap() instead.

vertical
: TRUE/FALSE for vertical/horizontal colour bar. Default is FALSE. Parameters 'width' and 'height' might need to be modified to accommodate the vertical colour bar.

draw_separators, triangle_ends_scale, bar_label_digits
: Set of parameters to control the visual aspect of the drawn colour bar (2/3). See ?ColorBar for a full explanation.

bar_label_scale, units_scale, bar_tick_scale, bar_extra_margin
: Set of parameters to control the visual aspect of the drawn colour bar (3/3). See ?ColorBar for a full explanation.

include_lower_boundary
: Logical value indicating whether to include the minimum value of the field. Takes TRUE by default.

include_upper_boundary
: Logical value indicating whether to include the maximum value of the field. Takes TRUE by default.

`hatching_mask` Logical or binary (0/1) array with two named dimensions: c(latitude, longitude). Hatching is applied to grid cells where 'hatching_mask' is TRUE (or 1). Arrays with dimensions c(longitude, latitude) are also accepted, but the resulting hatching may appear transposed. To ensure correct alignment with the map, provide 'data'. The function will compare the dimension order of 'hatching_mask' and 'data', and automatically transpose 'hatching_mask' if the latitude and longitude dimensions appear to be reversed.

`hatching_density` The density of shading lines, in lines per inch. A zero value of density means no shading nor filling, whereas negative values and NA suppress shading (and so allow color filling). NULL means that no shading lines are drawn. Default is 10.

`hatching_angle` The slope of shading lines, given as an angle in degrees (counter-clockwise). Default is 45.

`hatching_color` Color of the hatching lines. Default is `"#252525"`.

`hatching_lwd` The line width, a positive number. The interpretation is device-specific, and some devices do not implement line widths less than one. Default is 0.5.

`hatching_cross` A logical value indicating crosshatching. If TRUE, adds a second set of lines in the opposite angle. Default is FALSE.

`boxlim` Limits of a box to be added to the plot, in degrees: c(x1, y1, x2, y2). A list with multiple box specifications can also be provided.

`boxcol` Colour of the box lines. A vector with a colour for each of the boxes is also accepted. Defaults to 'purple2'.

`boxlwd` Line width of the box lines. A vector with a line width for each of the boxes is also accepted. Defaults to 5.

`margin_scale` Scale factor for the margins around the map plot, with the format c(y1, x1, y2, x2). Defaults to rep(1, 4). If drawleg = TRUE, then margin_scale[1] is subtracted 1 unit.

`title_scale` Scale factor for the figure top title. Defaults to 1.

`caption_size` Scale factor for the figure caption. Default is 0.8 (1 if vertical = TRUE).

`caption_height` Numeric value controlling the vertical position of the caption in margin lines. Positive values move it downwards, negative values move it upwards. Default is 1.

`numbfig` Number of figures in the layout the plot will be put into. A higher numbfig will result in narrower margins and smaller labels, axe labels, ticks, thinner lines, ... Defaults to 1.

`fileout` File where to save the plot. If not specified (default) a graphics device will pop up. Extensions allowed: eps/ps, jpeg, png, pdf, bmp and tiff.

`width` File width, in the units specified in the parameter 'size_units' (inches by default). Takes 8 by default.

`height` File height, in the units specified in the parameter 'size_units' (inches by default). Takes 5 by default.

`size_units` Units of the size of the device (file or window) to plot in. Inches ('in') by default. See ?Devices and the creator function of the corresponding device.

res	Resolution of the device (file or window) to plot in. See ?Devices and the creator function of the corresponding device.
var	Deprecated. Use 'data' instead.
...	Arguments to be passed to the method. Only accepts the following graphical parameters: adj ann ask bg bty cex.sub cin col.axis col.lab col.main col.sub cra crt csi cxy err family fg font font.axis font.lab font.main font.sub lend lheight ljoin lmitre mex mfcol mfrow mfg mkh omd omi page pch pin plt pty smo srt tcl usr xaxp xaxs xaxt xlog xpd yaxp yaxs yaxt ylbias ylog For more information about the parameters see 'par'.

Value

brks	Breaks used for colouring the map (and legend if drawleg = TRUE).
cols	Colours used for colouring the map (and legend if drawleg = TRUE). Always of length length(brks) - 1.
col_inf	Colour used to draw the lower triangle end in the colour bar (NULL if not drawn at all).
col_sup	Colour used to draw the upper triangle end in the colour bar (NULL if not drawn at all).

Examples

```
ano <- s2dv::Ano_CrossValid(map_temp$exp, map_temp$obs, memb = FALSE,
                            dat_dim = c('dat', 'member'), memb_dim = 'member')
data <- s2dv::MeanDims(ano$exp, "member")
lats <- attr(map_temp$exp, "Variables")$common$lat
lons <- attr(map_temp$exp, "Variables")$common$lon

VizEquiMap(data[1, 1, 1, 1, , ], lon = lons, lat = lats,
          toptitle = 'Near-surface temperature anomaly, Nov. 2000',
          filled.continents = FALSE, title_scale = 0.7,
          caption = paste0("This is a test caption."))
```

VizForecastPDF *Plot one or multiple ensemble forecast pdfs for the same event*

Description

This function plots the probability distribution function of several ensemble forecasts. Separate panels are used to plot forecasts valid or initialized at different times or by different models or even at different locations. Probabilities for tercile categories are computed, plotted in colors and annotated. An asterisk marks the tercile with higher probabilities. Probabilities for extreme categories (above P90 and below P10) can also be included as hatched areas. Individual ensemble members can be plotted as jittered points. The observed value is optionally shown as a diamond.

Usage

```
VizForecastPDF(
  fcst,
  tercile.limits,
  extreme.limits = NULL,
  obs = NULL,
  toptitle = "Set a title",
  title = NULL,
  var.name = "Varname (units)",
  title.legend = "Probability of terciles",
  strings.legend = c("Below normal", "Near normal", "Above normal"),
  ensemble.legend = "Ensemble members",
  obs.title = "Observations",
  strings.obs = NULL,
  title.extremes = "Probability of extremes",
  strings.extremes = c("Below p10", "Above p90"),
  xlab.title = "Probability density",
  fcst.names = NULL,
  title.cex = 1,
  labs.cex = 1,
  fcst.names.cex = 1,
  obs.lines = TRUE,
  obs.size = 3,
  add.ensmemb = c("above", "below", "no"),
  color.set = c("ggplot", "s2s4e", "hydro", "vitigeoss"),
  memb_dim = "member",
  width = 6,
  height = 5,
  fileout = NULL,
  plotfile = NULL,
  res = 300
)
```

Arguments

`fcst`	A dataframe or array containing all the ensember members for each forecast. If `'fcst'` is an array, it should have two labelled dimensions, and one of them should be `'members'`. If `'fcsts'` is a data.frame, each column shoul be a separate forecast, with the rows beeing the different ensemble members.
`tercile.limits`	An array or vector with P33 and P66 values that define the tercile categories for each panel. Use an array of dimensions (nforecasts,2) to define different terciles for each forecast panel, or a vector with two elements to reuse the same tercile limits for all forecast panels.
`extreme.limits`	(optional) An array or vector with P10 and P90 values that define the extreme categories for each panel. Use an array of (nforecasts,2) to define different extreme limits for each forecast panel, or a vector with two elements to reuse the same tercile limits for all forecast panels. (Default: extreme categories are not shown).

obs	(optional) A vector providing the observed values for each forecast panel or a single value that will be reused for all forecast panels. (Default: observation is not shown).
toptitle	A string with the plot main title.
title	Deprecated. Use 'toptitle' instead.
var.name	A string with the variable name and units.
title.legend	A string with the title of the legend showing the probabilistic categories. By default, it is set to "Probability of terciles".
strings.legend	A vector of strings with the name of the probabilistic categories. By default, it is set to c("Below normal", "Near normal", "Above normal").
ensemble.legend	A string with the title of the legend showing the ensemble members. By default, it is set to "Ensemble members".
obs.title	A string with the title of the legend showing the observations ('obs'). By default, it is set to "Observations".
strings.obs	A vector of strings with the name of the observations, which correspond to the values of 'obs'. By default, it is set to NULL.
title.extremes	A string with the title of the legend showing the extreme categories. By default, it is set to "Probability of extremes".
strings.extremes	A vector of strings with the name of the extreme categories. By default, it is set to c("Below p10", "Above p90").
xlab.title	A string with the title of the x-axis. By default, it is set to "Probability density".
fcst.names	(optional) A string or vector of strings with the titles of each individual forecast panel.
title.cex	A number with the size of 'toptitle'. By default, it is set to 1.
labs.cex	A number with the size of the labels in the axes. By default, it is set to 1.
fcst.names.cex	A number with the size of the 'fcst.names'. By default, it is set to 1.
obs.lines	A logical value indicating whether to draw horizontal lines at the height of the observations. The default value is TRUE.
obs.size	A number with the size of the observation dots. By default, it is set to 3.
add.ensmemb	Either to add the ensemble members 'above' (default) or 'below' the pdf, or not ('no').
color.set	A selection of predefined color sets: use 'ggplot' (default) for blue/green/red, 's2s4e' for blue/grey/orange, 'hydro' for yellow/gray/blue (suitable for precipitation and inflows) or the "vitigeoss" color set.
memb_dim	A character string indicating the name of the member dimension.
width	File width, in the units specified in the parameter size_units (inches by default). Takes 8 by default.
height	File height, in the units specified in the parameter size_units (inches by default). Takes 5 by default.

fileout	(optional) A filename (pdf, png...) where the plot will be saved. (Default: the plot is not saved).
plotfile	Deprecated. Use 'fileout' instead.
res	Resolution of the device (file or window) to plot in. See ?Devices and the creator function of the corresponding device.

Value

A ggplot object containing the plot.

Author(s)

Llorenç Lledó <llledo@bsc.es>

Examples

```
fcsts <- data.frame(fcst1 = rnorm(10), fcst2 = rnorm(10, 0.5, 1.2))
VizForecastPDF(fcsts,c(-1,1))
```

VizLayout *Arrange and Fill Multi-Pannel Layouts With Optional Colour Bar*

Description

This function takes an array or list of arrays and loops over each of them to plot all the sub-arrays they contain on an automatically generated multi-pannel layout. A different plot function (not necessarily from s2dv) can be applied over each of the provided arrays. The input dimensions of each of the functions have to be specified, either with the names or the indices of the corresponding input dimensions. It is possible to draw a common colour bar at any of the sides of the multi-pannel for all the s2dv plots that use a colour bar. Common plotting arguments for all the arrays in 'data' can be specified via the '...' parameter, and specific plotting arguments for each array can be fully adjusted via 'special_args'. It is possible to draw titles for each of the figures, layout rows, layout columns and for the whole figure. A number of parameters is provided in order to adjust the position, size and colour of the components. Blank cells can be forced to appear and later be filled in manually with customized plots.
This function pops up a blank new device and fills it in, so it cannot be nested in complex layouts.

Usage

```
VizLayout(
  fun,
  plot_dims,
  data,
  ...,
  special_args = NULL,
  nrow = NULL,
```

```
  ncol = NULL,
  toptitle = NULL,
  row_titles = NULL,
  col_titles = NULL,
  bar_scale = 1,
  title_scale = 1,
  title_margin_scale = 1,
  title_left_shift_scale = 1,
  subtitle_scale = 1,
  subtitle_margin_scale = 1,
  subplot_titles_scale = 1,
  brks = NULL,
  cols = NULL,
  drawleg = "S",
  titles = NULL,
  subsampleg = NULL,
  bar_limits = NULL,
  triangle_ends = NULL,
  col_inf = NULL,
  col_sup = NULL,
  color_fun = ClimColors,
  draw_bar_ticks = TRUE,
  draw_separators = FALSE,
  triangle_ends_scale = 1,
  bar_extra_labels = NULL,
  units = NULL,
  units_scale = 1,
  bar_label_scale = 1,
  bar_tick_scale = 1,
  bar_extra_margin = rep(0, 4),
  bar_left_shift_scale = 1,
  bar_label_digits = 4,
  extra_margin = rep(0, 4),
  layout_by_rows = TRUE,
  fileout = NULL,
  width = NULL,
  height = NULL,
  size_units = "in",
  res = 100,
  close_device = TRUE,
  var = NULL
)
```

Arguments

fun	Plot function (or name of the function) to be called on the arrays provided in 'data'. If multiple arrays are provided in 'data', a vector of as many function names (character strings!) can be provided in 'fun', one for each array in 'data'.

`plot_dims` Numeric or character string vector with identifiers of the input plot dimensions of the plot function specified in 'fun'. If character labels are provided, names(dim(data)) or attr('dimensions', data) will be checked to locate the dimensions. As many plots as prod(dim(data)[-plot_dims]) will be generated. If multiple arrays are provided in 'data', 'plot_dims' can be sent a list with a vector of plot dimensions for each. If a single vector is provided, it will be used for all the arrays in 'data'.

`data` Multi-dimensional array with at least the dimensions expected by the specified plot function in 'fun'. The dimensions reqired by the function must be specified in 'plot_dims'. The dimensions can be disordered and will be reordered automatically. Dimensions can optionally be labelled in order to refer to them with names in 'plot_dims'. All the available plottable sub-arrays will be automatically plotted and arranged in consecutive cells of an automatically arranged layout. A list of multiple (super-)arrays can be specified. The process will be repeated for each of them, by default applying the same plot function to all of them or, if properly specified in 'fun', a different plot function will be applied to each of them. NAs can be passed to the list: a NA will yield a blank cell in the layout, which can be populated after (see .SwitchToFigure).

`...` Parameters to be sent to the plotting function 'fun'. If multiple arrays are provided in 'data' and multiple functions are provided in 'fun', the parameters provided through ... will be sent to all the plot functions, as common parameters. To specify concrete arguments for each of the plot functions see parameter 'special_args'.

`special_args` List of sub-lists, each sub-list having specific extra arguments for each of the plot functions provided in 'fun'. If you want to fix a different value for each plot in the layout you can do so by a) splitting your array into a list of sub-arrays (each with the data for one plot) and providing it as parameter 'data', b) providing a list of named sub-lists in 'special_args', where the names of each sub-list match the names of the parameters to be adjusted, and each value in a sub-list contains the value of the corresponding parameter. For example, if the plots are two maps with different arguments, the structure would be like:
data:
List of 2
$: num [1:360, 1:181] 1 3.82 5.02 6.63 8.72 ...
$: num [1:360, 1:181] 2.27 2.82 4.82 7.7 10.32 ...
special_args:
List of 2
$:List of 2
..$ arg1: ...
..$ arg2: ...
$:List of 1
..$ arg1: ...

`nrow` Numeric value to force the number of rows in the automatically generated layout. If higher than the required, this will yield blank cells in the layout (which can then be populated). If lower than the required the function will stop. By default it is configured to arrange the layout in a shape as square as possible. Blank

cells can be manually populated after with customized plots (see SwitchTofigure).

`ncol` Numeric value to force the number of columns in the automatically generated layout. If higher than the required, this will yield blank cells in the layout (which can then be populated). If lower than the required the function will stop. By default it is configured to arrange the layout in a shape as square as possible. Blank cells can be manually populated after with customized plots (see SwitchTofigure).

`toptitle` Topt title for the multi-pannel. Blank by default.

`row_titles` Character string vector with titles for each of the rows in the layout. Blank by default.

`col_titles` Character string vector with titles for each of the columns in the layout. Blank by default.

`bar_scale` Scale factor for the common colour bar. Takes 1 by default.

`title_scale` Scale factor for the multi-pannel title. Takes 1 by default.

`title_margin_scale`
Scale factor for the margins surrounding the top title. Takes 1 by default.

`title_left_shift_scale`
When plotting row titles, a shift is added to the horizontal positioning of the top title in order to center it to the region of the figures (without taking row titles into account). This shift can be reduced. A value of 0 will remove the shift completely, centering the title to the total width of the device. This parameter will be disregarded if no 'row_titles' are provided.

`subtitle_scale` Scale factor for the row titles and column titles (specified in 'row_titles' and 'col_titles'). Takes 1 by default.

`subtitle_margin_scale`
Scale factor for the margins surrounding the subtitles. Takes 1 by default.

`subplot_titles_scale`
Scale factor for the subplots top titles. Takes 1 by default.

`brks`, `cols`, `bar_limits`, `triangle_ends`
Usually only providing 'brks' is enough to generate the desired colour bar. These parameters allow to define n breaks that define n - 1 intervals to classify each of the values in 'data'. The corresponding grid cell of a given value in 'data' will be coloured in function of the interval it belongs to. These parameters are sent to `ColorBarContinuous()` to generate the breaks and colours. Additional colours for values beyond the limits of the colour bar are also generated and applied to the plot if 'bar_limits' or 'brks' and 'triangle_ends' are properly provided to do so. See ?ColorBarContinuous for a full explanation.

`drawleg` Where to draw the common colour bar. Can take values TRUE, FALSE or:
'up', 'u', 'U', 'top', 't', 'T', 'north', 'n', 'N'
'down', 'd', 'D', 'bottom', 'b', 'B', 'south', 's', 'S' (default)
'right', 'r', 'R', 'east', 'e', 'E'
'left', 'l', 'L', 'west', 'w', 'W'

`titles` Character string vector with titles for each of the figures in the multi-pannel, from top-left to bottom-right. Blank by default.

col_inf, col_sup
: Colour identifiers to colour the values in 'data' that go beyond the extremes of the colour bar and to colour NA values, respectively. 'colNA' takes 'white' by default. 'col_inf' and 'col_sup' will take the value of 'colNA' if not specified. See ?ColorBarContinuous for a full explanation on 'col_inf' and 'col_sup'.

color_fun, subsampleg, bar_extra_labels, draw_bar_ticks, draw_separators, triangle_ends_scale, bar_label_digits, bar_label_scale, units_scale, bar_tick_scale, bar_extra_margin
: Set of parameters to control the visual aspect of the drawn colour bar. See ?ColorBarContinuous for a full explanation.

units
: Title at the top of the colour bar, most commonly the units of the variable provided in parameter 'data'.

bar_left_shift_scale
: When plotting row titles, a shift is added to the horizontal positioning of the colour bar in order to center it to the region of the figures (without taking row titles into account). This shift can be reduced. A value of 0 will remove the shift completely, centering the colour bar to the total width of the device. This parameter will be disregarded if no 'row_titles' are provided.

extra_margin
: Extra margins to be added around the layout, in the format c(y1, x1, y2, x2). The units are margin lines. Takes rep(0, 4) by default.

layout_by_rows
: Logical indicating wether the panels should be filled by columns (FALSE) or by rows (TRUE, default).

fileout
: File where to save the plot. If not specified (default) a graphics device will pop up. Extensions allowed: eps/ps, jpeg, png, pdf, bmp and tiff.

width
: Width in inches of the multi-pannel. 7 by default, or 11 if 'fielout' has been specified.

height
: Height in inches of the multi-pannel. 7 by default, or 11 if 'fileout' has been specified.

size_units
: Units of the size of the device (file or window) to plot in. Inches ('in') by default. See ?Devices and the creator function of the corresponding device.

res
: Resolution of the device (file or window) to plot in. See ?Devices and the creator function of the corresponding device.

close_device
: Whether to close the graphics device after plotting the layout and a 'fileout' has been specified. This is useful to avoid closing the device when saving the layout into a file and willing to add extra elements or figures. Takes TRUE by default. Disregarded if no 'fileout' has been specified.

var
: Deprecated. Use 'data' instead.

Value

brks
: Breaks used for colouring the map (and legend if drawleg = TRUE).

cols
: Colours used for colouring the map (and legend if drawleg = TRUE). Always of length length(brks) - 1.

col_inf
: Colour used to draw the lower triangle end in the colour bar (NULL if not drawn at all).

col_sup	Colour used to draw the upper triangle end in the colour bar (NULL if not drawn at all).
layout_matrix	Underlying matrix of the layout. Useful to later set any of the layout cells as current figure to add plot elements. See .SwitchToFigure.

Examples

```
ano <- s2dv::Ano_CrossValid(map_temp$exp, map_temp$obs, memb = FALSE,
                            dat_dim = c('dat', 'member'), memb_dim = 'member')
data <- s2dv::MeanDims(ano$exp, "member")
lats <- attr(map_temp$exp, "Variables")$common$lat
lons <- attr(map_temp$exp, "Variables")$common$lon

VizLayout(fun = VizEquiMap, plot_dims = c('lat', 'lon'),
          data = data[, 1, 1, 1, , ], lon = lons, lat = lats,
          filled.continents = FALSE,
          toptitle = 'Near-surface temperature Nov.',
          titles = paste(2000:2005))
```

VizMatrix *Function to convert any numerical table to a grid of coloured squares.*

Description

This function converts a numerical data matrix into a coloured grid. It is useful for a slide or article to present tabular results as colors instead of numbers.

Usage

```
VizMatrix(
  data,
  brks = NULL,
  cols = NULL,
  toptitle = NULL,
  title.color = "royalblue4",
  xtitle = NULL,
  ytitle = NULL,
  xlabels = NULL,
  xvert = FALSE,
  ylabels = NULL,
  line = 3,
  figure.width = 1,
  drawleg = TRUE,
  legend = NULL,
  legend.width = 0.15,
  xlab_dist = NULL,
  ylab_dist = NULL,
```

```
  fileout = NULL,
  size_units = "px",
  res = 100,
  var = NULL,
  ...
)
```

Arguments

data	A numerical matrix containing the values to be displayed in a colored image.
brks	A vector of the color bar intervals. The length must be one more than the parameter 'cols'. Use ColorBarContinuous() to generate default values.
cols	A vector of valid color identifiers for color bar. The length must be one less than the parameter 'brks'. Use ColorBarContinuous() to generate default values.
toptitle	A string of the title of the grid. Set NULL as default.
title.color	A string of valid color identifier to decide the title color. Set "royalblue4" as default.
xtitle	A string of title of the x-axis. Set NULL as default.
ytitle	A string of title of the y-axis. Set NULL as default.
xlabels	A vector of labels of the x-axis. The length must be length of the column of parameter 'data'. Set the sequence from 1 to the length of the column of parameter 'data' as default.
xvert	A logical value to decide whether to place x-axis labels vertically. Set FALSE as default, which keeps the labels horizontally.
ylabels	A vector of labels of the y-axis The length must be length of the row of parameter 'data'. Set the sequence from 1 to the length of the row of parameter 'data' as default.
line	An integer specifying the distance between the title of the x-axis and the x-axis. Set 3 as default. Adjust if the x-axis labels are long.
figure.width	A positive number as a ratio adjusting the width of the grids. Set 1 as default.
drawleg	A logical value to decide to draw the grid color legend or not. Set TRUE as default.
legend	Deprecated. Use 'drawleg' instead.
legend.width	A number between 0 and 0.5 to adjust the legend width. Set 0.15 as default.
xlab_dist	A number specifying the distance between the x labels and the x axis. If not specified, it equals to -1 - (nrow(data) / 10 - 1).
ylab_dist	A number specifying the distance between the y labels and the y axis. If not specified, it equals to 0.5 - ncol(data) / 10.
fileout	A string of full directory path and file name indicating where to save the plot. If not specified (default), a graphics device will pop up.
size_units	A string indicating the units of the size of the device (file or window) to plot in. Set 'px' as default. See ?Devices and the creator function of the corresponding device.

res	A positive number indicating resolution of the device (file or window) to plot in. See ?Devices and the creator function of the corresponding device.
var	Deprecated. Use 'data' instead.
...	The additional parameters to be passed to function ColorBarContinuous() in s2dv for color legend creation.

Value

A figure in popup window by default, or saved to the specified path. via `fileout`.

Examples

```
# Example with random data
VizMatrix(data = matrix(rnorm(n = 120, mean = 0.3), 10, 12),
          cols = c('white','#fef0d9','#fdd49e','#fdbb84','#fc8d59',
                   '#e34a33','#b30000', '#7f0000'),
          brks = c(-1, 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 1),
          toptitle = "Mean Absolute Error",
          xtitle = "Forecast time (month)", ytitle = "Start date",
          xlabels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul",
                      "Aug", "Sep", "Oct", "Nov", "Dec"))
```

VizMostLikelyQuantileMap

Plot Maps of Most Likely Quantiles

Description

This function receives as main input (via the parameter `probs`) a collection of longitude-latitude maps, each containing the probabilities (from 0 to 1) of the different grid cells of belonging to a category. As many categories as maps provided as inputs are understood to exist. The maps of probabilities must be provided on a common rectangular regular grid, and a vector with the longitudes and a vector with the latitudes of the grid must be provided. The input maps can be provided in two forms, either as a list of multiple two-dimensional arrays (one for each category) or as a three-dimensional array, where one of the dimensions corresponds to the different categories.

Usage

```
VizMostLikelyQuantileMap(
  probs,
  lon,
  lat,
  cat_dim = "bin",
  bar_titles = NULL,
  col_unknown_cat = "white",
  drawleg = TRUE,
```

```
  margin_scale = NULL,
  bar_extra_margin = c(1.5, 0, 1.5, 0),
  ...
)
```

Arguments

probs	A list of bi-dimensional arrays with the named dimensions 'latitude' (or 'lat') and 'longitude' (or 'lon'), with equal size and in the same order, or a single tri-dimensional array with an additional dimension (e.g. 'bin') for the different categories. The arrays must contain probability values between 0 and 1, and the probabilities for all categories of a grid cell should not exceed 1 when added.
lon	A numeric vector with the longitudes of the map grid, in the same order as the values along the corresponding dimension in probs.
lat	A numeric vector with the latitudes of the map grid, in the same order as the values along the corresponding dimension in probs.
cat_dim	The name of the dimension along which the different categories are stored in probs. This only applies if probs is provided in the form of 3-dimensional array. The default expected name is 'bin'.
bar_titles	Vector of character strings with the names to be drawn on top of the color bar for each of the categories. As many titles as categories provided in probs must be provided.
col_unknown_cat	Character string with a colour representation of the colour to be used to paint the cells for which no category can be clearly assigned. Takes the value 'white' by default.
drawleg	Where to draw the common colour bar. Can take values TRUE, FALSE or: 'up', 'u', 'U', 'top', 't', 'T', 'north', 'n', 'N' 'down', 'd', 'D', 'bottom', 'b', 'B', 'south', 's', 'S' (default) 'right', 'r', 'R', 'east', 'e', 'E' 'left', 'l', 'L', 'west', 'w', 'W'
margin_scale	Numeric vector of length 4 specifying the plot margins (excluding the color bar), in the order: bottom, left, top, and right. If not specified (NULL), the default margins from par("mar"), which are c(5.1, 4.1, 4.1, 2.1), are used. Default is NULL.
bar_extra_margin	Numeric vector of length 4 specifying the margins to be added around the color bar, in the order: bottom, left, top, and right. The units are margin lines. The default values are c(1.5, 0, 1.5, 0).
...	Additional parameters to be sent to VizCombinedMap and VizEquiMap.

Value

Invisibly returns NULL after producing the plot.

Author(s)

Veronica Torralba, <veronica.torralba@bsc.es>, Nicolau Manubens, <nicolau.manubens@bsc.es>

See Also

VizCombinedMap and VizEquiMap

Examples

```
# Simple example:

x <- array(1:(20 * 10), dim = c(lat = 10, lon = 20)) / 200
a <- x * 0.6
b <- (1 - x) * 0.6
c <- 1 - (a + b)
lons <- seq(0, 359.5, length = 20)
lats <- seq(-89.5, 89.5, length = 10)

VizMostLikelyQuantileMap(list(a, b, c), lons, lats,
                         toptitle = 'Most likely tercile map',
                         bar_titles = paste('% of belonging to', c('a', 'b', 'c')),
                         brks = 20, width = 10, height = 8,
                         bar_extra_margin = c(1, 0, 1.5, 0))


# More complex example:

# 1. Generation of sample data
lons <- seq(0, 359.5, length = 40)
lats <- seq(-89.5, 89.5, length = 20)

# Generate sample data (with dimensions time, lat, lon)
sample_data <- sample(1:5, 50 * 20 * 40, replace = TRUE)
dim(sample_data) <- c(time = 50, lat = 20, lon = 40)

# 2. Binning sample data
n_bins <- 4
prob_thresholds <- 1:n_bins / n_bins
prob_thresholds <- prob_thresholds[1:(n_bins - 1)]
thresholds <- quantile(sample_data, prob_thresholds)

binning <- function(x, thresholds) {
  n_samples <- length(x)
  n_bins <- length(thresholds) + 1
  thresholds <- c(thresholds, max(x))
  result <- 1:n_bins
  lower_threshold <- min(x) - 1
  for (i in 1:n_bins) {
    result[i] <- sum(x > lower_threshold & x <= thresholds[i]) / n_samples
    lower_threshold <- thresholds[i]
  }
  dim(result) <- c(bin = n_bins)
  result
}
bins <- apply(sample_data, 2:3, binning, thresholds)
names(dim(bins))[1] <- "bin"
```

```
VizMostLikelyQuantileMap(bins, lons, lats,
                         toptitle = 'Most likely quantile map',
                         bar_titles = paste('% of belonging to', letters[1:n_bins]),
                         brks = 20, width = 10, height = 10,
                         bar_extra_margin = c(1, 0, 1.5, 0))
```

`VizPDFsOLE` *Plotting two probability density gaussian functions and the optimal linear estimation (OLE) as result of combining them.*

Description

This function plots two probability density gaussian functions and the optimal linear estimation (OLE) as result of combining them.

Usage

```
VizPDFsOLE(
  pdf_1,
  pdf_2,
  nsigma = 3,
  legendPos = "bottom",
  legendSize = 1,
  fileout = NULL,
  plotfile = NULL,
  width = 30,
  height = 15,
  units = "cm",
  dpi = 300
)
```

Arguments

`pdf_1`	A numeric array with a dimension named 'statistic', containg two parameters: mean' and 'standard deviation' of the first gaussian pdf to combining.
`pdf_2`	A numeric array with a dimension named 'statistic', containg two parameters: mean' and 'standard deviation' of the second gaussian pdf to combining.
`nsigma`	(optional) A numeric value for setting the limits of X axis. (Default nsigma = 3).
`legendPos`	(optional) A character value for setting the position of the legend ("bottom", "top", "right" or "left")(Default 'bottom').
`legendSize`	(optional) A numeric value for setting the size of the legend text. (Default 1.0).
`fileout`	(optional) A filename where the plot will be saved. (Default: the plot is not saved).

plotfile	Deprecated. Use 'fileout' instead
width	(optional) A numeric value indicating the plot width in units ("in", "cm", or "mm"). (Default width = 30).
height	(optional) A numeric value indicating the plot height. (Default height = 15).
units	(optional) A character value indicating the plot size unit. (Default units = 'cm').
dpi	(optional) A numeric value indicating the plot resolution. (Default dpi = 300).

Value

VizPDFsOLE() returns a ggplot object containing the plot.

Author(s)

Eroteida Sanchez-Garcia - AEMET, <esanchezg@aemet.es>

Examples

```
pdf_1 <- c(1.1,0.6)
attr(pdf_1, "name") <- "NAO1"
dim(pdf_1) <-  c(statistic = 2)
pdf_2 <- c(1,0.5)
attr(pdf_2, "name") <- "NAO2"
dim(pdf_2) <-  c(statistic = 2)

VizPDFsOLE(pdf_1, pdf_2)
```

VizPolygonsMap *Plot polygon-based maps in different projections*

Description

Plot spatial data defined over polygon geometries using a specified map projection. Values in 'data' are associated with polygon features in 'shapefile' through a shared region dimension, and the resulting map is rendered using **ggplot2** and **sf**. The order of regions in data must correspond to the row order of 'shapefile'.

Usage

```
VizPolygonsMap(
  data,
  shapefile,
  region_dim = "region",
  target_proj = NULL,
  bkg_shapefile = NULL,
  drawleg = "bar",
  dots = NULL,
```

```
  mask = NULL,
  brks = NULL,
  cols = NULL,
  bar_limits = NULL,
  triangle_ends = NULL,
  col_inf = NULL,
  col_sup = NULL,
  colNA = NULL,
  col_mask = "grey80",
  color_fun = ClimPalette(),
  bar_extra_margin = rep(0, 4),
  plot.margin = margin(0, 0, 0, 0, "in"),
  vertical = TRUE,
  borders = TRUE,
  toptitle = NULL,
  caption = NULL,
  units = NULL,
  xlab = NULL,
  ylab = NULL,
  xlab_interval = 4,
  ylab_interval = 4,
  box = NULL,
  title_scale = 16,
  dot_size = 0.5,
  dot_symbol = 16,
  ocean_color = "white",
  shapefile_color = "grey30",
  shapefile_width = 0.3,
  bkg_shapefile_fill = "grey40",
  bkg_shapefile_color = "grey30",
  bkg_shapefile_width = 0.3,
  default_land_color = "grey40",
  default_border_color = "grey40",
  default_border_width = 0.2,
  default_coast_color = "grey30",
  default_coast_width = 0.3,
  fileout = NULL,
  width = 8,
  height = 5,
  size_units = "in",
  res = 300
)
```

Arguments

data	A numeric array with a dimension corresponding to spatial regions.
shapefile	A shapefile of class "sf" containing the polygon geometries.
region_dim	A character string indicating the dimension name in 'data' corresponding to spa-

tial regions. The values in 'data' are matched with geometries in 'shapefile', so its length must be equal to `nrow(shapefile)`. The default value is 'region'.

`target_proj` A character string indicating the target projection. It should be a valid CRS string. The default projection is Robinson: "ESRI:54030". Note that the character string may work differently depending on PROJ and GDAL module version. If package version 'sf' is lower than "1.0.10" and an error appears regarding the target crs, you can try with numeric crs (e.g. target_proj = 54030).

`bkg_shapefile` A second (optional) shapefile for the background map of the plot. If this is provided, the default background from **rnaturalearth** is not drawn.

`drawleg` A character string indicating the legend style. It can be 'bar' (color bar by `ColorBarContinuous()`), 'ggplot2' (discrete legend by ggplot2), or FALSE (no legend). The default value is 'bar'.

`dots` An array with the same dimensions as 'data' of [0, 1] or logical values indicating the regions to dot. Regions where 'dots' is 1 or TRUE are dotted.

`mask` An array with the same dimensions as 'data' of [0, 1] or logical values indicating the regions to mask. Regions where 'mask' is 0 or FALSE are masked.

`brks`, `cols`, `bar_limits`, `triangle_ends`
Usually only providing 'brks' is enough to generate the desired color bar. These parameters allow to define n breaks that define n - 1 intervals to classify each of the values in 'data'. The corresponding region of a given value in 'data' will be colored in function of the interval it belongs to. These parameters are sent to `ColorBarContinuous()` to generate the breaks and colours. Additional colors for values beyond the limits of the colour bar are also generated and applied to the plot if 'bar_limits' or 'brks' and 'triangle_ends' are properly provided to do so. See ?ColorBarContinuous for a full explanation.

`col_inf`, `col_sup`, `colNA`
Color identifiers to color the values that exceed the extremes of the color bar and to color NAs, respectively. 'colNA' takes attr(cols, 'na_color') if available by default, where cols is the parameter 'cols' if provided or the vector of colors returned by 'color_fun'. 'col_inf' and 'col_sup' will take the value of 'colNA' if not specified. See ?ColorBarContinuous for a full explanation.

`col_mask` Colour used for masked regions (where mask is FALSE). Default is "grey80".

`color_fun`, `bar_extra_margin`
Set of parameters to control the visual aspect of the drawn colour bar (1/3). See ?ColorBarContinuous for a full explanation.

`plot.margin` A `ggplot2` margin object specifying the plot margins.

`vertical` A logical value indicating the direction of colorbar if parameter 'drawleg' is 'bar'. The default value is TRUE.

`borders` A logical value indicating whether polygon borders from 'shapefile' should be drawn. If TRUE (default), borders are drawn using 'shapefile_color' and 'shapefile_width'.

`toptitle` A character string of the top title of the figure, scalable with parameter 'title_scale'.

`caption` A character string of the caption located at left-bottom of the plot.

`units` A character string indicating the data units. Used as the legend title.

`xlab` A character string specifying the label of the x-axis (typically longitude). If NULL (default), no x-axis label is drawn.

`ylab` A character string specifying the label of the y-axis (typically latitude). If NULL (default), no x-axis label is drawn.

`xlab_interval` Numeric value indicating the spacing (in degrees) between longitude axis ticks. Default is 4.

`ylab_interval` Numeric value indicating the spacing (in degrees) between latitude axis ticks. Default is 4.

`box` A named numeric vector defining the longitude–latitude bounding box of the plot extent, with elements 'xmin', 'xmax', 'ymin', and 'ymax'. The box is interpreted in geographic coordinates (EPSG:4326) and transformed to the target projection. If NULL (default), the plot extent is taken from `sf::st_bbox(shapefile)`.

`title_scale` A number of the size of the top title. The default is 16.

`dot_size` A number of the size of the dots. The default is 0.5.

`dot_symbol` A number indicating the dot shape recognized by parameter 'shape' in `geom_point()`.

`ocean_color` Ocean color. Default is "white".

`shapefile_color`
Color of 'borders' within the data 'shapefile'. Default is "grey30".

`shapefile_width`
A number of the width of 'borders' within the data 'shapefile'. Default is 0.3.

`bkg_shapefile_fill`
Fill color of the 'bkg_shapefile' polygons. Default is "grey40".

`bkg_shapefile_color`
Border color of the 'bkg_shapefile' polygons. Default is "grey30".

`bkg_shapefile_width`
A number indicating the border width of the 'bkg_shapefile' polygons. Default is 0.3.

`default_land_color`
Fill color of land in the default background used when no 'bkg_shapefile' is provided. Set to "transparent" to hide the default land. Default is "grey40".

`default_border_color`
Color of country borders outside in the default background when no 'bkg_shapefile' is provided. Set to "transparent" to hide the default external borders. Default is "grey40".

`default_border_width`
A number of the width of 'default_border_color' when no 'bkg_shapefile' is provided. Default is 0.2.

`default_coast_color`
Color of the default coastline when no 'bkg_shapefile' is provided. It is derived from **rnaturalearth**, using `ne_coastline()`. Set to "transparent" to hide the default coastline. The default is "grey30".

`default_coast_width`
A number indicating the width of the default

fileout	A character string of the path to save the plot. If not specified (default), a graphic device will pop up. The extension should be accepted by ggsave().
width	A number of the plot width, in the units specified in parameter 'size_units'. The default is 8.
height	A number of the plot height, in the units specified in parameter 'size_units'. The default is 4.
size_units	A character string of the units of the size of the device (file or window) to plot in. The default is 'in' (inches). See ?ggsave and ?Devices for details of the corresponding device.
res	Resolution of the device (file or window) to plot in. The default value is 300. See ?ggsave 'dpi' and ?Devices for details of the corresponding device.

Details

Polygon geometries are re-projected from geographic coordinates (longitude– latitude) to a target coordinate reference system (CRS), specified via target_proj. The default projection is Robinson (ESRI:54030). Optional elements such as legends, borders, coastlines, and significance dots can be added, and graphical parameters are provided to customise colours, borders, titles and layout.

Value

A map plot with specified projection, either in pop-up window or a saved file.

Examples

```
data <- array(seq(-10, 10, length.out = 1509), dim = c(region = 1509))
dots <- data
dots[which(dots < 4 & dots > -4)] <- 0
dots[which(dots != 0)] <- 1
mask <- sample(c(TRUE, FALSE), size = 1509, replace = TRUE)
mask <- array(mask, dim = c(region = 1509))
bbox <- sf::st_as_sfc(sf::st_bbox(c(xmin = -180, ymin = -90, xmax = 180,
                                    ymax = 90), crs = 4326))
grid <- sf::st_make_grid(bbox, n = c(39, 39), what = "polygons")[seq_len(1509)]
shapefile <- sf::st_sf(region = seq_len(1509), geometry = grid)

VizPolygonsMap(data, shapefile, dots = dots, brks = seq(-10, 10, length.out = 11),
               toptitle = 'synthetic example', vertical = FALSE,
               caption = 'Robinson Projection',
               bar_extra_margin = c(0, 1, 0, 1), width = 8, height = 6)
VizPolygonsMap(data, shapefile, drawleg = 'ggplot2', target_proj = "+proj=moll",
               brks = seq(-10, 10, length.out = 11),
               color_fun = ClimPalette("purpleorange"), colNA = 'green',
               toptitle = 'synthetic example',
               caption = 'Mollweide Projection', width = 8, height = 6)
```

VizRobinson *Plot map in Robinson or other projections*

Description

Transform a regular grid longitude-latitude data to a different projection and plot the map. The target projection must be a valid CRS string, preferrably be EPSG or ESRI code; check st_crs for more explanation. This function is mainly tested for Robinson projection (ESRI:54030), but it can work with other projection types in theory.

The map can be plotted by points or polygon. A legend can be plotted as either a color bar or a discrete ggplot legend. Dots can be drawn on top of the data, which can be used for significance test. A mask can be added to not plot the data specified. A number of options is provided to adjust aesthetics, like position, size, colors, etc.

Usage

```
VizRobinson(
  data,
  lon,
  lat,
  lon_dim = NULL,
  lat_dim = NULL,
  target_proj = NULL,
  drawleg = "bar",
  style = "point",
  dots = NULL,
  mask = NULL,
  brks = NULL,
  cols = NULL,
  bar_limits = NULL,
  triangle_ends = NULL,
  col_inf = NULL,
  col_sup = NULL,
  colNA = NULL,
  color_fun = ClimPalette(),
  bar_extra_margin = rep(0, 4),
  vertical = TRUE,
  toptitle = NULL,
  caption = NULL,
  units = NULL,
  crop_coastlines = NULL,
  point_size = "auto",
  title_scale = 16,
  title_size = NULL,
  dot_size = 0.5,
  dots_size = NULL,
  dot_symbol = 47,
```

```
  dots_shape = NULL,
  coast_width = 0.3,
  coastlines_width = NULL,
  fileout = NULL,
  width = 8,
  height = 4,
  size_units = "in",
  res = 300
)
```

Arguments

data	A numeric array with longitude and latitude dimensions. The grid should be regular grid. It can contain NA values.
lon	A numeric vector of longitude locations of the cell centers of the grid of 'data'. Expected to be regularly spaced, within the range of either [-180, 180] or [0, 360].
lat	A numeric vector of latitude locations of the cell centers of the grid of 'data'. Expected to be regularly spaced, within the range [-90, 90] of ascending or descending order.
lon_dim	A character string indicating the longitude dimension name in 'data'. If it is NULL, the function tries to find the name in `esviz:::.KnownLonNames`. The default value is NULL.
lat_dim	A character string indicating the latitude dimension name in 'data'. If it is NULL, the function tries to find the name in `esviz:::.KnownLatNames`. The default value is NULL.
target_proj	A character string indicating the target projection. It should be a valid crs string. The default projection is Robinson: "ESRI:54030". Note that the character string may work differently depending on PROJ and GDAL module version. If package version 'sf' is lower than "1.0.10" and an error appears regarding the target crs, you can try with numeric crs (e.g. target_proj = 54030).
drawleg	A character string indicating the legend style. It can be 'bar' (color bar by `ColorBarContinuous()`), 'ggplot2' (discrete legend by ggplot2), or FALSE (no legend). The default value is 'bar'.
style	A character string indicating the plotting style. It can be 'point' or 'polygon'. The default value is 'point'. Note that 'polygon' may be time- and memory-consuming for global or high-resolution data.
dots	An array with the same dimensions as 'data' of [0, 1] or logical indicating the grids to plot dots. The value 0 or FALSE is the point to be dotted.
mask	An array with the same dimensions as 'data' of [0, 1] or logical indicating the grids to not plot data. The value 0 or FALSE is the point not to be plotted.
brks, cols, bar_limits, triangle_ends	Usually only providing 'brks' is enough to generate the desired color bar. These parameters allow to define n breaks that define n - 1 intervals to classify each of the values in 'data'. The corresponding grid cell of a given value in 'data' will be colored in function of the interval it belongs to. These parameters are sent to

ColorBarContinuous() to generate the breaks and colours. Additional colors for values beyond the limits of the colour bar are also generated and applied to the plot if 'bar_limits' or 'brks' and 'triangle_ends' are properly provided to do so. See ?ColorBarContinuous for a full explanation.

col_inf, col_sup, colNA
: Colour identifiers to color the values that excess the extremes of the color bar and to color NAs, respectively. 'colNA' takes attr(cols, 'na_color') if available by default, where cols is the parameter 'cols' if provided or the vector of colors returned by 'color_fun'. 'col_inf' and 'col_sup' will take the value of 'colNA' if not specified. See ?ColorBarContinuous for a full explanation.

color_fun, bar_extra_margin
: Set of parameters to control the visual aspect of the drawn colour bar (1/3). See ?ColorBarContinuous for a full explanation.

vertical
: A logical value indicating the direction of colorbar if parameter 'drawleg' is 'bar'. The default value is TRUE.

toptitle
: A character string of the top title of the figure, scalable with parameter 'title_scale'.

caption
: A character string of the caption located at left-bottom of the plot.

units
: A character string of the data units, which is the title of the legend.

crop_coastlines
: A named numeric vector [lonmin, lonmax, latmin, latmax] indicating the region to plot coastlines. Note that the longitude range cannot exceed 180 degrees.

point_size
: A number of the size of the data points if "style = 'point'". The default is 'auto' and the function tries to find the appropriate size.

title_scale
: A number of the size of the top title. The default is 16.

title_size
: Deprecated. Use 'title_scale' instead.

dot_size
: A number of the size of the dots. The default is 0.5.

dots_size
: Deprecated. Use 'dot_size' instead.

dot_symbol
: A number indicating the dot shape recognized by parameter 'shape' in geom_point().

dots_shape
: Deprecated. Use 'dot_symbol' instead.

coast_width
: A number indicating the width of the coastlines. Default is 0.3.

coastlines_width
: Deprecated. Use 'coast_width' instead.

fileout
: A character string of the path to save the plot. If not specified (default), a graphic device will pop up. The extension should be accepted by ggsave().

width
: A number of the plot width, in the units specified in parameter 'size_units'. The default is 8.

height
: A number of the plot height, in the units specified in parameter 'size_units'. The default is 4.

size_units
: A character string of the units of the size of the device (file or window) to plot in. The default is 'in' (inches). See ?ggsave and ?Devices for details of the corresponding device.

res
: Resolution of the device (file or window) to plot in. The default value is 300. See ?ggsave 'dpi' and ?Devices for details of the corresponding device.

Value

A map plot with speficied projection, either in pop-up window or a saved file.

Examples

```
data <- array(rep(seq(-10, 10, length.out = 181), 360) + rnorm(360),
              dim = c(lat = 181, lon = 360))
dots <- data
dots[which(dots < 4 & dots > -4)] <- 0
dots[which(dots != 0)] <- 1
VizRobinson(data, lon = 0:359, lat = -90:90, dots = dots,
            brks = seq(-10, 10, length.out = 11), drawleg = FALSE,
            toptitle = 'synthetic example', vertical = FALSE,
            caption = 'Robinson Projection',
            bar_extra_margin = c(0, 1, 0, 1), width = 8, height = 6)


VizRobinson(data, lon = 0:359, lat = -90:90, mask = dots, drawleg = FALSE,
            target_proj = "+proj=moll", brks = seq(-10, 10, length.out = 11),
            color_fun = ClimPalette("purpleorange"), colNA = 'green',
            toptitle = 'synthetic example', caption = 'Mollweide Projection',
            width = 8, height = 6)
```

`VizScorecard` *Function to plot Scorecard tables This function renders a scorecard table from a multidimensional array in HTML style. The structure of the table is based on the assignment of each dimension of the array as a structure element: row, subrow, column or subcolumn. It is useful to present tabular results with colors in a nice way. Note: Module PhantomJS is required.*

Description

Function to plot Scorecard tables

This function renders a scorecard table from a multidimensional array in HTML style. The structure of the table is based on the assignment of each dimension of the array as a structure element: row, subrow, column or subcolumn. It is useful to present tabular results with colors in a nice way.

Note: Module PhantomJS is required.

Usage

```
VizScorecard(
  data,
  sign = NULL,
  row_dim = "region",
  subrow_dim = "time",
```

```
  col_dim = "metric",
  subcol_dim = "sdate",
  legend_dim = "metric",
  row_names = NULL,
  subrow_names = NULL,
  col_names = NULL,
  subcol_names = NULL,
  row_title = NULL,
  subrow_title = NULL,
  col_title = NULL,
  table_title = NULL,
  table_subtitle = NULL,
  legend_breaks = NULL,
  plot_legend = TRUE,
  label_scale = 1.4,
  legend_width = NULL,
  legend_height = 50,
  palette = NULL,
  colorunder = NULL,
  colorsup = NULL,
  round_decimal = 2,
  font_size = 1.1,
  legend_white_space = 6,
  columns_width = 1.2,
  col1_width = NULL,
  col2_width = NULL,
  fileout = NULL
)
```

Arguments

data	A multidimensional array containing the data to be plotted with at least four dimensions. Each dimension will have assigned a structure element: row, subrow, column and subcolumn.
sign	A multidimensional boolean array with the same dimensions as 'data', indicting which values to be highlighted. If set to NULL no values will be highlighted.
row_dim	A character string indicating the dimension name to show in the rows of the plot. It is set as 'region' by default.
subrow_dim	A character string indicating the dimension name to show in the sub-rows of the plot. It is set as 'time' by default.
col_dim	A character string indicating the dimension name to show in the columns of the plot. It is set as 'metric' by default.
subcol_dim	A character string indicating the dimension name to show in the sub-columns of the plot. It is set as 'sdate' by default.
legend_dim	A character string indicating the dimension name to use for the legend. It is set as 'metric' by default.

row_names	A vector of character strings with row display names. It is set as NULL by default.
subrow_names	A vector of character strings with sub-row display names. It is set as NULL by default.
col_names	A vector of character strings with column display names. It is set as NULL by default.
subcol_names	A vector of character strings with sub-column display names. It is set as NULL by default.
row_title	A character string for the title of the row names. It is set as NULL by default.
subrow_title	A character string for the title of the sub-row names. It is set as NULL by default.
col_title	A character string for the title of the column names. It is set as NULL by default.
table_title	A character string for the title of the plot. It is set as NULL by default.
table_subtitle	A character string for the sub-title of the plot. It is set as NULL by default.
legend_breaks	A vector of numerics or a list of vectors of numerics, containing the breaks for the legends. If a vector is given as input, then these breaks will be repeated for each 'legend_dim'. A list of vectors can be given as input if the 'legend_dims' require different breaks. This parameter is required even if the legend is not plotted, to define the colors in the scorecard table. It is set as NULL by default.
plot_legend	A logical value to determine if the legend is plotted. It is set as TRUE by default.
label_scale	A numeric value to define the size of the legend labels. It is set as 1.4 by default.
legend_width	A numeric value to define the width of the legend bars. By default it is set to NULL and calculated internally from the table width.
legend_height	A numeric value to define the height of the legend bars. It is set as 50 by default.
palette	A vector of character strings or a list of vectors of character strings containing the colors to use in the legends. If a vector is given as input, then these colors will be used for each legend_dim. A list of vectors can be given as input if different colors are desired for the legend_dims. This parameter must be included even if the legend is not plotted, to define the colors in the scorecard table.
colorunder	A character string, a vector of character strings or a list with single character string elements defining the colors to use for data values with are inferior to the lowest breaks value. This parameter will also plot a inferior triangle in the legend bar. The parameter can be set to NULL if there are no inferior values. If a character string is given this color will be applied to all 'legend_dims'. It is set as NULL by default.
colorsup	A character string, a vector of character strings or a list with single character string elements defining the colors to use for data values with are superior to the highest breaks value. This parameter will also plot a inferior triangle in the legend bar. The parameter can be set to NULL if there are no superior values. If a character string is given this color will be applied to all legend_dims. It is set as NULL by default.
round_decimal	A numeric indicating to which decimal point the data is to be displayed in the scorecard table. It is set as 2 by default.

font_size A numeric indicating the font size on the scorecard table. Default is 2.

legend_white_space A numeric value defining the initial starting position of the legend bars, the white space infront of the legend is calculated from the left most point of the table as a distance in cm. The default value is 6.

columns_width A numeric value defining the width all columns within the table in cm (excluding the first and second columns containing the titles).

col1_width A numeric value defining the width of the first table column in cm. It is set as NULL by default.

col2_width A numeric value defining the width of the second table column in cm. It is set as NULL by default.

fileout A path of the location to save the scorecard plots. By default the plots will be saved to the working directory.

Value

An image file containing the scorecard.

Examples

```
data <- array(rnorm(1000), dim = c('sdate' = 12, 'metric' = 4, 'region' = 3,
                                   'time' = 6))
row_names <- c('Tropics', 'Extra-tropical NH', 'Extra-tropical SH')
col_names <- c('Mean bias (K)', 'Correlation', 'RPSS','CRPSS')

tmp <- tempfile()
VizScorecard(data = data, row_names = row_names, col_names = col_names,
             subcol_names =  month.abb[as.numeric(1:12)],
             row_title = 'Region', subrow_title = 'Forecast Month',
             col_title = 'Start date',
             table_title = "Temperature of ECMWF System 5",
             table_subtitle = "(Ref: ERA5 1994-2016)",
             plot_legend = FALSE, fileout = tmp)
unlink(paste0(tmp, "*"))
```

VizSection *Plots A Vertical Section*

Description

Plot a (longitude,depth) or (latitude,depth) section.

Usage

```
VizSection(
  data,
  horiz,
  depth,
  toptitle = "",
  title_scale = 1,
  sizetit = NULL,
  units = "",
  brks = NULL,
  cols = NULL,
  axelab = TRUE,
  intydep = 200,
  intxhoriz = 20,
  drawleg = TRUE,
  fileout = NULL,
  width = 10,
  height = 8,
  size_units = "in",
  res = 100,
  var = NULL,
  ...
)
```

Arguments

data	Matrix to plot with (longitude/latitude, depth) dimensions.
horiz	Array of longitudes or latitudes.
depth	Array of depths.
toptitle	Title, optional.
title_scale	Scale factor for the figure top title. Defaults to 1.
sizetit	Deprecated. Use 'title_scale' instead.
units	Units, optional.
brks	Colour levels, optional.
cols	List of colours, optional.
axelab	TRUE/FALSE, label the axis. Default = TRUE.
intydep	Interval between depth ticks on y-axis. Default: 200m.
intxhoriz	Interval between longitude/latitude ticks on x-axis. Default: 20deg.
drawleg	Draw colorbar. Default: TRUE.
fileout	Name of output file. Extensions allowed: eps/ps, jpeg, png, pdf, bmp and tiff. Default = NULL
width	File width, in the units specified in the parameter size_units (inches by default). Takes 8 by default.

height	File height, in the units specified in the parameter size_units (inches by default). Takes 5 by default.
size_units	Units of the size of the device (file or window) to plot in. Inches ('in') by default. See ?Devices and the creator function of the corresponding device.
res	Resolution of the device (file or window) to plot in. See ?Devices and the creator function of the corresponding device.
var	Deprecated. Use 'data' instead.
...	Arguments to be passed to the method. Only accepts the following graphical parameters: adj ann ask bg bty cex.lab cex.sub cin col.axis col.lab col.main col.sub cra crt csi cxy err family fg fig fin font font.axis font.lab font.main font.sub lend lheight ljoin lmitre lty lwd mex mfcol mfrow mfg mkh oma omd omi page pch pin plt pty smo srt tcl usr xaxp xaxs xaxt xlog xpd yaxp yaxs yaxt ylbias ylog For more information about the parameters see 'par'.

Value

A figure in popup window by default, or saved to the specified path via `fileout`.

Examples

```
# Synthetic data
data <- array(rep(seq(25, 10, length.out = 7), each = 21) - rnorm(147),
              dim = c(lat = 21, depth = 7))
VizSection(data, horiz = 0:20, depth = seq(0, 300, length.out = 7),
           toptitle = 'Temperature cross-section', units = "degC")
```

VizStereoMap *Maps A Two-Dimensional Variable On A Polar Stereographic Projection*

Description

Map longitude-latitude array (on a regular rectangular or gaussian grid) on a polar stereographic world projection with coloured grid cells. Only the region within a specified latitude interval is displayed. A colour bar (legend) can be plotted and adjusted. It is possible to draw superimposed dots, symbols, boxes, contours, and arrows. A number of options is provided to adjust the position, size and colour of the components. This plot function is compatible with figure layouts if colour bar is disabled.

Usage

```
VizStereoMap(
  data = NULL,
  lon,
  lat,
```

```
varu = NULL,
varv = NULL,
latlims = c(60, 90),
toptitle = NULL,
sizetit = NULL,
units = NULL,
brks = NULL,
cols = NULL,
bar_limits = NULL,
triangle_ends = NULL,
col_inf = NULL,
col_sup = NULL,
colNA = NULL,
color_fun = ClimPalette(),
filled.continents = FALSE,
coast_color = NULL,
coast_width = 1,
contours = NULL,
brks2 = NULL,
contour_lwd = 0.5,
contour_color = "black",
contour_lty = 1,
contour_draw_label = TRUE,
contour_label_draw = NULL,
contour_label_scale = 0.6,
dots = NULL,
dot_symbol = 4,
dot_size = 0.8,
intlat = 10,
arr_subsamp = floor(length(lon)/30),
arr_scale = 1,
arr_ref_len = 15,
arr_units = "m/s",
arr_scale_shaft = 1,
arr_scale_shaft_angle = 1,
drawleg = TRUE,
subsampleg = NULL,
bar_extra_labels = NULL,
draw_bar_ticks = TRUE,
draw_separators = FALSE,
triangle_ends_scale = 1,
bar_label_digits = 4,
bar_label_scale = 1,
units_scale = 1,
bar_tick_scale = 1,
bar_extra_margin = rep(0, 4),
boxlim = NULL,
boxcol = "purple2",
```

```
  boxlwd = 5,
  margin_scale = rep(1, 4),
  title_scale = 1,
  numbfig = NULL,
  fileout = NULL,
  width = 6,
  height = 5,
  size_units = "in",
  res = 100,
  var = NULL,
  ...
)
```

Arguments

data	Array with the values at each cell of a grid on a regular rectangular or gaussian grid. The array is expected to have two dimensions: c(latitude, longitude). Longitudes can be in ascending or descending order and latitudes in any order. It can contain NA values (coloured with 'colNA'). Arrays with dimensions c(longitude, latitude) will also be accepted but 'lon' and 'lat' will be used to disambiguate so this alternative is not appropriate for square arrays.
lon	Numeric vector of longitude locations of the cell centers of the grid of 'data', in ascending or descending order (same as 'data'). Expected to be regularly spaced, within either of the ranges [-180, 180] or [0, 360]. Data for two adjacent regions split by the limits of the longitude range can also be provided, e.g. `lon = c(0:50, 300:360)` ('data' must be provided consitently).
lat	Numeric vector of latitude locations of the cell centers of the grid of 'data', in any order (same as 'data'). Expected to be from a regular rectangular or gaussian grid, within the range [-90, 90].
varu	Array of the zonal component of wind/current/other field with the same dimensions as 'data'.
varv	Array of the meridional component of wind/current/other field with the same dimensions as 'data'.
latlims	Latitudinal limits of the figure. Example : c(60, 90) for the North Pole c(-90,-60) for the South Pole
toptitle	Top title of the figure, scalable with parameter 'title_scale'.
sizetit	Scale factor for the figure top title provided in parameter 'toptitle'. Deprecated. Use 'title_scale' instead.
units	Title at the top of the colour bar, most commonly the units of the variable provided in parameter 'data'.
brks, cols, bar_limits, triangle_ends	Usually only providing 'brks' is enough to generate the desired colour bar. These parameters allow to define n breaks that define n - 1 intervals to classify each of the values in 'data'. The corresponding grid cell of a given value in 'data' will be coloured in function of the interval it belongs to. These parameters

are sent to ColorBarContinuous() to generate the breaks and colours. Additional colours for values beyond the limits of the colour bar are also generated and applied to the plot if 'bar_limits' or 'brks' and 'triangle_ends' are properly provided to do so. See ?ColorBarContinuous for a full explanation.

col_inf, col_sup, colNA
: Colour identifiers to colour the values in 'data' that go beyond the extremes of the colour bar and to colour NA values, respectively. 'colNA' takes attr(cols, 'na_color') if available by default, where cols is the parameter 'cols' if provided or the vector of colors returned by 'color_fun'. If not available, it takes 'pink' by default. 'col_inf' and 'col_sup' will take the value of 'colNA' if not specified. See ?ColorBarContinuous for a full explanation on 'col_inf' and 'col_sup'.

color_fun, subsampleg, bar_extra_labels, draw_bar_ticks, draw_separators, triangle_ends_scale, bar_label_digits, bar_label_scale, units_scale, bar_tick_scale, bar_extra_margin
: Set of parameters to control the visual aspect of the drawn colour bar. See ?ColorBarContinuous for a full explanation.

filled.continents
: Colour to fill in drawn projected continents. Takes the value gray(0.5) by default. If set to FALSE, continents are not filled in.

coast_color
: Colour of the coast line of the drawn projected continents. Takes the value gray(0.5) by default.

coast_width
: Line width of the coast line of the drawn projected continents. Takes the value 1 by default.

contours
: Array of same dimensions as 'data' to be added to the plot and displayed with contours. Parameter 'brks2' is required to define the magnitude breaks for each contour curve.

brks2
: A numeric value or vector of magnitude breaks where to draw contour curves for the array provided in 'contours'. If it is a number, it represents the number of breaks (n) that defines (n - 1) intervals to classify 'contours'.

contour_lwd
: Line width of the contour curves provided via 'contours' and 'brks2'. The default value is 0.5.

contour_color
: Line color of the contour curves provided via 'contours' and 'brks2'.

contour_lty
: Line type of the contour curves. Takes 1 (solid) by default. See help on 'lty' in par() for other accepted values.

contour_draw_label
: A logical value indicating whether to draw the contour labels (TRUE) or not (FALSE) when 'contours' is used. The default value is TRUE.

contour_label_draw
: Deprecated. Use 'contour_draw_label' instead.

contour_label_scale
: Scale factor for the superimposed labels when drawing contour levels. The default value is 0.6.

dots
: Array of same dimensions as 'data' or with dimensions c(n, dim(data)), where n is the number of dot/symbol layers to add to the plot. A value of TRUE at a grid cell will draw a dot/symbol on the corresponding square of the plot. By default

all layers provided in 'dots' are plotted with dots, but a symbol can be specified for each of the layers via the parameter 'dot_symbol'.

`dot_symbol` Single character/number or vector of characters/numbers that correspond to each of the symbol layers specified in parameter 'dots'. If a single value is specified, it will be applied to all the layers in 'dots'. Takes 15 (centered square) by default. See 'pch' in par() for additional accepted options.

`dot_size` Scale factor for the dots/symbols to be plotted, specified in 'dots'. If a single value is specified, it will be applied to all layers in 'dots'. Takes 1 by default.

`intlat` Interval between latitude lines (circles), in degrees. Defaults to 10.

`arr_subsamp` A number as subsampling factor to select a subset of arrows in 'varu' and 'varv' to be drawn. Only one out of arr_subsamp arrows will be drawn. The default value is 1.

`arr_scale` A number as scale factor for drawn arrows from 'varu' and 'varv'. The default value is 1.

`arr_ref_len` A number of the length of the refence arrow to be drawn as legend at the bottom of the figure (in same units as 'varu' and 'varv', only affects the legend for the wind or variable in these arrays). The default value is 15.

`arr_units` Units of 'varu' and 'varv', to be drawn in the legend. Takes 'm/s' by default.

`arr_scale_shaft`
A number for the scale of the shaft of the arrows (which also depend on the number of figures and the arr_scale parameter). The default value is 1.

`arr_scale_shaft_angle`
A number for the scale of the angle of the shaft of the arrows (which also depend on the number of figure and the arr_scale parameter). The default value is 1.

`drawleg` Whether to plot a color bar (legend, key) or not. Defaults to TRUE.

`boxlim` Limits of a box to be added to the plot, in degrees: c(x1, y1, x2, y2). A list with multiple box specifications can also be provided.

`boxcol` Colour of the box lines. A vector with a colour for each of the boxes is also accepted. Defaults to 'purple2'.

`boxlwd` Line width of the box lines. A vector with a line width for each of the boxes is also accepted. Defaults to 5.

`margin_scale` Scale factor for the margins to be added to the plot, with the format c(y1, x1, y2, x2). Defaults to rep(1, 4). If drawleg = TRUE, margin_scale[1] is subtracted 1 unit.

`title_scale` Scale factor for the figure top title. Defaults to 1.

`numbfig` Number of figures in the layout the plot will be put into. A higher numbfig will result in narrower margins and smaller labels, axe labels, ticks, thinner lines, ... Defaults to 1.

`fileout` File where to save the plot. If not specified (default) a graphics device will pop up. Extensions allowed: eps/ps, jpeg, png, pdf, bmp and tiff.

`width` File width, in the units specified in the parameter size_units (inches by default). Takes 8 by default.

height	File height, in the units specified in the parameter size_units (inches by default). Takes 5 by default.
size_units	Units of the size of the device (file or window) to plot in. Inches ('in') by default. See ?Devices and the creator function of the corresponding device.
res	Resolution of the device (file or window) to plot in. See ?Devices and the creator function of the corresponding device.
var	Deprecated. Use 'data' instead.
...	Arguments to be passed to the method. Only accepts the following graphical parameters: adj ann ask bg bty cex.sub cin col.axis col.lab col.main col.sub cra crt csi cxy err family fg font font.axis font.lab font.main font.sub lend lheight ljoin lmitre mex mfcol mfrow mfg mkh omd omi page pch pin plt pty smo srt tcl usr xaxp xaxs xaxt xlog xpd yaxp yaxs yaxt ylbias ylog For more information about the parameters see 'par'.

Value

brks	Breaks used for colouring the map (and legend if drawleg = TRUE).
cols	Colours used for colouring the map (and legend if drawleg = TRUE). Always of length length(brks) - 1.
col_inf	Colour used to draw the lower triangle end in the colour bar (NULL if not drawn at all).
col_sup	Colour used to draw the upper triangle end in the colour bar (NULL if not drawn at all).

Examples

```
data <- matrix(rnorm(100 * 50), 100, 50)
x <- seq(from = 0, to = 360, length.out = 100)
y <- seq(from = -90, to = 90, length.out = 50)
VizStereoMap(data = data, lon = x, lat = y, latlims = c(60, 90), brks = 50,
            toptitle = "This is the title")
```

VizTriangles4Categories

Function to convert any 3-d numerical array to a grid of coloured triangles.

Description

This function converts a 3-d numerical data array into a coloured grid with triangles. It is useful for a slide or article to present tabular results as colors instead of numbers. This can be used to compare the outputs of two or four categories (e.g. modes of variability, clusters, or forecast systems).

Usage

```
VizTriangles4Categories(
  data,
  brks = NULL,
  cols = NULL,
  toptitle = NULL,
  sig_data = NULL,
  pch_sig = 18,
  col_sig = "black",
  cex_sig = 1,
  xlab = TRUE,
  ylab = TRUE,
  xlabels = NULL,
  xtitle = NULL,
  ylabels = NULL,
  ytitle = NULL,
  drawleg = TRUE,
  legend = NULL,
  lab_legend = NULL,
  cex_leg = 1,
  col_leg = "black",
  cex_axis = 1.5,
  mar = c(5, 4, 0, 0),
  fileout = NULL,
  size_units = "px",
  res = 100,
  figure.width = 1,
  ...
)
```

Arguments

data	Array with three named dimensions: 'dimx', 'dimy', 'dimcat', containing the values to be displayed in a coloured image with triangles.
brks	A vector of the color bar intervals. The length must be one more than the parameter 'cols'. Use ColorBar() to generate default values.
cols	A vector of valid colour identifiers for color bar. The length must be one less than the parameter 'brks'. Use ColorBar() to generate default values.
toptitle	A string of the title of the grid. Set NULL as default.
sig_data	Logical array with the same dimensions as 'data' to add layers to the plot. A value of TRUE at a grid cell will draw a dot/symbol on the corresponding triangle of the plot. Set NULL as default.
pch_sig	Symbol to be used to represent sig_data. Takes 18 (diamond) by default. See 'pch' in par() for additional accepted options.
col_sig	Colour of the symbol to represent sig_data.
cex_sig	Parameter to increase/reduce the size of the symbols used to represent sig_data.

xlab	A logical value (TRUE) indicating if xlabels should be plotted
ylab	A logical value (TRUE) indicating if ylabels should be plotted
xlabels	A vector of labels of the x-axis The length must be length of the col of parameter 'data'. Set the sequence from 1 to the length of the row of parameter 'data' as default.
xtitle	A string of title of the x-axis. Set NULL as default.
ylabels	A vector of labels of the y-axis The length must be length of the row of parameter 'data'. Set the sequence from 1 to the length of the row of parameter 'data' as default.
ytitle	A string of title of the y-axis. Set NULL as default.
drawleg	A logical value to decide to draw the color bar legend or not. Set TRUE as default.
legend	Deprecated. Use 'drawleg' instead.
lab_legend	A vector of labels indicating what is represented in each category (i.e. triangle). Set the sequence from 1 to the length of the categories (2 or 4).
cex_leg	A number to indicate the increase/reductuion of the lab_legend used to represent sig_data.
col_leg	Color of the legend (triangles).
cex_axis	A number to indicate the increase/reduction of the axis labels.
mar	A numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot.
fileout	A string of full directory path and file name indicating where to save the plot. If not specified (default), a graphics device will pop up.
size_units	A string indicating the units of the size of the device (file or window) to plot in. Set 'px' as default. See ?Devices and the creator function of the corresponding device.
res	A positive number indicating resolution of the device (file or window) to plot in. See ?Devices and the creator function of the corresponding device.
figure.width	a numeric value to control the width of the plot.
...	The additional parameters to be passed to function ColorBarContinuous() in for color legend creation.

Value

A figure in popup window by default, or saved to the specified path.

Author(s)

History:
1.0 - 2020-10 (V.Torralba, <veronica.torralba@bsc.es>) - Original code

Examples

```
# Example with random data
arr1 <- array(runif(n = 4 * 5 * 4, min = -1, max = 1), dim = c(4,5,4))
names(dim(arr1)) <- c('dimx', 'dimy', 'dimcat')
arr2 <- array(TRUE, dim = dim(arr1))
arr2[which(arr1 < 0.3)] <- FALSE
VizTriangles4Categories(data = arr1,
                        cols = c('white','#fef0d9','#fdd49e','#fdbb84','#fc8d59'),
                        brks = c(-1, 0, 0.1, 0.2, 0.3, 0.4),
                        lab_legend = c('NAO+', 'BL','AR','NAO-'),
                        xtitle = "Target month", ytitle = "Lead time",
                        xlabels = c("Jan", "Feb", "Mar", "Apr"))
```

`VizVsLTime` *Plot a score along the forecast time with its confidence interval*

Description

Plot the correlation (`Corr()`), the root mean square error (`RMS()`) between the forecast values and their observational counterpart, the slope of their trend (`Trend()`), the InterQuartile range, maximum-mininum, standard deviation or median absolute Deviation of the ensemble members (`Spread()`), or the ratio between the ensemble spread and the RMSE of the ensemble mean (`RatioSDRMS()`) along the forecast time for all the input experiments on the same figure with their confidence intervals.

Usage

```
VizVsLTime(
  data,
  toptitle = "",
  ytitle = "",
  monini = 1,
  freq = 12,
  nticks = NULL,
  limits = NULL,
  listexp = c("exp1", "exp2", "exp3"),
  listobs = c("obs1", "obs2", "obs3"),
  biglab = FALSE,
  hlines = NULL,
  drawleg = TRUE,
  leg = NULL,
  siglev = FALSE,
  title_scale = 1,
  sizetit = NULL,
  show_conf = TRUE,
  fileout = NULL,
```

```
  width = 8,
  height = 5,
  size_units = "in",
  res = 100,
  var = NULL,
  ...
)
```

Arguments

data	Matrix containing any Prediction Score with dimensions: (nexp/nmod, 3/4 ,nltime) or (nexp/nmod, nobs, 3/4 ,nltime).
toptitle	Main title, optional.
ytitle	Title of Y-axis, optional.
monini	Starting month between 1 and 12. Default = 1.
freq	1 = yearly, 12 = monthly, 4 = seasonal, ... Default = 12.
nticks	Number of ticks and labels on the x-axis, optional.
limits	c(lower limit, upper limit): limits of the Y-axis, optional.
listexp	List of experiment names, optional.
listobs	List of observation names, optional.
biglab	TRUE/FALSE for presentation/paper plot. Default = FALSE.
hlines	c(a,b, ..) Add horizontal black lines at Y-positions a,b, ... Default = NULL.
drawleg	TRUE/FALSE if legend should be added or not to the plot. Default = TRUE.
leg	Deprecated. Use 'drawleg' instead.
siglev	TRUE/FALSE if significance level should replace confidence interval. Default = FALSE.
title_scale	Scale factor for the figure top title. Defaults to 1.
sizetit	Deprecated. Use 'title_scale' instead.
show_conf	TRUE/FALSE to show/not confidence intervals for input variables.
fileout	Name of output file. Extensions allowed: eps/ps, jpeg, png, pdf, bmp and tiff. The default value is NULL.
width	File width, in the units specified in the parameter size_units (inches by default). Takes 8 by default.
height	File height, in the units specified in the parameter size_units (inches by default). Takes 5 by default.
size_units	Units of the size of the device (file or window) to plot in. Inches ('in') by default. See ?Devices and the creator function of the corresponding device.
res	Resolution of the device (file or window) to plot in. See ?Devices and the creator function of the corresponding device.
var	Deprecated. Use 'data' instead.

... Arguments to be passed to the method. Only accepts the following graphical parameters:
adj ann ask bg bty cex.sub cin col.axis col.lab col.main col.sub cra crt csi cxy err family fg fig font font.axis font.lab font.main font.sub lheight ljoin lmitre mar mex mfcol mfrow mfg mkh oma omd omi page pch plt smo srt tck tcl usr xaxp xaxs xaxt xlog xpd yaxp yaxs yaxt ylbias ylog
For more information about the parameters see 'par'.

Details

Examples of input:
Model and observed output from `Load()` then `Clim()` then `Ano()` then `Smoothing()`:
(nmod, nmemb, nsdate, nltime) and (nobs, nmemb, nsdate, nltime)
then averaged over the members
`Mean1Dim(var_exp/var_obs, posdim = 2)`:
(nmod, nsdate, nltime) and (nobs, nsdate, nltime)
then passed through
`Corr(exp, obs, posloop = 1, poscor = 2)` or
`RMS(exp, obs, posloop = 1, posRMS = 2)`:
(nmod, nobs, 3, nltime)
would plot the correlations or RMS between each exp & each obs as a function of the forecast time.

Value

A figure in popup window by default, or saved to the specified path via `fileout`.

Examples

```
clim <- s2dv::Clim(ts_temp$exp, ts_temp$obs, time_dim = "sdate",
                   dat_dim = c("dat", "member"))
ano_exp <- s2dv::Ano(ts_temp$exp, clim$clim_exp)
ano_obs <- s2dv::Ano(ts_temp$obs, clim$clim_obs)
corr_ano <- s2dv::Corr(s2dv::MeanDims(ano_exp, 'member'), ano_obs,
                       time_dim = 'sdate', dat_dim = 'dat')
input_cor <- array(dim = c(dat = 1, 4, time = 5))
input_cor[, 1, ] <- corr_ano$conf.lower[, 1, 1, ]
input_cor[, 2, ] <- corr_ano$corr[, 1, 1, ]
input_cor[, 3, ] <- corr_ano$conf.upper[, 1, 1, ]
input_cor[, 4, ] <- corr_ano$p.val[, 1, 1, ]
VizVsLTime(input_cor, toptitle = "Correlation",
           monini = 11, limits = c(-1, 2), listexp = 'SEAS5',
           listobs = 'ERA5', biglab = FALSE, hlines = c(-1, 0, 1))
```

`VizWeeklyClim` *Plots the observed weekly means and climatology of a timeseries data*

Description

This function plots the observed weekly means and climatology of a timeseries data using ggplot package. It compares the weekly climatology in a specified period (reference period) to the observed conditions during the target period analyzed in the case study.

Usage

```
VizWeeklyClim(
  data,
  first_date,
  ref_period,
  last_date = NULL,
  data_years = NULL,
  time_dim = "time",
  sdate_dim = "sdate",
  ylim = NULL,
  toptitle = NULL,
  title = NULL,
  subtitle = NULL,
  ytitle = NULL,
  legend = TRUE,
  palette = "Blues",
  fileout = NULL,
  device = NULL,
  width = 8,
  height = 6,
  units = "in",
  dpi = 300
)
```

Arguments

data	A multidimensional array with named start date and lead time dimensions; if other dimensions are found, the function drops them with the first index. It can also be a dataframe with computed percentiles as input for ggplot. If it's a dataframe, it must contain the following column names: 'week', 'clim', 'p10', 'p90', 'p33', 'p66', 'week_mean', 'day' and 'data'.
first_date	The first date of the observed values of timeseries. It can be of class 'Date', 'POSIXct' or a character string in the format 'yyyy-mm-dd'. If parameter 'data_years' is not provided, it must be a date included in the reference period.
ref_period	A vector of numeric values indicating the years of the reference period. If parameter 'data_years' is not specified, it must be of the same length of dimension 'sdate_dim' of parameter 'data'.
last_date	Optional parameter indicating the last date of the target period of the daily timeseries. It can be of class 'Date', 'POSIXct' or a character string in the format 'yyyy-mm-dd'. If it is NULL, the last date of the daily timeseries will be set as the last date of 'data'. As the data is plotted by weeks, only full groups of 7 days

will be plotted. If the last date of the timeseries is not a multiple of 7 days, the last week will not be plotted.

data_years — A vector of numeric values indicating the years of the data. It must be of the same length of dimension 'sdate_dim' of parameter 'data'. It is optional, if not specified, all the years will be used as the target period.

time_dim — A character string indicating the daily time dimension name. The default value is 'time'.

sdate_dim — A character string indicating the start year dimension name. The default value is 'sdate'.

ylim — A numeric vector of length two providing limits of the scale. Use NA to refer to the existing minimum or maximum. For more information, see 'ggplot2' documentation of 'scale_y_continuous' parameter.

toptitle — The text for the top title of the plot. It is NULL by default.

title — Deprecated. Use 'toptitle' instead.

subtitle — The text for the subtitle of the plot. It is NULL bu default.

ytitle — Character string to be drawn as y-axis title. It is NULL by default.

legend — A logical value indicating whether a legend should be included in the plot. If it is TRUE or NA, the legend will be included. If it is FALSE, the legend will not be included. It is TRUE by default.

palette — A palette name from the R Color Brewer's package. The default value is 'Blues'.

fileout — A character string indicating the file name where to save the plot. If not specified (default) a graphics device will pop up.

device — A character string indicating the device to use. Can either be a device function (e.g. png), or one of "eps", "ps", "tex" (pictex), "pdf", "jpeg", "tiff", "png", "bmp", "svg" or "wmf" (windows only).

width — A numeric value of the plot width in units ("in", "cm", "mm", or "px"). It is set to 8 by default.

height — A numeric value of the plot height in units ("in", "cm", "mm", or "px"). It is set to 6 by default.

units — Units of the size of the device (file or window) to plot in. Inches ('in') by default.

dpi — A numeric value of the plot resolution. It is set to 300 by default.

Value

A ggplot object containing the plot.

Examples

```
data <- array(rnorm(49*20*3, 274), dim = c(time = 49, sdate = 20, member = 3))
VizWeeklyClim(data = data, first_date = '2002-08-09',
             last_date = '2002-09-15', ref_period = 2010:2019,
             data_years = 2000:2019, time_dim = 'time', sdate_dim = 'sdate',
             toptitle = "Observed weekly means and climatology",
             subtitle = "Target years: 2010 to 2019",
             ytitle = paste0('tas', " (", "deg.C", ")"))
```

Index

BarPlotCat, 3

ClimColors *(*ClimPalette*)*, 6
ClimPalette, 6
ColorBarContinuous, 7
ColorBarDiscrete, 10

Hatching, 13

map_temp, 15

ShapeToMask, 17
st_crs, *64*

ts_temp, 20

Viz2VarsVsLTime, 20
VizACC, 23
VizAnimateMap, 25
VizAno, 28
VizBoxWhisker, 30
VizClim, 32
VizCombinedMap, 34
VizEquiMap, 37
VizForecastPDF, 45
VizLayout, 48
VizMatrix, 53
VizMostLikelyQuantileMap, 55
VizPDFsOLE, 58
VizPolygonsMap, 59
VizRobinson, 64
VizScorecard, 67
VizSection, 70
VizStereoMap, 72
VizTriangles4Categories, 77
VizVsLTime, 80
VizWeeklyClim, 82