Package 'GHRexplore'

November 6, 2026

Title Exploratory Analysis of Temporal and Spatio-Temporal Health Data

Version 0.3.0

Description A collection of commonly used visualizations of temporal and spatio-temporal health data including case counts, incidence rates, and covariates. The available plot types include time series, heatmaps, seasonality plots, maps and more. The package supports standard data transformations such as temporal and spatial aggregations, while offering extensive customization options for the resulting figures.

License GPL (>= 2)

URL https://gitlab.earth.bsc.es/ghr/ghrexplore, https://bsc-es.github.io/GHRtools/docs/GHRexplore/GHRexplore

BugReports https://gitlab.earth.bsc.es/ghr/ghrexplore/-/work_items

Depends R (>= 4.1.0)

Imports colorspace, cowplot, dplyr, ggplot2 (>= 3.5.0), grDevices, RColorBrewer, rlang, stats, tidyr, lubridate

Suggests knitr, rmarkdown, gganimate, magick, sf, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Carles Milà [aut, cre] (ORCID: <https://orcid.org/0000-0003-0470-0760>), Giovenale Moirano [aut] (ORCID: <https://orcid.org/0000-0001-8748-3321>), Anna B. Kawiecki [aut] (ORCID: <https://orcid.org/0000-0002-0499-2612>), Rachel Lowe [aut, cph] (ORCID: <https://orcid.org/0000-0003-3939-7343>)

Maintainer Carles Milà <carles.milagarcia@bsc.es>

Repository CRAN

Date/Publication 2026-11-06 15:50:02 UTC

Contents

aggregate_cases . . . 2
aggregate_cov . . . 4
animate_map . . . 5
check_consecutive . . . 8
date_to_epiyw . . . 9
dengue_MS . . . 10
dengue_SP . . . 11
epiyw_to_date . . . 12
get_time_interval . . . 13
GHR_palette . . . 13
log10p1_breaks_like . . . 14
log10p1_trans . . . 15
map_MS . . . 15
plot_bivariate . . . 16
plot_combine . . . 18
plot_compare . . . 20
plot_correlation . . . 21
plot_crosscorrelation . . . 23
plot_heatmap . . . 26
plot_map . . . 28
plot_multiple . . . 32
plot_seasonality . . . 34
plot_timeseries . . . 36
plot_timeseries2 . . . 40

Index **43**

`aggregate_cases` *Aggregate cases*

Description

Aggregates a data frame containing disease cases in space and/or time and provides case counts and incidence aggregations.

Usage

```
aggregate_cases(
  data,
  cases,
  pop,
  time,
  unit = NULL,
  pt = 1e+05,
  aggregate_space = NULL,
  aggregate_time = NULL,
```

```
  area = NULL
)
```

Arguments

data	Data frame containing equally spaced (daily, weekly, monthly) incident cases for one or multiple areas.
cases	Name of the variable that identifies the cases.
pop	Name of the variable that identifies the population.
time	Name of the variable that identifies the temporal dimension. The values must be in date format ("yyyy-mm-dd") representing the day of observation for daily data, the first day of the week for weekly, or the first day of the month for monthly observations.
unit	Optional name of the variable that identifies the different units for which a time series is available, typically a spatial unit.
pt	Scale of the person-time (default 100,000) for incidence rates.
aggregate_space	Name of variable used to define spatial aggregation groups that contain the different units.
aggregate_time	Temporal scale used to perform temporal aggregation. Options are: "week" (ISO 8601), "month", "year".
area	Deprecated, use unit instead.

Details

The temporal aggregation requires a number of time points depending on the time scale, e.g. from days to weeks it will require 7 points. Otherwise, it will return NA.

Value

A data frame with the aggregated cases and disease incidence.

Examples

```
# Load data
data("dengue_MS")

# Aggregate cases spatially from 'micro' to 'meso'
aggregate_cases(
  dengue_MS,
  time = "date",
  cases = "dengue_cases",
  pop = "population",
  unit = "micro_code",
  aggregate_space = "meso_name"
)
```

aggregate_cov *Aggregate covariates*

Description

Aggregates a data frame containing a covariate of interest in space and/or time.

Usage

```
aggregate_cov(
  data,
  var,
  time,
  unit = NULL,
  aggregate_space = NULL,
  aggregate_time = NULL,
  aggregate_space_fun = "mean",
  aggregate_time_fun = "mean",
  area = NULL
)
```

Arguments

data	Data frame containing equally spaced (daily, weekly, monthly) incident cases for one or multiple areas.
var	Name of the variable that identifies the covariate.
time	Name of the variable that identifies the temporal dimension. The values must be in date format ("yyyy-mm-dd"). representing the date of observation for daily data, the first day of the week for weekly, or the first day of the month for monthly observations.
unit	Optional name of the variable that identifies the different units for which a time series is available, typically a spatial unit.
aggregate_space	Name of variable used to define spatial aggregation groups that contain the different units.
aggregate_time	Temporal scale used to perform temporal aggregation. Options are: "week" (ISO 8601), "month", "year". See details.
aggregate_space_fun	Character indicating the function to be used in the aggregation over space, default is "mean".
aggregate_time_fun	Character indicating the function to be used in the aggregation over time, default is "mean".
area	Deprecated, use unit instead.

Details

The temporal aggregation requires a number of time points depending on the time scale, e.g. from days to weeks it will require 7 points. Otherwise, it will return NA.

Value

A data frame with the aggregated covariate.

Examples

```
# Load data
data("dengue_SP")

# Weekly to monthly total precipitation
aggregate_cov(
  dengue_SP,
  var = "precip_tot",
  time = "date",
  aggregate_time = "month",
  aggregate_time_fun = "sum"
)
```

---

animate_map *Choropleth map animation*

Description

Produces an animation of a choropleth map optionally complemented by a synchronised time series plot.

Usage

```
animate_map(
  data,
  var,
  time,
  unit = NULL,
  map,
  filename,
  type = "cov",
  pop = NULL,
  pt = 1e+05,
  map_unit = NULL,
  aggregate_time = NULL,
  aggregate_time_fun = "mean",
  transform = "identity",
  title = NULL,
```

```
  var_label = NULL,
  palette = NULL,
  centering = NULL,
  bins = NULL,
  bins_method = "quantile",
  bins_label = NULL,
  fps = 1,
  out_width = 600,
  out_height = 600,
  ts_include = FALSE,
  ts_color = "#168c81",
  ts_title = NULL,
  coordgrid = FALSE,
  ...,
  area = NULL,
  map_area = NULL
)
```

Arguments

data	Data frame containing equally spaced (daily, weekly, monthly) covariate or case observations for one or multiple locations.
var	Name of the column identifying the variable to be plotted.
time	Name of the variable that identifies the temporal dimension of the data frame. Its values must be in date format ("yyyy-mm-dd") representing the day of observation for daily data, the first day of the week for weekly, or the first day of the month for monthly observations.
unit	Name of the variable that identifies the different units for which a time series is available, typically a spatial unit.
map	sf object corresponding to the spatial unit specified in 'unit'.
filename	Path indicating the filename where the resulting gif animation should be saved.
type	Character that specifies the type of variable in var. Possible values include 'cov' (covariate, default), 'counts' (case counts), and 'inc' (case incidence). If type='inc', pop is required.
pop	Character identifying the variable name for population. Only needed if type='inc'.
pt	Scale of the person-time (default 100,000) for incidence rates.
map_unit	Name of the variable that identifies the different units in the map object. If not specified, it assumes the same name as in unit.
aggregate_time	Temporal scale for visualization and aggregation. Options include "year", "month", "week".
aggregate_time_fun	Character indicating the function to be used in the aggregation over time for type="cov". Options are "mean" (default), "median", "sum". For case counts and incidence, "sum" is always applied.

transform	Character, defaults to "identity" (i.e., no transformation). Transforms the color ramp for better visualization. Useful options include "log10p1" `log10(x+1)` for case counts and incidence with 0s, or any of the in-built ggplot2 options such as "log10" `log10(x)`, "log1p" `log(x+1)`, and "sqrt" `sqrt(x)` (check all possible options using `?scale_y_continuous`). I also applies to the optional time series plot.
title	Optional title of the plot.
var_label	Character with a custom name for the case or covariate variable.
palette	GHR, RColorBrewer or colorspace palette. Use "-" before the palette name (e.g., "-Reds") to reverse it.
centering	Numerical or "median", defaults to NULL. If set, it centers the palette on that value.
bins	Number of bins for categorization of numerical variables. Defaults to NULL (no binning).
bins_method	Method to compute the bins, only used when `bins` is not NULL. Possible values are "quantile" (default) and "equal".
bins_label	Optional labels for the bins. They must have the same length as the number of bins. Defaults to NULL (default interval labels).
fps	Animation frames per second, defaults to 1.
out_width	Horizontal size in pixels for the map gif.
out_height	Vertical size in pixels for the map gif.
ts_include	If TRUE, include a time series with the average value of the covariate across all units (type='cov'), the total number of cases (type='counts') or total incidence (type='inc'). See Details.
ts_color	Color for the progress bar of the time series plot. See details.
ts_title	Optional title for the time series plot. Defaults to NULL (no title). Only used when `ts_include=TRUE`.
coordgrid	Boolean indicating whether the coordinate grid should be included in the visualization. Defaults to FALSE.
...	Additional aesthetics to be passed to geom_sf. Possible values include `colour` (e.g., `colour="black"`), linewidth (e.g., `linewidth=0.1`), linetype (e.g., `linetype=2`), and alpha (e.g., `alpha=0.8`).
area	Deprecated, use `unit` instead.
map_area	Deprecated, use `map_unit` instead.

Details

The `magick` package is used to merge the two gifs when `ts_include=TRUE`, which may run into memory issues. Please reduce the number of time points, perform a temporal aggregation, or reduce the gif size if that happens.

Value

Invisibly returns the path to the saved gif file; called for its side effect of writing the animation to disk.

Examples

```
library("sf")
data("dengue_MS")
data("map_MS")

# Animation of monthly minimum temperature in 2002
animate_map(
  data = dengue_MS[dengue_MS$year == 2002, ],
  var = "tmin",
  time = "date",
  unit = "micro_code",
  map = map_MS,
  map_unit = "code",
  type = "cov",
  var_label = "Min Temp",
  filename = tempfile(fileext = ".gif")
)
```

`check_consecutive` *check_consecutive*

Description

Verifies whether time values in a dataset are consecutive based on the detected time interval (e.g., daily, weekly, monthly, yearly). If a `unit` column is specified, the check is performed within each unit.

Usage

```
check_consecutive(data, time, unit = NULL, strict = FALSE, area = NULL)
```

Arguments

`data`	A data frame containing the dataset. Must include the time variable and optionally a unit variable.
`time`	A string representing the name of the column in `data` that contains time values. The column should be in "yyyy-mm-dd" format or any format recognizable by `as.Date()`.
`unit`	(Optional) A single character representing the name of the columns in `data` that identifies the different unit for which a time series is available, typically a spatial unit. If provided, the check will be performed separately within each unit.
`strict`	If FALSE (default) return a warning, if TRUE return an error.
`area`	Deprecated, use `unit` instead.

Details

Different time intervals per unit are not allowed.

Value

If strict = FALSE, returns a logical value: TRUE if all time values are consecutive, and FALSE otherwise. If strict = TRUE, the function throws an error if the check is not met.

Examples

```
df <- data.frame(
  date = seq(as.Date("2020-01-01"), by = "day", length.out = 10),
  cases = 1:10
)
check_consecutive(df, time = "date")
```

---

date_to_epiyw *Convert a date to epiyearweek format*

Description

The conversion is done following the ISO 8601 standard, whereby the first week of the year is the one that contains at least four days of the new year. This coincides with the approach taken by lubridate::epiweek and lubridate::isoweek but may not coincide with other definitions of epidemiological weeks.

Usage

```
date_to_epiyw(date, weekday = "Monday")
```

Arguments

date	A vector of type 'Date'.
weekday	The weekday that determines the start of the week. Defaults to 'Monday'.

Value

The transformed dates to epiyearweek.

Examples

```
# Convert dates in 'dengue_SP' in epiyw
data(dengue_SP)
head(dengue_SP$date)
dengue_SP$epiyw <- date_to_epiyw(as.Date(dengue_SP$date), "Sunday")
head(dengue_SP$epiyw)
```

dengue_MS *Dengue cases from the "Mato Grosso do Sul" state of Brazil*

Description

The dengue_MS example data set contains monthly counts of notified dengue cases by microregion, along with a range of spatial and spatiotemporal covariates (e.g., environmental, socio-economic and meteo-climatic factors). This data set represents a subset of a larger national data set that covers the entire territory of Brazil. The subset focuses on a specific region, *Mato Grosso do Sul*, for the purposes of illustration and computational efficiency. See the source below for access to the complete data set.

Usage

```
dengue_MS
```

Format

A data frame with 2,640 rows and 28 columns:

`micro_code` Unique ID number to each micro region (11 units)

`micro_name` Name of each micro region

`micro_name_ibge` Name of each micro region following IBGE

`meso_code` Unique ID number to each meso region (4 units)

`meso_name` Name of each meso region

`state_code` Unique ID number to each state (1 unit)

`state_name` Name of each state

`region_code` Unique ID number given to each Brazilian Region, In this data frame all observations come from the "Centre-West"

`region_name` Name of each Brazilian Region, In this data frame all observations come from the "Centre-West"

`biome_code` Biome code

`biome_name` Biome name

`ecozone_code` Ecozone code

`ecozone_name` Ecozone name

`main_climate` Most prevalent climate regime in the microregion. Based on Koppen Geiger climate regimes

`month` Calendar month index, 1 = January, 12 = December

`year` Year 2000 - 2019

`time` Time index starting at 1 for January 2000

`dengue_cases` Number of notified dengue cases registered in the notifiable diseases system in Brazil (SINAN) in the microregion of reference, at the month of first symptoms

`population` Estimated population, based on projections calculated using the 2000 and 2010 censuses, and counts taken in 2007 and 2017

`pop_density` Population density (number of people per km2)

`tmax` Monthly average daily maximum temperature; gridded values (at a 0.5 deg resolution) averaged across each microregion

`tmin` Monthly average daily minimum temperature; gridded values (at a 0.5 deg resolution) averaged across each microregion

`pdsi` Self-calibrated Palmer drought severity index for each microregion. It measures how wet or dry a region is relative to usual conditions. Negative values represent periods of drought, positive values represent wetter periods. Calculated by taking the mean value within each microregion

`urban` Percentage of inhabitants living in urban areas (2010 census)

`water_network` Percentage of inhabitants with access to the piped water network according to the 2010 census

`water_shortage` Frequency of reported water shortages per microregion between 2000 - 2016

`date` First day of the Month, in date format ("%Y-%m-%d")

`pop100000` Population divided by 100000.

Source

source code on GitHub; source code on Zenodo;

`dengue_SP` *Dengue cases from the "São Paulo" state of Brazil*

Description

The `dengue_SP` example data set reports the weekly number of notified dengue cases in the municipality of São Paulo together with climatic covariates. Data was sourced from Infodengue (see source below).

Usage

```
dengue_SP
```

Format

A data frame with 624 rows and 9 columns:

`date` First day of the week, in date format ("%Y-%m-%d")

`geocode` IBGE municipality code for the city of São Paulo

`cases` Number of notified dengue cases

`year` Year 2010 - 2021

`temp_med` Weekly average daily mean temperature

`precip_tot` Weekly cumulative precipitation

`enso` El Niño-Southern Oscillation Index

`pop` Number of inhabitants

`pop100000` Population divided by 100000.

Source

Infodengue API

`epiyw_to_date` *Convert a epiyearweek to date format*

Description

The conversion is done following the ISO 8601 standard, whereby the first week of the year is the one that contains at least four days of the new year. This coincides with the approach taken by `lubridate::epiweek` and `lubridate::isoweek` but may not coincide with other definitions of epidemiological weeks.

Usage

```
epiyw_to_date(epiyw, weekday = "Monday")
```

Arguments

`epiyw`	An epiyear-epiweek character vector in 'YYYY-WW' format.
`weekday`	Starting day of the epiweek. Defaults to Monday.

Value

The dates corresponding to each epiyearweek.

Examples

```
epiyw <- c("2025-01", "2025-10", "2025-20", "2025-30")
epiyw_to_date(epiyw)
```

get_time_interval *get_time_interval*

Description

This function determines the time interval (e.g., daily, weekly, monthly, yearly) of a dataset based on the differences between the first two time points.

Usage

```
get_time_interval(data, time, unit = NULL, area = NULL)
```

Arguments

data	an object of class 'data.frame' containing equally spaced time series data (daily, weekly, monthly data) for one or multiple spatial units.
time	A string representing the name of the column in data that contains time information. The column should be in "yyyy-mm-dd" format.
unit	(Optional) A string representing the names of the columns in data that identifies the different units for which a time series is available, typically a spatial unit. Required if data contains multiple units.
area	Deprecated, use unit instead.

Value

A character string indicating the time interval of the dataset. Possible values: "day", "week", "month", "year", or "unknown".

Examples

```
df <- data.frame(
  date = seq(as.Date("2020-01-01"), by = "week", length.out = 10),
  cases = 1:10
)
get_time_interval(df, time = "date")
```

GHR_palette *Generate GHR color palettes*

Description

Generates color palettes including custom, ColorBrewer and colorspace palettes.

Creates a visualization of all custom GHR palettes.

Usage

```
GHR_palette(palette, ncols = 30)

GHR_palettes()
```

Arguments

palette	Name of the GHR, RColorBrewer or colorspace palette. Use "-" before the palette name (e.g., "-Reds") to reverse it. A vector of custom colors is also possible.
ncols	Number of colors to sample.

Details

See all available options by running `GHR_palettes()`, `RColorBrewer::display.brewer.all()` and `colorspace::hcl_palettes(plot=TRUE)`.

Value

GHR_palette() returns the function that generates the color palette.
GHR_palettes() returns a plot with the custom GHR palettes.

Examples

```
GHR_palette("IDE1", 5)(5)

GHR_palettes()
```

---

log10p1_breaks_like *log10p1_breaks_like*

Description

Breaks and label generator for log10p1_trans

Usage

```
log10p1_breaks_like(x)
```

Arguments

x	the numeric vector for which labels are computed

Value

A numeric vector of breaks (powers of 10, preceded by 0 if x contains values below 1).

Examples

```
log10p1_breaks_like(c(0, 5, 2000))
log10p1_breaks_like(c(3, 50, 800))

library(ggplot2)
df <- data.frame(x = 1:5, y = c(0, 3, 30, 300, 3000))
ggplot(df, aes(x, y)) +
  geom_point() +
  scale_y_continuous(trans = log10p1_trans, breaks = log10p1_breaks_like)
```

`log10p1_trans` *log10p1_trans*

Description

Transform to execute a log10(x+1) scale transformation

Usage

```
log10p1_trans
```

Value

A transformation object of class `transform` (see `scales::new_transform()`) with `transform` and `inverse` functions, to be used in the `trans` or `transform` argument of ggplot2 scales.

Examples

```
log10p1_trans$transform(c(0, 9, 99))
log10p1_trans$inverse(c(0, 1, 2))

library(ggplot2)
df <- data.frame(x = 1:5, y = c(0, 3, 30, 300, 3000))
ggplot(df, aes(x, y)) +
  geom_point() +
  scale_y_continuous(trans = log10p1_trans)
```

`map_MS` *Administrative boundaries of* Mato Grosso do Sul

Description

Administrative boundaries (polygon geometries) of the 11 municipalities in *Mato Grosso do Sul* (Brazil) to be used with the `dengue_MS` data set.

Usage

```
data(map_MS)
```

Format

A simple feature (sf) object including 11 rows and 2 columns:

code Unique ID number to each micro region (11 units).

geometry Geometries of the sf multipolygon.

`plot_bivariate` *Bivariate plot*

Description

Plots a bivariate graph to visually assess associations. It will be a scatterplot if both variables are numeric and grouped boxplots if one of them is categorical.

Usage

```
plot_bivariate(
  data,
  var,
  unit = NULL,
  facet = FALSE,
  free_x_scale = FALSE,
  free_y_scale = FALSE,
  title = NULL,
  var_label = NULL,
  legend = NULL,
  palette = NULL,
  ...,
  area = NULL
)
```

Arguments

`data`	Data frame containing equally spaced (daily, weekly, monthly) covariate or disease case observations for one or multiple locations.
`var`	Character vector of covariate names with length 2 (x, y). One of them can be a factor.
`unit`	Character, the name of the variable that identifies the different units for which a time series is available, typically a spatial unit, or any other grouping of interest. If specified, results are grouped by this variable. Defaults to NULL (no grouping).
`facet`	If TRUE, plot each grouping in a different facet.

free_x_scale	If TRUE and facet=TRUE, the x-axis scale is free in each facet.
free_y_scale	If TRUE and facet=TRUE, the y-axis scale is free in each facet.
title	Optional title of the plot.
var_label	A 2 character vector with a custom name for the variables.
legend	A character vector with a custom name for the legend.
palette	GHR, RColorBrewer or colorspace palette (e.g. "Purp"). Single R colors in colors() or hex codes can be used when there is no grouping or facets are used. Use "-" before the palette name (e.g., "-Reds") to reverse it. Defaults to a dark green when unit is NULL or when facet is TRUE, otherwise defaults to the "IDE2" palette.
...	Additional arguments to be passed to geom_point or geom_boxplot, e.g. alpha=0.5 and size=2.
area	Deprecated, use unit instead.

Value

A ggplot2 scatterplot or boxplot graph.

Examples

```
# Load data
data("dengue_MS")

# Scatter (two numeric variables) - No grouping
plot_bivariate(dengue_MS, var = c("pop_density", "tmin"), palette = "#d04a2d")

# Scatter (two numeric variables) - Grouping in the same graph
plot_bivariate(
  dengue_MS,
  var = c("pop_density", "tmin"),
  var_label = c("Pop. density", "Min temp."),
  unit = "micro_code"
)

# Scatter  (two numeric variables) - Grouping in facets
plot_bivariate(
  dengue_MS,
  var = c("pop_density", "tmin"),
  var_label = c("Pop. density", "Min temp."),
  unit = "micro_code",
  facet = TRUE,
  free_x_scale = TRUE
)

# Boxplots (one numeric, one categorical) - No grouping
plot_bivariate(
  dengue_MS,
  var = c("pop_density", "biome_name"),
  var_label = c("Pop. density", "Biome"),
```

```
  palette = "royalblue"
)

# Boxplots (one numeric, one categorical) - Grouping
plot_bivariate(
  dengue_MS,
  var = c("biome_name", "tmin"),
  unit = "meso_code",
  palette = "Accent"
)
```

`plot_combine` *Combine plots*

Description

Combines plots, each representing one variable, into a single plot.

Usage

```
plot_combine(
  plot_list,
  combine_legend = FALSE,
  combine_xaxis = FALSE,
  ncol = 1,
  align = "v",
  ...,
  ncol_l = 2,
  nrow_l = NULL,
  rel_widths_l = c(3, 1),
  rel_heights_l = c(1, 1),
  ncol_legend = 1
)
```

Arguments

`plot_list`	A list of plots to be combined.
`combine_legend`	Logical. If TRUE, assumes the legend of all plots is the same as the legend of the first plot in `plot_list` and final plot shows only one instance of the common legend. Default is FALSE.
`combine_xaxis`	Logical. If TRUE, removes x axis labels from all but the last plot. Default is FALSE.
`ncol`	(from cowplot) Number of colums in the plot grid. Default is 1.
`align`	(from cowplot) Specifies how plots should be aligned Options are "none", "hv" (align in both directions), "h", and "v" (default).
`...`	Additional arguments passed to `cowplot::plot_grid`.

ncol_l	When combine_legend = TRUE, number of colums in which to align plots and the common legend. Default is 2.
nrow_l	When combine_legend = TRUE, number of rows in which to align plots and the common legend. Default is NULL.
rel_widths_l	When combine_legend = TRUE, vector of widths in which to align plots and the common legend. Default is c(3, 1).
rel_heights_l	When combine_legend = TRUE, vector of heights in which to align plots and the common legend. Default is c(1, 1).
ncol_legend	When combine_legend = TRUE, number of columns the legend should be distributed in. Default is one column.

Details

This function takes any input from the `cowplot::plot_grid` function to customize the organization of the plots.

Value

A single (cow)plot including the provided multiple plots.

See Also

plot_compare, plot_multiple

Examples

```
# Load data
data("dengue_MS")

# Multiple time series plot
plots <- plot_multiple(
  plot_function = plot_timeseries,
  data = dengue_MS,
  var = c("tmax", "tmin", "pdsi"),
  type = c("cov", "cov", "cov"),
  aggregate_space = "meso_code",
  pop = "population",
  var_label = c("Max Temp", "Min Temp", "PDSI"),
  time = "date",
  unit = "micro_code"
)

# Combine them with a shared legend
plot_combine(
  plot_list = plots,
  ncol = 1,
  align = "v",
  combine_legend = TRUE,
  combine_xaxis = TRUE,
  rel_widths_l = c(7, 1)
)
```

`plot_compare` *Compare plots*

Description

Combines multiple plots of several variables in a single graph.

Usage

```
plot_compare(plot_function, data, var, type, ...)
```

Arguments

`plot_function`	Indicates which of the plot types to use. Options are: plot_timeseries, plot_heatmap, plot_seasonality, and plot_map.
`data`	Data frame containing equally spaced (daily, weekly, monthly) covariate or disease case observations for one or multiple locations.
`var`	Character vector with the name of the variables to be plotted.
`type`	Character vector with the same length of `var` that specifies the types of variable for each element in `var`. Possible values include 'cov' (covariate, default), 'counts' (case counts), and 'inc' (case incidence). If `type='inc'`, `pop` is required.
`...`	Additional arguments for `plot_multiple`, `plot_combine`, and the selected `plot_function`.

Details

This function takes any input arguments from `plot_combine()` and `plot_multiple()` to customize the plots and their organization in a grid.

Value

A single (cow)plot containing plots of several variables.

See Also

`plot_multiple`, `plot_combine`

Examples

```
# Load data
library("sf")
data("dengue_MS")
data("map_MS")

# Comparing time series plots
plot_compare(
  plot_function = plot_timeseries,
```

```
  data = dengue_MS,
  var = c("dengue_cases", "pdsi"),
  type = c("inc", "cov"),
  pop = "population",
  time = "date",
  unit = "micro_code",
  var_label = c("Dengue inc", "PDSI"),
  combine_legend = TRUE,
  ncol_legend = 1,
  ncol = 1,
  align = "h"
)

# Comparing heatmaps plots
plot_compare(
  plot_function = plot_heatmap,
  data = dengue_MS,
  var = c("dengue_cases", "pdsi"),
  type = c("inc", "cov"),
  pop = "population",
  time = "date",
  unit = "micro_code",
  var_label = c("Dengue Cases", "Min Temp"),
  palette = c("Reds", "Blues"),
  ncol_legend = 1,
  combine_xaxis = TRUE
)

# Comparing map plots
plot_compare(
  plot_function = plot_map,
  data = dengue_MS,
  var = c("dengue_cases", "tmax"),
  type = c("inc", "cov"),
  pop = "population",
  time = "date",
  unit = "micro_code",
  var_label = c("Dengue Incidence", "Max Temperature"),
  palette = c("Reds", "Blues"),
  map = map_MS,
  map_unit = "code",
  aggregate_time = "all",
  ncol_legend = 1,
  combine_xaxis = TRUE
)
```

`plot_correlation` *Correlation plot*

Description

Plots a correlation matrix of a series of variables.

Usage

```
plot_correlation(
  data,
  var,
  var_label = NULL,
  method = "pearson",
  plot_type = c("circle", "number"),
  scale = 1,
  title = NULL,
  palette = "IDE1"
)
```

Arguments

data	Data frame containing equally spaced (daily, weekly, monthly) covariate or disease case observations for one or multiple locations.
var	Character vector containing at least two, non-duplicated variables in data to include in the correlation matrix.
var_label	Optional character vector of the same length as var containing custom names for the variables.
method	Correlation computation method. Options include "pearson" (default), "spearman" or "kendall".
plot_type	Character vector of length 2 indicating the type of plot to use in the lower triangular and diagonal (1st element) and the upper triangular (2nd element). Options include "circle", "number" and "raster".
scale	Circle and number size multiplier, e.g. 1.1 increases the size by 10% while 0.9 decreases it by 10%.
title	Optional title of the plot.
palette	GHR, RColorBrewer or colorspace palette. Use "-" before the palette name (e.g., "-Reds") to reverse it.

Value

A plot of the correlation matrix.

Examples

```
# Load data
data("dengue_MS")

# Pearson correlation plot
plot_correlation(
  dengue_MS,
  method = "pearson",
  var = c(
    "dengue_cases",
    "pop_density",
```

```
    "tmax",
    "tmin",
    "pdsi",
    "urban",
    "water_network",
    "water_shortage"
  ),
  var_label = c(
    "dengue cases",
    "pop. density",
    "max temp",
    "min temp",
    "drought index",
    "urbanization",
    "water network",
    "water shortage"
  ),
  title = "Correlation matrix"
)

# Spearman correlation plot of type 'raster' and 'number'
plot_correlation(
  dengue_MS,
  method = "spearman",
  var = c(
    "dengue_cases",
    "pop_density",
    "tmax",
    "tmin",
    "pdsi",
    "urban",
    "water_network",
    "water_shortage"
  ),
  var_label = c(
    "dengue cases",
    "pop. density",
    "max temp",
    "min temp",
    "drought index",
    "urbanization",
    "water network",
    "water shortage"
  ),
  plot_type = c("raster", "number"),
  palette = "-Blue-Red 3"
)
```

`plot_crosscorrelation` *Cross-correlation function (CCF) plot for two time series*

Description

Plots the correlation between one time series and lagged versions of another across a range of lags.

Usage

```
plot_crosscorrelation(
  data,
  var,
  time,
  type = c("cov", "cov"),
  method = "pearson",
  lags = 0:9,
  pop = NULL,
  unit = NULL,
  aggregate_space = NULL,
  aggregate_space_fun = "mean",
  facet = FALSE,
  title = NULL,
  var_label = NULL,
  legend = NULL,
  ylab = NULL,
  xlab = NULL,
  palette = NULL,
  area = NULL
)
```

Arguments

data	Data frame containing equally spaced (daily, weekly, monthly) covariate or disease case observations for one or multiple locations.
var	A character vector of length 2 identifying the variables to be included. The second variable is the one that is lagged.
time	Name of the variable that identifies the temporal dimension of the data frame. Its values must be in date format ("yyyy-mm-dd") representing the day of observation for daily data, the first day of the week for weekly, or the first day of the month for monthly observations.
type	A character vector of length 2 that specifies the types of variable in var. Possible values include 'cov' (covariate, default), 'counts' (case counts), and 'inc' (case incidence). If type='inc', pop is required.
method	Correlation method. Options include "pearson" (default), "spearman" or "kendall".
lags	Numerical vector indicating the lags at which the cross-correlation should be computed. The second variable in 'var' will be lagged.
pop	Character identifying the variable name for population. Only needed if type='inc'.
unit	Name of the variable that identifies the different units for which a time series is available, typically a spatial unit.

aggregate_space
: Name of variable used to define spatial aggregation groups that contain the different units.

aggregate_space_fun
: Character indicating the function to be used in the aggregation over space for `type="cov"`. Options are "mean" (default), "median", "sum". For case counts and incidence, "sum" is always applied.

facet
: If TRUE a separate time series for each space unit is plotted in different facets.

title
: Optional title of the plot.

var_label
: A character vector of length 2 with custom names for 'var' (in the same order).

legend
: Character with a custom name for the legend.

ylab
: Label for the y-axis.

xlab
: Label for the x-axis.

palette
: GHR, RColorBrewer or colorspace palette. Use "-" before the palette name (e.g., "-Reds") to reverse it.

area
: Deprecated, use `unit` instead.

Details

Correlations are computed after deleting missing rows in the lag data so that every correlation is computed with the same observations.

Value

A ggplot2 cross-correlation plot.

Examples

```
# Load data
data("dengue_MS")
data("dengue_SP")

# Cross-correlation (Pearson) between dengue counts and enso (single unit)
plot_crosscorrelation(
  data = dengue_SP,
  var = c("cases", "enso"),
  type = c("counts", "cov"),
  time = "date"
)

# Cross-correlation (Spearman) between dengue inc and tmax for multiple areas
plot_crosscorrelation(
  data = dengue_MS,
  var = c("dengue_cases", "tmax"),
  type = c("inc", "cov"),
  pop = "population",
  time = "date",
  unit = "micro_name",
  method = "spearman"
```

```
)

# Cross-correlation (Kendall) between dengue inc and pdsi for multiple areas
# aggregated by mesoregion, faceted, with custom labels
plot_crosscorrelation(
  data = dengue_MS,
  var = c("dengue_cases", "pdsi"),
  type = c("inc", "cov"),
  pop = "population",
  time = "date",
  unit = "micro_name",
  method = "kendall",
  aggregate_space = "meso_name",
  facet = TRUE,
  title = "Cross-correlation for meso areas"
)
```

`plot_heatmap` *Heatmap plot*

Description

Plots temporal heatmaps of covariates, case counts, or incidence rates.

Usage

```
plot_heatmap(
  data,
  var,
  time,
  type = "cov",
  pop = NULL,
  pt = 1e+05,
  unit = NULL,
  aggregate_space = NULL,
  aggregate_time = "month",
  aggregate_space_fun = "mean",
  aggregate_time_fun = "mean",
  transform = "identity",
  title = NULL,
  var_label = NULL,
  ylab = NULL,
  xlab = NULL,
  palette = NULL,
  centering = NULL,
  area = NULL
)
```

Arguments

data	Data frame containing equally spaced (daily, weekly, monthly) covariate or disease case observations for one or multiple locations.
var	Name of the column identifying the variable to be plotted.
time	Name of the variable that identifies the temporal dimension of the data frame. Its values must be in date format ("yyyy-mm-dd") representing the day of observation for daily data, the first day of the week for weekly, or the first day of the month for monthly observations.
type	Character that specifies the type of variable in `var`. Possible values include 'cov' (covariate, default), 'counts' (case counts), and 'inc' (case incidence). If `type='inc'`, `pop` is required.
pop	Character identifying the variable name for population. Only needed if `type='inc'`.
pt	Numerical only used for `type='inc'`. It represents the scale of the person-time (default 100,000) for incidence rates.
unit	Name of the variable that identifies the different units for which a time series is available, typically a spatial unit.
aggregate_space	Name of variable used to define spatial aggregation groups that contain the different units.
aggregate_time	Temporal scale used to perform temporal aggregation. Options are: "week" (ISO 8601) or "month".
aggregate_space_fun	Character indicating the function to be used in the aggregation over space for `type="cov"`. Options are "mean" (default), "median", "sum". For case counts and incidence, "sum" is always applied.
aggregate_time_fun	Character indicating the function to be used in the aggregation over time for `type="cov"`. Options are "mean" (default), "median", "sum". For case counts and incidence, "sum" is always applied.
transform	Character, defaults to "identity" (i.e., no transformation). Transforms the color ramp for better visualization. Useful options include "log10p1" `log10(x+1)` useful for case counts and incidence with 0s, or any of the in-built ggplot2 options such as "log10" `log10(x)`, "log1p" `log(x+1)`, and "sqrt" `sqrt(x)` (check all possible options using `?scale_y_continuous`).
title	Optional title of the plot.
var_label	Character with a custom name for 'var'.
ylab	Label for the y-axis.
xlab	Label for the x-axis.
palette	GHR, RColorBrewer or colorspace palette. Use "-" before the palette name (e.g., "-Reds") to reverse it.
centering	Numerical or "median", defaults to NULL. If set, it centers the palette on that value.
area	Deprecated, use `unit` instead.

Value

A ggplot2 heatmap plot.

Examples

```
# Load data
data("dengue_MS")

# Covariate heatmap with space aggregation
plot_heatmap(
  dengue_MS,
  var = "tmin",
  time = "date",
  var_label = "Minimum\ntemp.",
  type = "cov",
  unit = "micro_code",
  aggregate_space = "meso_code",
  palette = "Blue-Red"
)

# Case count heatmap with log scale
plot_heatmap(
  dengue_MS,
  var = "dengue_cases",
  time = "date",
  type = "counts",
  unit = "micro_code",
  palette = "Reds",
  title = "Dengue counts",
  var_label = "Dengue \ncounts",
  transform = "log10p1"
)

# Case incidence (for 1,000 persons) heatmap with space aggregation
plot_heatmap(
  dengue_MS,
  var = "dengue_cases",
  time = "date",
  type = "inc",
  pop = "population",
  pt = 1000,
  unit = "micro_code",
  aggregate_space = "meso_code",
  palette = "Purp"
)
```

---

plot_map *Choropleth map*

Description

Plots a choropleth map of covariates, case counts, or incidence rates.

Usage

```
plot_map(
  data,
  var,
  time,
  unit = NULL,
  map,
  type = "cov",
  pop = NULL,
  pt = 1e+05,
  map_unit = NULL,
  aggregate_time = "year",
  aggregate_time_fun = "mean",
  transform = "identity",
  title = NULL,
  var_label = NULL,
  palette = NULL,
  centering = NULL,
  bins = NULL,
  bins_method = "quantile",
  bins_label = NULL,
  coordgrid = FALSE,
  ...,
  area = NULL,
  map_area = NULL,
  by_year = NULL
)
```

Arguments

`data`	Data frame containing equally spaced (daily, weekly, monthly) covariate or case observations for one or multiple locations.
`var`	Name of the column identifying the variable to be plotted.
`time`	Name of the variable that identifies the temporal dimension of the data frame. Its values must be in date format ("yyyy-mm-dd") representing the day of observation for daily data, the first day of the week for weekly, or the first day of the month for monthly observations.
`unit`	Name of the variable that identifies the different units for which a time series is available, typically a spatial unit.
`map`	`sf` object corresponding to the spatial unit specified in 'unit'.
`type`	Character that specifies the type of variable in `var`. Possible values include 'cov' (covariate, default), 'counts' (case counts), and 'inc' (case incidence). If `type='inc'`, `pop` is required.

pop	Character identifying the variable name for population. Only needed if type='inc'.
pt	Scale of the person-time (default 100,000) for incidence rates.
map_unit	Name of the variable that identifies the different units in the map object. If not specified, it assumes the same name as in unit.
aggregate_time	Temporal scale for visualization and aggregation. Options include "all" (across all time points) and "year" (default).
aggregate_time_fun	Character indicating the function to be used in the aggregation over time for type="cov". Options are "mean" (default), "median", "sum". For case counts and incidence, "sum" is always applied.
transform	Character, defaults to "identity" (i.e., no transformation). Transforms the color ramp for better visualization. Useful options include "log10p1" log10(x+1) for case counts and incidence with 0s, or any of the in-built ggplot2 options such as "log10" log10(x), "log1p" log(x+1), and "sqrt" sqrt(x) (check all possible options using ?scale_y_continuous).
title	Optional title of the plot.
var_label	Character with a custom name for the case or covariate variable.
palette	GHR, RColorBrewer or colorspace palette. Use "-" before the palette name (e.g., "-Reds") to reverse it.
centering	Numerical or "median", defaults to NULL. If set, it centers the palette on that value.
bins	Number of bins for categorization of numerical variables. Defaults to NULL (no binning). Can't be used if using centering or transform.
bins_method	Method to compute the bins, only used when bins is not NULL. Possible values are "quantile" (default) and "equal".
bins_label	Optional labels for the bins. They must have the same length as the number of bins. Defaults to NULL (default interval labels).
coordgrid	Boolean indicating whether the coordinate grid should be included in the visualization. Defaults to FALSE.
...	Additional arguments to be passed to geom_sf. Possible values include colour (e.g., colour="black"), linewidth (e.g., linewidth=0.1), linetype (e.g., linetype=2), and alpha (e.g., alpha=0.8).
area	Deprecated, use unit instead.
map_area	Deprecated, use map_unit instead.
by_year	Defunct. Use 'aggregate_time' instead.

Value

A ggplot2 choropleth map.

Examples

```
# Load data
library("sf")
data("dengue_MS")
data("map_MS")

# Temporal average of a covariate
plot_map(
  data = dengue_MS,
  var = "tmin",
  time = "date",
  type = "cov",
  unit = "micro_code",
  map = map_MS,
  map_unit = "code",
  aggregate_time = "all",
  aggregate_time_fun = "mean",
  palette = "Reds",
  var_label = "Min Temp."
)

# Categorical covariate
plot_map(
  data = dengue_MS,
  var = "biome_name",
  time = "date",
  unit = "micro_code",
  aggregate_time = "all",
  map = map_MS,
  map_unit = "code",
  palette = "Viridis",
  var_label = "Biome"
)

# Case counts by year (log)
dengue_MS |>
  plot_map(
    var = "dengue_cases",
    time = "date",
    type = "counts",
    unit = "micro_code",
    pop = "population",
    map = map_MS,
    map_unit = "code",
    palette = "Reds",
    transform = "log10p1"
  )

# Case incidence by year, binned
plot_map(
  dengue_MS,
  var = "dengue_cases",
```

```
  type = "inc",
  time = "date",
  unit = "micro_code",
  pop = "population",
  pt = 1000,
  map = map_MS,
  map_unit = "code",
  bins = 5,
  palette = "Viridis"
)
```

`plot_multiple` *Multiple plot*

Description

Produces a list of multiple plots of the same type, each representing one variable.

Usage

```
plot_multiple(plot_function, ...)
```

Arguments

`plot_function`	Indicates which of the plot types to use. Options are: plot_timeseries, plot_heatmap, plot_seasonality, and plot_map.
`...`	Additional arguments to pass to the plotting function.

Details

Variable names, types, labels, palette and transform can be customized for each plot, the rest of parameters will be the same for all variables (options depend on the chosen plot type).

Value

A list of the different generated plots.

See Also

`plot_compare`, `plot_combine`

Examples

```
# Load data
library("sf")
data("dengue_MS")
data("map_MS")

# Multiple time series plots
```

```
plots <- plot_multiple(
  plot_function = plot_timeseries,
  data = dengue_MS,
  var = c("dengue_cases", "dengue_cases", "tmax"),
  type = c("counts", "inc", "cov"),
  pop = "population",
  var_label = c("Dengue Cases", "Dengue inc", "Max Temp"),
  palette = c("blue", "red", "darkgreen"),
  time = "date",
  unit = "micro_code",
  facet = TRUE
)

# Acess individual plots
print(plots[[1]])

# Multiple heatmap plots
plots <- plot_multiple(
  plot_function = plot_heatmap,
  data = dengue_MS,
  var = c("dengue_cases", "dengue_cases", "tmax"),
  type = c("counts", "inc", "cov"),
  pop = "population",
  var_label = c("Dengue Cases", "Dengue inc", "Max Temp"),
  palette = c("Blues", "Reds", "BrBG"),
  time = "date",
  unit = "micro_code"
)

# Multiple seasonality plots
plots <- plot_multiple(
  plot_function = plot_seasonality,
  data = dengue_MS,
  var = c("dengue_cases", "dengue_cases", "tmax"),
  type = c("counts", "inc", "cov"),
  pop = "population",
  var_label = c("Dengue Cases", "Dengue inc", "Max Temp"),
  palette = c("Blues", "Reds", "BrBG"),
  time = "date",
  unit = "micro_code"
)

# Multiple map plots
plots <- plot_multiple(
  plot_function = plot_map,
  data = dengue_MS,
  var = c("dengue_cases", "dengue_cases", "tmax"),
  type = c("counts", "inc", "cov"),
  pop = "population",
  var_label = c("Dengue Cases", "Dengue inc", "Max Temp"),
  palette = c("Reds", "Blues", "Viridis"),
  map = map_MS,
  map_unit = "code",
```

```
  time = "date",
  unit = "micro_code"
)
```

`plot_seasonality` *Seasonality plot*

Description

Plots yearly time series of covariates, case counts, or incidence rates to explore seasonality patterns.

Usage

```
plot_seasonality(
  data,
  var,
  time,
  type = "cov",
  pop = NULL,
  pt = 1e+05,
  unit = NULL,
  aggregate_space = NULL,
  aggregate_time = "month",
  aggregate_space_fun = "mean",
  aggregate_time_fun = "mean",
  transform = "identity",
  title = NULL,
  var_label = NULL,
  ylab = NULL,
  xlab = NULL,
  free_y_scale = FALSE,
  palette = "IDE1",
  area = NULL
)
```

Arguments

`data`	Data frame containing equally spaced (daily, weekly, monthly) covariate or disease case observations for one or multiple locations.
`var`	Name of the column identifying the variable to be plotted.
`time`	Name of the variable that identifies the temporal dimension of the data frame. Its values must be in date format ("yyyy-mm-dd") representing the day of observation for daily data, the first day of the week for weekly, or the first day of the month for monthly observations.
`type`	Character that specifies the type of variable in `var`. Possible values include 'cov' (covariate, default), 'counts' (case counts), and 'inc' (case incidence). If `type='inc'`, `pop` is required.

pop	Character identifying the variable name for population. Only needed if type='inc'.
pt	Scale of the person-time (default 100,000) for incidence rates.
unit	Name of the variable that identifies the different units for which a time series is available, typically a spatial unit.
aggregate_space	Name of variable used to define spatial aggregation groups that contain the different units.
aggregate_time	Temporal scale used to perform temporal aggregation. Options are: "week" (ISO 8601) or "month".
aggregate_space_fun	Character indicating the function to be used in the aggregation over space for type="cov". Options are "mean" (default), "median", "sum". For case counts and incidence, "sum" is always applied.
aggregate_time_fun	Character indicating the function to be used in the aggregation over time for type="cov". Options are "mean" (default), "median", "sum". For case counts and incidence, "sum" is always applied.
transform	Character, defaults to "identity" (i.e., no transformation). Transforms the y-axis for better visualization. Useful options include "log10p1" log10(x+1) for case counts and incidence with 0s, or any of the in-built ggplot2 options such as "log10" log10(x), "log1p" log(x+1), and "sqrt" sqrt(x) (check all possible options using ?scale_y_continuous).
title	Optional title of the plot.
var_label	Character with a custom name for the case or covariate variable.
ylab	Label for the y-axis.
xlab	Label for the x-axis.
free_y_scale	If TRUE, the y-axis scale is free in each facet.
palette	GHR, RColorBrewer or colorspace palette. Use "-" before the palette name (e.g., "-Reds") to reverse it.
area	Deprecated, use unit instead.

Value

A ggplot2 seasonality plot.

Examples

```
# Load data
data("dengue_MS")

# Seasonality plot of a covariate with space aggregation
plot_seasonality(
  dengue_MS,
  var = "tmax",
  time = "date",
  var_label = "Max temp.",
```

```
  type = "cov",
  unit = "micro_code",
  aggregate_space = "region_code"
)

# Plot case counts (log scale) with space aggregation
plot_seasonality(
  dengue_MS,
  var = "dengue_cases",
  time = "date",
  type = "counts",
  unit = "micro_code",
  aggregate_space = "meso_code",
  transform = "log10p1",
  var_label = "Monthly Dengue Cases",
  xlab = "Month",
  ylab = "Number of cases",
  free_y_scale = TRUE
)

# Seasonality plot of incidence
plot_seasonality(
  dengue_MS,
  var = "dengue_cases",
  time = "date",
  type = "inc",
  pop = "population",
  unit = "micro_code",
  pt = 1000,
  title = "Monthly Dengue Incidence",
  palette = "Reds"
)
```

`plot_timeseries` *Time series plot*

Description

Plots time series of covariates, case counts, or incidence rates.

Usage

```
plot_timeseries(
  data,
  var,
  time,
  type = "cov",
  pop = NULL,
  pt = 1e+05,
```

```
  unit = NULL,
  aggregate_space = NULL,
  aggregate_time = NULL,
  aggregate_space_fun = "mean",
  aggregate_time_fun = "mean",
  facet = FALSE,
  highlight = NULL,
  transform = "identity",
  title = NULL,
  var_label = NULL,
  legend = NULL,
  ylab = NULL,
  xlab = NULL,
  free_y_scale = FALSE,
  palette = NULL,
  area = NULL
)
```

Arguments

`data`	Data frame containing equally spaced (daily, weekly, monthly) covariate or disease case observations for one or multiple locations.
`var`	Name of the column identifying the variable to be plotted.
`time`	Name of the variable that identifies the temporal dimension of the data frame. Its values must be in date format ("yyyy-mm-dd") representing the day of observation for daily data, the first day of the week for weekly, or the first day of the month for monthly observations.
`type`	Character that specifies the type of variable in `var`. Possible values include 'cov' (covariate, default), 'counts' (case counts), and 'inc' (case incidence). If `type='inc'`, `pop` is required.
`pop`	Character identifying the variable name for population. Only needed if `type='inc'`.
`pt`	Numerical only used for `type='inc'`. It represents the scale of the person-time (default 100,000) for incidence rates.
`unit`	Name of the variable that identifies the different units for which a time series is available, typically a spatial unit.
`aggregate_space`	Name of variable used to define spatial aggregation groups that contain the different units.
`aggregate_time`	Temporal scale used to perform temporal aggregation. Options are: "week" (ISO 8601), "month", "year".
`aggregate_space_fun`	Character indicating the function to be used in the aggregation over space for `type="cov"`. Options are "mean" (default), "median", "sum". For case counts and incidence, "sum" is always applied.

`aggregate_time_fun`
: Character indicating the function to be used in the aggregation over time for `type="cov"`. Options are "mean" (default), "median", "sum". For case counts and incidence, "sum" is always applied.

`facet`
: If TRUE a separate time series for each space unit is plotted in different facets.

`highlight`
: ID of the `unit` to be highlighted. Using this option will only color the selected spatial unit and set all the rest to grey.

`transform`
: Character, defaults to "identity" (i.e., no transformation). Transforms the y-axis for better visualization. Useful options include "log10p1" `log10(x+1)` useful for case counts and incidence with 0s, or any of the in-built ggplot2 options such as "log10" `log10(x)`, "log1p" `log(x+1)`, and "sqrt" `sqrt(x)` (check all possible options using `?scale_y_continuous`).

`title`
: Optional title of the plot.

`var_label`
: Character with a custom name for the case or covariate variable.

`legend`
: Character with a custom name for the legend.

`ylab`
: Label for the y-axis.

`xlab`
: Label for the x-axis.

`free_y_scale`
: Logical, default FALSE. Allows different scales in the y_axis when facets are used.

`palette`
: GHR, RColorBrewer or colorspace palette (e.g. "Purp"). Single R colors in `colors()` or hex codes can be used for single time series or facets. Use "-" before the palette name (e.g., "-Reds") to reverse it. Defaults to a dark green when `unit` is NULL, when `facet` is TRUE or when `highlight` is used (i.e. single time series), otherwise defaults to the "IDE2" palette.

`area`
: Deprecated, use `unit` instead.

Value

A ggplot2 time series plot.

See Also

plot_timeseries2 for dual-axis time series plots.

Examples

```
# Load data
data("dengue_MS")

# Plotting a covariate, all units in a single graph
plot_timeseries(
  dengue_MS,
  var = "tmin",
  time = "date",
  type = "cov",
  unit = "micro_code",
  title = "Minimun Temperature"
```

```
)

# Plotting a covariate with space aggregation and different facets
plot_timeseries(
  dengue_MS,
  var = "tmin",
  time = "date",
  type = "cov",
  unit = "micro_code",
  aggregate_space = "meso_code",
  aggregate_space_fun = "mean",
  facet = TRUE,
  var_label = "Minimum Temperature",
  palette = "violetred"
)

# Plotting counts, highlight a single unit
plot_timeseries(
  dengue_MS,
  var = "dengue_cases",
  time = "date",
  type = "counts",
  pop = "population",
  unit = "micro_code",
  title = "Dengue cases",
  highlight = "50001"
)

# Plot disease counts (log scale) with temporal and spatial aggregation
plot_timeseries(
  dengue_MS,
  var = "dengue_cases",
  time = "date",
  type = "counts",
  unit = "micro_code",
  aggregate_space = "meso_code",
  aggregate_time = "year",
  title = "Yearly Cases",
  transform = "log10"
)

# Plot incidence for 1,000 people and log y axis
plot_timeseries(
  dengue_MS,
  var = "dengue_cases",
  time = "date",
  type = "inc",
  pop = "population",
  unit = "micro_code",
  pt = 1000,
  transform = "log10p1"
)
```

plot_timeseries2 *Time series plot of two variables in two different axes*

Description

Plots dual-axis time series of two covariates, case counts, or incidence rates.

Usage

```
plot_timeseries2(
  data,
  var,
  time,
  type = c("cov", "cov"),
  pop = NULL,
  pt = 1e+05,
  unit = NULL,
  aggregate_space = NULL,
  aggregate_time = NULL,
  aggregate_space_fun = "mean",
  aggregate_time_fun = "mean",
  align = "min",
  title = NULL,
  var_label = NULL,
  legend = "Variable",
  ylab = NULL,
  xlab = NULL,
  palette = c("#168c81", "#B98AFB"),
  alpha = 0.9,
  area = NULL
)
```

Arguments

data	Data frame containing equally spaced (daily, weekly, monthly) covariate or disease case observations for one or multiple locations.
var	A character vector of length 2 (left axis, right axis) identifying the variables to be plotted. Neither variable can be constant over the plotted range, as this makes it impossible to compute a secondary axis scale.
time	Name of the variable that identifies the temporal dimension of the data frame. Its values must be in date format ("yyyy-mm-dd") representing the day of observation for daily data, the first day of the week for weekly, or the first day of the month for monthly observations.
type	A character vector of length 2 (left axis, right axis) that specifies the types of variable in var. Possible values include 'cov' (covariate, default), 'counts' (case counts), and 'inc' (case incidence). If type='inc', pop is required.

pop	Character identifying the variable name for population. Only needed if type='inc'.
pt	Numerical only used for type='inc'. It represents the scale of the person-time (default 100,000) for incidence rates.
unit	Name of the variable that identifies the different units for which a time series is available, typically a spatial unit.
aggregate_space	Name of variable used to define spatial aggregation groups that contain the different units.
aggregate_time	Temporal scale used to perform temporal aggregation. Options are: "week" (ISO 8601), "month", "year".
aggregate_space_fun	Character indicating the function to be used in the aggregation over space for type="cov". Options are "mean" (default), "median", "sum". For case counts and incidence, "sum" is always applied.
aggregate_time_fun	Character indicating the function to be used in the aggregation over time for type="cov". Options are "mean" (default), "median", "sum". For case counts and incidence, "sum" is always applied.
align	Options to align the two plots. Defaults to "min", which forces the minimum of the two variables to be aligned. Other options include "mean" and "median".
title	Optional title of the plot.
var_label	A character vector of length 2 (left axis, right axis) with custom names for the case or covariate variable.
legend	Character with a custom name for the legend.
ylab	A character vector of length 2 (left, right) for the y-axes.
xlab	Label for the x-axis.
palette	A character vector of length 2 (left axis, right axis) indicating the colours (R or hex codes) to use for each of the two variables.
alpha	Numerical between 0 and 1 determining the transparency of the lines. Defaults to 0.9.
area	Deprecated, use unit instead.

Value

A dual-axis ggplot2 time series plot.

See Also

plot_timeseries for single-axis time series plots.

Examples

```
# Load data
data("dengue_MS")
data("dengue_SP")
```

```
# Plotting two covariates with temporal aggregation, align using the mean
plot_timeseries2(
  dengue_SP,
  var = c("temp_med", "precip_tot"),
  time = "date",
  align = "mean",
  aggregate_time = "month"
)

# Plotting case incidence and a covariate with temporal aggregation
# and customized colours and labels
plot_timeseries2(
  dengue_SP,
  var = c("cases", "precip_tot"),
  type = c("inc", "cov"),
  var_label = c("Incidence", "Precipitation"),
  title = "Precipitation and dengue incidence in Sao Paulo",
  time = "date",
  pop = "pop",
  aggregate_time = "month",
  palette = c("darkgreen", "royalblue"),
  alpha = 0.8
)

# Plotting case incidence and a covariate with spatial aggregation
plot_timeseries2(
  dengue_MS,
  var = c("dengue_cases", "pdsi"),
  type = c("inc", "cov"),
  pop = "population",
  time = "date",
  unit = "micro_code",
  aggregate_space = "meso_code"
)
```

Index

∗ **datasets**
dengue_MS, 10
dengue_SP, 11
map_MS, 15

aggregate_cases, 2
aggregate_cov, 4
animate_map, 5

check_consecutive, 8

date_to_epiyw, 9
dengue_MS, 10
dengue_SP, 11

epiyw_to_date, 12

get_time_interval, 13
GHR_palette, 13
GHR_palettes *(*GHR_palette*)*, 13

log10p1_breaks_like, 14
log10p1_trans, 15

map_MS, 15

plot_bivariate, 16
plot_combine, 18, *20*, *32*
plot_compare, *19*, 20, *32*
plot_correlation, 21
plot_crosscorrelation, 23
plot_heatmap, *20*, 26, *32*
plot_map, *20*, 28, *32*
plot_multiple, *19, 20*, 32
plot_seasonality, *20*, *32*, 34
plot_timeseries, *20*, *32*, 36, *41*
plot_timeseries2, *38*, 40

scales::new_transform(), *15*