Package ‘GHRexplore’

July 10, 2025

Title Exploratory Analysis of Temporal and Spatio-Temporal Health Data

Version 0.1.1

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://earth.bsc.es/gitlab/ghr/ghrexplore

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

Depends R (>= 4.1.0)

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

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

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

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 2025-07-10 13:10:09 UTC

Contents

`aggregate_cases` *Aggregate cases*

Description

Aggregates a data frame containing disease cases in space and/or time.

Usage

```
aggregate_cases(
  data = NULL,
  cases = NULL,
  pop = NULL,
  time = NULL,
  area = NULL,
  pt = 1e+05,
  aggregate_space = NULL,
  aggregate_time = 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.
`area`	Name of variable that identifies the different locations (e.g., areal units) for which a time series is available.
`pt`	Scale of the person-time (default 100,000) for incidence rates.
`aggregate_space`	Name of variable used to define spatial aggregation groups.
`aggregate_time`	Temporal scale used to perform temporal aggregation. Options are: "week" (ISO 8601), "month", "year".

Value

A data frame with the aggregated cases.

`aggregate_cov` *Aggregate covariates*

Description

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

Usage

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

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.

`area`	Name of variable that identifies the different locations (i.e., areal units) for which a time series is available.
`aggregate_space`	Name of variable used to define spatial aggregation groups.
`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, default is "mean".
`aggregate_time_fun`	Character indicating the function to be used in the aggregation over time, default is "mean".

Value

A data frame with the aggregated covariate.

`dengue_MS` *Dengue cases in* Mato Grosso do Sul

Description

Data frame containing the monthly number of notified dengue cases by municipality in the *Mato Grosso do Sul* state of Brazil, as well as set of spatial and spatio-temporal covariates (e.g., environmental, socio-economic and meteo-climatic factors). This dataset is a subset of the original containing observations over the entire Brazil.

Usage

```
dengue_MS
```

Format

A data frame with 2,640 rows and 27 columns:

micro_code Unique ID number for each micro region (11 units).

micro_name Name of each micro region.

micro_name_ibge Name of each micro region in IBGE format.

meso_code Unique ID number for each meso region (4 units).

meso_name Name of each meso region.

state_code Unique ID number for each state (1 unit).

state_name Name of each state.

region_code Unique ID number given to each Brazilian Region. All observations come from the "Southeast Region".

region_name Name of each Brazilian Region. All observations come from the "Southeast Region".

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, as well as population counts from 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 population living in urban areas (2010 census).

water_network Percentage of population with access to the piped water network according to the 2010 census.

water_shortage Frequency of reported water shortages per microregion between 2000 and 2016.

date First day of the month in date format ("%d-%m-%Y").

Source

`https://github.com/drrachellowe/hydromet_dengue`

`dengue_SP` *Dengue cases in* Sao Paulo

Description

Data frame containing the weekly number of notified dengue cases in the municipality of *Sao Paulo*, as well as a set of climatic covariates.

Usage

```
dengue_SP
```

Format

A data frame with 678 rows and 8 columns:

date First day of the week in date format.

geocode Unique ID code for Sao Paulo microregion.

cases Number of notified dengue cases.

year Year (2010 - 2022).

temp_med Weekly average daily mean temperature.

precip_tot Weekly cumulative precipitation.

enso El Niño-Southern Oscillation index.

pop Number of inhabitants.

Source

`https://info.dengue.mat.br/services/api`

`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 and the attribute 'na_color'.
GHR_palettes() returns a plot with the custom GHR palettes.

Examples

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

GHR_palettes()
```

`map_MS` *Municipality 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

```
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,
  area = NULL,
  facet = FALSE,
  free_x_scale = FALSE,
  free_y_scale = FALSE,
  title = NULL,
  var_label = NULL,
  legend = NULL,
  palette = 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.
`area`	Character, the name of the variable that identifies the different areal units 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 `area` is NULL or when `facet` is TRUE, otherwise defaults to the "IDE2" palette.

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."),
               area = "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."),
               area = "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", "Min temp."),
               palette = "royalblue")

# Boxplots (one numeric, one categorical) - Grouping
plot_bivariate(dengue_MS,
               var = c("biome_name", "tmin"),
               area = "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",
  area = "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', '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")

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

# Comparing seasonality plots
plot_compare(
  plot_function = plot_seasonality,
  data = dengue_MS,
  var = c("dengue_cases", "dengue_cases", "pdsi"),
  type = c("counts", "inc", "cov"),
  pop = "population",
  time = "date",
  area = "micro_code",
  aggregate_space = "region_code",
  pt = 100,
  var_label = c("Dengue Cases", "Dengue inc", "Min Temp"),
  ncol_legend = 1,
  combine_legend = TRUE)

# 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",
    area = "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",
    area = "micro_code",
    var_label= c("Dengue Incidence", "Max Temperature"),
    palette = c("Reds", "Blues"),
    map = map_MS,
    map_area = "code",
    by_year = FALSE,
    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",
  print = FALSE
)
```

Arguments

`data`	Data frame containing equally spaced (daily, weekly, monthly) covariate or disease case observations for one or multiple locations.
`var`	Character vector containing 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 a 10% while 0.9 decreases it a 10%.
`title`	Optional title of the plot.
`palette`	GHR, RColorBrewer or colorspace palette. Use "-" before the palette name (e.g., "-Reds") to reverse it.
`print`	Logical. If TRUE, print the correlation matrix.

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")

# Print spearman correlation plot of type 'raster' and 'number'
# with another palette
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_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,
  area = 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
)
```

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.
`area`	Name of variable that identifies the different locations (i.e., areal units) for which a time series is available.
`aggregate_space`	Name of variable used to define spatial aggregation groups.
`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.
`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 the case or covariate variable.
`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.

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",
```

```
             area = "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",
             area = "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,
             area = "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,
  type = "cov",
  pop = NULL,
  pt = 1e+05,
  area = NULL,
  map = NULL,
  map_area = NULL,
  by_year = TRUE,
  aggregate_time_fun = "mean",
  transform = "identity",
  title = NULL,
```

```
  var_label = NULL,
  palette = NULL,
  centering = NULL,
  bins = NULL,
  bins_method = "quantile",
  bins_label = 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.
`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.
`area`	Name of variable that identifies the different locations (e.g., areal units) for which a time series is available.
`map`	Name of the sf object corresponding to the spatial unit specified in 'area'.
`map_area`	Name of the variable that identifies the different locations (e.g., areal units) in the map object. If not specified, it assumes the same name as in `area`.
`by_year`	Logical, if TRUE a map for each year is produced.
`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).
`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).

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",
         area = "micro_code",
         map = map_MS,
         map_area = "code",
         aggregate_time_fun = "mean",
         palette ="Reds",
         by_year = FALSE,
         var_label= "Min Temp.")

# Categorical covariate
plot_map(data = dengue_MS,
         var = "biome_name",
         time = "date",
         area = "micro_code",
         map = map_MS,
         map_area = "code",
         palette ="Viridis",
         by_year = FALSE,
         var_label= "Biome")

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

```
           transform = "log10p1")

# Case incidence by year, binned
plot_map(dengue_MS,
         var = "dengue_cases",
         type = "inc",
         time = "date",
         area = "micro_code",
         pop = "population",
         pt = 1000,
         map = map_MS,
         map_area = "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 and palette 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")

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",
  area = "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",
  area = "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",
  area = "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_area = "code",
  time = "date",
  area = "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,
  area = 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"
)
```

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.
`area`	Name of variable that identifies the different locations (e.g., areal units) for which a time series is available.
`aggregate_space`	Name of variable used to define spatial aggregation groups.
`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.
`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.

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",
                 area = "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",
                  area = "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",
                 area = "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,
  area = 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
)
```

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.
`area`	Name of variable that identifies the different locations (e.g., areal units) for which a time series is available.
`aggregate_space`	Name of variable used to define spatial aggregation groups.
`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 `area` 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 `area` is NULL, when `facet` is TRUE or when `highlight` is used (i.e. single time series), otherwise defaults to the "IDE2" palette.

Value

A ggplot2 time series plot.

Examples

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

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

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

# Plotting counts, highlight a single area
plot_timeseries(dengue_MS,
```

```
                var = "dengue_cases",
                time = "date",
                type = "counts",
                pop = "population",
                area = "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",
                area = "micro_code",
                aggregate_space = "meso_code",
                aggregate_time = "year",
                title = "Yearly Cases",
                transform = "log10")

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

Index