Package: insetplot (via r-universe)

June 2, 2026

Title Inset Plots for Spatial Data Visualization

Version 1.4.0

Description Tools for easily and flexibly creating 'ggplot2' maps with inset maps. One crucial feature of maps is that they have fixed coordinate ratios, i.e., they cannot be distorted, which makes it difficult to manually place inset maps. This package provides functions to automatically position inset maps based on user-defined parameters, making it extremely easy to create maps with inset maps with minimal code.

License GPL (>= 2)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports sf, ggplot2, patchwork,

Suggests testthat, knitr, rmarkdown, cowplot

VignetteBuilder knitr

BugReports https://github.com/fncokg/insetplot/issues

URL https://fncokg.github.io/insetplot/

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository https://fncokg.r-universe.dev

Date/Publication 2026-02-02 01:50:45 UTC

RemoteUrl https://github.com/fncokg/insetplot

RemoteRef HEAD

RemoteSha b646d2087103200a10560c32deac77af50626daf

Contents

`insetplot-package` *insetplot: Compose ggplot2 maps with insets*

Description

insetplot lets you create ggplot2 maps with inset maps easily and flexibly. It handles spatial configuration, aspect ratios, and plot composition automatically.

Core workflow

1. Build a configuration with `config_insetmap` and `inset_spec` by specifying necessary parameters (position and size).
2. Pass your `ggplot` object to `with_inset` to generate the composed figure.
3. Save the final plot with `ggsave_inset` to maintain correct aspect ratio.

Main functions

- `inset_spec`: Define bbox, position (`loc` or `loc_left`/`loc_bottom`), and size (prefer `scale_factor`; or provide one of `width`/`height`).
- `config_insetmap`: Create and store the configuration.
- `with_inset`: Crop each subplot, compose subplots and calculate sizes and positions automatically.
- `ggsave_inset`: Save with the correct aspect ratio derived from `with_inset`, with optional `ratio_scale` for fine-tuning.

Example

```
library(sf)
library(ggplot2)
library(insetplot)

nc <- st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)
```

```
# Approach 1: shared base plot for all subplots
config_insetmap(
  bbox = st_bbox(nc),
  specs = list(
    inset_spec(main = TRUE),
    inset_spec(
      xmin = -84, xmax = -75, ymin = 33, ymax = 37,
      loc = "left bottom", scale_factor = 0.5
    )
  )
)
base_map <- ggplot(nc, aes(fill = AREA)) +
  geom_sf() +
  scale_fill_viridis_c() +
  guides(fill = "none") +
  theme_void()
p <- with_inset(base_map)

# Approach 2: provide custom plots in each spec
config_insetmap(
  bbox = st_bbox(nc),
  specs = list(
    inset_spec(main = TRUE, plot = base_map),
    inset_spec(
      xmin = -84, xmax = -75, ymin = 33, ymax = 37,
      loc = "left bottom", scale_factor = 0.5,
      plot = base_map + ggtitle("Detail")
    )
  )
)
p <- with_inset()  # plot argument is optional here

# Save with the correct aspect ratio
ggsave_inset("map.png", p, width = 10)
```

Author(s)

Maintainer: Chao Kong <kongchao1998@gmail.com> (ORCID) [copyright holder]

See Also

inset_spec, config_insetmap, with_inset, ggsave_inset, map_border, last_insetcfg

---

config_insetmap *Configure inset map settings*

Description

Create and store an inset configuration used by `with_inset()`. The configuration contains subplot specifications, aspect ratio of the main plot, CRS settings, and border appearance for insets.

Usage

```
config_insetmap(
  specs,
  to_crs = sf::st_crs("EPSG:4326"),
  from_crs = sf::st_crs("EPSG:4326"),
  bbox = NULL,
  lims_method = "cross",
  border_args = list()
)
```

Arguments

`specs`	A non-empty list of `inset_spec()` objects.
`to_crs`	Coordinate reference system to transform to, passed to `ggplot2::coord_sf()` as `crs`. Default `"EPSG:4326"`.
`from_crs`	Coordinate reference system of bboxes in `specs`. Default `"EPSG:4326"`.
`bbox`	An optional bounding box (compatible with `sf::st_bbox()`) to define the default extent for subplots that do not specify their own coordinates. If NULL, subplots must specify their own valid dimensions/coordinates.
`lims_method`	Method to calculate limits from bbox. See also `ggplot2::coord_sf()`. Default `"cross"`.
`border_args`	A list of named arguments passed to `map_border()` to style the borders around inset plots. See `map_border()` for details (defaults: `color = "black"`, `linewidth = 1`).

Value

An object of class `insetcfg`. Also stored as the last configuration, retrievable via `last_insetcfg()`.

See Also

`inset_spec()`, `with_inset()`, `last_insetcfg()`

Examples

```
library(sf)

nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)

config_insetmap(
    bbox = sf::st_bbox(nc),
    specs = list(
        inset_spec(main = TRUE),
```

```
            inset_spec(
                xmin = -84, xmax = -75, ymin = 33, ymax = 37,
                loc = "left bottom", scale_factor = 0.5
            )
        )
    )
```

`get_bbox_features` *Extract width, height, and aspect ratio from a bounding box*

Description

Computes spatial range and aspect ratio metrics from a bounding box.

Usage

```
get_bbox_features(bbox)
```

Arguments

`bbox`	A named numeric vector with elements `xmin`, `xmax`, `ymin`, `ymax`.

Value

A list with elements:

`x_range`	Width (xmax - xmin) of the bounding box
`y_range`	Height (ymax - ymin) of the bounding box
`xy_ratio`	Aspect ratio (x_range / y_range)

Examples

```
# Create a sample bounding box
bbox <- c(xmin = -84, xmax = -75, ymin = 33, ymax = 37)

# Extract width, height, and aspect ratio
features <- get_bbox_features(bbox)
features

# Access individual components
features$x_range
features$y_range
features$xy_ratio
```

get_widest_bbox *Compute the union bounding box from multiple shapes*

Description

Calculates the overall bounding box that encompasses all provided spatial shapes.

Usage

```
get_widest_bbox(shapes)
```

Arguments

shapes	A list of sf objects.

Value

A named numeric vector with elements: `ymin`, `xmin`, `xmax`, `ymax` representing the union of all input bounding boxes.

Examples

```
library(sf)

# Load sample data
nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)

# Get the bounding box of the entire dataset
bbox <- get_widest_bbox(list(nc))
bbox
```

ggsave_inset *Save a composed inset plot with appropriate dimensions*

Description

A wrapper around `ggplot2::ggsave()` that automatically calculates the output dimensions based on the full ratio defined in the inset configuration. This ensures the saved image maintains the correct aspect ratio for proper rendering of all subplots.

Usage

```
ggsave_inset(
  filename,
  plot = last_plot(),
  device = NULL,
  path = NULL,
  scale = 1,
  width = NA,
  height = NA,
  ...,
  ratio_scale = 1,
  .cfg = last_insetcfg()
)
```

Arguments

filename	Filename to save the plot to. Passed directly to ggplot2::ggsave().
plot	The plot to save. Default ggplot2::last_plot().
device	Device to save to (e.g., "png", "pdf"). Default NULL (inferred from filename).
path	Directory path for saving. Default NULL (current directory).
scale	Scaling factor. Default 1.
width, height	Width and height in inches. You only need to provide one; the other will be calculated automatically. Default NA.
...	Additional arguments passed to ggplot2::ggsave().
ratio_scale	Optional scaling factor to adjust the aspect ratio. Default 1.0. Use when there are extra elements (e.g., titles, legends) that affect the overall image dimensions. For example, set to 1.1 for extra width when a legend is present on the left/right side.
.cfg	An inset configuration (class insetcfg) created by config_insetmap().

Details

All parameters are the same as ggplot2::ggsave(), except that you only need to provide either width or height, and the other dimension will be calculated automatically to match the aspect ratio defined in the inset configuration.

The function automatically calculates width and height based on .cfg$main_ratio to maintain aspect ratio consistency. If both width and height are provided, a warning is issued as the output aspect ratio may not match the configuration.

Value

NULL (invisibly). Saves the plot to disk.

See Also

with_inset()

Examples

```
library(sf)
library(ggplot2)

nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)

config_insetmap(
    bbox = st_bbox(nc),
    specs = list(
        inset_spec(main = TRUE),
        inset_spec(
            xmin = -84, xmax = -75, ymin = 33, ymax = 37,
            loc = "left bottom", width = 0.3
        )
    )
)

base <- ggplot(nc, aes(fill = AREA)) +
    geom_sf() +
    scale_fill_viridis_c() +
    guides(fill = "none") +
    theme_void()
with_inset(base)

# Save with automatically calculated height

ggsave_inset(paste0(tempdir(), "/inset_map.png"), width = 10)
```

---

inset_spec *Create a plot specification for insets*

Description

Define the spatial extent and positioning for each subplot (main or inset).

Usage

```
inset_spec(
  xmin = NA,
  xmax = NA,
  ymin = NA,
  ymax = NA,
  loc = "right bottom",
  loc_left = NA,
  loc_bottom = NA,
  width = NA,
  height = NA,
  scale_factor = NA,
  main = FALSE,
```

```
  plot = NULL
)
```

Arguments

xmin, xmax, ymin, ymax	Numeric bbox coordinates for the subplot in the coordinate system of the data, normally longitude/latitude. Any may be NA and will be inferred from the overall extent if possible.
loc	A convenience string like "left bottom", "center top", etc. to specify the position of the inset on the full canvas. Horizontal position must be one of "left", "center", or "right"; vertical position must be one of "bottom", "center", or "top". Ignored when loc_left and loc_bottom are provided.
loc_left, loc_bottom	Numbers in [0, 1] for the bottom-left position of the inset on the full canvas.
width, height	Numeric values in (0, 1] for the size of the inset. It is recommended to provide only one of these; the other dimension will be inferred to maintain the aspect ratio of the spatial extent. It is also recommended to use scale_factor to automatically size the inset relative to the main plot instead of specifying width/height directly.
scale_factor	Numeric value in (0, Inf) indicating the scale of the inset relative to the main plot. If not NA, the inset's width/height are automatically derived from the spatial ranges relative to the main plot multiplied by this factor. For example, the scale of the main plot is 1:10,000, the inset's dimensions will be 1:20,000 if scale_factor is 0.5.
main	Logical. TRUE marks this spec as the main plot (exactly one). Default FALSE.
plot	Optional ggplot object to use for this spec instead of the base plot passed to with_inset().

Value

A list with elements bbox, loc_left, loc_bottom, width, height, scale_factor, main, plot, hpos, and vpos. You do not normally need to interact with this object directly; it is used internally.

Examples

```
specs <- list(
    # Create a main plot specification
    inset_spec(main = TRUE),
    # Create an inset plot specification with explicit dimensions
    inset_spec(
        xmin = -120, xmax = -100, ymin = 30, ymax = 50,
        loc = "right bottom",
        width = 0.3
    ),
    # Create an inset with scale factor
    inset_spec(
        xmin = -120, xmax = -100, ymin = 30, ymax = 50,
        loc = "left bottom",
```

```
        scale_factor = 0.5
    )
)
```

`last_insetcfg` *Get Last Inset Configuration*

Description

Retrieves the most recently created inset configuration object. This is used internally by `with_inset()` when no configuration is explicitly provided.

Usage

```
last_insetcfg()
```

Value

An inset configuration object of class `insetcfg`, or NULL if no configuration has been set.

Examples

```
library(sf)

# Load some spatial data
nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)

# Configure inset map
config_insetmap(
    bbox = sf::st_bbox(nc),
    specs = list(
        inset_spec(main = TRUE),
        inset_spec(
            xmin = -84, xmax = -75, ymin = 33, ymax = 37,
            loc = "left bottom",
            width = 0.3
        )
    )
)

# Retrieve the configuration
cfg <- last_insetcfg()
```

map_border *Add a border around a map plot*

Description

Returns a small theme that draws a rectangular border around the plot area. Handy for visually separating inset plots from the main plot.

Usage

```
map_border(color = "black", linewidth = 1, fill = "white", ...)
```

Arguments

color	Border color. Default "black".
linewidth	Border line width. Default 1.
fill	Background fill color. Default "white".
...	Passed to ggplot2::element_rect().

Value

A ggplot2 theme object to add to a ggplot with +.

Examples

```
library(ggplot2)

ggplot(mtcars, aes(mpg, wt)) +
    geom_point() +
    map_border(color = "red", linewidth = 2)
```

with_inset *Compose a main plot with inset(s)*

Description

Build a combined plot using an inset configuration created by config_insetmap(). For each plot specification in the configuration, the function either uses the provided spec$plot or the supplied plot parameter and adds spatial coordinates via ggplot2::coord_sf() with the given bounding box. Non-main subplots receive a border from map_border(). Insets are composed using patchwork::inset_element().

Usage

```
with_inset(
  plot = NULL,
  .cfg = last_insetcfg(),
  .as_is = FALSE,
  .return_details = FALSE
)
```

Arguments

`plot`	Optional. Either: • A single ggplot object to use as the base plot for all subplots (unless a spec has its own plot) • A list of ggplot objects matching the length of `.cfg$specs`, where each element corresponds to a subplot in the configuration. • NULL if all specs have their own plot defined (plot is fully optional in this case) NOTE: you SHOULD NOT pass `ggplot2::coord_sf()` into this plot manually. The coordinate system is handled internally. Default NULL.
`.cfg`	An inset configuration (class "insetcfg") created by `config_insetmap()`. Defaults to `last_insetcfg()`.
`.as_is`	Logical. If TRUE, return `plot` as-is without creating insets. Useful when debugging or code reuse outside the inset workflow. Default FALSE.
`.return_details`	Logical. If FALSE (default), returns a combined plot with the main plot and inset layers. If TRUE, returns a list. See 'Value' section for details.

Value

If `.return_details = FALSE`, a ggplot object containing the main plot plus inset layers. If TRUE, a list with elements:

`full`	The combined plot
`subplots`	Individual ggplot objects for each subplot
`subplot_layouts`	A `list` of layout information (`x`, `y`, `width`, `height`) for each inset
`main_ratio`	Width-to-height ratio of the main plot's data extent

See Also

`config_insetmap()`

Examples

```
library(sf)
library(ggplot2)

nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)

config_insetmap(
    bbox = st_bbox(nc),
    specs = list(
        inset_spec(main = TRUE),
        inset_spec(
            xmin = -82, xmax = -80.5, ymin = 35.5, ymax = 36,
            loc = "left bottom", scale_factor = 2
        )
    )
)

# Supply base plot for all subplots
base <- ggplot(nc, aes(fill = AREA)) +
    geom_sf() +
    scale_fill_viridis_c() +
    guides(fill = "none") +
    theme_void()
with_inset(base)

# Or supply custom plots in each inset_spec, then call with_inset() without plot
config_insetmap(
    bbox = st_bbox(nc),
    specs = list(
        inset_spec(main = TRUE, plot = base),
        inset_spec(
            xmin = -82, xmax = -80.5, ymin = 35.5, ymax = 36,
            loc = "left bottom", scale_factor = 2,
            plot = base # Each spec has its own plot
        )
    )
)
with_inset() # plot parameter is optional now
```

Index