Package 'ggmapinset'

December 18, 2024

Type Package

Title Add Inset Panels to Maps

Version 0.4.0

Description Helper to add insets based on geom_sf() from 'ggplot2'.
This package gives you a drop-in replacement for geom_sf() that supports
adding a zoomed inset map without having to create and embed a separate plot.

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.2

Depends R (>= 3.6), sf (>= 1.0)

Imports cli (>= 3.4.0), ggplot2 (>= 3.4.0), lifecycle, rlang (>=
1.0.0), vctrs (>= 0.6.5)

Suggests dplyr, forcats, ggrepel, knitr, nswgeo, purrr, rmarkdown,
stringr, tibble, tidyr,

VignetteBuilder knitr

URL https://github.com/cidm-ph/ggmapinset,
https://cidm-ph.github.io/ggmapinset/

BugReports https://github.com/cidm-ph/ggmapinset/issues

LazyData true

NeedsCompilation no

Author Carl Suster [aut, cre] (<https://orcid.org/0000-0001-7021-9380>),
Western Sydney Local Health District, NSW Health [cph]

Maintainer Carl Suster <Carl.Suster@health.nsw.gov.au>

Repository CRAN

Date/Publication 2024-12-18 06:40:02 UTC

Contents

`ggmapinset-package` *Add Inset Panels to Maps*

Description

This package helps with making zoomed map insets. See `geom_sf_inset()`.

Author(s)

Maintainer: Carl Suster <Carl.Suster@health.nsw.gov.au> (ORCID)

Other contributors:

- Western Sydney Local Health District, NSW Health [copyright holder]

See Also

Useful links:

- https://github.com/cidm-ph/ggmapinset
- https://cidm-ph.github.io/ggmapinset/
- Report bugs at https://github.com/cidm-ph/ggmapinset/issues

`build_sf_inset_layers` *Build layers to implement an inset-compatible geometry*

Description

For plotting, use `geom_sf_inset()` instead. This helper is intended to be used when implementing custom geometries based on `geom_sf_inset()` so that they can provide parameters to control the inset.

Usage

```
build_sf_inset_layers(
  data,
  mapping,
  stat,
  position,
  show.legend,
  inherit.aes,
  params,
  inset,
  map_base = "normal",
  map_inset = "auto"
)
```

Arguments

`data`, `mapping`, `stat`, `position`, `show.legend`, `inherit.aes`, `params`	See `ggplot2::layer()`.
`inset`	Inset configuration; see `configure_inset()`. If `NA` (the default), this is inherited from the coord (see `coord_sf_inset()`).
`map_base`	Controls the layer with the base map. Possible values are `"normal"` to create a layer as though the inset were not specified, `"clip"` to create a layer with the inset viewport cut out, and `"none"` to prevent the insertion of a layer for the base map.
`map_inset`	Controls the layer with the inset map. Possible values are `"auto"` to choose the behaviour based on whether `inset` is specified, `"normal"` to create a layer with the viewport cut out and transformed, and `"none"` to prevent the insertion of a layer for the viewport map.

Value

A `ggplot` layer, or a pair of layers.

Examples

```
my_custom_geom <- function(
  mapping = ggplot2::aes(),
  data = NULL,
  stat = "my_custom_stat",
  position = "identity",
  ...,
  inset = NA,
  map_base = "normal",
  map_inset = "auto",
  na.rm = TRUE,
  inherit.aes = TRUE
) {
  params <- rlang::list2(na.rm = na.rm, ...)
  build_sf_inset_layers(
    data = data, mapping = mapping,
    stat = stat, position = position,
    show.legend = show.legend,
    inherit.aes = inherit.aes,
    params = params,
    inset = inset,
    map_base = map_base,
    map_inset = map_inset
  )
}
```

---

configure_inset *Configure transformations underpinning a map inset*

Description

The configuration returned by this function will often be passed to the coordinate system via `coord_sf_inset()` so that it is propagated to all relevant layers.

Usage

```
configure_inset(
  shape,
  scale = NULL,
  translation = NULL,
  units = "km",
  crs_working = NULL,
  centre = deprecated(),
  radius = deprecated()
)
```

Arguments

shape	Inset shape: see shape_circle(), shape_rectangle(), or shape_sf().
scale	Zoom scale: values larger than one will make the inset bigger.
translation	Translation (shift) of the inset relative to the centre. This can be an sf::st_point() or simply a vector of length 2 containing the x and y offsets respectively. Units are specified by crs_working.
units	Base length unit (e.g. "km" or "mi"). Ignored if crs_working is provided. See Details for supported values.
crs_working	The coordinate reference system to use internally when applying the transformations. See Details.
centre, radius	**[Deprecated]** Use shape = shape_circle(centre, radius) instead.

Details

The default crs_working uses the equidistant cylindrical coordinate reference system with the latitude of true scale set to match the latitude of centre. This ensures that circular insets will appear circular in most cases since the projection is not distorted near the centre. The geometries are converted to this CRS for the inset transformation and constructing the inset frame, and are converted back to the CRS of centre at the end.

The default units are kilometres but can be changed with units instead of specifying the whole projection. The possible values for units are those understood by proj:

- "mm": millimetre
- "cm": centimetre
- "m": metre
- "ft": foot
- "us-ft": US survey foot
- "fath": fathom
- "kmi": nautical mile
- "us-ch": US survey chain
- "us-mi": US survey mile
- "km": kilometre
- "ind-ft": Indian foot (1937)
- "ind-yd": Indian yard (1937)
- "mi": Statute mile
- "yd": yard
- "ch": chain
- "link": link
- "dm": decimeter
- "in": inch
- "ind-ch": Indian chain
- "us-in": US survey inch
- "us-yd": US survey yard

Value

An inset configuration object of class `inset_config`.

Examples

```
library(sf)

# circular inset with a 2x enlargement
cfg <- configure_inset(
  shape_circle(centre = c(-82, 35), radius = 50),
  scale = 2,
  translation = c(70, -180),
  units = "mi"
)
```

`coord_sf_inset` *Specify an inset configuration for the whole plot*

Description

This allows a default inset configuration to be provided to avoid having to repeat it for each layer. Any layer that is inset-aware can use this as the default configuration if none is specifically provided to that layer. This coord also expands the axis limits to include the inset area.

Usage

```
coord_sf_inset(inset, ...)
```

Arguments

`inset`	Inset configuration; see `configure_inset()`.
`...`	Arguments passed to `ggplot2::coord_sf()`

Value

A ggplot coordinate object to be added to a plot.

See Also

`geom_sf_inset()`

Examples

```
library(ggplot2)

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

ggplot(nc) +
  geom_sf_inset(aes(fill = AREA)) +
  geom_inset_frame() +
  coord_sf_inset(inset = configure_inset(
    centre = sf::st_sfc(sf::st_point(c(-80, 35.5)), crs = 4326),
    scale = 1.5, translation = c(-50, -140), radius = 50, units = "mi"))
```

`geom_inset_frame` *Add a frame and burst lines for an inset*

Description

The frame is computed from the inset configuration, so any `data` passed to this layer is ignored. The frame is an sf object consisting of three features: the source area, the target area (a scaled and translated version of the source area), and the connecting/burst lines.

Usage

```
geom_inset_frame(
  mapping = ggplot2::aes(),
  data = NULL,
  stat = "sf_inset",
  position = "identity",
  ...,
  inset = NA,
  na.rm = FALSE,
  source.aes = list(),
  target.aes = list(),
  lines.aes = list(),
  show.legend = NA,
  inherit.aes = FALSE
)
```

Arguments

`mapping`, `data`, `stat`, `position`, `na.rm`, `show.legend`, `inherit.aes`, `...`
: See `ggplot2::geom_sf()`.

`inset`
: Inset configuration; see `configure_inset()`. If `NA` (the default), this is inherited from the coord (see `coord_sf_inset()`).

`source.aes`, `target.aes`, `lines.aes`
: Override the aesthetics of the inset source, target, and lines respectively. The value should be a list named by the aesthetics, and the values should be scalars of length one.

Details

Burst lines for circular insets are bitangenets (tangent to both the source and target circles) or absent if the circles are nested. Burst lines for rectangular insets are the shortest line from each corner of the source rectangle to any corner of the target rectangle, after excluding any such lines that intersect either rectangle or each other. When the burst lines are absent due to geometrical constraints, there will still be a corresponding (empty) feature in the frame layer's data.

Value

A ggplot layer holding the inset frame.

Limitation

The frame cannot be drawn without another sf layer that contains data due to a limitation of the ggplot layout evaluation. Attempting to plot a frame by itself will result in the error: "Scale limits cannot be mapped onto spatial coordinates in `coord_sf()`".

Examples

```
library(ggplot2)

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

ggplot(nc) +
  geom_sf_inset() +
  geom_inset_frame(
    source.aes = list(fill = "red", alpha = 0.2, linewidth = 0),
    target.aes = list(colour = "blue"),
    lines.aes = list(linetype = 2, linewidth = 2)
  ) +
  coord_sf_inset(inset = configure_inset(
    shape_circle(centre = c(-82, 35), radius = 50),
    scale = 5, translation = c(0, -260), units = "mi"
  ))
```

---

geom_sf_inset *Visualise sf objects with insets*

Description

These geoms are wrappers around `ggplot2::geom_sf()` and its relatives that assist with creating map insets. In many cases all that is needed is to use `coord_sf_inset()` with `configure_inset()` to configure the location and transformation of the inset, and then replace the sf-related geoms with their `_inset` counterparts. Use `geom_inset_frame()` to add a frame around the inset that connects it to the main map.

Usage

```
geom_sf_inset(
  mapping = ggplot2::aes(),
  data = NULL,
  stat = "sf_inset",
  position = "identity",
  ...,
  inset = NA,
  map_base = "normal",
  map_inset = "auto",
  na.rm = TRUE,
  show.legend = NA,
  inherit.aes = TRUE
)

stat_sf_inset(
  mapping = ggplot2::aes(),
  data = NULL,
  geom = "sf_inset",
  position = "identity",
  ...,
  inset = NA,
  na.rm = TRUE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

`mapping`, `data`, `stat`, `geom`, `position`, `na.rm`, `show.legend`, `inherit.aes`, `...`	See `ggplot2::geom_sf()`.
`inset`	Inset configuration; see `configure_inset()`. If `NA` (the default), this is inherited from the coord (see `coord_sf_inset()`).
`map_base`	Controls the layer with the base map. Possible values are `"normal"` to create a layer as though the inset were not specified, `"clip"` to create a layer with the inset viewport cut out, and `"none"` to prevent the insertion of a layer for the base map.
`map_inset`	Controls the layer with the inset map. Possible values are `"auto"` to choose the behaviour based on whether `inset` is specified, `"normal"` to create a layer with the viewport cut out and transformed, and `"none"` to prevent the insertion of a layer for the viewport map.

Details

Internally this works by creating two layers: one for the base map, and one for the inset. These can be separately controlled by the `map_base` and `map_inset` parameters. If `inset` is not specified, this geom will instead behave like `ggplot2::geom_sf()`.

When an inset is configured, the default creates both base and inset layers using the same aesthetic mapping and params:

```
geom_sf_inset(...)
```

You can alternatively specify the two layers separately:

```
# draw the base map only (both versions are equivalent):
geom_sf(...)
geom_sf_inset(..., map_inset = "none")

# separately, draw the inset map only:
geom_sf_inset(..., map_base = "none")
```

`stat_sf_inset()` works the same `ggplot2::stat_sf()` except that it also expands the axis limits to account for the inset area.

Value

A ggplot layer similar to `ggplot2::geom_sf()` but transformed according to the inset configuration.

Examples

```
library(ggplot2)

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

ggplot(nc) +
  geom_sf_inset(aes(fill = AREA)) +
  geom_inset_frame() +
  coord_sf_inset(configure_inset(
    shape_circle(
      centre = sf::st_sfc(sf::st_point(c(-80, 35.5)), crs = sf::st_crs(nc)),
      radius = 50
    ),
    scale = 1.5, translation = c(-50, -140), units = "mi"
  ))
```

`geom_sf_text_inset` *Extract coordinates from 'sf' objects (inset-aware)*

Description

Reduce spatial data to coordinates in the same way as `stat_sf_coordinates()`. The result can then be used by `geom_sf()` or `geom_sf_inset()` or any geom that needs `x` and `y` aesthetics. As this is particularly useful for text, wrappers are provided for `geom_text()` and `geom_label()`.

Usage

```
geom_sf_text_inset(
  mapping = aes(),
  data = NULL,
  stat = "sf_coordinates_inset",
  position = "identity",
  ...,
  where = "inset",
  parse = FALSE,
  check_overlap = FALSE,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  fun.geometry = NULL
)

geom_sf_label_inset(
  mapping = aes(),
  data = NULL,
  stat = "sf_coordinates_inset",
  position = "identity",
  ...,
  where = "inset",
  parse = FALSE,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  fun.geometry = NULL
)

stat_sf_coordinates_inset(
  mapping = ggplot2::aes(),
  data = NULL,
  geom = "point",
  position = "identity",
  ...,
  inset = NA,
  fun.geometry = NULL,
  where = "inset",
  na.rm = TRUE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

mapping, data, geom, position, na.rm, show.legend, inherit.aes, fun.geometry, ...	See ggplot2::stat_sf_coordinates().
stat, parse, check_overlap	See ggplot2::geom_sf_text().
where	Specifies how the text position interacts with the inset. "inset" means that any points in the inset area are drawn on the inset map, "base" puts them on the base map. This setting is merely a shorthand for setting the position aesthetics to after_stat(x_inset) or after_stat(x) respectively, so will have no effect if these are specified in the mapping.
inset	Inset configuration; see configure_inset(). If NA (the default), this is inherited from the coord (see coord_sf_inset()).

Value

A plot layer

Required aesthetics

geometry The sf geometry column containing spatial features

Computed variables

x X dimension of the simple feature

y Y dimension of the simple feature

x_inset X dimension of the simple feature after inset transformation

y_inset Y dimension of the simple feature after inset transformation

inside_inset logical indicating points inside the inset viewport

inset_scale 1 for points outside the inset, otherwise the configured inset scale parameter

Examples

```
library(ggplot2)

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

ggplot(nc) +
  geom_sf_inset() +
  geom_inset_frame() +
  geom_sf_text(
    aes(x = after_stat(x_inset), y = after_stat(y_inset), label = NAME),
    stat = "sf_coordinates_inset") +
  coord_sf_inset(configure_inset(
    shape_circle(
      centre = sf::st_sfc(sf::st_point(c(-80, 35.5)), crs = 4326),
      radius = 50
    ),
```

```
  scale = 1.5, translation = c(-50, -140), units = "mi"
))
```

`get_inset_config` *Get the inset configuration from the params or coord*

Description

This is a helper for implementing inset-aware ggplot layers. If the `inset` is missing (`NA`) then the default inset configuration is retrieved from the coord.

Usage

```
get_inset_config(inset, coord)
```

Arguments

`inset`	Inset passed in as a param to the layer
`coord`	Coord object for the plot

Value

Inset configuration or `NULL`

Examples

```
# defining a new geom deriving from geom_sf()
GeomCustom <- ggplot2::ggproto("GeomCustom", ggplot2::GeomSf,
  draw_panel = function(self, data, panel_params, coord, inset = NA) {
    inset <- get_inset_config(inset, coord)

    # do something with the inset ...

    # note that this example doesn't pass on the remaining geom_sf params but
    # in real usage you would probably want to do that
    ggplot2::ggproto_parent(ggplot2::GeomSf, self)$draw_panel(data, panel_params, coord)
  },
)
```

`mozzies_nsw2301` *Mosquito counts from NSW Arbovirus Surveillance program*

Description

This dataset is derived from the NSW Arbovirus Surveillance and Mosquito Monitoring program. The program monitors mosquito-borne diseases in the state of New South Wales, Australia. A number of mosquito traps are managed by the program during the spring to autumn months when mosquitoes are active.

Usage

```
mozzies_nsw2301
```

Format

Data frame with the following fields:

location Location of the mosquito trap

week_ending Date of the end of the week of observation

species Mosquito species counted, or "total" for the total count

count Binned mosquito abundance

type Category of the site

lat Latitude of trap in WGS 84 coordinates

long Longitude of trap in WGS 84 coordinates

Details

Each week traps are collected and the mosquito species are identified and counted. This is analysed alongside climate conditions, and arbovirus detections in the traps to inform public health management of human disease risk from arboviruses in NSW. This dataset includes the mosquito abundance tables for January 2023. Additional context and analysis can be found in the original report published by NSW Health.

The trap locations are classified as inland or coastal (since the species found will depend on the environmental conditions). A separate group of sites are labelled as being in the Sydney region (i.e. with the highest human population density).

The counts are binned with the following definition:

`NA` No observation

low < 50

medium 50 - 100

high 101 - 1,000

very high 1,001 - 10,000

extreme > 10,000

Source

Surveillance and Risk Unit, Environmental Health Branch, Health Protection NSW, NSW Health. "NSW Arbovirus Surveillance and Mosquito Monitoring 2022-2023; Weekly Update: Week ending 25 February 2023 (Report Number 19)" https://www.health.nsw.gov.au/environment/pests/vector/Publications/nswasp-weekly-report-2023-02-25.pdf, accessed 15 January 2024.

`shape_circle` *Circular insets*

Description

Circular insets

Usage

```
shape_circle(centre, radius)
```

Arguments

`centre`	Coordinates of the inset centre. Ideally this should be an `sfc` object (see `sf::st_sfc()`) including a coordinate reference system. An `sf::st_point()` or a vector of longitude and latitude are also accepted. If a CRS cannot be determined, WGS 84 is assumed.
`radius`	Radius of the inset circle in the units of the inset's `crs_working`.

Value

A shape definition suitable for use with `configure_inset()`.

See Also

`configure_inset()`

Other shapes: `shape_rectangle()`, `shape_sf()`

Examples

```
library(ggplot2)
nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)
make_demo <- function(...) {
  ggplot(nc) +
    geom_sf(fill = "grey95", colour = "grey85") +
    # For a filled frame, we want to interleave it between the base layer
    # (above this line) and the target layer (below the following line).
    geom_inset_frame(target.aes = list(fill = "white")) +
```

```
    geom_sf_inset(map_base = "none") +
    coord_sf_inset(inset = configure_inset(...)) +
    theme_void()
}
circle <- shape_circle(sf::st_centroid(nc[21,]), radius = 50)

make_demo(circle, scale = 3, translation = c(-200, -200))
make_demo(circle, scale = 3, translation = c(-100, -100))
make_demo(circle, scale = 3, translation = c(0, 0))
make_demo(circle, scale = 0.5, translation = c(0, 0))
```

`shape_rectangle` *Rectangular insets*

Description

Rectangular insets

Usage

```
shape_rectangle(centre, hwidth, hheight = NULL)
```

Arguments

`centre`	Coordinates of the inset centre. Ideally this should be an `sfc` object (see `sf::st_sfc()`) including a coordinate reference system. An `sf::st_point()` or a vector of longitude and latitude are also accepted. If a CRS cannot be determined, WGS 84 is assumed.
`hwidth`	Half width of the inset in the units of the inset's `crs_working`.
`hheight`	Half height of the inset in the units of the inset's `crs_working`. Defaults to the same value as `hwidth`.

Value

A shape definition suitable for use with `configure_inset()`.

See Also

`configure_inset()`

Other shapes: `shape_circle()`, `shape_sf()`

Examples

```
library(ggplot2)
nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)
make_demo <- function(...) {
  ggplot(nc) +
    geom_sf(fill = "grey95", colour = "grey85") +
    # For a filled frame, we want to interleave it between the base layer
    # (above this line) and the target layer (below the following line).
    geom_inset_frame(target.aes = list(fill = "white")) +
    geom_sf_inset(map_base = "none") +
    coord_sf_inset(inset = configure_inset(...)) +
    theme_void()
}
rectangle <- shape_rectangle(sf::st_centroid(nc[21,]), hwidth = 50, hheight = 40)

make_demo(rectangle, scale = 3, translation = c(-300, 0))
make_demo(rectangle, scale = 3, translation = c(-250, -200))
make_demo(rectangle, scale = 3, translation = c(-150, -100))
make_demo(rectangle, scale = 3, translation = c(0, 0))
make_demo(rectangle, scale = 0.5, translation = c(0, 0))
```

---

shape_sf *Arbitrary geometry as insets*

Description

You can use any polygon to define the shape of the inset, including a feature from your base map layer.

Usage

```
shape_sf(geometry)
```

Arguments

geometry	A simple features geometry that is either a polygon or multipolygon, and is valid and simple.

Value

A shape definition suitable for use with `configure_inset()`.

See Also

`configure_inset()`

Other shapes: `shape_circle()`, `shape_rectangle()`

Examples

```
library(ggplot2)
nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)
make_demo <- function(...) {
  ggplot(nc) +
    geom_sf(fill = "grey95", colour = "grey85") +
    # For a filled frame, we want to interleave it between the base layer
    # (above this line) and the target layer (below the following line).
    geom_inset_frame(target.aes = list(fill = "white")) +
    geom_sf_inset(map_base = "none", colour = NA) +
    coord_sf_inset(inset = configure_inset(...)) +
    theme_void()
}
shape <- shape_sf(nc[21,])

make_demo(shape, scale = 6, translation = c(-200, -200))
make_demo(shape, scale = 6, translation = c(-100, -100))
make_demo(shape, scale = 6, translation = c(100, 100))
make_demo(shape, scale = 0.5, translation = c(0, 0))
```

---

transform_to_inset *Transform coordinates according to inset configuration*

Description

This helper operates on an sf object to scale and translate its geometry according to the inset specification.

Usage

```
transform_to_inset(x, inset)
```

Arguments

x	Spatial data frame or other sf object; see sf::st_geometry().
inset	Inset configuration; see configure_inset().

Value

A copy of x with the geometry replaced by the transformed version.

Examples

```
library(sf)

nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)
cfg <- configure_inset(
  centre = st_sfc(st_point(c(-82, 35)), crs = 4326),
  scale = 2,
```

```
  translation = c(10, -60),
  radius = 50,
  units = "mi")

transform_to_inset(nc, cfg)
```

Index