Package: ggmapinset (via r-universe)

September 12, 2024

Type Package

Title Add Inset Panels to Maps

Version 0.3.0.9000

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 7.3.0

Roxygen list(markdown = TRUE)

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

Imports cli (>= 3.4.0), ggplot2 (>= 3.4.0), rlang (>= 1.0.0)

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

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

Repository https://cidm-ph.r-universe.dev

RemoteUrl https://github.com/cidm-ph/ggmapinset

RemoteRef HEAD

RemoteSha fd6ddb8d2d0cc1037a2dff84d98a7d212a753729

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	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot(). A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).
mapping	Set of aesthetic mappings created by aes(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
stat	The statistical transformation to use on the data for this layer, either as a ggproto Geom subclass or as a string naming the stat stripped of the stat_ prefix (e.g. "count" rather than "stat_count")
position	Position adjustment, either as a string naming the adjustment (e.g. "jitter" to use position_jitter), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.

show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders().
params	Additional parameters to the geom and stat.
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 normally be passed to the coordinate system via coord_sf_inset(). Currently only circular insets are supported, and only one inset per plot.

Usage

```
configure_inset(
  centre,
  scale = NULL,
  translation = NULL,
  radius = NULL,
  units = "km",
  crs_working = 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, crs_working is assumed.
scale	Zoom scale: values larger than one will make the circle bigger.
translation	Translation (shift) of the inset relative to the centre. This can be an st_point or simply a vector of length 2 containing the x and y offsets respectively. Units are specified by crs_working.
radius	Radius of the inset circle in the units of 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.

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 the circle 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(
  centre = st_sfc(st_point(c(-82, 35)), crs = 4326),
  scale = 2,
  translation = c(70, -180),
  radius = 50,
  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

Add a frame and burst lines for an inset

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.

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() +
  coord_sf_inset(inset = configure_inset(
    centre = sf::st_sfc(sf::st_point(c(-82, 35)), crs = 4326),
    scale = 2, translation = c(0, -300), radius = 50, 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
)

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_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.

`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.

`parse` If `TRUE`, the labels will be parsed into expressions and displayed as described in `?plotmath`.

`check_overlap` If `TRUE`, text that overlaps previous text in the same layer will not be plotted. `check_overlap` happens at draw time and in the order of the data. Therefore data should be arranged by the label column before calling `geom_text()`. Note that this argument is not supported by `geom_label()`.

`fun.geometry` A function that takes a `sfc` object and returns a `sfc_POINT` with the same length as the input. If `NULL`, `function(x) sf::st_point_on_surface(sf::st_zm(x))` will be used. Note that the function may warn about the incorrectness of the result if the data is not projected, but you can ignore this except when you really care about the exact locations.

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(inset = configure_inset(
    centre = sf::st_sfc(sf::st_point(c(-80, 35.5)), crs = sf::st_crs(nc)),
    scale = 1.5, translation = c(-50, -140), radius = 50, 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.

---

stat_sf_coordinates_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.

Usage

```
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`, `...`
: See `ggplot2::stat_sf_coordinates()`.

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

`fun.geometry`
: A function that takes a `sfc` object and returns a `sfc_POINT` with the same length as the input. If `NULL`, `function(x) sf::st_point_on_surface(sf::st_zm(x))` will be used. Note that the function may warn about the incorrectness of the result if the data is not projected, but you can ignore this except when you really care about the exact locations.

`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.

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(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"))
```

`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