Package 'sugarglider'

September 3, 2024

Type Package

Title ``sugarglider'' ggplot2 extension for geom sugarglider

Version 0.0.0.9000

Description The ``sugarglider'' package extends ggplot2 by introducing ribbon and segment plots, designed for glyph plot creation.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports dplyr, ggplot2

RoxygenNote 7.3.1

Depends R (>= 2.10)

Suggests testthat (>= 3.0.0), ggplotify, tidyr, grid, lubridate, knitr, rmarkdown, gridExtra, ozmaps, sf, tidyverse, viridis, ggthemes, vdiffr

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Maliny Po [aut, cre] (<https://orcid.org/0009-0008-4686-6631>), Nathan Yang [aut], H. Sherry Zhang [ctb] (<https://orcid.org/0000-0002-7122-1463>), Dianne Cook [ctb] (<https://orcid.org/0000-0002-3813-7155>)

Maintainer Maliny Po <malinypo12@gmail.com>

R topics documented:

`add_glyph_boxes` *Add Glyph Boxes layer to glyph plot*

Description

This function introduces a custom layer to a ggplot, employing 'glyph boxes' to visually represent individual glyph. Users can specify various aesthetics including alpha, height, width, color, line type, and fill to customize the appearance.

Usage

```
add_glyph_boxes(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  x_major = NULL,
  y_major = NULL,
  alpha = 1,
  height = ggplot2::rel(2.5),
  width = ggplot2::rel(4),
  fill = "white",
  inherit.aes = TRUE,
  show.legend = NA,
  ...
)
```

Arguments

`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.
`data`	The data to be displayed in this layer. If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`.
`stat`	The statistical transformation to use on the data for this layer. When using a `geom_*()` function to construct a layer, the `stat` argument can be used the override the default coupling between geoms and stats. The `stat` argument accepts the following: • A `Stat` ggproto subclass, for example `StatCount`. • A string naming the stat. To give the stat as a string, strip the function name of the `stat_` prefix. For example, to use `stat_count()`, give the stat as `"count"`. • For more information and other ways to specify the stat, see the layer stat documentation.

`position` A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The `position` argument accepts the following:

- The result of calling a position function, such as `position_jitter()`. This method allows for passing extra arguments to the position.
- A string naming the position adjustment. To give the position as a string, strip the function name of the `position_` prefix. For example, to use `position_jitter()`, give the position as `"jitter"`.
- For more information and other ways to specify the position, see the layer position documentation.

`x_major, y_major` Aesthetics to map plot coordinates for major and minor glyph components.

`alpha` The transparency level of the glyph box (ranges between 0 and 1).

`height, width` The relative height and width of each glyph box.

`fill` The color used to fill the glyph box.

`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()`.

`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.

`...` Additional arguments passed on to function.

Value

A layer object that can be added to a ggplot.

`add_glyph_legend` *Add Legend Layer to a ggplot*

Description

This function adds a custom legend layer to a ggplot object using the specified aesthetics and parameters.

Usage

```
add_glyph_legend(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  show.legend = NA,
  x_minor = NULL,
  x_scale = identity,
  y_scale = identity,
  global_rescale = TRUE,
  inherit.aes = TRUE,
  ...
)
```

Arguments

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.
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)).
stat	The statistical transformation to use on the data for this layer. When using a geom_*() function to construct a layer, the stat argument can be used the override the default coupling between geoms and stats. The stat argument accepts the following: • A Stat ggproto subclass, for example StatCount. • A string naming the stat. To give the stat as a string, strip the function name of the stat_ prefix. For example, to use stat_count(), give the stat as "count". • For more information and other ways to specify the stat, see the layer stat documentation.
position	A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following: • The result of calling a position function, such as position_jitter(). This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the position_ prefix. For example, to use position_jitter(), give the position as "jitter". • For more information and other ways to specify the position, see the layer position documentation.
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.
x_minor	Aesthetics to map plot coordinates for major and minor glyph components.
x_scale, y_scale	The scaling function applied to each set of minor values within a grid cell. Defaults to ‘identity‘.
global_rescale	A setting that determines whether to perform rescaling globally or on individual glyphs.
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().
...	Additional arguments passed on to function.

Value

A ggplot2 layer.

`add_ref_lines` *Add reference lines to glyph plot*

Description

This function draw reference lines that include both major and minor division markers.

Usage

```
add_ref_lines(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  show.legend = NA,
  x_major = NULL,
  y_major = NULL,
  height = ggplot2::rel(2.5),
  width = ggplot2::rel(4),
  inherit.aes = TRUE,
  ...
)
```

Arguments

`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.
`data`	The data to be displayed in this layer. If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`.
`stat`	The statistical transformation to use on the data for this layer. When using a `geom_*()` function to construct a layer, the `stat` argument can be used the override the default coupling between geoms and stats. The `stat` argument accepts the following: • A `Stat` ggproto subclass, for example `StatCount`. • A string naming the stat. To give the stat as a string, strip the function name of the `stat_` prefix. For example, to use `stat_count()`, give the stat as `"count"`. • For more information and other ways to specify the stat, see the layer stat documentation.
`position`	A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The `position` argument accepts the following: • The result of calling a position function, such as `position_jitter()`. This method allows for passing extra arguments to the position.

- A string naming the position adjustment. To give the position as a string, strip the function name of the `position_` prefix. For example, to use `position_jitter()`, give the position as `"jitter"`.
- For more information and other ways to specify the position, see the layer position documentation.

`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.

`x_major, y_major` Aesthetics to map plot coordinates for major and minor glyph components.

`height, width` he relative height and width of each glyph box.

`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()`.

`...` Additional arguments passed on to function.

Value

A ggplot2 layer.

`aus_temp` *Australian Weather Data for 2022*

Description

This dataset contains aggregated monthly average temperatures (minimum and maximum) and precipitation for selected Australian weather stations for the year 2022. Stations were selected based on specific criteria such as operational status and completeness of data for the year.

Usage

```
aus_temp
```

Format

A data frame with the following columns:

id Station ID.

long Longitude of the station.

lat Latitude of the station.

month Month for the aggregated data.

tmin Monthly average minimum temperature (in degrees Celsius).

tmax Monthly average maximum temperature (in degrees Celsius).

prcp Monthly average precipitation (in mm).

Source

GHCN Daily data via 'meteo_pull_monitors' from the 'rnoaa' package.

flights *Flight Data from 2019-2023*

Description

Minimum and maximum number of flights originated from the top 10 airports with the most canceled flights. The included airports are DEN, MCO, SEA, ATL, DFW, ORD, LAS, LAX, and PHX.

Usage

```
flights
```

Format

'flights' A data frame with 120 rows and 6 columns:

month The month of the flight

origin The origin airport for that flight ...

GeomGlyphBox *GeomGlyphBox*

Description

GeomGlyphBox

GeomGlyphLegend *GeomGlyphLegend*

Description

GeomGlyphLegend

GeomGlyphLine *GeomGlyphLine*

Description

GeomGlyphLine

GeomGlyphRibbon *GeomGlyphRibbon*

Description

GeomGlyphRibbon

GeomGlyphSegment *GeomSegmentGlyph*

Description

GeomSegmentGlyph

geom_glyph_ribbon *Create a Glyph Ribbon plot using ggplot2*

Description

This function creates a ribbon geometry designed to display glyphs based on the combination of ‘x_major‘ and ‘y_major‘. For each ‘x_minor‘ value, ‘geom_glyph_ribbon()‘ displays a y interval defined by ‘ymin_minor‘ and ‘ymax_minor‘.

Usage

```
geom_glyph_ribbon(
  mapping = NULL,
  data = NULL,
  show.legend = NA,
  stat = "identity",
  position = "identity",
  x_major = NULL,
  y_major = NULL,
  x_minor = NULL,
  ymin_minor = NULL,
  ymax_minor = NULL,
  height = ggplot2::rel(2.5),
  width = ggplot2::rel(4),
  x_scale = identity,
  y_scale = identity,
  global_rescale = TRUE,
  inherit.aes = TRUE,
  ...
)
```

Arguments

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.
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)).
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.
stat	The statistical transformation to use on the data for this layer. When using a geom_*() function to construct a layer, the stat argument can be used the override the default coupling between geoms and stats. The stat argument accepts the following: • A Stat ggproto subclass, for example StatCount. • A string naming the stat. To give the stat as a string, strip the function name of the stat_ prefix. For example, to use stat_count(), give the stat as "count". • For more information and other ways to specify the stat, see the layer stat documentation.
position	A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following: • The result of calling a position function, such as position_jitter(). This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the position_ prefix. For example, to use position_jitter(), give the position as "jitter". • For more information and other ways to specify the position, see the layer position documentation.
x_major, y_major, x_minor, ymin_minor, ymax_minor	Each combination of 'x_major' and 'y_major' forms a unique grid cell. 'ymin_minor' and 'ymax_minor' define the lower and upper bounds of the geom_ribbon.
height, width	The height and width of each glyph.
x_scale, y_scale	The scaling function applied to each set of minor values within a grid cell. Defaults to 'identity'.
global_rescale	A setting that determines whether to perform rescaling globally or on individual glyphs.
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().
...	Additional arguments passed on to function.

Value

A ggplot object.

Examples

```
library(ggplot2)

# Basic glyph map with base map and custom theme
aus_temp |>
  ggplot(aes(x_major = long, y_major = lat,
         x_minor = month, ymin_minor = tmin, ymax_minor = tmax)) +
  geom_sf(data = ozmaps::abs_ste, fill = "grey95",
          color = "white",inherit.aes = FALSE) +
  geom_glyph_ribbon() +
  ggthemes::theme_map()


# Adjust width and height of the glyph
aus_temp |>
  ggplot(aes(x_major = long, y_major = lat,
         x_minor = month, ymin_minor = tmin, ymax_minor = tmax)) +
  geom_sf(data = ozmaps::abs_ste, fill = "grey95",
          color = "white",inherit.aes = FALSE) +
  geom_glyph_ribbon(width = rel(4.5), height = rel(3)) +
 ggthemes::theme_map()

# Extend glyph map with reference box and line
aus_temp |>
 ggplot(aes(x_major = long, y_major = lat,
         x_minor = month, ymin_minor = tmin, ymax_minor = tmax)) +
  geom_sf(data = ozmaps::abs_ste, fill = "grey95",
          color = "white",inherit.aes = FALSE) +
  add_glyph_boxes() +
  add_ref_lines() +
  geom_glyph_ribbon() +
  ggthemes::theme_map()
```

---

geom_glyph_segment *Create a Glyph Segment plot using ggplot2*

Description

This function enables the creation of segment glyphs by defining major coordinates (longitude and latitude) and minor segment structures within a grid cell. Each glyph's appearance can be customized by specifying its height, width, and scaling, allowing for flexible data representation in a visual context.

Usage

```
geom_glyph_segment(
  mapping = NULL,
  data = NULL,
  stat = "identity",
```

```
  position = "identity",
  ...,
  x_major = NULL,
  x_minor = NULL,
  y_major = NULL,
  y_minor = NULL,
  yend_minor = NULL,
  width = ggplot2::rel(4),
  x_scale = identity,
  y_scale = identity,
  height = ggplot2::rel(2.5),
  global_rescale = TRUE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

`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.
`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)`).
`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.
`...`	Other arguments passed on to `layer()`. These are often aesthetics, used to set an aesthetic to a fixed value, like `colour = "red"` or `size = 3`. They may also be parameters to the paired geom/stat.
`x_major, x_minor, y_major, y_minor, yend_minor`	The name of the variable (as a string) for the major and minor x and y axes. `x_major` and `y_major` specify a longitude and latitude on a map while `x_minor`, `y_minor`, and `yend_minor` provide the structure for glyph.
`y_scale, x_scale`	The scaling function to be applied to each set of minor values within a grid cell. The default is `identity` which produces a result without scaling.
`height, width`	The height and width of each glyph.
`global_rescale`	Determines whether or not the rescaling is performed globally or separately for each individual glyph.

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().

Value

a ggplot object

Examples

```
library(ggplot2)

# Basic glyph map with base map and custom theme
aus_temp |>
  ggplot(aes(x_major = long, y_major = lat,
         x_minor = month, y_minor = tmin, yend_minor = tmax)) +
  geom_sf(data = ozmaps::abs_ste, fill = "grey95",
          color = "white",inherit.aes = FALSE) +
  geom_glyph_segment() +
  ggthemes::theme_map()


# Adjust width and height of the glyph
aus_temp |>
  ggplot(aes(x_major = long, y_major = lat,
         x_minor = month, y_minor = tmin, yend_minor = tmax)) +
  geom_sf(data = ozmaps::abs_ste, fill = "grey95",
          color = "white",inherit.aes = FALSE) +
  geom_glyph_segment(width = rel(4.5), height = rel(3)) +
 ggthemes::theme_map()

# Extend glyph map with reference box and line
aus_temp |>
 ggplot(aes(x_major = long, y_major = lat,
         x_minor = month, y_minor = tmin, yend_minor = tmax)) +
  geom_sf(data = ozmaps::abs_ste, fill = "grey95",
          color = "white",inherit.aes = FALSE) +
  add_glyph_boxes() +
  add_ref_lines() +
  geom_glyph_segment() +
  ggthemes::theme_map()
```

---

historical_temp *Historical Australian Weather Data for 2021-2022*

Description

This dataset contains aggregated monthly average temperatures (minimum and maximum) and precipitation for selected Australian weather stations for the years 2021 and 2022. It provides a broader historical perspective compared to 'aus_temp'. Stations were selected based on operational status and data completeness.

Usage

```
historical_temp
```

Format

A data frame with the following columns:

id Station ID.

long Longitude of the station.

lat Latitude of the station.

month Month for the aggregated data.

year Year for the aggregated data, either 2021 or 2022.

tmin Monthly average minimum temperature (in degrees Celsius).

tmax Monthly average maximum temperature (in degrees Celsius).

prcp Monthly average precipitation (in mm).

Source

GHCN Daily data via ‘meteo_pull_monitors‘ from the ‘rnoaa‘ package.

Index