Package 'ggfields'

February 26, 2024

Title Add Vector Field Layers to Ggplots

Version 0.0.6

Date 2024-02-26

Author Pepijn de Vries [aut, cre] (<https://orcid.org/0000-0002-7961-6646>)

Maintainer Pepijn de Vries <pepijn.devries@outlook.com>

Description Add vector field layers to ggplots. Ideal for visualising wind speeds, water currents, electric/magnetic fields, etc. Accepts data.frames, simple features (sf), and spatiotemporal arrays (stars) objects as input. Vector fields are depicted as arrows starting at specified locations, and with specified angles and radii.

Depends R (>= 4.1.0)

Imports dplyr (>= 1.1.4), ggplot2 (>= 3.4.4), grid (>= 4.1.0), rlang (>= 1.1.2), sf (>= 1.0-15), scales (>= 1.3.0)

Suggests CopernicusMarine, knitr, rmarkdown, stars (>= 0.6-4), testthat (>= 3.0.0), vdiffr (>= 1.0.7), svglite (>= 2.1.3)

License GPL (>= 3)

URL https://pepijn-devries.github.io/ggfields/, https://github.com/pepijn-devries/ggfields/

Encoding UTF-8

RoxygenNote 7.2.3

Collate 'angle_correction.r' 'helpers.r' 'stat_fields.r' 'draw_key_fields.r' 'geom_fields.r' 'annotation_fields.r' 'data.r' 'ggfields-package.r' 'import.r' 'pythagoras.r' 'scale.r'

LazyData true

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Repository CRAN

Date/Publication 2024-02-26 14:40:03 UTC

R topics documented:

`angle_correction` *Calculate correction for angle in the plot coordinate system*

Description

The angle of a vector may be distorted when your plot uses a different coordinate system than the one for which the angle is specified. If `data` is a simple feature object (sf), the angle will be corrected for the displayed coordinate reference system (crs). When the crs is missing, an aspect ratio of 1 is assumed. For any other data, the angle is corrected for the aspect ratio in the plot.

Usage

```
angle_correction(data, panel_params, coord)
```

Arguments

`data`	fortified data used in a `geom_fields()`. Should at least contain `numeric` columns `x`, `y` and `angle`.
`panel_params`	panel parameters as returned by GeomFields$setup_params()
`coord`	A coord object.

Details

This function is used by default by `geom_fields()`. For more details on why this correction is required and how to customize corrections please see `vignette("angle_correction")`.

Value

A `data.frame` with an additional `angle_correction` column. The corrected angle is given by `angle_correction + angle`.

Author(s)

Pepijn de Vries

Examples

```
## Create a data.frame with some xy-coordinates and all angles pointing North (0 degrees)
d <-
  data.frame(
    x = seq(1, 2, 0.1),
    y = seq(50, 51, 0.1),
    angle = 0
  ) |>
  sf::st_as_sf(coords = c("x", "y"), crs = 4326, remove = FALSE)

## Create a mockup of ggplot params. Normally this is handled automatically by ggplot2
params_mockup <-
  c(
    ggplot2::ggplot() + geom_fields(),
    list(
      x_range = range(d$x),
      y_range = range(d$y),
      crs = sf::st_crs(4326),
      default_crs = 4326
    )
  )

## When plotting as lon-lat, the angle correction will be zero
angle_correction(d, params_mockup, ggplot2::coord_sf(default_crs = 4326))

## Transform to UTM zone 31N in meters
d2 <- d |> sf::st_transform(32631)

## Again get parameter mockup values
params_mockup2 <-
  c(
    ggplot2::ggplot() + geom_fields(),
      list(
        x_range = range(sf::st_coordinates(d2)[,1]),
        y_range = range(sf::st_coordinates(d2)[,1]),
        crs = sf::st_crs(32631),
        default_crs = 4326
      )
    )
## in UTM projection in this area (which is slightly tilted) the correction is
## larger than zero
angle_correction(d2, params_mockup2,
                 ggplot2::coord_sf(crs = 32631, default_crs = 4326))
```

`annotation_fields` *Annotate a ggplot with vector fields*

Description

Functions exactly the same as `geom_fields()`, with that difference that this function does not train the `x` and `y` scales. This makes the data central, and uses this layer to support it. Consequently,

`annotation_fields()` does not accept a `stat` argument.

Usage

```
annotation_fields(
  mapping = NULL,
  data = NULL,
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  max_radius = ggplot2::unit(0.5, "cm"),
  .angle_correction = angle_correction,
  arrow = grid::arrow(length = ggplot2::unit(0.2, "cm")),
  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`	Can be one of four things: • `NULL`: in that case data from the parent ggplot call is inherited. • `data.frame`: you need to assign the `x` and `y` aesthetics. • sf object: it should contain a `geometry` column with only `POINT` geometries. • stars object: it will be converted automatically to an `sf` object.
`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.
`na.rm`	If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.
`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.
`max_radius`	Maximum radius to which the `radius` aesthetic is scaled in the plot. You can use absolute ("e.g., "cm", "in", "pt") and relative ("npc") units to set its value. Default is `0.5 cm`.
`.angle_correction`	Function to correct the angle in the aesthetics for the projection and/or aspect ratio used in the plot. When set to `NULL` the angle is not corrected and is treated as the angle in the final plot. A custom function can be provided which should accept at least three arguments (`data`, `panel_params` and `coord`). See `angle_correction()` and `vignette("angle_correction")` for more details.
`arrow`	specification for arrow heads, as created by `grid::arrow()`.

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().
...	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.

Value

A ggplot2::layer_sf().

Author(s)

Pepijn de Vries

Examples

```
if (requireNamespace("stars") && requireNamespace("ggplot2")) {
  library(stars)
  library(ggplot2)

  data("seawatervelocity")
  sw_sub <- seawatervelocity[,8:13,1:5]

  ## Note that the `seawatervelocity` spans a much larger area,
  ## but the plot only focuses on `sw_sub`
  ggplot() +
    geom_stars(data = sw_sub) +
    annotation_fields(data = seawatervelocity,
                      aes(angle = as.numeric(atan2(vo, uo)),
                          radius = as.numeric(pythagoras(uo, vo)))) +
    labs(radius = "v [m/s]")
}
```

---

draw_key_fields *Key glyphs for 'radius' legends*

Description

Each geom has an associated function that draws the key when the geom needs to be displayed in a legend. These functions are called draw_key_*(), where * stands for the name of the respective key glyph. The key glyphs can be customized for individual geoms by providing a geom with the key_glyph argument (see layer() or examples below.)

Usage

```
draw_key_fields(data, params, size)
```

Arguments

data	A single row data frame containing the scaled aesthetics to display in this key
params	A list of additional parameters supplied to the geom.
size	Width and height of key in mm.

Details

The layer geom_fields() allows for a special aesthetic radius. This function draws a key glyph for this aesthetics, where the radius of the arrow corresponds with the scalar value listed with this radius. Note that the width of the key glyph cannot be adjusted by the aesthetic itself. Therefore, if your max_radius parameter exceeds the glyph width, you need to change the width of the guides yourself, see vignette("radius_aes") for more details.

Value

A grid grob

Author(s)

Pepijn de Vries

Examples

```
if (requireNamespace("ggplot2")) {
  library(ggplot2)
  p <- ggplot(economics, aes(date, psavert, color = "savings rate"))
  p + geom_line(key_glyph = "fields")
}
```

GeomFields *Arrows depicting a vector field*

Description

Visualise vector fields (such as, electric/magnetic fields, wind speed, or water currents) with arrows as a layer in a ggplot.

Usage

```
GeomFields

geom_fields(
  mapping = NULL,
  data = NULL,
  stat = "fields",
  position = "identity",
  na.rm = FALSE,
```

```
  show.legend = NA,
  max_radius = ggplot2::unit(0.5, "cm"),
  .angle_correction = angle_correction,
  arrow = grid::arrow(length = ggplot2::unit(0.2, "cm")),
  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	Can be one of four things: • NULL: in that case data from the parent ggplot call is inherited. • data.frame: you need to assign the x and y aesthetics. • sf object: it should contain a geometry column with only POINT geometries. • stars object: it will be converted automatically to an sf object.
stat	The statistical transformation to use on the data for this layer. By default it is set to GeomFields() ("fields").
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.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
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.
max_radius	Maximum radius to which the radius aesthetic is scaled in the plot. You can use absolute ("e.g., "cm", "in", "pt") and relative ("npc") units to set its value. Default is 0.5 cm.
.angle_correction	Function to correct the angle in the aesthetics for the projection and/or aspect ratio used in the plot. When set to NULL the angle is not corrected and is treated as the angle in the final plot. A custom function can be provided which should accept at least three arguments (data, panel_params and coord). See angle_correction() and vignette("angle_correction") for more details.
arrow	specification for arrow heads, as created by grid::arrow().
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().
...	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.

Format

An object of class `GeomFields` (inherits from `GeomSegment`, `Geom`, `ggproto`, `gg`) of length 8.

Details

Adds a layer with vector fields to a ggplot. In order to achieve this two special aesthetic are required: `radius` and `angle`.

Value

A layer which can be added to a ggplot.

Aesthetics

- `geometry|x`: Either a `geometry` column or `x` coordinate. In case of `geometry` the column should be of class sf::sfc_POINT. In case of `x`, it should be a `numeric vector`, and the aesthetic `y` needs to be specified as well. It specifies the location of the origin of each vector.
- `radius`: This aesthetic will be used to scale the radius of the vector arrows in the field you wish to display. The maximum radius of the arrows is given by parameter `max_radius`. See `vignette("radius_aes")` for more details.
- `angle`: This aesthetic represent the angles of the vectors in your field in radians. Contrary to the mathematical definition, an angle of 0 radians will point upwards (instead of to the right). This was chosen such because in most geographical applications an angle of zero degrees points Northwards. Before plotting these angles are corrected by the function passed to the `.angle_correction` argument. See `vignette("angle_corrections)` for more details.
- `y`: This aesthetic needs to be used in combination with the `x` aesthetic. It needs to be a `numeric vector`.
- `fill`: See `vignette("ggplot2-specs", "ggplot2")`
- `colour`: See `vignette("ggplot2-specs", "ggplot2")`
- `linetype`: See `vignette("ggplot2-specs", "ggplot2")`
- `linewidth`: See `vignette("ggplot2-specs", "ggplot2")`
- `alpha`: A variable to control the opacity of an element.

Author(s)

Pepijn de Vries

Examples

```
data(seawatervelocity)

if (requireNamespace("ggplot2") && requireNamespace("stars") &&
      requireNamespace("scales")) {
  library(ggplot2)
  library(stars)

  sw_df <- as.data.frame(seawatervelocity)
```

```
  ggplot(sw_df, aes(x = x, y = y, radius = as.numeric(v), angle = as.numeric(angle))) +
    geom_fields(max_radius = unit(0.5, "cm"), na.rm = TRUE)

  ggplot() +
    geom_fields(data    = seawatervelocity,
                mapping = aes(radius = as.numeric(v),
                              angle  = as.numeric(angle),
                              col    = as.numeric(v)),
                max_radius = unit(0.5, "cm")) +
    scale_colour_viridis_c()
}
```

`pythagoras` *A helper function to calculate vector lengths*

Description

Calculates the length of a vector using the Pythagorean theorem.

Usage

```
pythagoras(x, y)
```

Arguments

`x`	A `numeric` `vector` with the same length as `y`. It should represent the lengths of the first leg (cathetus) of right triangles.
`y`	A `numeric` `vector` with the same length as `x`. It should represent the lengths of the second leg (cathetus) of right triangles.

Value

Returns a `numeric` `vector` with the same length as `x` and `y`, reflecting the lengths of the hypotenuse of the right triangles.

Author(s)

Pepijn de Vries

Examples

```
pythagoras(x = c(1, 2), y = c(1, 2))
```

scale_radius_continuous *Vector field radius scales*

Description

Scales to set up the visualisation of the `radius` aesthetic. These scales are also automatically used in plot guides. Note that `scale_radius_identity()` does *not* exist as it would be impossible to relate such a scale to the `max_radius` parameter. For more details see `vignette("radius_aes")`.

Usage

```
scale_radius_continuous(..., range = c(1e-08, 1))

scale_radius_binned(..., range = c(1e-08, 1))

scale_radius_discrete(..., range = c(1e-08, 1))
```

Arguments

`...`	Arguments passed on to underpinning `ggplot2::scale_*` functions.
`range`	Relative output range of radii. Must lie between 0 and 1.

Value

An object of class Scale.

Author(s)

Pepijn de Vries

Examples

```
if (requireNamespace("ggplot2")) {
  library(ggplot2)
  data(seawatervelocity)

  g_num <-
    ggplot() +
    geom_fields(data = seawatervelocity,
                aes(radius = as.numeric(v), angle = as.numeric(angle)))
  g_discr <-
    ggplot() +
    geom_fields(data = seawatervelocity,
                aes(radius = cut(as.numeric(v), 4), angle = as.numeric(angle)))

  g_num + scale_radius_continuous()
  g_num + scale_radius_binned()
```

```
  g_discr + scale_radius_discrete()
}
```

`seawatervelocity` *A small subset of the global ocean physics analysis and forecast product*

Description

A small subset of ocean currents data retrieved with CopernicusMarine from the source listed below serving as an example.

Format

A stars object with `x`, `y`, `depth` and `time` dimensions. It has the attributes `vo` (northward seawater velocity [m/s]) and `uo` (eastward seawater velocity [m/s]).

References

E.U. Copernicus Marine Service Information; Global Ocean Physics Analysis and Forecast - GLOBAL_ANALYSISFORECA
(2016-10-14). doi:10.48670/moi00016

Examples

```
data("seawatervelocity")
```

`StatFields` *Stat method for geom_fields*

Description

Prepares data before being handled by `geom_fields()`

Usage

```
StatFields

stat_fields(
  mapping = NULL,
  data = NULL,
  geom = "fields",
  position = "identity",
  na.rm = FALSE,
  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	Can be one of four things: • NULL: in that case data from the parent ggplot call is inherited. • data.frame: you need to assign the x and y aesthetics. • sf object: it should contain a geometry column with only POINT geometries. • stars object: it will be converted automatically to an sf object.
geom	The layer type for which the data is prepared. In this case "fields".
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.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
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().
...	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.

Format

An object of class StatFields (inherits from StatSf, Stat, ggproto, gg) of length 3.

Value

Returns a layer that can be further modified by geom_fields().

Author(s)

Pepijn de Vries

Examples

```
stat_fields()
```

Index