

Package ‘ggstar’

February 10, 2021

Title Star Layer for 'ggplot2'

Version 1.0.1

Description To create the regular polygon layer for easily discernible shapes,
we developed the package, it can be easily used if you know the 'ggplot2'.

Depends R (>= 3.5)

Imports grid, utils, ggplot2, scales, gridExtra

Suggests knitr, markdown, rmarkdown, prettydoc

License Artistic-2.0

Encoding UTF-8

URL <https://github.com/xiangpin/ggstar/>

BugReports <https://github.com/xiangpin/ggstar/issues>

VignetteBuilder knitr

LazyData false

RoxygenNote 7.1.1

NeedsCompilation no

Author Shuangbin Xu [aut, cre] (<<https://orcid.org/0000-0003-3513-5362>>)

Maintainer Shuangbin Xu <xshuangbin@163.com>

Repository CRAN

Date/Publication 2021-02-10 10:40:29 UTC

R topics documented:

draw_key	2
GeomStar	2
geom_star	3
scale_manual	5
scale_starshape	6
scale_starshape_identity	7
show_starshapes	9

Index	11
--------------	-----------

draw_key	<i>Key drawing functions</i>
----------	------------------------------

Description

Each Geom has an associated function that draws the key when the geom needs to be displayed in a legend. These are the options built into ggplot2.

Usage

```
draw_key_star(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.

Value

A grid grob.

GeomStar	<i>GeomStar</i>
----------	-----------------

Description

GeomStar

Author(s)

Shuangbin Xu

geom_star

*Star layer***Description**

geom_star provides the siogon layer to easily discernible starshapes for ggplot2, you can use it to create scatterplots.

Usage

```
geom_star(
  mapping = NULL,
  data = NULL,
  na.rm = FALSE,
  stat = "identity",
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or 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. <code>~ head(.x, 10)</code>).
na.rm	If 'FALSE', the default, missing values are removed with a warning. If 'TRUE', missing values are silently removed.
stat	The statistical transformation to use on the data for this layer, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
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`.

Details

Note: the 'left-triangle' and 'right-triangle' are developed to plot the 'triangle-heatmap'. Their centers are not in their internal, but the center of hypotenuse.

Value

polygonal layer

Aesthetics

`geom_star()` understands the following aesthetics (required aesthetics are in bold):

- **x**
- **y**
- alpha
- angle
- colour
- fill
- group
- phase
- size
- starshape
- starstroke

Learn more about setting these aesthetics in `vignette("ggplot2-specs")`.

Author(s)

Shuangbin Xu

Examples

```
library(ggplot2)
p <- ggplot(iris, aes(x=Sepal.Length,
                     y=Sepal.Width,
                     starshape=Species)) +
  geom_star(size=4)
p
```

scale_manual	<i>Create your own discrete scale</i>
--------------	---------------------------------------

Description

Create your own discrete scale

Usage

```
scale_starshape_manual(values, ...)
```

Arguments

values a set of aesthetic values to map data values to. If this is a named vector, then the values will be matched based on the names. If unnamed, values will be matched in order (usually alphabetical) with the limits of the scale. Any data values that don't match will be given 'na.value'.

... Arguments passed on to `ggplot2::discrete_scale`

scale_name The name of the scale that should be used for error messages associated with this scale.

palette A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., `scales::hue_pal()`).

name The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.

breaks One of:

- `NULL` for no breaks
- `waiver()` for the default breaks (the scale limits)
- A character vector of breaks
- A function that takes the limits as input and returns breaks as output

labels One of:

- `NULL` for no labels
- `waiver()` for the default labels computed by the transformation object
- A character vector giving labels (must be same length as breaks)
- A function that takes the breaks as input and returns labels as output

limits One of:

- `NULL` to use the default scale values
- A character vector that defines possible values of the scale and their order
- A function that accepts the existing (automatic) values and returns new ones

`na.translate` Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`.

`na.value` If `na.translate = TRUE`, what aesthetic value should the missing values be displayed as? Does not apply to position scales where NA is always placed at the far right.

`drop` Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE uses all the levels in the factor.

`guide` A function used to create a guide or its name. See `guides()` for more information.

`super` The super class to use for the constructed scale

Value

starshape scale constructor

scale_starshape	<i>Scales for starshapes, aka glyphs</i>
-----------------	--

Description

'scale_starshape' maps discrete variables to six easily discernible starshapes. If you have more than 6 levels, you will get a warning message, and the seventh and subsequent levels will not appear on the plot. Use `[scale_starshape_manual()]` to supply your own values. You can not map a continuous variable to shape.

Usage

```
scale_starshape(default = TRUE, ...)
```

Arguments

`default` should the starshapes be default?

`...` Arguments passed on to `ggplot2::discrete_scale`

`aesthetics` The names of the aesthetics that this scale works with.

`scale_name` The name of the scale that should be used for error messages associated with this scale.

`palette` A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., `scales::hue_pal()`).

`name` The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If NULL, the legend title will be omitted.

`breaks` One of:

- NULL for no breaks

- `waiver()` for the default breaks (the scale limits)
 - A character vector of breaks
 - A function that takes the limits as input and returns breaks as output
- labels** One of:
- `NULL` for no labels
 - `waiver()` for the default labels computed by the transformation object
 - A character vector giving labels (must be same length as breaks)
 - A function that takes the breaks as input and returns labels as output
- limits** One of:
- `NULL` to use the default scale values
 - A character vector that defines possible values of the scale and their order
 - A function that accepts the existing (automatic) values and returns new ones
- na.translate** Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`.
- na.value** If `na.translate = TRUE`, what aesthetic value should the missing values be displayed as? Does not apply to position scales where `NA` is always placed at the far right.
- drop** Should unused factor levels be omitted from the scale? The default, `TRUE`, uses the levels that appear in the data; `FALSE` uses all the levels in the factor.
- guide** A function used to create a guide or its name. See [guides\(\)](#) for more information.
- super** The super class to use for the constructed scale

scale_starshape_identity

Use values without scaling for ggstar

Description

Use values without scaling for ggstar

Usage

```
scale_starshape_identity(..., guide = "none")
```

Arguments

`...` Arguments passed on to [ggplot2::continuous_scale](#)

aesthetics The names of the aesthetics that this scale works with.

scale_name The name of the scale that should be used for error messages associated with this scale.

- palette** A palette function that when called with a numeric vector with values between 0 and 1 returns the corresponding output values (e.g., `scales::area_pal()`).
- name** The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.
- breaks** One of:
- `NULL` for no breaks
 - `waiver()` for the default breaks computed by the [transformation object](#)
 - A numeric vector of positions
 - A function that takes the limits as input and returns breaks as output (e.g., a function returned by `scales::extended_breaks()`)
- minor_breaks** One of:
- `NULL` for no minor breaks
 - `waiver()` for the default breaks (one minor break between each major break)
 - A numeric vector of positions
 - A function that given the limits returns a vector of minor breaks.
- n.breaks** An integer guiding the number of major breaks. The algorithm may choose a slightly different number to ensure nice break labels. Will only have an effect if `breaks = waiver()`. Use `NULL` to use the default number of breaks given by the transformation.
- labels** One of:
- `NULL` for no labels
 - `waiver()` for the default labels computed by the transformation object
 - A character vector giving labels (must be same length as breaks)
 - A function that takes the breaks as input and returns labels as output
- limits** One of:
- `NULL` to use the default scale range
 - A numeric vector of length two providing limits of the scale. Use `NA` to refer to the existing minimum or maximum
 - A function that accepts the existing (automatic) limits and returns new limits Note that setting limits on positional scales will **remove** data outside of the limits. If the purpose is to zoom, use the `limit` argument in the coordinate system (see `coord_cartesian()`).
- rescaler** A function used to scale the input values to the range [0, 1]. This is always `scales::rescale()`, except for diverging and n colour gradients (i.e., `scale_colour_gradient2()`, `scale_colour_gradientn()`). The rescaler is ignored by position scales, which always use `scales::rescale()`.
- oob** One of:
- Function that handles limits outside of the scale limits (out of bounds).
 - The default (`scales::censor()`) replaces out of bounds values with `NA`.
 - `scales::squish()` for squishing out of bounds values into range.
 - `scales::squish_infinite()` for squishing infinite values into range.

expand For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function `expansion()` to generate the values for the `expand` argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.

na.value Missing values will be replaced with this value.

trans For continuous scales, the name of a transformation object or the object itself. Built-in transformations include "asn", "atanh", "boxcox", "date", "exp", "hms", "identity", "log", "log10", "log1p", "log2", "logit", "modulus", "probability", "probit", "pseudo_log", "reciprocal", "reverse", "sqrt" and "time".

A transformation object bundles together a transform, its inverse, and methods for generating breaks and labels. Transformation objects are defined in the scales package, and are called `<name>_trans` (e.g., `scales::boxcox_trans()`).

You can create your own transformation with `scales::trans_new()`.

position For position scales, The position of the axis. left or right for y axes, top or bottom for x axes.

super The super class to use for the constructed scale

guide Guide to use for this scale. Defaults to "none".

Value

identical (default) starshape scale constructor

See Also

[scale_shape_identity](#)

show_starshapes	<i>Show the total star shapes</i>
-----------------	-----------------------------------

Description

Show the total star shapes

Usage

```
show_starshapes(...)
```

Arguments

... see also [theme](#).

Value

gg object

Author(s)

Shuangbin Xu

Examples

```
p <- show_starshapes()  
p
```

Index

- * **datasets**
 - GeomStar, 2
- aes(), 3
- aes_(), 3
- borders(), 4
- coord_cartesian(), 8
- draw_key, 2
- draw_key_star (draw_key), 2
- expansion(), 9
- fortify(), 3
- geom_star, 3
- GeomStar, 2
- ggplot(), 3
- ggplot2::continuous_scale, 7
- ggplot2::discrete_scale, 5, 6
- guides(), 6, 7
- layer, 4
- scale_colour_gradient2(), 8
- scale_colour_gradientn(), 8
- scale_manual, 5
- scale_shape_identity, 9
- scale_starshape, 6
- scale_starshape_continuous (scale_starshape), 6
- scale_starshape_discrete (scale_starshape), 6
- scale_starshape_identity, 7
- scale_starshape_manual (scale_manual), 5
- scale_starshape_ordinal (scale_starshape), 6
- scales::area_pal(), 8
- scales::boxcox_trans(), 9
- scales::censor(), 8
- scales::extended_breaks(), 8
- scales::hue_pal(), 5, 6
- scales::rescale(), 8
- scales::squish(), 8
- scales::squish_infinite(), 8
- scales::trans_new(), 9
- show_starshapes, 9
- theme, 9
- transformation object, 8