Package 'gglyph'

June 15, 2021

Title Multivariate Data Visualization using Glyphs

Version 0.0.0.9000

Description Provides geoms for visualizing multivariate data as glyphs using 'ggplot2'.

License GPL-2 | GPL-3

Encoding UTF-8

RoxygenNote 7.1.1

LazyData true

RdMacros Rdpack

Depends R (>= 3.5.0)

Imports dplyr,
ggplot2,
grid,
Rdpack,
rlang,
scales

Suggests RColorBrewer

R topics documented:

dotglyphGrob 1
geom_starglyph 4
metroglyphGrob 7
pieglyphGrob 11
profileglyphGrob 13
starglyphGrob 18
tileglyphGrob 22

Index **25**

dotglyphGrob *Draw a Dot Profile Glyph*

Description

Uses Grid graphics to draw a dot profile glyph (Chambers et al. 1983; DuToit et al. 1986).

Usage

```
dotglyphGrob(
  x = 0.5,
  y = 0.5,
  z,
  radius = 10,
  col = "black",
  fill = NA,
  lwd = 1,
  alpha = 1,
  mirror = FALSE,
  flip.axes = FALSE
)
```

Arguments

x	A numeric vector or unit object specifying x-locations.
y	A numeric vector or unit object specifying y-locations.
z	A numeric vector specifying the distance of star glyph points from the center.
radius	The radius of the glyphs.
col	The colour of whisker and contours.
fill	The fill colour.
lwd	The line width.
alpha	The alpha transparency value.
mirror	logical. If TRUE, mirror profile is plotted.
flip.axes	logical. If TRUE, axes are flipped.

Value

A grob object.

References

Chambers JM, Cleveland WS, Kleiner B, Tukey PA (1983). *Graphical Methods for Data Analysis*. Chapman and Hall/CRC, Boca Raton. ISBN 978-1-351-07230-4.

DuToit SHC, Steyn AGW, Stumpf RH (1986). *Graphical Exploratory Data Analysis*, Springer Texts in Statistics. Springer-Verlag, New York. ISBN 978-1-4612-9371-2, doi: 10.1007/97814612-49504, https://doi.org/10.1007/978-1-4612-4950-4.

See Also

geom_dotprofileglyph

Examples

```
dg1 <- dotglyphGrob(x = 150, y = 200,
                    z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                    radius = 10)

dg2 <- dotglyphGrob(x = 450, y = 200,
                    z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                    radius = 10, mirror = TRUE)

dg3 <- dotglyphGrob(x = 150, y = 450,
                    z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                    radius = 10, flip.axes = TRUE)

dg4 <- dotglyphGrob(x = 450, y = 450,
                    z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                    radius = 10, mirror = TRUE,
                    flip.axes = TRUE)

grid::grid.newpage()
grid::grid.draw(dg1)
grid::grid.draw(dg2)
grid::grid.draw(dg3)
grid::grid.draw(dg4)

dg1 <- dotglyphGrob(x = 150, y = 200,
                    z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                    radius = 10, fill = "black", col = "white")

dg2 <- dotglyphGrob(x = 450, y = 200,
                    z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                    radius = 10, mirror = TRUE,
                    fill = "salmon", col = "black")

dg3 <- dotglyphGrob(x = 150, y = 450,
                    z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                    radius = 10, flip.axes = TRUE,
                    fill = "cyan", col = "grey")

dg4 <- dotglyphGrob(x = 450, y = 450,
                    z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                    radius = 10, mirror = TRUE,
                    flip.axes = TRUE,
                    fill = "green", col = "grey")

grid::grid.newpage()
grid::grid.draw(dg1)
grid::grid.draw(dg2)
grid::grid.draw(dg3)
grid::grid.draw(dg4)

clrs <- mapply(function(a, b) rep(a, b),
               RColorBrewer::brewer.pal(6, "Dark2"),
```

```
              round(c(4, 3.5, 2.7, 6.8, 3.4, 5.7)))
clrs <- unlist(clrs)

dg1 <- dotglyphGrob(x = 150, y = 200,
                    z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                    radius = 10, fill = clrs, col = "white")

dg2 <- dotglyphGrob(x = 450, y = 200,
                    z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                    radius = 10, mirror = TRUE,
                    fill = clrs, col = "black")

dg3 <- dotglyphGrob(x = 150, y = 450,
                    z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                    radius = 10, flip.axes = TRUE,
                    fill = "black", col = clrs)

dg4 <- dotglyphGrob(x = 450, y = 450,
                    z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                    radius = 10, mirror = TRUE,
                    flip.axes = TRUE,
                    col = clrs)

grid::grid.newpage()
grid::grid.draw(dg1)
grid::grid.draw(dg2)
grid::grid.draw(dg3)
grid::grid.draw(dg4)
```

`geom_starglyph` *Add Star Glyphs as a Scatterplot*

Description

The starglyph geom is used to plot multivariate data as star glyphs (Siegel et al. 1972; Chambers et al. 1983; DuToit et al. 1986) in a scatterplot.

Usage

```
geom_starglyph(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  cols = character(0L),
  whisker = TRUE,
  contour = TRUE,
  colour.whisker = NULL,
  linewidth.whisker = 1,
  linewidth.contour = 1,
  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. ~ head(.x,10)).
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.
...	additional parameters
cols	Name of columns specifying the variables to be plotted in the glyphs as a character vector.
whisker	logical. If TRUE, plots the star glyph whiskers.
contour	logical. If TRUE, plots the star glyph contours. glyph.
colour.whisker	The colour of whisker.
linewidth.whisker	The whisker line width.
linewidth.contour	The contour line width.
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 geom layer.

Aesthetics

geom_starglyph() understands the following aesthetics (required aesthetics are in bold):

- **x**
- **y**
- alpha
- colour
- fill
- group

- shape
- size
- stroke
- linetype

References

Chambers JM, Cleveland WS, Kleiner B, Tukey PA (1983). *Graphical Methods for Data Analysis*. Chapman and Hall/CRC, Boca Raton. ISBN 978-1-351-07230-4.

DuToit SHC, Steyn AGW, Stumpf RH (1986). *Graphical Exploratory Data Analysis*, Springer Texts in Statistics. Springer-Verlag, New York. ISBN 978-1-4612-9371-2, doi: 10.1007/97814612-49504, https://doi.org/10.1007/978-1-4612-4950-4.

Siegel JH, Farrell EJ, Goldwyn RM, Friedman HP (1972). "The surgical implications of physiologic patterns in myocardial infarction shock." *Surgery*, **72**(1), 126–141.

See Also

starglyphGrob

Examples

```
# Scale the data
zs <- c("hp", "drat", "wt", "qsec", "vs", "am", "gear", "carb")
mtcars[ , zs] <- lapply(mtcars[ , zs], scales::rescale)

mtcars$cyl <- as.factor(mtcars$cyl)
mtcars$lab <- row.names(mtcars)

library(ggplot2)

# Both whiskers and contour
ggplot(data = mtcars) +
  geom_starglyph(aes(x = mpg, y = disp, fill = cyl),
                 cols = zs, whisker = TRUE, contour = TRUE,
                 size = 0.1, alpha =  0.5) +
  ylim(c(-0, 550))

ggplot(data = mtcars) +
  geom_starglyph(aes(x = mpg, y = disp, fill = cyl),
                 cols = zs, whisker = TRUE, contour = TRUE,
                 size = 0.1, alpha =  0.5,
                 linewidth.whisker = 3, linewidth.contour = 0.1) +
  ylim(c(-0, 550))

ggplot(data = mtcars) +
  geom_starglyph(aes(x = mpg, y = disp, fill = cyl),
                 cols = zs, whisker = TRUE, contour = TRUE,
                 size = 0.1, alpha =  0.5,
                 linewidth.whisker = 1, linewidth.contour = 3) +
  ylim(c(-0, 550))


# Only contours (polygon)
```

```
ggplot(data = mtcars) +
  geom_starglyph(aes(x = mpg, y = disp, fill = cyl),
                 cols = zs, whisker = FALSE, contour = TRUE,
                 size = 0.1, alpha =  0.5) +
  ylim(c(-0, 550))

ggplot(data = mtcars) +
  geom_starglyph(aes(x = mpg, y = disp, fill = cyl),
                 cols = zs, whisker = FALSE, contour = TRUE,
                 size = 0.1, alpha =  0.5, linewidth.contour = 3) +
  ylim(c(-0, 550))

# Only whiskers
ggplot(data = mtcars) +
  geom_starglyph(aes(x = mpg, y = disp, colour = cyl),
                 cols = zs, whisker = TRUE, contour = FALSE,
                 size = 0.1) +
  geom_point(data = mtcars, aes(x = mpg, y = disp, colour = cyl)) +
  ylim(c(-0, 550))

# Whiskers with colours
ggplot(data = mtcars) +
  geom_starglyph(aes(x = mpg, y = disp),
                 cols = zs, whisker = TRUE, contour = FALSE,
                 size = 0.1,
                 colour.whisker = RColorBrewer::brewer.pal(8, "Dark2")) +
  geom_point(data = mtcars, aes(x = mpg, y = disp)) +
  ylim(c(-0, 550))

# With text annotations
ggplot(data = mtcars) +
  geom_starglyph(aes(x = mpg, y = disp, colour = cyl),
                 cols = zs, whisker = TRUE, contour = FALSE,
                 size = 0.1) +
  geom_point(data = mtcars, aes(x = mpg, y = disp, colour = cyl)) +
  geom_text(data = mtcars, aes(x = mpg, y = disp, label = lab), cex = 2) +
  ylim(c(-0, 550))

# Faceted
ggplot(data = mtcars) +
  geom_starglyph(aes(x = mpg, y = disp, fill = cyl),
                 cols = zs, whisker = TRUE, contour = TRUE,
                 size = 0.1, alpha =  0.5) +
  ylim(c(-0, 550)) +
  facet_grid(. ~ cyl)

ggplot(data = mtcars) +
  geom_starglyph(aes(x = mpg, y = disp, colour = cyl),
                 cols = zs, whisker = TRUE, contour = TRUE,
                 size = 0.1) +
  ylim(c(-0, 550)) +
  facet_grid(. ~ cyl)
```

`metroglyphGrob` *Draw a Metroglyph*

Description

Uses Grid graphics to draw a metroglyph (Anderson 1957; DuToit et al. 1986).

Usage

```
metroglyphGrob(
  x = 0.5,
  y = 0.5,
  z,
  size = 1,
  circle.size = 10,
  col.circle = "black",
  col.ray = "black",
  fill = NA,
  lwd.circle = 1,
  lwd.ray = 1,
  alpha = 1,
  angle.start = 0,
  angle.stop = 2 * base::pi
)
```

Arguments

x	A numeric vector or unit object specifying x-locations.
y	A numeric vector or unit object specifying y-locations.
z	A numeric vector specifying the distance of star glyph points from the center.
size	The size of rays.
circle.size	The size of the central circle.
col.circle	The circle colour.
col.ray	The colour of rays.
fill	The circle fill colour.
lwd.circle	The circle line width.
lwd.ray	The ray line width.
alpha	The alpha transparency value.
angle.start	The start angle for the glyph rays in radians. Default is zero.
angle.stop	The stop anlge for the glyph rays in radians. Default is 2π.

Value

A grobTree object.

References

Anderson E (1957). "A semigraphical method for the analysis of complex problems." *Proceedings of the National Academy of Sciences of the United States of America*, **43**(10), 923.

DuToit SHC, Steyn AGW, Stumpf RH (1986). *Graphical Exploratory Data Analysis*, Springer Texts in Statistics. Springer-Verlag, New York. ISBN 978-1-4612-9371-2, doi: 10.1007/97814612-49504, https://doi.org/10.1007/978-1-4612-4950-4.

See Also

geom_metroglyph

Examples

```
mglyph1 <- metroglyphGrob(x = 200, y = 100,
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 10)

mglyph2 <- metroglyphGrob(x = 500, y = 100,
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 25)

mglyph3 <- metroglyphGrob(x = 200, y = 300,
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 0,
                          angle.start = base::pi, angle.stop = -base::pi)

mglyph4 <- metroglyphGrob(x = 500, y = 300,
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 50,
                          angle.start = base::pi, angle.stop = -base::pi)

grid::grid.newpage()
grid::grid.draw(mglyph1)
grid::grid.draw(mglyph2)
grid::grid.draw(mglyph3)
grid::grid.draw(mglyph4)

mglyph1 <- metroglyphGrob(x = 200, y = 100,
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 10,
                          angle.start = base::pi, angle.stop = 0)

mglyph2 <- metroglyphGrob(x = 500, y = 100,
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 25,
                          angle.start = base::pi, angle.stop = 0)

mglyph3 <- metroglyphGrob(x = 200, y = 400,
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 0,
                          angle.start = 0, angle.stop = -base::pi)

mglyph4 <- metroglyphGrob(x = 500, y = 400,
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 50,
                          angle.start = 0, angle.stop = -base::pi)

grid::grid.newpage()
grid::grid.draw(mglyph1)
grid::grid.draw(mglyph2)
grid::grid.draw(mglyph3)
grid::grid.draw(mglyph4)

mglyph1 <- metroglyphGrob(x = 200, y = 100,
```

```
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 10, lwd.circle = 3)

mglyph2 <- metroglyphGrob(x = 500, y = 100,
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 25, lwd.circle = 3)

mglyph3 <- metroglyphGrob(x = 200, y = 300,
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 0,
                          angle.start = base::pi, angle.stop = -base::pi,
                          lwd.ray = 3)

mglyph4 <- metroglyphGrob(x = 500, y = 300,
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 50,
                          angle.start = base::pi, angle.stop = -base::pi,
                          lwd.ray = 3)

grid::grid.newpage()
grid::grid.draw(mglyph1)
grid::grid.draw(mglyph2)
grid::grid.draw(mglyph3)
grid::grid.draw(mglyph4)

mglyph1 <- metroglyphGrob(x = 200, y = 100,
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 10, lwd.circle = 3,
                          col.ray = RColorBrewer::brewer.pal(6, "Dark2"),
                          col.circle = "gray")

mglyph2 <- metroglyphGrob(x = 500, y = 100,
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 25, lwd.circle = 3,
                          col.ray = RColorBrewer::brewer.pal(6, "Dark2"),
                          col.circle = "white", fill = "black")

mglyph3 <- metroglyphGrob(x = 200, y = 300,
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 0,
                          angle.start = base::pi, angle.stop = -base::pi,
                          lwd.ray = 3,
                          col.ray = RColorBrewer::brewer.pal(6, "Dark2"))

mglyph4 <- metroglyphGrob(x = 500, y = 300,
                          z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                          size = 100, circle.size = 50,
                          angle.start = base::pi, angle.stop = -base::pi,
                          lwd.ray = 5, lwd.circle = 15,
                          col.ray = RColorBrewer::brewer.pal(6, "Dark2"),
                          col.circle = "white", fill = "gray")

grid::grid.newpage()
grid::grid.draw(mglyph1)
grid::grid.draw(mglyph2)
grid::grid.draw(mglyph3)
grid::grid.draw(mglyph4)
```

pieglyphGrob *Draw a Pie Glyph*

Description

Uses Grid graphics to draw a circular pie or clock glyph (Ward and Lipchak 2000; Fuchs et al. 2013).

Usage

```
pieglyphGrob(
  x = 0.5,
  y = 0.5,
  z,
  size = 1,
  edges = 200,
  col = "black",
  fill = NA,
  lwd = 1,
  alpha = 1,
  angle.start = 0,
  angle.stop = 2 * base::pi,
  scale.segment = FALSE,
  scale.radius = TRUE
)
```

Arguments

x	A numeric vector or unit object specifying x-locations.
y	A numeric vector or unit object specifying y-locations.
z	A numeric vector specifying the distance of star glyph points from the center.
size	The size of glyphs.
edges	The number of edges of the polygon to depict the circular glyph outline.
col	The colour of whisker and contours.
fill	The fill colour.
lwd	The line width.
alpha	The alpha transparency value.
angle.start	The start angle for the glyph in radians. Default is zero.
angle.stop	The stop anlge for the glyph in radians. Default is 2π.
scale.segment	logical. If TRUE, the segments (pie slices) are scaled according to value of z.
scale.radius	logical. If TRUE, the radius of segments (pie slices) are scaled according to value of z.

Value

A grobTree object.

References

Fuchs J, Fischer F, Mansmann F, Bertini E, Isenberg P (2013). "Evaluation of alternative glyph designs for time series data in a small multiple setting." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 3237–3246. ISBN 978-1-4503-1899-0.

Ward MO, Lipchak BN (2000). "A visualization tool for exploratory analysis of cyclic multivariate data." *Metrika*, **51**(1), 27–37.

Fuchs J, Fischer F, Mansmann F, Bertini E, Isenberg P (2013). "Evaluation of alternative glyph designs for time series data in a small multiple setting." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 3237–3246. ISBN 978-1-4503-1899-0.

Ward MO, Lipchak BN (2000). "A visualization tool for exploratory analysis of cyclic multivariate data." *Metrika*, **51**(1), 27–37.

See Also

geom_pieglyph

Examples

```
p1 <- pieglyphGrob(x = 150, y = 150,
             z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
             size = 50)

p2 <- pieglyphGrob(x = 300, y = 150,
                   z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                   size = 50, scale.radius = FALSE)

p3 <- pieglyphGrob(x = 450, y = 150,
                   z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                   size = 50, scale.segment = TRUE, scale.radius = FALSE)

p4 <- pieglyphGrob(x = 150, y = 350,
                   z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                   size = 50, angle.start = 0, angle.stop = -base::pi)

p5 <- pieglyphGrob(x = 300, y = 350,
                   z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                   size = 50, scale.radius = FALSE,
                   angle.start = 0, angle.stop = -base::pi)

p6 <- pieglyphGrob(x = 450, y = 350,
                   z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                   size = 50, scale.segment = TRUE, scale.radius = FALSE,
                   angle.start = 0, angle.stop = -base::pi)

grid::grid.newpage()
grid::grid.draw(p1)
grid::grid.draw(p2)
grid::grid.draw(p3)
grid::grid.draw(p4)
grid::grid.draw(p5)
grid::grid.draw(p6)

p1 <- pieglyphGrob(x = 150, y = 150,
```

```
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                    size = 50, fill = RColorBrewer::brewer.pal(6, "Dark2"))

p2 <- pieglyphGrob(x = 300, y = 150,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                    size = 50, scale.radius = FALSE,
                    fill = RColorBrewer::brewer.pal(6, "Dark2"))

p3 <- pieglyphGrob(x = 450, y = 150,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                    size = 50, scale.segment = TRUE, scale.radius = FALSE,
                    fill = RColorBrewer::brewer.pal(6, "Dark2"))

p4 <- pieglyphGrob(x = 150, y = 350,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                    size = 50, angle.start = 0, angle.stop = -base::pi,
                    fill = RColorBrewer::brewer.pal(6, "Dark2"))

p5 <- pieglyphGrob(x = 300, y = 350,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                    size = 50, scale.radius = FALSE,
                    angle.start = 0, angle.stop = -base::pi,
                    fill = RColorBrewer::brewer.pal(6, "Dark2"))

p6 <- pieglyphGrob(x = 450, y = 350,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33),
                    size = 50, scale.segment = TRUE, scale.radius = FALSE,
                    angle.start = 0, angle.stop = -base::pi,
                    fill = RColorBrewer::brewer.pal(6, "Dark2"))

grid::grid.newpage()
grid::grid.draw(p1)
grid::grid.draw(p2)
grid::grid.draw(p3)
grid::grid.draw(p4)
grid::grid.draw(p5)
grid::grid.draw(p6)
```

`profileglyphGrob` *Draw a Profile Glyph*

Description

Uses `Grid` graphics to draw a profile glyph (Chambers et al. 1983; DuToit et al. 1986).

Usage

```
profileglyphGrob(
  x = 0.5,
  y = 0.5,
  z,
  size = 1,
  col = "black",
```

```
  fill = NA,
  lwd = 1,
  alpha = 1,
  width = 10,
  flip.axes = FALSE,
  bar = TRUE,
  line = TRUE,
  mirror = TRUE
)
```

Arguments

x	A numeric vector or unit object specifying x-locations.
y	A numeric vector or unit object specifying y-locations.
z	A numeric vector specifying the distance of star glyph points from the center.
size	The size of glyphs.
col	The colour of whisker and contours.
fill	The fill colour.
lwd	The line width.
alpha	The alpha transparency value.
width	The width of the bars.
flip.axes	logical. If TRUE, axes are flipped.
bar	logical. If TRUE, profile bars are plotted.
line	logical. If TRUE, profile line is plotted.
mirror	logical. If TRUE, mirror profile is plotted.

Value

A grobTree object.

References

Chambers JM, Cleveland WS, Kleiner B, Tukey PA (1983). *Graphical Methods for Data Analysis*. Chapman and Hall/CRC, Boca Raton. ISBN 978-1-351-07230-4.

DuToit SHC, Steyn AGW, Stumpf RH (1986). *Graphical Exploratory Data Analysis*, Springer Texts in Statistics. Springer-Verlag, New York. ISBN 978-1-4612-9371-2, doi: 10.1007/97814612-49504, https://doi.org/10.1007/978-1-4612-4950-4.

See Also

geom_profileglyph

Examples

```
# mirror = TRUE
dims = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33)
barglyph <- profileglyphGrob(x = 100, y = 100, z = dims,
                             size = 100)
```

```
barprofileglyph <- profileglyphGrob(x = 300, y = 100, z = dims,
                                    size = 100, line = FALSE)

profileglyph <- profileglyphGrob(x = 500, y = 100, z = dims,
                                 size = 100, line = TRUE, bar = FALSE)
grid::grid.newpage()
grid::grid.draw(barglyph)
grid::grid.draw(barprofileglyph)
grid::grid.draw(profileglyph)

barglyph <- profileglyphGrob(x = 100, y = 250, z = dims,
                             size = 100,
                             col = "salmon")

barprofileglyph <- profileglyphGrob(x = 300, y = 250, z = dims,
                                    size = 100, line = FALSE,
                                    col = "cyan")

profileglyph <- profileglyphGrob(x = 500, y = 250, z = dims,
                                 size = 100, line = TRUE, bar = FALSE,
                                 col = "green")

grid::grid.draw(barglyph)
grid::grid.draw(barprofileglyph)
grid::grid.draw(profileglyph)

barglyph <- profileglyphGrob(x = 100, y = 400, z = dims, size = 100,
                             fill = "salmon")

barprofileglyph <- profileglyphGrob(x = 300, y = 400, z = dims,
                                    size = 100, line = FALSE,
                                    fill = "cyan")

profileglyph <- profileglyphGrob(x = 500, y = 400, z = dims, size = 100,
                                 line = TRUE, bar = FALSE,
                                 fill = "green")

grid::grid.draw(barglyph)
grid::grid.draw(barprofileglyph)
grid::grid.draw(profileglyph)


# mirror = FALSE
dims = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33)
barglyph <- profileglyphGrob(x = 100, y = 200, z = dims,
                             size = 100,
                             mirror = FALSE)

barprofileglyph <- profileglyphGrob(x = 300, y = 200, z = dims,
                                    size = 100, line = FALSE,
                                    mirror = FALSE)

profileglyph <- profileglyphGrob(x = 500, y = 200, z = dims,
                                 size = 100, line = TRUE, bar = FALSE,
                                 mirror = FALSE)
grid::grid.newpage()
grid::grid.draw(barglyph)
```

```
grid::grid.draw(barprofileglyph)
grid::grid.draw(profileglyph)

barglyph <- profileglyphGrob(x = 100, y = 350, z = dims,
                             size = 100, mirror = FALSE,
                             col = "salmon")

barprofileglyph <- profileglyphGrob(x = 300, y = 350, z = dims,
                                    size = 100, line = FALSE, mirror = FALSE,
                                    col = "cyan")

profileglyph <- profileglyphGrob(x = 500, y = 350, z = dims,
                                 size = 100, line = TRUE, bar = FALSE,
                                 mirror = FALSE, col = "green")

grid::grid.draw(barglyph)
grid::grid.draw(barprofileglyph)
grid::grid.draw(profileglyph)

barglyph <- profileglyphGrob(x = 100, y = 500, z = dims, size = 100,
                             fill = "salmon", mirror = FALSE)

barprofileglyph <- profileglyphGrob(x = 300, y = 500, z = dims,
                                    size = 100, line = FALSE, mirror = FALSE,
                                    fill = "cyan")

profileglyph <- profileglyphGrob(x = 500, y = 500, z = dims, size = 100,
                                 line = TRUE, bar = FALSE,
                                 mirror = FALSE, fill = "green")

grid::grid.draw(barglyph)
grid::grid.draw(barprofileglyph)
grid::grid.draw(profileglyph)


# mirror = TRUE, flip.axes = TRUE
dims = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33)
barglyph <- profileglyphGrob(x = 100, y = 100, z = dims,
                             size = 100, flip.axes = TRUE)

barprofileglyph <- profileglyphGrob(x = 300, y = 100, z = dims,
                                    size = 100, line = FALSE,
                                    flip.axes = TRUE)

profileglyph <- profileglyphGrob(x = 500, y = 100, z = dims,
                                 size = 100, line = TRUE, bar = FALSE,
                                 flip.axes = TRUE)
grid::grid.newpage()
grid::grid.draw(barglyph)
grid::grid.draw(barprofileglyph)
grid::grid.draw(profileglyph)

barglyph <- profileglyphGrob(x = 100, y = 250, z = dims,
                             size = 100, flip.axes = TRUE,
                             col = "salmon")

barprofileglyph <- profileglyphGrob(x = 300, y = 250, z = dims,
```

```
                                    size = 100, line = FALSE,
                                    flip.axes = TRUE,
                                    col = "cyan")

profileglyph <- profileglyphGrob(x = 500, y = 250, z = dims,
                                 size = 100, line = TRUE, bar = FALSE,
                                 flip.axes = TRUE,
                                 col = "green")

grid::grid.draw(barglyph)
grid::grid.draw(barprofileglyph)
grid::grid.draw(profileglyph)

barglyph <- profileglyphGrob(x = 100, y = 400, z = dims, size = 100,
                             flip.axes = TRUE,
                             fill = "salmon")

barprofileglyph <- profileglyphGrob(x = 300, y = 400, z = dims,
                                    size = 100, line = FALSE,
                                    flip.axes = TRUE,
                                    fill = "cyan")

profileglyph <- profileglyphGrob(x = 500, y = 400, z = dims, size = 100,
                                 line = TRUE, bar = FALSE,
                                 flip.axes = TRUE,
                                 fill = "green")

grid::grid.draw(barglyph)
grid::grid.draw(barprofileglyph)
grid::grid.draw(profileglyph)


# mirror = FALSE, flip.axes = TRUE
dims = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33)
barglyph <- profileglyphGrob(x = 100, y = 100, z = dims,
                             size = 100, flip.axes = TRUE,
                             mirror = FALSE)

barprofileglyph <- profileglyphGrob(x = 300, y = 100, z = dims,
                                    size = 100, line = FALSE,
                                    flip.axes = TRUE,
                                    mirror = FALSE)

profileglyph <- profileglyphGrob(x = 500, y = 100, z = dims,
                                 size = 100, line = TRUE, bar = FALSE,
                                 flip.axes = TRUE,
                                 mirror = FALSE)
grid::grid.newpage()
grid::grid.draw(barglyph)
grid::grid.draw(barprofileglyph)
grid::grid.draw(profileglyph)

barglyph <- profileglyphGrob(x = 100, y = 250, z = dims,
                             size = 100, mirror = FALSE,
                             flip.axes = TRUE,
                             col = "salmon")
```

```
barprofileglyph <- profileglyphGrob(x = 300, y = 250, z = dims,
                                    size = 100, line = FALSE, mirror = FALSE,
                                    flip.axes = TRUE,
                                    col = "cyan")

profileglyph <- profileglyphGrob(x = 500, y = 250, z = dims,
                                 size = 100, line = TRUE, bar = FALSE,
                                 flip.axes = TRUE,
                                 mirror = FALSE, col = "green")

grid::grid.draw(barglyph)
grid::grid.draw(barprofileglyph)
grid::grid.draw(profileglyph)

barglyph <- profileglyphGrob(x = 100, y = 400, z = dims, size = 100,
                             flip.axes = TRUE,
                             fill = "salmon", mirror = FALSE)

barprofileglyph <- profileglyphGrob(x = 300, y = 400, z = dims,
                                    size = 100, line = FALSE, mirror = FALSE,
                                    flip.axes = TRUE,
                                    fill = "cyan")

profileglyph <- profileglyphGrob(x = 500, y = 400, z = dims, size = 100,
                                 line = TRUE, bar = FALSE,
                                 flip.axes = TRUE,
                                 mirror = FALSE, fill = "green")

grid::grid.draw(barglyph)
grid::grid.draw(barprofileglyph)
grid::grid.draw(profileglyph)
```

`starglyphGrob` *Draw a Star Glyph*

Description

Uses Grid graphics to draw a star glyph (Siegel et al. 1972; Chambers et al. 1983; DuToit et al. 1986).

Usage

```
starglyphGrob(
  x = 0.5,
  y = 0.5,
  z,
  size = 1,
  col.whisker = "black",
  col.contour = "black",
  fill = NA,
  lwd.whisker = 1,
  lwd.contour = 1,
  alpha = 1,
```

```
  angle.start = 0,
  angle.stop = 2 * base::pi,
  whisker = TRUE,
  contour = TRUE
)
```

Arguments

x	A numeric vector or unit object specifying x-locations.
y	A numeric vector or unit object specifying y-locations.
z	A numeric vector specifying the distance of star gylph points from the center.
size	The size of glyphs.
col.whisker	The colour of whisker.
col.contour	The colour of contours.
fill	The fill colour.
lwd.whisker	The whisker line width.
lwd.contour	The contour line width.
alpha	The alpha transparency value.
angle.start	The start angle for the glyph in radians. Default is zero.
angle.stop	The stop anlge for the glyph in radians. Default is 2π.
whisker	logical. If TRUE, plots the star glyph whiskers.
contour	logical. If TRUE, plots the star glyph contours. glyph.

Value

A grobTree object.

References

Chambers JM, Cleveland WS, Kleiner B, Tukey PA (1983). *Graphical Methods for Data Analysis*. Chapman and Hall/CRC, Boca Raton. ISBN 978-1-351-07230-4.

DuToit SHC, Steyn AGW, Stumpf RH (1986). *Graphical Exploratory Data Analysis*, Springer Texts in Statistics. Springer-Verlag, New York. ISBN 978-1-4612-9371-2, doi: 10.1007/97814612-49504, https://doi.org/10.1007/978-1-4612-4950-4.

Siegel JH, Farrell EJ, Goldwyn RM, Friedman HP (1972). "The surgical implications of physiologic patterns in myocardial infarction shock." *Surgery*, **72**(1), 126–141.

See Also

geom_starglyph

Examples

```
sg1 <- starglyphGrob(x = 250, y = 150,
                     z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100)

sg2 <- starglyphGrob(x = 250, y = 300,
                     z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
```

```
                    lwd.whisker = 3,
                    lwd.contour = 0.01)

sg3 <- starglyphGrob(x = 250, y = 450,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    lwd.whisker = 0.1,
                    lwd.contour = 3)

sg4 <- starglyphGrob(x = 500, y = 150,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    angle.start = 0, angle.stop = -base::pi)

sg5 <- starglyphGrob(x = 500, y = 300,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    lwd.whisker = 3,
                    lwd.contour = 0.01,
                    angle.start = 0, angle.stop = -base::pi)

sg6 <- starglyphGrob(x = 500, y = 450,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    lwd.whisker = 0.1,
                    lwd.contour = 3,
                    angle.start = 0, angle.stop = -base::pi)

grid::grid.newpage()
grid::grid.draw(sg1)
grid::grid.draw(sg2)
grid::grid.draw(sg3)
grid::grid.draw(sg4)
grid::grid.draw(sg5)
grid::grid.draw(sg6)

sg1 <- starglyphGrob(x = 250, y = 150,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    fill = "salmon")

sg2 <- starglyphGrob(x = 250, y = 300,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    lwd.whisker = 3,
                    lwd.contour = 0.01,
                    fill = "cyan")

sg3 <- starglyphGrob(x = 250, y = 450,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    lwd.whisker = 0.1,
                    lwd.contour = 3,
                    fill = "green")

sg4 <- starglyphGrob(x = 500, y = 150,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    angle.start = 0, angle.stop = -base::pi,
                    fill = "salmon")

sg5 <- starglyphGrob(x = 500, y = 300,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    lwd.whisker = 3,
                    lwd.contour = 0.01,
```

```
                    angle.start = 0, angle.stop = -base::pi,
                    fill = "cyan")

sg6 <- starglyphGrob(x = 500, y = 450,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    lwd.whisker = 0.1,
                    lwd.contour = 3,
                    angle.start = 0, angle.stop = -base::pi,
                    fill = "green")

grid::grid.newpage()
grid::grid.draw(sg1)
grid::grid.draw(sg2)
grid::grid.draw(sg3)
grid::grid.draw(sg4)
grid::grid.draw(sg5)
grid::grid.draw(sg6)

sg1 <- starglyphGrob(x = 250, y = 150,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    col.whisker = RColorBrewer::brewer.pal(6, "Dark2"),
                    col.contour = "gray")

sg2 <- starglyphGrob(x = 250, y = 300,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    lwd.whisker = 3,
                    lwd.contour = 0.01,
                    col.whisker = RColorBrewer::brewer.pal(6, "Dark2"),
                    col.contour = "gray")

sg3 <- starglyphGrob(x = 250, y = 450,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    lwd.whisker = 0.1,
                    lwd.contour = 3,
                    col.whisker = RColorBrewer::brewer.pal(6, "Dark2"),
                    col.contour = "gray")

sg4 <- starglyphGrob(x = 500, y = 150,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    angle.start = 0, angle.stop = -base::pi,
                    col.whisker = RColorBrewer::brewer.pal(6, "Dark2"),
                    col.contour = "gray")

sg5 <- starglyphGrob(x = 500, y = 300,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    lwd.whisker = 3,
                    lwd.contour = 0.01,
                    angle.start = 0, angle.stop = -base::pi,
                    col.whisker = RColorBrewer::brewer.pal(6, "Dark2"),
                    col.contour = "gray")

sg6 <- starglyphGrob(x = 500, y = 450,
                    z = c(0.24, 0.3, 0.8, 1.4, 0.6, 0.33), size = 100,
                    lwd.whisker = 0.1,
                    lwd.contour = 3,
                    angle.start = 0, angle.stop = -base::pi,
                    col.whisker = RColorBrewer::brewer.pal(6, "Dark2"),
```

```
                    col.contour = "gray")

grid::grid.newpage()
grid::grid.draw(sg1)
grid::grid.draw(sg2)
grid::grid.draw(sg3)
grid::grid.draw(sg4)
grid::grid.draw(sg5)
grid::grid.draw(sg6)
```

`tileglyphGrob` *Draw a Tile Glyph*

Description

Uses Grid graphics to draw a tile glyph similar to 'autoglyph' (Beddow 1990) or 'stripe glyph' (Fuchs et al. 2013).

Usage

```
tileglyphGrob(
  x = 0.5,
  y = 0.5,
  z,
  size = 10,
  ratio = 1,
  nrow = 1,
  col = "black",
  fill = NA,
  lwd = 1,
  alpha = 1
)
```

Arguments

x	A numeric vector or unit object specifying x-locations.
y	A numeric vector or unit object specifying y-locations.
z	A numeric vector specifying the distance of star glyph points from the center.
size	The size of glyphs.
ratio	The aspect ratio (height / width).
nrow	The number of rows.
col	The line colour.
fill	The fill colour.
lwd	The line width.
alpha	The alpha transparency value.

Value

A grob object.

References

Beddow J (1990). "Shape coding of multidimensional data on a microcomputer display." In *Proceedings of the First IEEE Conference on Visualization: Visualization '90*, 238–246,. ISBN 978-0-8186-2083-6.

Fuchs J, Fischer F, Mansmann F, Bertini E, Isenberg P (2013). "Evaluation of alternative glyph designs for time series data in a small multiple setting." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 3237–3246. ISBN 978-1-4503-1899-0.

See Also

geom_tileglyph

Examples

```
tg1 <- tileglyphGrob(x = 150, y = 150,
                     z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7, 4.3),
                     size = 30)

tg2 <- tileglyphGrob(x = 450, y = 150,
                       z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                       size = 30)

tg3 <- tileglyphGrob(x = 150, y = 250,
                       z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7, 4.3),
                       size = 30, nrow = 2)

tg4 <- tileglyphGrob(x = 450, y = 250,
                       z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                       size = 30, nrow = 2)

tg5 <- tileglyphGrob(x = 150, y = 350,
                       z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7, 4.3),
                       size = 30,
                       fill = RColorBrewer::brewer.pal(7, "Dark2"))

tg6 <- tileglyphGrob(x = 450, y = 350,
                       z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                       size = 30,
                       fill = RColorBrewer::brewer.pal(7, "Dark2"))

tg7 <- tileglyphGrob(x = 150, y = 450,
                       z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7, 4.3),
                       size = 30, nrow = 2,
                       fill = RColorBrewer::brewer.pal(7, "Dark2"))

tg8 <- tileglyphGrob(x = 450, y = 450,
                       z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                       size = 30, nrow = 2,
                       fill = RColorBrewer::brewer.pal(7, "Dark2"))

grid::grid.newpage()
grid::grid.draw(tg1)
grid::grid.draw(tg2)
grid::grid.draw(tg3)
grid::grid.draw(tg4)
```

```
grid::grid.draw(tg5)
grid::grid.draw(tg6)
grid::grid.draw(tg7)
grid::grid.draw(tg8)

tg1 <- tileglyphGrob(x = 150, y = 150,
                     z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7, 4.3),
                     size = 5, ratio = 10)

tg2 <- tileglyphGrob(x = 450, y = 150,
                       z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                       size = 5, ratio = 10)

tg3 <- tileglyphGrob(x = 150, y = 250,
                       z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7, 4.3),
                       size = 5, nrow = 2, ratio = 10)

tg4 <- tileglyphGrob(x = 450, y = 250,
                       z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                       size = 5, nrow = 2, ratio = 10)

tg5 <- tileglyphGrob(x = 150, y = 350,
                       z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7, 4.3),
                       size = 5, ratio = 10,
                       fill = RColorBrewer::brewer.pal(7, "Dark2"))

tg6 <- tileglyphGrob(x = 450, y = 350,
                       z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                       size = 5, ratio = 10,
                       fill = RColorBrewer::brewer.pal(7, "Dark2"))

tg7 <- tileglyphGrob(x = 150, y = 450,
                       z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7, 4.3),
                       size = 5, nrow = 2, ratio = 10,
                       fill = RColorBrewer::brewer.pal(7, "Dark2"))

tg8 <- tileglyphGrob(x = 450, y = 450,
                       z = c(4, 3.5, 2.7, 6.8, 3.4, 5.7),
                       size = 5, nrow = 2, ratio = 10,
                       fill = RColorBrewer::brewer.pal(7, "Dark2"))

grid::grid.newpage()
grid::grid.draw(tg1)
grid::grid.draw(tg2)
grid::grid.draw(tg3)
grid::grid.draw(tg4)
grid::grid.draw(tg5)
grid::grid.draw(tg6)
grid::grid.draw(tg7)
grid::grid.draw(tg8)
```

Index

aes(), *4*
aes_(), *4*

borders(), *5*

dotglyphGrob, 1

fortify(), *5*

geom_dotprofileglyph, *2*
geom_metroglyph, *8*
geom_pieglyph, *12*
geom_profileglyph, *14*
geom_starglyph, 4, *19*
geom_tileglyph, *23*
ggplot(), *5*
Grid, *1*, *7*, *11*, *13*, *18*, *22*
grob, *2*, *22*
grobTree, *8*, *11*, *14*, *19*

metroglyphGrob, 7

pieglyphGrob, 11
profileglyphGrob, 13

starglyphGrob, *6*, 18

tileglyphGrob, 22