Package 'fillpattern'

November 18, 2026

Type Package

Title Patterned Fills for 'ggplot2' and 'grid' Graphics

Version 1.0.5

Maintainer Daniel P. Smith <dansmith01@gmail.com>

Description Provides geometric patterns to replace solid color fills in statistical graphics. These patterns ensure figures remain distinguishable when viewed by colorblind readers or when printed in black and white. The included patterns can be customized in terms of scale, rotation, color, fill, line type, and line width. Compatible with the 'ggplot2' package as well as 'grid' graphics.

URL https://cmmr.github.io/fillpattern/, https://github.com/cmmr/fillpattern

BugReports https://github.com/cmmr/fillpattern/issues

License MIT + file LICENSE

Encoding UTF-8

Config/Needs/website rmarkdown

Config/testthat/edition 3

Config/testthat/parallel true

Depends R (>= 4.2.0)

Imports ggplot2 (>= 3.5), grDevices, grid, methods, utils

Suggests ragg, testthat (>= 3.0.0)

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Daniel P. Smith [aut, cre] (ORCID: <https://orcid.org/0000-0002-2479-2044>), Alkek Center for Metagenomics and Microbiome Research [cph, fnd]

Repository CRAN

Date/Publication 2026-11-18 18:30:02 UTC

Contents

fill_pattern 2
scale_fill_pattern 5

Index **9**

`fill_pattern` *Patterned Fills for Grobs*

Description

Patterned Fills for Grobs

Usage

```
fill_pattern(
  patterns = "brick",
  fg = "black",
  bg = "transparent",
  angle = 0,
  width = 5,
  height = NA,
  lwd = 1,
  lty = "solid",
  fun = NULL,
  min_size = 2
)

fillPatternGrob(
  pattern = "brick",
  fg = "black",
  bg = "transparent",
  angle = 0,
  width = 5,
  height = NA,
  lwd = 1,
  lty = "solid",
  fun = NULL,
  min_size = 2
)
```

Arguments

`patterns`, `pattern`
: The pattern specification. Options are `"brick"`, `"chevron"`, `"fish"`, `"grid"`, `"herringbone"`, `"hexagon"`, `"octagon"`, `"rain"`, `"saw"`, `"shingle"`, `"rshingle"`, `"stripe"`, `"wave"`, and `"empty"`, optionally abbreviated and/or suffixed with modifiers. See "Pattern Names" section below. Default: `"brick"`

fg	Foreground color, for the pattern's lines, or NA to use the outline color (col) of the shape being filled. Default: "black"
bg	Background color (or grob), or NA to use the outline color (col) of the shape being filled. Default: "transparent"
angle	How much the rotate the pattern, given in degrees clockwise. Default: 0
width	The width of the pattern tile. Assumed to be millimeters unless set otherwise with unit(). Default: 5
height	The height of the pattern tile, or NA to match width. Assumed to be millimeters unless set otherwise with unit(). Default: NA
lwd	Line width. A positive number. See graphics::par() for additional details. Default: 1
lty	Line type. One of "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash". See graphics::par() for additional details. Default: "solid"
fun	A function for modifying graphical parameters immediately before rendering. Should accept two parameters: env, an environment that the function should modify, and row, the row of transformed data that ggbuild has constructed for this grob (including aes mappings). The function should return a gTree or an error to force returning from the parent function immediately, or NULL to continue processing with the updated env. Default: NULL
min_size	Minimum size of the pattern to draw. Applies to both width and height. Useful for avoiding CPU and memory overhead on tiny graphical elements. Assumed to be millimeters unless set otherwise with unit(). Default: 2

Details

fillPatternGrob() expects a single value for each parameter. fill_pattern() can accept a vector of values for each parameter which are subset or recycled as needed to obtain the same number as length(patterns).

When fg or bg is NA, the color is looked up when the pattern is drawn: it is the col of the shape the pattern fills, or "black" (for fg) / "white" (for bg) if that shape has no single, visible col. A grob that draws several shapes, such as rectGrob(x = 1:3/4, ...), only qualifies if all of them have the same col.

Value

fill_pattern() returns a list of grid::pattern() objects; fillPatternGrob() returns a grid::gTree() object.

Pattern Names

Base name:

- Pattern names must always begin with one of "brick", "chevron", "fish", "grid", "herringbone", "hexagon", "octagon", "rain", "saw", "shingle", "rshingle", "stripe", "wave", or "empty".
- "empty" draws no pattern lines, only the bg color. This is useful for a baseline or control group shown next to patterned groups. Modifiers are accepted but have no visible effect.

- These names support partial matching, e.g. `"her"`, `"herring"`, and `"herringbone"` are all valid. However, tiling designs may be added in the future, so it is recommended to use the full names in finished code.

Angle modifier:

- A number immediately following the tiling design, such as `"stripe45"`, `"fish180"`, or `"saw20"`.
- Is added to the `angle` argument; `fill_pattern("brick45", angle=45)` is equivalent to `fill_pattern("brick90")`.

Width and height modifier:

- An underscore followed by a single size to be used for both width and height.
- Or, an underscore followed by the new width and height separated by a colon.
- Can be absolute sizes (`"grid_4"` or `"hex_5mm:0.1npc"`) or relative to the `width` and `height` arguments (`"saw_sm"` or `"brick_*2:/2"`). The shorthand values `"xs"`, `"sm"`, `"md"`, `"lg"`, and `"xl"` are equivalent to `"/4"`, `"/2"`, `"1"`, `"*2"`, and `"*4"`, respectively.

Line width and style:

- An underscore, followed by a number, followed by one of `"solid"`, `"dashed"`, `"dotted"`, `"dotdash"`, `"longdash"`, or `"twodash"`. For example, `"shingle_0.5dashed"` or `"wave_2solid"`.
- The number component is optional, so `"oct_longdash"` is also valid, and will use `lwd` for the line width.
- To specify just the line width, suffix the number with "lwd": `"grid_2lwd"` will use `lty` for the line style.

Combinations:

- Modifiers can be combined in any order. For example, `"hex_lg:xl_2dotted"` or `"grid45_dashed_1.4lwd_:6mm_sm:`

See Also

`scale_fill_pattern()` for ggplot2 integration.

Examples

```
library(grid)
library(fillpattern)

grid.newpage()
grid.rect(gp = gpar(fill = fill_pattern("brick", bg = "gray", angle = 90)))

grid.newpage()
gp <- Map(gpar, fill = fill_pattern(
  patterns = c("grid_3lwd", "stripe_longdash", "herringbone45", "hexagon_lg"),
  fg       = c("black",     "white",           "black",         "blue"),
  bg       = c("white",     "black",           "cyan",          "beige") ))
grid.circle( gp = gp[[1]], x = 1/4, y = 3/4, r = 1/5)
grid.polygon(gp = gp[[2]], x = c(9,12,15)/16, y = c(15,9,15)/16)
grid.rect(   gp = gp[[3]], x = 1/4, y = 1/4, width = 2/5, height = 2/5)
```

```
grid.rect(   gp = gp[[4]], x = 3/4, y = 1/4, width = 2/5, height = 2/5)

# fg = NA picks up the outline color of each shape.
grid.newpage()
fill <- fill_pattern("stripe45", fg = NA)
grid.circle(x = 1/4, r = 1/5, gp = gpar(col = "red",  fill = fill))
grid.circle(x = 3/4, r = 1/5, gp = gpar(col = "blue", fill = fill))
```

`scale_fill_pattern` *Patterned Fills for ggplot.*

Description

Patterned Fills for ggplot.

Usage

```
scale_fill_pattern(
  patterns = seq_len,
  fg = NA,
  bg = ifelse(is.na(fg), "transparent", NA),
  fade = ifelse(is.na(fg), 1, 0.6),
  alpha = 1,
  angle = 0,
  width = unit(1/10, "npc"),
  height = NA,
  lwd = 1,
  lty = "solid",
  fun = NULL,
  min_size = 2,
  aesthetics = "fill",
  ...
)
```

Arguments

`patterns`	A vector of pattern names that will be subset or recycled as needed to match the levels of the `aes()` fill variable. If integers are provided, they are mapped to predefined patterns. See "Details" and "Pattern Names" sections below. Default: `seq_len`
`fg`	Foreground color for the pattern's lines, or `NA` to use the color scale for the `aes()` color variable. Default: `NA`
`bg`	Background color (or grob), or `NA` to use the color scale for the `aes()` color variable. Default: `ifelse(is.na(fg), "transparent", NA)`
`fade, alpha`	Modify the color from the `aes()` color scale. Fade will make it more white, and alpha will make it more transparent. Both values must be between 0 and 1, inclusive, where 1 means unchanged. Default: `fade = ifelse(is.na(fg), 1, 0.6), alpha = 1`

angle	How much the rotate the pattern, given in degrees clockwise. Default: 0
width	The width of the pattern tile. Assumed to be millimeters unless set otherwise with unit(). Default: unit(1/10, 'npc')
height	The height of the pattern tile, or NA to match width. Assumed to be millimeters unless set otherwise with unit(). Default: NA
lwd	Line width. A positive number. See graphics::par() for additional details. Default: 1
lty	Line type. One of "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash". See graphics::par() for additional details. Default: "solid"
fun	A function for modifying graphical parameters immediately before rendering. Should accept two parameters: env, an environment that the function should modify, and row, the row of transformed data that ggbuild has constructed for this grob (including aes mappings). The function should return a gTree or an error to force returning from the parent function immediately, or NULL to continue processing with the updated env. Default: NULL
min_size	Minimum size of the pattern to draw. Applies to both width and height. Useful for avoiding CPU and memory overhead on tiny graphical elements. Assumed to be millimeters unless set otherwise with unit(). Default: 2
aesthetics	The names of the aesthetics that this scale works with, for instance "pattern_fill" for geoms from the **ggpattern** package. Default: "fill"
...	Other arguments passed on to ggplot2::discrete_scale(), such as name, breaks, labels, limits, na.value, drop, or guide.

Details

All of the parameters can accept a vector of values or a function that takes n as an argument and returns the value(s) to use. The values are subset or recycled as needed to obtain the same number as nlevels(fill), where fill is the variable defined by aes(fill =).

Value

A ggplot2::discrete_scale() object.

Color Mapping

When fg or bg is NA, the pattern takes its color from the colour aesthetic of the same data row. Geoms that apply alpha to their fill with ggplot2::fill_alpha() (all of ggplot2's own geoms, and many extensions) let scale_fill_pattern() look up that row directly. For other geoms, the pattern uses the outline color (col) that the geom gives the filled shape, if there is exactly one; otherwise it uses black lines on a white background. For example, ggalluvial::geom_alluvium() only gives its flows an outline color with outline.type = "full".

Pattern Names

Base name:

- Pattern names must always begin with one of `"brick"`, `"chevron"`, `"fish"`, `"grid"`, `"herringbone"`, `"hexagon"`, `"octagon"`, `"rain"`, `"saw"`, `"shingle"`, `"rshingle"`, `"stripe"`, `"wave"`, or `"empty"`.
- `"empty"` draws no pattern lines, only the `bg` color. This is useful for a baseline or control group shown next to patterned groups. Modifiers are accepted but have no visible effect.
- These names support partial matching, e.g. `"her"`, `"herring"`, and `"herringbone"` are all valid. However, tiling designs may be added in the future, so it is recommended to use the full names in finished code.

Angle modifier:

- A number immediately following the tiling design, such as `"stripe45"`, `"fish180"`, or `"saw20"`.
- Is added to the `angle` argument; `fill_pattern("brick45", angle=45)` is equivalent to `fill_pattern("brick90")`.

Width and height modifier:

- An underscore followed by a single size to be used for both width and height.
- Or, an underscore followed by the new width and height separated by a colon.
- Can be absolute sizes (`"grid_4"` or `"hex_5mm:0.1npc"`) or relative to the `width` and `height` arguments (`"saw_sm"` or `"brick_*2:/2"`). The shorthand values `"xs"`, `"sm"`, `"md"`, `"lg"`, and `"xl"` are equivalent to `"/4"`, `"/2"`, `"1"`, `"*2"`, and `"*4"`, respectively.

Line width and style:

- An underscore, followed by a number, followed by one of `"solid"`, `"dashed"`, `"dotted"`, `"dotdash"`, `"longdash"`, or `"twodash"`. For example, `"shingle_0.5dashed"` or `"wave_2solid"`.
- The number component is optional, so `"oct_longdash"` is also valid, and will use `lwd` for the line width.
- To specify just the line width, suffix the number with "lwd": `"grid_2lwd"` will use `lty` for the line style.

Combinations:

- Modifiers can be combined in any order. For example, `"hex_lg:xl_2dotted"` or `"grid45_dashed_1.4lwd_:6mm_sm:`

See Also

`fill_pattern()` for base `grid` graphics integration.

Examples

```
library(ggplot2)
library(fillpattern)

ggplot(mpg, aes(x = class, y = hwy, color = class, fill = class)) +
  geom_boxplot() +
  scale_fill_pattern()

ggplot(mpg, aes(x = drv, y = hwy, color = drv, fill = drv)) +
```

```
  geom_violin() +
  scale_colour_brewer(palette = "Set1") +
  scale_fill_pattern(c("brick", "stripe45", "grid45_lg"), fg = "black")

ggplot(mpg, aes(x = drv, color = drv, fill = drv)) +
  geom_bar() +
  scale_fill_pattern(
    patterns = c("hex_sm", "brick90_xl", "fish"),
    lty      = c("solid", "twodash", "dotted"),
    lwd      = c(2, 3, 1) ) +
  theme(legend.key.size = unit(2, 'cm'))
```

Index

fill_pattern, 2
fill_pattern(), *7*
fillPatternGrob *(*fill_pattern*)*, 2

ggplot2::discrete_scale(), *6*
ggplot2::fill_alpha(), *6*
graphics::par(), *3*, *6*

scale_fill_pattern, 5
scale_fill_pattern(), *4*

unit(), *3*, *6*