

# Package ‘fillpattern’

July 22, 2025

**Type** Package

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

**Version** 1.0.2

**Date** 2024-06-24

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

**Description** Adds distinctive yet unobtrusive geometric patterns where solid color fills are normally used. Patterned figures look just as professional when viewed by colorblind readers or when printed in black and white. The dozen 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

**RoxygenNote** 7.3.1

**Config/Needs/website** rmarkdown

**Config/testthat/edition** 3

**Config/testthat/parallel** true

**Depends** R (>= 4.1.0)

**Imports** ggplot2 (>= 3.5), grDevices, grid (>= 4.1), methods, utils

**Suggests** ragg, testthat (>= 3.0.0)

**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** 2024-06-24 22:10:02 UTC

Contents

fill_pattern . . . . .	2
scale_fill_pattern . . . . .	5
<b>Index</b>	<b>8</b>

---

fill_pattern	<i>Patterned Fills for Grobs</i>
--------------	----------------------------------

---

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", and "wave", optionally abbreviated and/or suffixed with modifiers. See "Pattern Names" section below. Default: "brick"

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

## 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", or "wave".
- 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)
```

---

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
)
```

## Arguments

patterns	A vector of pattern names that will be subset or recycled as needed to match the levels of the <code>aes()</code> fill variable. If integers are provided, they are mapped to predefined patterns. See "Details" and "Pattern Names" sections below. Default: <code>seq_len</code>
fg	Foreground color for the pattern's lines, or NA to use the color scale for the <code>aes()</code> color variable. Default: NA
bg	Background color (or grob), or NA to use the color scale for the <code>aes()</code> color variable. Default: <code>ifelse(is.na(fg), "transparent", NA)</code>
fade, alpha	Modify the color from the <code>aes()</code> 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: <code>fade = ifelse(is.na(fg), 1, 0.6)</code> , <code>alpha = 1</code>
angle	How much to rotate the pattern, given in degrees clockwise. Default: 0
width	The width of the pattern tile. Assumed to be millimeters unless set otherwise with <code>unit()</code> . Default: <code>unit(1/10, 'npc')</code>
height	The height of the pattern tile, or NA to match width. Assumed to be millimeters unless set otherwise with <code>unit()</code> . Default: NA
lwd	Line width. A positive number. See <code>graphics::par()</code> for additional details. Default: 1

lty	Line type. One of "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash". See <code>graphics::par()</code> for additional details. Default: "solid"
fun	A function for modifying graphical parameters immediately before rendering. Should accept two parameters: <code>env</code> , an environment that the function should modify, and <code>row</code> , the row of transformed data that <code>ggbuild</code> has constructed for this grob (including aes mappings). The function should return a <code>gTree</code> or an error to force returning from the parent function immediately, or <code>NULL</code> to continue processing with the updated <code>env</code> . Default: <code>NULL</code>
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 <code>unit()</code> . Default: 2

### 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 `length(levels(fill))`, where `fill` is the variable defined by `aes(fill = )`.

### Value

A `ggplot2::discrete_scale()` 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", or "wave".
- 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:x1\_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](#)  
`graphics::par()`, [3](#), [5](#), [6](#)  
  
`scale_fill_pattern`, [5](#)  
`scale_fill_pattern()`, [4](#)  
  
`unit()`, [3](#), [5](#), [6](#)