

Package: aRt (via r-universe)

October 22, 2026

Title Generative Art with R

Version 1.5.2

Description Creates generative art using R.

License Apache License (>= 2)

URL <https://nrennie.github.io/aRt/>

Depends R (>= 4.1)

Imports cowplot, dplyr, ggchicklet, ggforce, ggfx, ggpattern, ggplot2
(>= 3.4.0), ggvoronoi, grDevices, igraph, lwgeom, patchwork,
purrr, scales, sf, stats, stringr, tibble, tidyr, treemapify,
withr

Suggests deldir, elevatr, rayshader

Remotes hrbrmstr/ggchicklet, thomasp85/ggfx, nrennie/ggvoronoi

Config/Needs/website nrennie/nrenniepkgdown

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs

libabsl-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev libgdal-dev gdal-bin libgeos-
dev libglpk-dev libharfbuzz-dev libmagick++-dev gsfonts libicu-dev libjpeg-dev libpng-
dev libtiff-dev libwebp-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://trevorworld.r-universe.dev>

Date/Publication 2026-06-07 20:11:45 UTC

RemoteUrl <https://github.com/nrennie/aRt>

RemoteRef HEAD

RemoteSha c88a69d4987b3821a9b4596c9e8a2199b2a3688d

Contents

abacus	3
black_hole	3
blending	4
boxes	5
bricks	6
bubbles	6
bullseye	7
chaos	8
crosshatch	9
disarray	10
distort	11
divide	12
fading	13
fractals	13
fragmentum	14
gradients	15
lockers	16
mesh	17
mirrored	18
mosaic	19
network	20
perpendicular	21
polygons	22
puzzles	23
rectangles	23
riley	24
rings	25
riso	26
shatter	27
smudge	28
squares	29
stack	30
stackture	30
tiles	31
vortex	32
wander	33

Index

35

abacus	<i>Abacus</i>
--------	---------------

Description

This function generates a generative art ggplot object using points and lines.

Usage

```
abacus(  
  nx = 30,  
  ny = 100,  
  max_size = 2,  
  col_palette = "black",  
  bg_col = "white",  
  s = 123  
)
```

Arguments

<code>nx</code>	Number of lines in x direction. Default 30.
<code>ny</code>	Number of points per line. Default 100.
<code>max_size</code>	Maximum size of points. Default 2.
<code>col_palette</code>	Vector of colours (or single colour). Default "black".
<code>bg_col</code>	Background colour. Default "white".
<code>s</code>	Random seed. Default 123.

Value

A ggplot object.

Examples

```
abacus()
```

black_hole	<i>Black Hole</i>
------------	-------------------

Description

This function generates a generative art ggplot object using points.

Usage

```
black_hole(
  r_max = c(50, 150, 250, 350),
  n = 10000,
  lim = 300,
  col_palette = c("#FCDE9C", "#FAA476", "#F0746E", "#E34F6F", "#DC3977", "#B9257A",
    "#7C1D6F"),
  bg_col = "black",
  size = 0.01,
  a = 0.5,
  s = 1234
)
```

Arguments

<code>r_max</code>	Vector of radii for the internal circle. Default <code>c(50, 150, 250, 350)</code> .
<code>n</code>	Number of points per circle. Default 10000.
<code>lim</code>	Numeric specifying size of grid. Default 400.
<code>col_palette</code>	Vector of colours (or single colour). Default <code>c("#FCDE9C", "#FAA476", "#F0746E", "#E34F6F", "#DC3977", "#B9257A", "#7C1D6F")</code> .
<code>bg_col</code>	Background colour. Default "black".
<code>size</code>	Size of points. Default 0.01.
<code>a</code>	Transparency of points. Default 0.5.
<code>s</code>	Random seed. Default 1234.

Value

A ggplot object.

Examples

```
black_hole()
```

 blending

Blending

Description

This function generates a generative art ggplot object using a random walk.

Usage

```
blending(n = 100, down = "white", up = "black", s = 1234)
```

Arguments

n	Number of point. Default 100.
down	Colour to use on bottom. Default "white".
up	Colour to use on top. Default "black".
s	Seed value. Default 1234.

Value

A ggplot object.

Examples

blending()

boxes	<i>Boxes</i>
-------	--------------

Description

This function generates a coloured generative art ggplot object from treemaps.

Usage

```
boxes(  
  n = 100,  
  perc = 0.1,  
  col_palette = c("#D2FBD4", "#A5DBC2", "#7BBCB0", "#559C9E", "#3A7C89", "#235D72",  
    "#123F5A"),  
  bg_col = "black",  
  s = 1234  
)
```

Arguments

n	Number of boxes
perc	Relationship between box size and colour. Value between 0 and 1 where 0 represents randomness and 1 perfect identical. Default 0.1.
col_palette	Vector of colours. Default "#D2FBD4" "#A5DBC2" "#7BBCB0" "#559C9E" "#3A7C89" "#235D72" "#123F5A".
bg_col	Background colour. Default "black".
s	Seed value. Default 1234.

Value

A ggplot object.

Examples

boxes()

bricks

*Bricks***Description**

This function generates a coloured generative art ggplot object using polygons.

Usage

```
bricks(
  n_y = 20,
  col_palette = c("#9B1D20", "#3D2B3D", "#CBEFB6", "#635D5C"),
  bg_col = "gray97",
  s = 1234
)
```

Arguments

<code>n_y</code>	Number of rows. Default 20.
<code>col_palette</code>	Vector of colours. Can be any length. Default <code>c("#9B1D20", "#3D2B3D", "#CBEFB6", "#635D5C")</code> .
<code>bg_col</code>	Background colour. Default "gray97".
<code>s</code>	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
bricks()
```

bubbles

*Bubbles***Description**

This function generates a generative art ggplot object consisting of circles filled with ellipses.

Usage

```
bubbles(
  num_circles = 20,
  main_col = "black",
  col_palette = c("#7F3C8D", "#11A579", "#3969AC", "#F2B701", "#E73F74", "#80BA5A"),
  col_prob = 0.3,
  bg_col = "white",
  s = 1234
)
```

Arguments

num_circles	Number of circles. Default 20.
main_col	Colour of non-highlighted rectangles. Default "black".
col_palette	Vector of colours. Default c("#7F3C8D", "#11A579", "#3969AC", "#F2B701", "#E73F74", "#80BA5A").
col_prob	Probability of choosing colour from col_palette instead of main_col.
bg_col	Background colour. Default "white".
s	Seed value. Default 1234.

Value

A ggplot object

Examples

```
bubbles()
```

bullseye

Bullseye

Description

This function generates a layered generative art ggplot object using polar bar charts.

Usage

```
bullseye(
  col_palette = c("#E01A4F", "#F15946", "#F9C22E", "#53B3CB", "#7DCFB6"),
  bg_col = "white",
  alpha = 0.3,
  s = 1234
)
```

Arguments

col_palette	Colour palette. Default <code>c("#E01A4F", "#F15946", "#F9C22E", "#53B3CB", "#7DCFB6")</code> .
bg_col	Background colour. Default <code>"white"</code> .
alpha	Transparency. Default <code>0.3</code> .
s	Seed value. Default <code>1234</code> .

Value

A ggplot object.

Examples

```
bullseye()
```

chaos

Chaos

Description

This function generates a generative art ggplot object using a lines, points, and circles.

Usage

```
chaos(
  n_lines = 75,
  n_points = 10,
  n_circles = 20,
  line_col = "grey70",
  point_col = "black",
  circle_col = "white",
  circle_line_col = "black",
  bg_col = "white",
  min_circle = 0.01,
  max_circle = 0.1,
  linewidth = 0.2,
  alpha = 0.5,
  size = 0.3,
  s = 1234
)
```

Arguments

n_lines	Number of lines. Default 75.
n_points	Number of points. Default 10.
n_circles	Number of circles. Default 20.
line_col	Line colour. Default "grey70".
point_col	Point colour. Default "black".
circle_col	Circle fill colour. Default "white".
circle_line_col	Circle line colour. Default "black".
bg_col	Background colour. Default "white".
min_circle	Minimum circle radius. Default 0.01.
max_circle	Maximum circle radius. Default 0.1.
linewidth	Linewidth of lines and circles. Default 0.2.
alpha	Transparency of circles. Default 0.5.
size	Size of points. Default 0.3.
s	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
chaos()
```

crosshatch

Crosshatch

Description

This function generates a coloured generative art ggplot object using overlapping lines in a grid.

Usage

```
crosshatch(
  n_x = 4,
  n_y = 4,
  n_lines = 10,
  line_overlap = 0.1,
  line_slope = 0.1,
  linewidth = 2,
  col_palette = c("#413C58", "#D1495B", "#EDA49", "#00798C", "#003D5B"),
  bg_col = "#fafafa",
  interpolate = TRUE,
  s = 1234
)
```

Arguments

n_x	Number of columns in grid. Default 4.
n_y	Number of rows in grid. Default 4.
n_lines	Number of lines per grid square. Default 10.
line_overlap	Line overlap outside grid. Default 0.1.
line_slope	Line slope within grid. Default 0.1.
linewidth	Thickness of lines. Default 2.
col_palette	Vector of colours. Default c("#6497b1", "#6a359c", "#FFB04F", "#679c35", "#cd1076").
bg_col	Background colour. Default "gray10".
interpolate	Boolean indicating if colours should be interpolated. Default TRUE.
s	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
crosshatch()
```

disarray

Disarray

Description

This function generates a generative art ggplot object using polygons

Usage

```
disarray(
  n_x = 4,
  n_y = 4,
  col_palette = c("#FF8811", "#9DD9D2", "#046E8F", "#D44D5C"),
  grid_col = "white",
  grid_width = 1,
  s = 1234
)
```

Arguments

n_x	Number of columns. Default 4.
n_y	Number of rows. Default 4.
col_palette	Vector of colours. Must be at least length 4. Default c("#FF8811", "#9DD9D2", "#046E8F", "#D44D5C")
grid_col	Colour of grid lines. Default "white".
grid_width	Linewidth of grid lines. Default 1.
s	Random seed. Default 1234.

Value

A ggplot object.

Examples

```
disarray()
```

distort	<i>Distort</i>
---------	----------------

Description

This function generates a generative art ggplot object using polygons

Usage

```
distort(
  n_x = 5,
  n_y = 5,
  deg_jitter = 0.4,
  col_palette = c("#4D4D4D", "#888888", "#AEAEAE", "#CCCCCC", "#E6E6E6"),
  line_col = "transparent",
  linewidth = 1,
  bg_col = "#333333",
  s = 1234
)
```

Arguments

n_x	Number of columns. Default 4.
n_y	Number of rows. Default 4.
deg_jitter	Degree of jitter. Default 0.4.
col_palette	Vector of colours. Must be at least length 4. Default c("#4D4D4D", "#888888", "#AEAEAE", "#CCCCCC", "#E6E6E6")

<code>line_col</code>	Colour of lines. Default "transparent".
<code>linewidth</code>	Width of lines. Default 1.
<code>bg_col</code>	Background colour. Default "#333333".
<code>s</code>	Random seed. Default 1234.

Value

A ggplot object.

Examples

```
distort()
```

<code>divide</code>	<i>Divide</i>
---------------------	---------------

Description

This function generates a coloured generative art ggplot object from intersecting lines.

Usage

```
divide(
  num_lines = 30,
  col_palette = c("#552000", "#8a4d00", "#c17d17", "#f8b150", "#f5f5f5", "#93c6e1",
    "#5f93ac", "#2e627a", "#00344a"),
  rayshade = FALSE,
  s = 1234
)
```

Arguments

<code>num_lines</code>	Number of intersecting lines. Default 30.
<code>col_palette</code>	Vector of colours. Default <code>c("#552000", "#8a4d00", "#c17d17", "#f8b150", "#f5f5f5", "#93c6e1", "#5f93ac", "#2e627a", "#00344a")</code> .
<code>rayshade</code>	Boolean determining whether the returned plot should be converted to three dimensional using rayshader. If TRUE, <code>{rayshader}</code> is required to be installed. Default FALSE.
<code>s</code>	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
divide()
```

fading

*Fading***Description**

This function generates a coloured generative art ggplot object using voronoi tiles.

Usage

```
fading(
  n_layers = 6,
  n_points = 10,
  col_palette = c("#FCDE9C", "#FAA476", "#F0746E", "#E34F6F", "#DC3977", "#B9257A",
    "#7C1D6F"),
  s = 1234
)
```

Arguments

<code>n_layers</code>	Number of layers. Default 6.
<code>n_points</code>	Number of points per layer area. Default 10.
<code>col_palette</code>	Vector of colours. Default <code>c("#FCDE9C", "#FAA476", "#F0746E", "#E34F6F", "#DC3977", "#B9257A", "#7C1D6F")</code> .
<code>s</code>	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
fading()
```

fractals

*Fractals***Description**

This function generates a generative art ggplot object using fractal patterns. Inspired by <https://www.r-bloggers.com/2010/08/fractals-in-r/>

Usage

```
fractals(
  N = 25,
  col_palette = c("#9b332b", "#b64f32", "#f7c267", "#b9b9b8", "#5d6174", "#41485f"),
  shift = 0,
  left = -1,
  right = 1,
  y_param = 3,
  resolution = 0.005,
  dist_max = 4
)
```

Arguments

N	Number of iterations. Default 25.
col_palette	Vector of colours. Default c("#9b332b", "#b64f32", "#f7c267", "#b9b9b8", "#5d6174", "#41485f").
shift	Offset of y-values. Default 0.
left	Start range of x-axis. Default -1.
right	End range of x-axis. Default 1.
y_param	Rate of y growth. Default 3.
resolution	Resolution of grid. Default 0.005.
dist_max	Size of center area. Default 4.

Value

A ggplot object.

Examples

```
fractals()
```

fragmentum

Fragmentum

Description

This function generates a coloured generative art ggplot object using randomly tessellated polygons.

Usage

```
fragmentum(
  n_x = 10,
  n_y = 10,
  deg_jitter = 0.1,
  linewidth = 0.5,
  line_col = "black",
  bg_col = "black",
  col_palette = c("#413C58", "#D1495B", "#EDAE49", "#00798C", "#003D5B"),
  s = 1234
)
```

Arguments

<code>n_x</code>	Number of polygons per row. Default 10.
<code>n_y</code>	Number of polygons per column. Default 10.
<code>deg_jitter</code>	Numeric between 0 and 0.5 specifying the degree of jitter. Default 0.1.
<code>linewidth</code>	Width of lines between polygons. Default 0.5.
<code>line_col</code>	Colour of lines between polygons. Default "black".
<code>bg_col</code>	Background colour. Default "black".
<code>col_palette</code>	Vector of colours. Can be any length. Default <code>c("#413C58", "#D1495B", "#EDAE49", "#00798C", "#003D5B")</code> .
<code>s</code>	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
fragmentum()
```

gradients

Generates generative art as a grid of gradient colour fades

Description

Generates generative art as a grid of gradient colour fades

Usage

```
gradients(nx = 4, ny = 4, bg_col = "white", linewidth = 1.5, s = 1234)
```

Arguments

<code>nx</code>	Number of columns. Default 4.
<code>ny</code>	Number of rows. Default 4.
<code>bg_col</code>	Background colour. Default "white".
<code>linewidth</code>	Linewidth. Default 1.5.
<code>s</code>	Random seed. Default 1234.

Value

Tibble

Examples

```
gradients()
```

lockers

Draw a generative locker-grid plot

Description

Builds a grid of randomly sized, optionally subdivided rectangles with rounded corners, coloured from a supplied palette.

Usage

```
lockers(
  n_col = 5,
  min_rows = 5,
  max_rows = 9,
  r = 3,
  subdivide_prob = 0.2,
  min_c = 3,
  max_c = 8,
  flip = FALSE,
  col_palette = c("#413C58", "#D1495B", "#EDAE49", "#00798C", "#003D5B"),
  bg_col = "black",
  linewidth = 1,
  s = 1234
)
```

Arguments

n_col	Number of columns. Defaults to 5.
min_rows	Minimum number of rows per column. Defaults to 5.
max_rows	Maximum number of rows per column. Defaults to 9.
r	Corner radius in points passed to <code>ggchicklet::geom_rrect()</code> . Defaults to 3.
subdivide_prob	Probability that any given cell is horizontally subdivided. Defaults to 0.2.
min_c	Minimum number of subdivisions when a cell is divided. Defaults to 3.
max_c	Maximum number of subdivisions when a cell is divided. Defaults to 8.
flip	Logical. If TRUE, axes are flipped via <code>ggplot2::coord_flip()</code> . Defaults to FALSE.
col_palette	Character vector of colours used to build the fill palette. Defaults to <code>c("#413C58", "#D1495B", "#EDA49", "#00798C", "#003D5B")</code> .
bg_col	Background and border colour. Defaults to "black".
linewidth	Width of the border lines. Defaults to 1.
s	Random seed for reproducibility. Defaults to 1234.

Value

A ggplot object.

Examples

```
lockers()
```

mesh	<i>Mesh</i>
------	-------------

Description

This function generates a coloured generative art ggplot object using rectangles.

Usage

```
mesh(
  n_x = 10,
  n_y = 10,
  col_palette = c("#0e7c7b", "#17bebb", "#d4f4dd", "#d62246", "#4b1d3f"),
  linewidth = 2,
  rayshade = FALSE
)
```

Arguments

<code>n_x</code>	Number of squares per row. Default 10.
<code>n_y</code>	Number of squares per column. Default 10.
<code>col_palette</code>	Vector of colours. Default <code>c("#0e7c7b", "#17bebb", "#d4f4dd", "#d62246", "#4b1d3f")</code> .
<code>linewidth</code>	Width of borders around squares. Default 2.
<code>rayshade</code>	Boolean determining whether the returned plot should be converted to three dimensional using rayshader. If TRUE, <code>{rayshader}</code> is required to be installed. Default FALSE.

Value

A ggplot object.

Examples

```
mesh()
```

mirrored

Mirrored

Description

This function generates a coloured generative art ggplot object from rectangles.

Usage

```
mirrored(
  n = 15,
  w = 4,
  col_palette = c("#420f75", "#7640a9", "#ad72d6", "#e7a8fb", "#f5f5f5", "#f8b150",
    "#c17d17", "#8a4d00", "#552000"),
  s = 1234
)
```

Arguments

<code>n</code>	Number of boxes per quadrant. Default 15.
<code>w</code>	Weighting towards first colour of palette. Minimum of 2. Default 4.
<code>col_palette</code>	Vector of colours. Default <code>c("#420f75", "#7640a9", "#ad72d6", "#e7a8fb", "#f5f5f5", "#f8b150", "#c17d17", "#8a4d00", "#552000")</code> .
<code>s</code>	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
mirrored()
```

mosaic	<i>Mosaic</i>
--------	---------------

Description

This function generates a generative art ggplot object from voronoi tiles.

Usage

```
mosaic(
  n = 10,
  col_palette = c("#4B3F72", "#CBB3BF", "#FFC857", "#119DA4", "#19647E"),
  line_col = "white",
  line_size = 1,
  x_means = c(0, 10, 5),
  y_means = c(0, 7, 8),
  xy_var = 2,
  s = 1234
)
```

Arguments

n	Number of points to generate tiles from. Default 100.
col_palette	Vector of colours to fill tiles with, Default c("#4B3F72", "#CBB3BF", "#FFC857", "#119DA4", "#19647E").
line_col	Colour of lines between tiles, Default "white".
line_size	Thickness of lines between tiles. Default 1.
x_means	Vector of any number of means for the x-coordinate. Default c(0, 10, 5).
y_means	Vector of any number of means for the y-coordinate. Default c(0, 7, 8).
xy_var	Numeric variance of x and y points. Default 2.
s	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
mosaic()
```

network*Network*

Description

This function generates a generative art ggplot object using a minimum spanning tree.

Usage

```
network(  
  n_x = 7,  
  n_y = 7,  
  prop = 0.3,  
  col_palette = c("#E01A4F", "#F15946", "#F9C22E", "#53B3CB", "#7DCFB6"),  
  bg_col = "white",  
  bg_line_col = "grey70",  
  line_col = "black",  
  s = 1234  
)
```

Arguments

<code>n_x</code>	Number of columns. Default 7.
<code>n_y</code>	Number of rows. Default 7.
<code>prop</code>	Proportion of squares to be nodes. Default 0.3.
<code>col_palette</code>	Colour palette. Default <code>c("#E01A4F", "#F15946", "#F9C22E", "#53B3CB", "#7DCFB6")</code> .
<code>bg_col</code>	Background colour. Default <code>"white"</code> .
<code>bg_line_col</code>	Background line colour. Default <code>"grey70"</code> .
<code>line_col</code>	Line colour. Default <code>"black"</code> .
<code>s</code>	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
network()
```

perpendicular	<i>Perpendicular</i>
---------------	----------------------

Description

This function generates a generative art ggplot object featuring multiple coloured perpendicular lines.

Usage

```
perpendicular(  
  n = 100,  
  max_length = 7,  
  linewidth = 0.5,  
  main_col = "black",  
  bg_col = "white",  
  s = 123  
)
```

Arguments

n	Number of rectangles. Default 100.
max_length	Maximum length of line. Default 7.
linewidth	Line width of line. Default 0.5.
main_col	Colour of lines. Default "black".
bg_col	Background colour. Default "white".
s	Seed value. Default 123.

Value

A ggplot object

Examples

```
perpendicular()
```

polygons

*Polygons***Description**

This function generates a coloured generative art ggplot object using polygons.

Usage

```
polygons(
  n_x = 16,
  n_y = 16,
  gap_size = 0.5,
  deg_jitter = 0.2,
  col_palette = c("#ff9062", "#fd6598", "#cb64c0", "#3294dd", "#75fb8a", "#d0eb60"),
  random = TRUE,
  bg_col = "black",
  s = 1234
)
```

Arguments

<code>n_x</code>	Number of polygons per row. Default 12.
<code>n_y</code>	Number of polygons per column. Default 18.
<code>gap_size</code>	Numeric between 0 and 1 specifying the size of the gap in the polygons. Default 0.5.
<code>deg_jitter</code>	Numeric between 0 and 0.5 specifying the degree of jitter. Default 0.2.
<code>col_palette</code>	Vector of colours. Can be any length. Default <code>c("#ff9062", "#fd6598", "#cb64c0", "#3294dd", "#75fb8a", "#d0eb60")</code> .
<code>random</code>	Boolean for whether colours should be random or ordered. Default TRUE.
<code>bg_col</code>	Background colour. Default "black".
<code>s</code>	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
polygons()
```

puzzles

Puzzles

Description

This function generates a coloured generative art ggplot object from treemaps.

Usage

```
puzzles(  
  n = 200,  
  num_groups = 30,  
  col_palette = c("#0e7c7b", "#17bebb", "#d4f4dd", "#d62246", "#4b1d3f"),  
  bg_col = "white",  
  s = 1234  
)
```

Arguments

n	Number of boxes. Default 200.
num_groups	Number of larger boxes. Default 30.
col_palette	Vector of colours. Default c("#0e7c7b", "#17bebb", "#d4f4dd", "#d62246", "#4b1d3f").
bg_col	Background colour. Default "white".
s	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
puzzles()
```

rectangles

Rectangles

Description

This function generates a generative art ggplot object featuring multiple coloured rectangles.

Usage

```
rectangles(
  n = 100,
  max_height = 7,
  max_width = 5,
  size = 2,
  main_col = "lightgrey",
  col_palette = c("#80BA5A", "#E68310", "#008695", "#CF1C90", "#F97B72", "#4B4B8F"),
  bg_col = "white",
  weight = 0.25,
  s = 1234
)
```

Arguments

n	Number of rectangles. Default 100.
max_height	Maximum height of rectangle. Default 7.
max_width	Maximum width of rectangle. Default 5.
size	Line width of rectangles. Default 2.
main_col	Colour of non-highlighted rectangles. Default "lightgrey".
col_palette	Vector of colours. Default c("#80BA5A", "#E68310", "#008695", "#CF1C90", "#F97B72", "#4B4B8F").
bg_col	Background colour. Default "white".
weight	Sampling weight. Default 0.25.
s	Seed value. Default 1234.

Value

A ggplot object

Examples

```
rectangles()
```

riley

Riley

Description

This function generates a coloured generative art ggplot object from intersecting lines. Inspired by Balm by Bridget Riley.

Usage

```
riley(n_x = 9, n_y = 9, offset = 3, main_col = "black", bg_col = "white")
```

Arguments

n_x	Number of circles in x-direction. Default 9.
n_y	Number of circles in y-direction. Default 9.
offset	Numeric giving offset of y-axis. Default 3.
main_col	Main colour. Default "black".
bg_col	Background colour. Default "white".

Value

A ggplot object.

Examples

```
riley()
```

rings	<i>Rings</i>
-------	--------------

Description

This function generates a generative art ggplot object using pie charts and circles.

Usage

```
rings(
  col_palette = c("#413C58", "#D1495B", "#EDAE49", "#00798C", "#003D5B"),
  bg_col = "#343046",
  x_ring = c(0.2, 0.9),
  y_ring = c(0.2, 1.8),
  r_ring = c(0.6, 0.4),
  x0 = c(0, 1),
  y0 = c(0, 2),
  r = c(1, 0.7),
  n = c(80, 80),
  s = 1234
)
```

Arguments

col_palette	Vector of colours. Default c("#413C58", "#D1495B", "#EDAE49", "#00798C", "#003D5B").
bg_col	Background colour. Default "#343046".
x_ring	Vector of x-co-ordinates for centre of gaps. Default c(0.2, 0.9).
y_ring	Vector of y-co-ordinates for centre of gaps. Default c(0.2, 1.8).
r_ring	Vector of radii for centre of gaps. Default c(0.6, 0.4).

<code>x0</code>	Vector of x-co-ordinates for pie charts. Default <code>c(0, 1)</code> .
<code>y0</code>	Vector of y-co-ordinates for pie charts. Default <code>c(0, 2)</code> .
<code>r</code>	Vector of radii for pie charts. Default <code>c(1, 0.7)</code> .
<code>n</code>	Vector of slices per pie chart. Default <code>c(80, 80)</code> .
<code>s</code>	Seed value. Default 1234.

Value

A ggplot object.

Examples

`rings()`

<code>riso</code>	<i>Riso</i>
-------------------	-------------

Description

This function generates a coloured generative art ggplot object using overlapping semi-transparent circles.

Usage

```
riso(  
  n_x = 4,  
  n_y = 4,  
  n_circles = 2,  
  r = 0.5,  
  jitter_x = 0.25,  
  jitter_y = 0.3,  
  alpha = 0.5,  
  col_palette = c("#6497b1", "#6a359c", "#FFB04F", "#679c35", "#cd1076"),  
  circle_bg = "#fafafa",  
  bg_col = "gray10",  
  interpolate = TRUE,  
  s = 1234  
)
```

Arguments

<code>n_x</code>	Number of columns in grid. Default 4.
<code>n_y</code>	Number of rows in grid. Default 4.
<code>n_circles</code>	Number of circles per grid square. Default 2.
<code>r</code>	Radius of circles. Default 0.5.

jitter_x	Jitter in x direction for circle positions. Default 0.25.
jitter_y	Jitter in y direction for circle positions. Default 0.3.
alpha	Transparency of circles. Default 0.5.
col_palette	Vector of colours. Default c("#6497b1", "#6a359c", "#FFB04F", "#679c35", "#cd1076").
circle_bg	Background colour of circles. Default "#fafafa".
bg_col	Background colour. Default "gray10".
interpolate	Boolean indicating if colours should be interpolated. Default TRUE.
s	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
riso()
```

shatter	<i>Shatter</i>
---------	----------------

Description

This function generates a generative art ggplot object using polygons.

Usage

```
shatter(
  n_x = 25,
  n_y = 25,
  decay = 0.8,
  col_palette = "black",
  bg_col = "gray97",
  s = 1234
)
```

Arguments

n_x	Number of polygons per row. Default 25.
n_y	Number of polygons per column. Default 25.
decay	Numeric between 0 and 1 specifying the rate of decay if square shapes. Default 0.8.
col_palette	colour palette. Default "black".
bg_col	Single colour for background. Default "gray97".
s	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
shatter()
```

smudge	<i>Smudge</i>
--------	---------------

Description

This function generates a coloured generative art ggplot object from contours.

Usage

```
smudge(
  n = 25,
  binwidth = 0.01,
  col_palette = c("#552000", "#8a4d00", "#c17d17", "#f8b150", "#f5f5f5", "#93c6e1",
    "#5f93ac", "#2e627a", "#00344a"),
  s = 1234
)
```

Arguments

n	Number of grid boxes. Default 25.
binwidth	Binwidth for colours. Default 0.01.
col_palette	Vector of colours. Default c("#552000", "#8a4d00", "#c17d17", "#f8b150", "#f5f5f5", "#93c6e1", "#5f93ac", "#2e627a", "#00344a").
s	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
smudge()
```

squares*Squares*

Description

This function generates a generative art ggplot object using pattern-filled squares

Usage

```
squares(  
  n = 7,  
  line_col = "white",  
  pattern_col = "white",  
  pattern_fill = "black",  
  pattern_size = 0.4,  
  linewidth = 1.5,  
  s = 1234  
)
```

Arguments

n	Number of squares per row. Default 7
line_col	Colour of lines between squares. Default "white".
pattern_col	Colour of pattern lines. Default "white".
pattern_fill	Colour of pattern background. Default "black".
pattern_size	Size of pattern. Default 0.4.
linewidth	Size of lines between squares. Default 1.5.
s	Random seed. Default 1234.

Value

A ggplot object.

Examples

```
squares()
```

stack

Stack

Description

This function generates an sf object.

Usage

```
stack(x0, y0, min_height, max_height, min_width, max_width)
```

Arguments

x0	x-coordinate of middle of rectangle.
y0	y-coordinate of middle of rectangle.
min_height	minimum height.
max_height	maximum height.
min_width	minimum width.
max_width	maximum width.

Value

An sf object.

stackture

Stackture

Description

This function generates a coloured generative art ggplot object using overlapping semi-transparent circles.

Usage

```
stackture(
  n_x = 8,
  n_y = 8,
  min_height = 1,
  max_height = 1.5,
  min_width = 1,
  max_width = 1.5,
  interpolate = TRUE,
  col_palette = c("#A053A1", "#DB778F", "#E69F52", "#09A39A", "#5869C7"),
  bg_col = "#004B67",
  alpha = 1,
  s = 1234
)
```

Arguments

<code>n_x</code>	Number of columns in grid. Default 8.
<code>n_y</code>	Number of rows in grid. Default 8.
<code>min_height</code>	minimum height.
<code>max_height</code>	maximum height.
<code>min_width</code>	minimum width.
<code>max_width</code>	maximum width.
<code>interpolate</code>	Boolean indicating if colours should be interpolated. Default TRUE.
<code>col_palette</code>	Vector of colours. Default <code>c("#A053A1", "#DB778F", "#E69F52", "#09A39A", "#5869C7")</code> .
<code>bg_col</code>	Background colour. Default <code>"#004B67"</code> .
<code>alpha</code>	Transparency. Default 1.
<code>s</code>	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
stackture()
```

<code>tiles</code>	<i>Tiles</i>
--------------------	--------------

Description

This function generates a coloured generative art ggplot object using square polygons.

Usage

```
tiles(
  n_x = 12,
  n_y = 12,
  col_palette = c("#67322e", "#99610a", "#6e948c", "#2c6b67", "#122c43"),
  rayshade = FALSE,
  shadow_intensity = 0.5,
  sunangle = 315,
  s = 1234
)
```

Arguments

n_x	Number of polygons per row. Default 12.
n_y	Number of polygons per column. Default 18.
col_palette	Vector of colours. Default c("#67322e", "#99610a", "#6e948c", "#2c6b67", "#122c43") which is the Veronese palette from MetBrewer.
rayshade	Boolean determining whether the returned plot should be converted to three dimensional using rayshader. If TRUE, {rayshader} is required to be installed. Default FALSE.
shadow_intensity	Intensity of shading for 3D elements, Default 0.5.
sunangle	Angle of the sun. Default 315.
s	Seed value. Default 1234.

Value

A ggplot object.

Examples

tiles()

vortex	<i>Vortex</i>
--------	---------------

Description

This function generates a greyscale or rainbow coloured generative art ggplot object shaped like a vortex.

Usage

```
vortex(  
  n = 25,  
  start_val = 90,  
  col_palette = c("#700460", "#a02c5d", "#ec0f47", "#ee6b3b", "#fbbf54", "#abd96d",  
    "#15c286", "#087353", "#045459", "#4b0082"),  
  bg_col = "black",  
  s = 1234  
)
```

Arguments

<code>n</code>	Number of points. Default 25.
<code>start_val</code>	Starting position for polar coordinates. Default 90.
<code>col_palette</code>	Colour palette. Default <code>c("#700460", "#a02c5d", "#ec0f47", "#ee6b3b", "#fbbf54", "#abd96d", "#15c286", "#087353", "#045459", "#4b0082")</code> .
<code>bg_col</code>	Background colour. Default "black".
<code>s</code>	Seed value. Default 1234.

Value

A ggplot object.

Examples

```
vortex()
```

wander	<i>Wander</i>
--------	---------------

Description

This function generates a generative art ggplot object using random walks

Usage

```
wander(
  n_lines = 100,
  n_points = 350,
  r_outer = 8,
  r_inner = 3,
  line_var = 0.01,
  deg_jitter = 0.1,
  linewidth = 0.1,
  bg_col = "#462255",
  col_palette = c("#FF8811", "#9DD9D2", "#046E8F", "#D44D5C"),
  n_cols = 20,
  s = 123
)
```

Arguments

<code>n_lines</code>	Number of lines. Default 100.
<code>n_points</code>	Number of points. Default 350.
<code>r_outer</code>	Radius of outer circle. Default 8.
<code>r_inner</code>	Radius of inner circle. Default 3.

<code>line_var</code>	Variance of random walk noise. Default 0.01.
<code>deg_jitter</code>	Degree of jitter for multiple lines. Default 0.1.
<code>linewidth</code>	Width of lines. Default 0.1.
<code>bg_col</code>	Background colour. Default "#462255".
<code>col_palette</code>	Vector of colours. Default <code>c("#FF8811", "#9DD9D2", "#046E8F", "#D44D5C")</code>
<code>n_cols</code>	Number of colours to create. Default 20.
<code>s</code>	Random seed. Default 123.

Value

A ggplot object.

Examples

```
wander()
```

Index

abacus, [3](#)

black_hole, [3](#)
blending, [4](#)
boxes, [5](#)
bricks, [6](#)
bubbles, [6](#)
bullseye, [7](#)

chaos, [8](#)
crosshatch, [9](#)

disarray, [10](#)
distort, [11](#)
divide, [12](#)

fading, [13](#)
fractals, [13](#)
fragmentum, [14](#)

ggchicklet::geom_rrect(), [17](#)
ggplot2::coord_flip(), [17](#)
gradients, [15](#)

lockers, [16](#)

mesh, [17](#)
mirrored, [18](#)
mosaic, [19](#)

network, [20](#)

perpendicular, [21](#)
polygons, [22](#)
puzzles, [23](#)

rectangles, [23](#)
riley, [24](#)
rings, [25](#)
riso, [26](#)

shatter, [27](#)

smudge, [28](#)
squares, [29](#)
stack, [30](#)
stackture, [30](#)

tiles, [31](#)

vortex, [32](#)

wander, [33](#)