Package: aRt (via r-universe)

November 2, 2024

Title Generative Art with R

Version 1.4.7

Description Creates generative art using R.

License Apache License (>= 2)

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

Depends R (>= 4.1)

Imports ape, cowplot, dplyr, ggforce, ggfx, ggpattern, ggplot2 (>= 3.4.0), ggraph, ggvoronoi, grDevices, igraph, lwgeom, magick, patchwork, purrr, sf, stats, stringr, terra, tibble, tidygraph, tidyr, tidyterra, treemapify

Suggests ambient, deldir, elevatr, ggstream, MetBrewer, particles, PrettyCols, rayshader, rcartocolor, RColorBrewer

Remotes thomasp85/ggfx, nrennie/ggvoronoi

Config/Needs/website nrennie/nrenniepkgdown

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Repository https://trevorld.r-universe.dev

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

RemoteRef HEAD

RemoteSha d9b526a5e831f33914733bd8694ac4bc2798eccf

Contents

`abacus` *Abacus*

Description

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

Usage

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

Arguments

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

Value

A ggplot object.

attraction *Attraction*

Description

This function generates a generative art ggplot object using strange attractors.

Usage

```
attraction(
  n = 5000,
  a = -3,
  b = 1,
  c = 0.5,
  d = -1,
  main_col = "black",
  bg_col = "white"
)
```

Arguments

n	Number of points. Default 5000.
a	Parameter. Default -3.
b	Parameter. Default 1.
c	Parameter. Default 0.5.
d	Parameter. Default -1.
main_col	Vector of colours (or single colour). Default "black".
bg_col	Background colour. Default "white".

Value

A ggplot object.

black_hole *Black Hole*

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,
  main_cols = rcartocolor::carto_pal(n = 7, name = "SunsetDark"),
  bg_col = "black",
  size = 0.01,
  a = 0.5,
  s = 1234
)
```

Arguments

`r_max`	Vector of radii for the internal circle. Default `c(50, 150, 250, 350)`.
`n`	Number of points per circle. Default 10000.
`lim`	Numeric specifying size of grid. Default 400.
`main_cols`	Vector of colours (or single colour). Default `rcartocolor::carto_pal(n = 7, name = "SunsetDark")`.
`bg_col`	Background colour. Default "black".
`size`	Size of points. Default 0.01.
`a`	Transparency of points. Default 0.5.
`s`	Random seed. Default 1234.

Value

A ggplot object.

`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.

---

boxes *Boxes*

Description

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

Usage

```
boxes(
  n = 100,
  perc = 0.1,
  col_palette = rcartocolor::carto_pal(n = 7, "DarkMint"),
  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 "DarkMint" colour palette from rcartocolor.
bg_col	Background colour. Default "black".
s	Seed value. Default 1234.

Value

A ggplot object.

---

bricks *Bricks*

Description

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

Usage

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

Arguments

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

Value

A ggplot object.

`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 = rcartocolor::carto_pal(n = 12, "Bold"),
  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 "Bold" colour palette from rcartocolor. Must have 12 colours.
`bg_col`	Background colour. Default "white".
`s`	Seed value. Default 1234.

Value

A ggplot object

`bullseye` *Bullseye*

Description

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

Usage

```
bullseye(main_col = "black", bg_col = "white", s = 1234)
```

Arguments

`main_col`	Colour scheme of art. One of `c("mono", "rainbow)`. Default "black".
`bg_col`	Background colour. Default "white".
`s`	Seed value. Default 1234.

Value

A ggplot object.

`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.

`circles` *Circles*

Description

This function generates a coloured generative art ggplot object using dendograms and circular graphs.

Usage

```
circles(
  n = 100,
  smoothness = 100,
  col_palette = rcartocolor::carto_pal(n = 12, "Bold"),
  line_col = NA,
  bg_col = "black",
  s = 1234
)
```

Arguments

`n`	Number of nodes. Default 10.
`smoothness`	Smoothness of lines on circles. Default 100.
`col_palette`	Vector of colours. Default "Bold" colour palette from rcartocolor.
`line_col`	Background colour. Default NA.
`bg_col`	Background colour. Default "black".
`s`	Seed value. Default 1234.

Value

A ggplot object.

`circular` *Circular*

Description

This function generates an abstract circular generative art ggplot object.

Usage

```
circular(n = 100, main_col = "black", bg_col = "white", s = 56)
```

Arguments

`n`	Number of steps from inside to outside. Default 100.
`main_col`	Colour of lines. Default black.
`bg_col`	Background colour. Default white.
`s`	Seed value. Default 56.

Value

A ggplot object.

connected *Connected*

Description

This function generates a generative art ggplot object connecting points through arcs on a circle.

Usage

```
connected(
  n = 100,
  n_geom = 2,
  random = FALSE,
  col_palette = RColorBrewer::brewer.pal(n = 9, "RdPu"),
  bg_col = "#ae217e",
  s = 1234
)
```

Arguments

`n`	Number of lines to start. Default 100.
`n_geom`	Number of points along path to create. Default 2.
`random`	Boolean value for whether to randomise plotting order of edges.
`col_palette`	Vector of colours. Default "RdPu" colour palette from RColorBrewer.
`bg_col`	Background colour. Default "#ae217e".
`s`	Seed value. Default 1234.

Value

A ggplot object.

contours *Contours*

Description

This function generates a generative art ggplot object based on elevation data of a geographic area

Usage

```
contours(
  xmin = -3.27,
  xmax = -3.15,
  ymin = 54.43,
  ymax = 54.49,
  col_palette = MetBrewer::met.brewer("Hokusai3"),
  res = 50,
  line_col = "#0a2e57",
  linewidth = 0.5
)
```

Arguments

`xmin`	Minimum x value og geographic area. Default -3.27.
`xmax`	Maximum x value og geographic area. Default -3.15.
`ymin`	Minimum y value og geographic area. Default 54.43.
`ymax`	Maximum y value og geographic area. Default 54.49.
`col_palette`	Colour palette to colour in contours. Default MetBrewer::met.brewer("Hokusai3").
`res`	Resolution of elevation. Default 50.
`line_col`	Colour of contours. Default "#0a2e57".
`linewidth`	Width of contours. 0.5.

Value

A ggplot object.

`crawling` *Crawling*

Description

This function generates a generative art ggplot object using dendograms.

Usage

```
crawling(
  n = 50,
  edge_colour = "black",
  node_size = 1,
  node_colour = "black",
  bg_col = "white",
  s = 1234
)
```

Arguments

n	Number of nodes.
edge_colour	Edge colour. Default "black".
node_size	Node size. Default 1.
node_colour	Node colour. Default "black".
bg_col	Background colour. Default "white".
s	Seed value. Default 1234.

Value

A ggplot object.

criss_cross *Criss-cross*

Description

This function generates a generative art ggplot object using intersecting lines.

Usage

```
criss_cross(
  n = 25,
  bg_col = "white",
  line_col = "grey50",
  linewidth = 0.1,
  outline_col = "black",
  outline_width = 1.5,
  s = 1234
)
```

Arguments

n	Number of lines per corner. Default 25.
bg_col	Background colour. Default "white".
line_col	Vector of colours (or single colour). Default "grey50".
linewidth	Width of lines. Default 0.1.
outline_col	Outline colour. Default "black".
outline_width	Linewidth of outline. Default 1.5.
s	Random seed. Default 1234.

Value

A ggplot object.

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", "#EDAE49", "#00798C", "#003D5B"),
  bg_col = "#fafafa",
  interpolate = TRUE,
  s = 123
)
```

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 123.

Value

A ggplot object.

`divide` *Divide*

Description

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

Usage

```
divide(
  num_lines = 30,
  col_palette = PrettyCols::prettycols("TangerineBlues"),
  rayshade = FALSE,
  s = 1234
)
```

Arguments

`num_lines` Number of intersecting lines. Default `30`.

`col_palette` Vector of colours. Default `PrettyCols::prettycols("TangerineBlues")`.

`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`.

`s` Seed value. Default `1234`.

Value

A ggplot object.

`dots` *Dots*

Description

This function generates a coloured generative art ggplot object using polar coordinates.

Usage

```
dots(
  n_x = 50,
  n_y = 100,
  jitter_size_width = 0.5,
  jitter_size_height = 0.5,
  col_palette = rcartocolor::carto_pal(n = 7, "Purp"),
  bg_col = "#63589f",
  s = 1234
)
```

Arguments

`n_x`	Number of rotational points. Default 50.
`n_y`	Number of outwards points. Default 100.
`jitter_size_width`	Size of jitter width. Default 0.5.
`jitter_size_height`	Size of jitter height. Default 0.5.
`col_palette`	Vector of colours. Default "Purp" colour palette from rcartocolor.
`bg_col`	Background colour. Default "#63589f".
`s`	Seed value. Default 1234.

Value

A ggplot object.

`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 = rcartocolor::carto_pal(n = 7, "SunsetDark"),
  s = 1234
)
```

Arguments

`n_layers`	Number of layers. Default 6.
`n_points`	Number of points per layer area. Default 10.
`col_palette`	Vector of colours. Default "SunsetDark" colour palette from rcartocolor.
`s`	Seed value. Default 1234.

Value

A ggplot object.

`flow_fields` *Flow fields*

Description

This function generates a generative art ggplot object from using particle traces. See also https://www.williamrchase.com/post fields-12-months-of-art-september/

Usage

```
flow_fields(
  n = 10000,
  granularity = 1000,
  x_freq = 1,
  y_freq = 1,
  alpha = 1,
  line_col = c("#edf8fb", "#bfd3e6", "#9ebcda", "#8c96c6", "#8c6bb1", "#88419d",
    "#6e016b"),
  bg_col = "lightgrey",
  s = 1234
)
```

Arguments

`n`	Number of lines. Default 10000.
`granularity`	How fine to draw the grid. Default 1000.
`x_freq`	Frequency of x simplex noise. Default 1.
`y_freq`	Frequency of y simplex noise. Default 1.
`alpha`	Transparency of lines. Default 1.
`line_col`	Line colours. Vector (or single element) of colours. Default c("#edf8fb","#bfd3e6","#9ebcda","#8c96c6",
`bg_col`	Background colour. Default "white".
`s`	Seed value. Default 1234.

Value

A ggplot object.

`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 = MetBrewer::met.brewer("Demuth", n = 25),
  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 of length >= N. Default MetBrewer::met.brewer("Demuth", n = 25).

`shift` Offset of y-values. Default 0.

`left` Start rangle 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.

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 = 5, ny = 5, s = 1234)
```

Arguments

nx	Number of columns. Default 5.
ny	Number of rows. Default 5.
s	Random seed. Default 1234.

Value

Tibble

heart *Heart*

Description

This function generates a greyscale or rainbow coloured generative art ggplot object in the shape of a heart.

Usage

```
heart(n = 25, col_scheme = "mono", bg_col = "black", s = 1234)
```

Arguments

n	Number of lines per colour. Default 25.
col_scheme	Colour scheme of art. One of c("mono", "rainbow). Default 0.1.
bg_col	Background colour. Default "black".
s	Seed value. Default 1234.

Value

A ggplot object.

imploding_hexagon *Imploding Hexagon*

Description

This function generates a generative art ggplot object using spatial hexagons and Magick.

Usage

```
imploding_hexagon(
  n = 25000,
  size = 0.05,
  bg_col = "grey10",
  col_palette = grDevices::grey.colors(n = 10, start = 0.1, end = 0.9),
  random = TRUE,
  implode_factor = 0.5,
  s = 1234
)
```

Arguments

`n`	Number of points. Default 25000.
`size`	Size of points. Default 0.1.
`bg_col`	Background colour. Default "grey10".
`col_palette`	Colour palette. Default `grDevices::grey.colors(n = 10, start = 0.1, end = 0.7)`.
`random`	Boolean. Should colours be arranged randomly or not? Default `TRUE`.
`implode_factor`	Image implode factor. Default 0.5.
`s`	Random seed. Default 123.

Value

A ggplot object.

infinity *Infinity*

Description

This function generates a greyscale or rainbow coloured generative art ggplot object in the shape of an infinity symbol.

Usage

```
infinity(n = 25, col_scheme = "mono", bg_col = "black", s = 1234)
```

Arguments

`n`	Number of lines per colour. Default 25.
`col_scheme`	Colour scheme of art. One of c("mono", "rainbow). Default 0.1.
`bg_col`	Background colour. Default "black".
`s`	Seed value. Default 1234.

Value

A ggplot object.

`make_lines` *Make lines*

Description

This function generates smooth random walks from a normal distribution.

Usage

```
make_lines(res = 100)
```

Arguments

`res`	Resolution of grid. Default 100.

Value

A numeric vector

make_square *Generates a tibble of line segments and colours*

Description

Generates a tibble of line segments and colours

Usage

```
make_square(xmin, xmax, ymin, ymax, low, high)
```

Arguments

xmin	x-min coord
xmax	x-max coord
ymin	y-min coord
ymax	y-max coord
low	Hex value for left colour
high	Hex value for right colour

Value

Tibble

mirrored *Mirrored*

Description

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

Usage

```
mirrored(
  n = 15,
  w = 4,
  col_palette = PrettyCols::prettycols("PurpleTangerines"),
  s = 1234
)
```

Arguments

n	Number of boxes per quadrant. Default 15.
w	Weighting towards first colour of palette. Minimum of 2. Default 4.
col_palette	Vector of colours. Default PrettyCols::prettycols("PurpleTangerines").
s	Seed value. Default 1234.

Value

A ggplot object.

---

moire *Moiré*

Description

This function creates generative art inspired by the Moiré effect.

Usage

```
moire(
  inner_n = 20,
  dist = 10,
  inner_col = "grey40",
  outer_col = "grey60",
  bg_col = "grey10",
  inner_r = 0.5,
  outer_r = 0.2
)
```

Arguments

`inner_n`	Number of bottom left circles. Default 20.
`dist`	Distance between bottom left and top right blocks. Default 4.
`inner_col`	Colour of bottom left circles. Default "grey40".
`outer_col`	Colour of top right circles. Default "grey60".
`bg_col`	Background colour. Default "grey10".
`inner_r`	Radius of bottom left circles. Default 0.5.
`outer_r`	Radius of top right circles. Default 0.2.

Value

A ggplot object.

mosaic *Mosiac*

Description

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

Usage

```
mosaic(
  n = 100,
  fill_cols = c("#4B3F72", "#CBB3BF", "#FFC857", "#119DA4", "#19647E"),
  line_col = "white",
  bg_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.
fill_cols	Vector of colours to fill tiles with, Default c("#4B3F72", "#CBB3BF", "#FFC857", "#119DA4", "#19647E").
line_col	Colour of lines between tiles, Default "white".
bg_col	Background colour. 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.

`mosaic_sketch` *Mosiac Sketch*

Description

This function generates a generative art ggplot object from voronoi tiles in a hand-sketched look.

Usage

```
mosaic_sketch(
  n = 10,
  fill_cols = c("#4B3F72", "#CBB3BF", "#FFC857", "#119DA4", "#19647E"),
  line_col = "white",
  bg_col = "white",
  line_size = 2,
  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.
`fill_cols`	Vector of colours to fill tiles with, Default c("#4B3F72", "#CBB3BF", "#FFC857", "#119DA4", "#19647E").
`line_col`	Colour of lines between tiles, Default "white".
`bg_col`	Background colour. 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 varaince of x and y points. Default 2.
`s`	Seed value. Default 1234.

Value

A ggplot object.

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

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

Value

A ggplot object.

perpendicular *Perpendicular*

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

polygons *Polygons*

Description

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

Usage

```
polygons(
  n_x = 12,
  n_y = 18,
  gap_size = 0.5,
  deg_jitter = 0.1,
  colours = c("#9B1D20", "#3D2B3D", "#CBEFB6", "#635D5C"),
  rand = FALSE,
  bg_col = "gray97",
  s = 1234
)
```

Arguments

`n_x`	Number of polygons per row. Default 12.
`n_y`	Number of polygons per column. Default 18.
`gap_size`	Numeric between 0 and 1 specifying the size of the gap in the polygons. Default 0.5.
`deg_jitter`	Numeric between 0 and 0.5 specifying the degree of jitter. Default 0.1.
`colours`	Vector of colours. Can be any length. Default `c("#9B1D20", "#3D2B3D", "#CBEFB6", "#635D5C")`.
`rand`	Boolean for whether colours should be random or ordered. Default FALSE.
`bg_col`	Background colour. Default "gray97".
`s`	Seed value. Default 1234.

Value

A ggplot object.

puzzles *Puzzles*

Description

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

Usage

```
puzzles(
  n = 200,
  num_groups = 30,
  col_palette = PrettyCols::prettycols("Beach"),
  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 PrettyCols::prettycols("Beach").
bg_col	Background colour. Default "white".
s	Seed value. Default 1234.

Value

A ggplot object.

random_hex *Generates a random hex code*

Description

Generates a random hex code

Usage

```
random_hex(n)
```

Arguments

n	Number of hex codes to generate

Value

Character string of hex codes

random_tessellation *Random Tessellation*

Description

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

Usage

```
random_tessellation(
  n_x = 10,
  n_y = 10,
  deg_jitter = 0.1,
  linewidth = 0.5,
  line_col = "black",
  bg_col = "black",
  col_palette = PrettyCols::prettycols("Lively"),
  s = 1234
)
```

Arguments

`n_x`	Number of polygons per row. Default 10.
`n_y`	Number of polygons per column. Default 10.
`deg_jitter`	Numeric between 0 and 0.5 specifying the degree of jitter. Default 0.1.
`linewidth`	Width of lines between polygons. Default 0.5.
`line_col`	Colour of lines between polygons. Default "black".
`bg_col`	Background colour. Default "black".
`col_palette`	Vector of colours. Can be any length. Default PrettyCols::prettycols("Lively").
`s`	Seed value. Default 1234.

Value

A ggplot object.

`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 = rcartocolor::carto_pal(n = 12, "Bold"),
  bg_col = "white",
  s = 123
)
```

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 "Bold" colour palette from rcartocolor. Must have 12 colours.
`bg_col`	Background colour. Default "white".
`s`	Seed value. Default 123.

Value

A ggplot object

`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.

rings *Rings*

Description

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

Usage

```
rings(
  col_palette = PrettyCols::prettycols("Lively"),
  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 PrettyCols::prettycols("Lively").
`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).
`x0`	Vector of x-co-ordinates for pie charts. Default c(0, 1).
`y0`	Vector of y-co-ordinates for pie charts. Default c(0, 2).
`r`	Vector of radii for pie charts. Default c(1, 0.7).
`n`	Vector of slices per pie chart. Default c(80, 80).
`s`	Seed value. Default 1234.

Value

A ggplot object.

`riso_circles` *Riso Circles*

Description

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

Usage

```
riso_circles(
  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

`n_x`	Number of columns in grid. Default 4.
`n_y`	Number of rows in grid. Default 4.
`n_circles`	Number of circles per grid square. Default 2.
`r`	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.

shatter *Shatter*

Description

This function generates a generative art ggplot object using polygons.

Usage

```
shatter(
  n_x = 25,
  n_y = 25,
  decay = 0.8,
  colour = "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.
colour	Single colour for fill colour of polygons. Default "black".
bg_col	Single colour for background. Default "gray97".
s	Seed value. Default 1234.

Value

A ggplot object.

shells *Shells*

Description

This function generates a layered generative art ggplot object sine and cosine waves and lines.

Usage

```
shells(n = 4, alpha = 1, main_col = "black", bg_col = "white")
```

Arguments

n	Number of spirals. Default 4.
alpha	Transparency of lines. Default 1.
main_col	Colour scheme of art. Default "black".
bg_col	Background colour. Default "white".

Value

A ggplot object.

---

smudge *Smudge*

Description

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

Usage

```
smudge(
  n = 25,
  binwidth = 0.01,
  col_palette = PrettyCols::prettycols("TangerineBlues"),
  s = 1234
)
```

Arguments

n	Number of grid boxes. Default 25.
binwidth	Binwidth for colours. Default 0.01.
col_palette	Vector of colours. Default `PrettyCols::prettycols("TangerineBlues")`.
s	Seed value. Default 1234.

Value

A ggplot object.

`spirals` *Spirals*

Description

This function generates a generative art ggplot object consisting of dots arranged in a spiral,

Usage

```
spirals(perc = 0.2, s = 1234)
```

Arguments

`perc`	Percentage of data points to be non-NA. Default 0.2.
`s`	Seed value. Default 1234.

Value

A ggplot object.

`spiro` *Spiro*

Description

This function generates a generative art ggplot object in a spirograph style.

Usage

```
spiro(
  n_x = 10,
  n_y = 10,
  d = 10,
  R = 4,
  r = 1,
  linewidth = 0.5,
  col_palette = "white",
  bg_col = "grey20",
  s = 1234
)
```

Arguments

`n_x`	Number of spirals per row. Default 10.
`n_y`	Number of spirals per column. Default 10.
`d`	Diameter. Default 10.
`R`	Outer radius. Default 4.
`r`	Inner radius. Default 1.
`linewidth`	Width on lines. Default 0.5.
`col_palette`	Vector of colours. Default "white".
`bg_col`	Background colour. Default "grey20".
`s`	Seed value. Default 1234.

Value

A ggplot object

`split_grid` *Split grid*

Description

This function generates a generative art ggplot object using polygons

Usage

```
split_grid(
  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.

split_jitter *Split jitter*

Description

This function generates a generative art ggplot object using polygons

Usage

```
split_jitter(
  n_x = 5,
  n_y = 5,
  deg_jitter = 0.4,
 col_palette = c("#416322", "#4e7628", "#5a892f", "#679c35", "#74af3b", "#80c044",
    "#8dc657"),
  line_col = "transparent",
  linewidth = 1,
  bg_col = "#679c35",
  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("#416322", "#4e7628", "#5a892f", "#679c35", "#74af3b", "#80c044", "#8dc657")
line_col	Colour of lines. Default "transparent".
linewidth	Width of lines. Default 1.
bg_col	Background colour. Default "#679c35".
s	Random seed. Default 1234.

Value

A ggplot object.

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,
  size = 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.
size	Size of lines between squares. Default 1.5.
s	Random seed. Default 1234.

Value

A ggplot object.

squiggles *Squiggles*

Description

This function generates generative art from multiple smooth lines.

Usage

```
squiggles(
  res = 100,
  num_lines = 100,
  perc = 0.1,
  alpha_low = 0.5,
  alpha_high = 1,
  line_col = "white",
  bg_col = "black",
  s = 1234
)
```

Arguments

`res`	Resolution of grid. Default 100.
`num_lines`	Number of lines to draw. Default 100.
`perc`	Percentage to colour darker. Default 0.1.
`alpha_low`	Transparency of majority of lines. Default 0.5.
`alpha_high`	Transparency of minority of lines. Default 1.
`line_col`	Colour of lines. Default "white".
`bg_col`	Background colour. Default "black".
`s`	Random seed. Default 1234.

Value

A ggplot object.

`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.

---

stacked *Stacked*

Description

This function generates a coloured generative art ggplot object using 3D square polygons.

Usage

```
stacked(
  n_x = 5,
  n_y = 5,
  col_palette = MetBrewer::met.brewer("Hiroshige", 9),
  rayshade = FALSE,
  shadow_intensity = 0.5,
  sunangle = 315,
  s = 1234
)
```

Arguments

`n_x`	Number of polygons per row. Default 5.
`n_y`	Number of polygons per column. Default 5.
`col_palette`	Vector of colours. Default Hiroshige 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.

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",
  s = 1234
)
```

Arguments

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

Value

A ggplot object.

`static` *Static*

Description

This function generates a greyscale generative art ggplot object.

Usage

```
static(perc = 0.1, n = 500, s = 1234)
```

Arguments

`perc`	Percentage of data points to be non-NA. Default 0.1.
`n`	Number of squares. Default 500.
`s`	Seed value. Default 1234.

Value

A ggplot object.

`streams` *Streams*

Description

This function generates a coloured generative art ggplot object from stream charts.

Usage

```
streams(
  bg_col = "white",
  line_col = "white",
  fill_col = c("#5F4690", "#1D6996", "#38A6A5", "#0F8554", "#73AF48", "#EDAD08",
    "#E17C05", "#CC503E", "#94346E", "#6F4070"),
  type = "right",
  s = 1234
)
```

Arguments

`bg_col`	Background colour. Default "white".
`line_col`	Line colour. Default "white".
`fill_col`	Vector of fill colours.
`type`	Rotation of stream. Default "right".
`s`	Seed value. Default 1234.

Value

A ggplot object.

stripes *Stripes*

Description

This function generates a generative art ggplot object featuring rows of stripes.

Usage

```
stripes(
  perc = 0.5,
  n = 3,
  col_palette = rcartocolor::carto_pal(n = 7, "TealGrn"),
  alpha = 1,
  s = 1234
)
```

Arguments

`perc`	Percentage of data points to be sorted. Default 0.5.
`n`	Number of rows. Default 3.
`col_palette`	Vector of colours. Default "TealGrn" colour palette from rcartocolor.
`alpha`	Transparancy of fill
`s`	Seed value. Default 1234.

Value

A ggplot object

sunbursts *Sunbursts*

Description

This function generates a generative art ggplot object from 2d density plots

Usage

```
sunbursts(
  n = 100,
  x_means = c(0, 10, 5),
  y_means = c(0, 7, 8),
  xy_var = 5,
  low = "#074050",
  high = "#d3f2a3",
  s = 1234
)
```

Arguments

`n`	Granularity. Default 100.
`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 varaince of x and y points. Default 5.
`low`	Colour of background. Default "#074050".
`high`	Colour of sunburst points. Default "#d3f2a3".
`s`	Seed value. Default 1234.

Value

A ggplot object.

`sunbursts_panel` *Sunbursts panel*

Description

This function generates a grid of sunbursts pieces

Usage

```
sunbursts_panel(
  n = 100,
  ncol = 4,
  nrow = 4,
  x_means = c(0, 10, 5),
  y_means = c(0, 7, 8),
  xy_var = 5,
  low = "#4e0550",
  high = "#facdfc",
  s = 1234
)
```

Arguments

n	Granularity. Default 100.
ncol	Number of column panels. Default 4.
nrow	Number of row panels. Default 4.
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 varaince of x and y points. Default 5.
low	Colour of background. Default "#4e0550".
high	Colour of sunburst points. Default "#facdfc".
s	Seed value. Default 1234.

Value

A ggplot object.

sunsets *Sunsets*

Description

This function creates generative art using faceted segments.

Usage

```
sunsets(
  num_bars = 8,
  n = 1000,
  col_palette = PrettyCols::prettycols("Lively"),
  bg_col = "#413C58",
  vertical = FALSE,
  fade_vertical = FALSE,
  alpha = 1,
  s = 2023
)
```

Arguments

num_bars	Number of bars. Default 8.
n	Number of lines per bar. Default 1000.
col_palette	Colour palette. Default PrettyCols::prettycols("Lively").
bg_col	Background colour. Default "#413C58".
vertical	Boolean indicating whether bars should be vertical. Default FALSE.
fade_vertical	Boolean indicating whether the colouring should be vertical. Default FALSE.
alpha	Transparency of coloured bars. Default 1.
s	Random seed. Default 2023.

Value

A ggplot object.

`theme_aRt` *aRt theme*

Description

Custom ggplot2 theme for aRt objects

Usage

```
theme_aRt
```

Format

An object of class `theme` (inherits from `gg`) of length 136.

`tiles` *Tiles*

Description

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

Usage

```
tiles(
  n_x = 12,
  n_y = 12,
  col_palette = MetBrewer::met.brewer("Veronese", 5),
  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 Veronese palette from MetBrewer.
`s`	Seed value. Default 1234.

Value

A ggplot object.

vortex *Vortex*

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_scheme = "mono", bg_col = "black", s = 1234)
```

Arguments

n	Number of points. Default 25.
start_val	Starting position for polar coordinates. Default 90.
col_scheme	Colour scheme of art. One of c("mono", "rainbow). Default 0.1.
bg_col	Background colour. Default "black".
s	Seed value. Default 1234.

Value

A ggplot object.

wander *Wander*

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

`n_lines`	Number of lines. Default 100.
`n_points`	Number of points. Default 350.
`r_outer`	Radius of outer circle. Default 8.
`r_inner`	Radius of inner circle. Default 3.
`line_var`	Variance of random walk noise. Default 0.01.
`deg_jitter`	Degree of jitter for multiple lines. Default 0.1.
`linewidth`	Width of lines. Default 0.1.
`bg_col`	Background colour. Default "#462255".
`col_palette`	Vector of colours. Default `c("#FF8811", "#9DD9D2", "#046E8F", "#D44D5C")`
`n_cols`	Number of colours to create. Default 20.
`s`	Random seed. Default 123.

Value

A ggplot object.

`waves` *Waves*

Description

This function generates a generative art ggplot object from sine and cosine waves.

Usage

```
waves(
  a = 23,
  b = 6,
  linewidth = 0.5,
  main_col = "black",
  bg_col = "white",
  s = 2021
)
```

Arguments

`a`	sine wave parameter. Default 23.
`b`	Cosine wave parameter. Default 6.
`linewidth`	Width of lines. Default 0.5.
`main_col`	Vector of colours (or single colour) for lines. Default "black".
`bg_col`	Background colour. Default "white".
`s`	Seed value. Default 2021.

Value

A ggplot object.

`window_boxes` *Window Boxes*

Description

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

Usage

```
window_boxes(
  n_x = 10,
  n_y = 10,
  col_palette = PrettyCols::prettycols("Beach", n = 5),
  linewidth = 2,
  rayshade = FALSE
)
```

Arguments

`n_x`	Number of squares per row. Default 10.
`n_y`	Number of squares per column. Default 10.
`col_palette`	Vector of colours. Default `PrettyCols::prettycols("Beach", n = 5)`.
`linewidth`	Width of borders around squares. Default 2.
`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`.

Value

A ggplot object.

Index