Package: ggbenjamini (via r-universe)

October 15, 2024

Title Generate ficus benjamina leaf shapes with bezier curves

Version 0.0.1

Description This package can be used to generate shapes in the form of ficus benjamina (weeping fig) leaves with bezier curves. The main output of the package are dataframes containing all the information of these bezier curves.

License MIT + file LICENSE

Suggests flametree, rsvg, minisvg, knitr, rmarkdown, testthat (>= 3.0.0), vdiffr, ambient, covr, usethis

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

VignetteBuilder knitr

Imports stringr, dplyr (>= 1.1.0), ggplot2, purrr, tidyr, ggforce, magrittr, tibble, rlang, grid, prismatic

Remotes coolbutuseless/minisvg, djnavarro/flametree

Depends R (>= 2.10)

URL https://urswilke.github.io/ggbenjamini/, https://github.com/urswilke/ggbenjamini/

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

RemoteUrl https://github.com/urswilke/ggbenjamini

RemoteRef HEAD

RemoteSha 77bdbe1649aca9364eed5142c61276a59b6e8e54

Contents

`benjamini_branch` *Generate a branch of benjamini leaves*

Description

Generate a branch of benjamini leaves

Usage

```
benjamini_branch(
  df_branch = tibble::tibble(x = c(70, 84, 126, 168), y = c(280, 245, 217, 217)),
  leaf_mean_dist_approx = 10,
  leaf_angle = 45,
  first_dir = sample(0:1, 1),
  leave_size_dist = spark_weibull(shape = 1.5, scale_factor = 0.8),
  leaf_angle_dist = spark_norm(mean = 0, sd = 3),
  last_angle_straight = TRUE,
  leaf_size_multiplicator = 1
)
```

Arguments

`df_branch`	dataframe containing the 4 x & y bezier coordinates of a branch
`leaf_mean_dist_approx`	approximate distance of two leaves
`leaf_angle`	angle between the leaf stalks and the branch
`first_dir`	direction of the first leaf in the branch (0 for right; 1 for left)
`leave_size_dist`	Manipulate the sizes of the leaves with a spark function This function returns a function which itself returns a numerical vector of length of the number of leaves on the branch. The function will be rescaled (divided by it's maximum value).

`leaf_angle_dist`
: Manipulate the leaf angles with a spark function This function returns a function which itself returns a numerical vector of length of the number of leaves on the branch.

`last_angle_straight`
: Logical if the angle of the last leaf on the branch is very sharp (defaults to TRUE).

`leaf_size_multiplicator`
: Multiply leaf size distribution with a factor.

Value

A dataframe containing the data for leaves on a branch (see example).

Examples

```
benjamini_branch() %>%
  tidyr::unite(b, i_part, i_leaf, element, remove = FALSE) %>%
  ggplot2::ggplot() +
  ggforce::geom_bezier(ggplot2::aes(x = x, y = y, group = b)) +
  ggplot2::coord_equal()
```

---

benjamini_leaf *Generate bezier curve coordinates of a benjamini leaf*

Description

Generate bezier curve coordinates of a benjamini leaf

Usage

```
benjamini_leaf(
  leaf_params = gen_leaf_parameters(),
  omega = 0,
  xrot = leaf_params$x0,
  yrot = leaf_params$y0,
  precision = 2
)
```

Arguments

`leaf_params`	parameter that control the leaf shape
`omega`	rotation angle of the leaf
`xrot`	x coordinate of pivot point (preset to leaf origin).
`yrot`	x coordinate of pivot point (preset to leaf origin).
`precision`	numeric precision of the output

Value

A dataframe conaining the data for the bezier curves of a leaf (see example).

Examples

```
df <- benjamini_leaf()
df
df %>%
  # This generated a unique identifier for the 4 rows of each bezier curve:
  tidyr::unite(b, element, i_part, remove = FALSE) %>%
  ggplot2::ggplot() +
  ggforce::geom_bezier(ggplot2::aes(x = x, y = y, group = b))
```

`bezier_to_polygon` *Transform bezier dataframe to dataframe with path coordinates*

Description

Transform bezier dataframe to dataframe with path coordinates

Usage

```
bezier_to_polygon(df_benjamini_leaf, ..., n = 100)
```

Arguments

`df_benjamini_leaf`	Dataframe returned by `benjamini_leaf()`
`...`	grouping variables in `df_benjamini_leaf` that will be kept in the transformation.
`n`	number of points per bezier

Value

Dataframe with the coordinates of the bezier curve interpolations.

Examples

```
df_coords <- benjamini_leaf() %>%
  tidyr::unite(b, i_part, element, remove = FALSE) %>%
  bezier_to_polygon(b, i_part, element)
df_coords
df_coords %>%
  ggplot2::ggplot(ggplot2::aes(x = x, y = y, group = element, fill = element)) +
  ggplot2::geom_polygon()
```

df_bezier_skeleton *Example bezier dataframe to grow leaves on*

Description

Generated with `vignette("import_svg_bezier")`

Usage

```
df_bezier_skeleton
```

Format

A data frame with 40 rows and 4 variables:

i_branch branch index

svg_point_type original point type in the svg file

x x coordinate

y y coordinate

gen_benjamini_points *Generate bezier end points*

Description

Generate bezier end points

Usage

```
gen_benjamini_points(leaf_params = gen_leaf_parameters())
```

Arguments

`leaf_params`	parameters generated by `gen_leaf_parameters()`

Value

tibble with absolute coordinates

Examples

```
gen_benjamini_points()
```

gen_benjamini_slopes *Generate bezier slopes coordinates*

Description

Generate bezier slopes coordinates

Usage

```
gen_benjamini_slopes(leaf_params = gen_leaf_parameters())
```

Arguments

leaf_params	parameters generated by gen_leaf_parameters()

Value

dataframe with sx1-4 in the x variable and sy1-4 in the y variable

Examples

```
gen_benjamini_slopes()
```

gen_leaf_parameters *Generate bezier coordinates of a leaf*

Description

Except the start point coordinates of the leaf origin x0 & y0 (arbitrarily set to 10 and 40) all coordinates are relative to this origin. Most of the parameters are random generated, except for some of them (e.g., to close polygons, or to keep the stalk and the mid vein on the same line).

Usage

```
gen_leaf_parameters(
  x0 = 10,
  y0 = 40,
  dx10 = sample(8:12, 1),
  dy10 = 0,
  dx21 = sample(12:20, 1),
  dy21 = sample(-4:-10, 1),
  dx32 = sample(10:18, 1),
  dy32 = stats::runif(1, 0.92 * (-dy21 - 1), 0.95 * (-dy21 - 1)),
  dx43 = sample(4:6, 1),
  dy43 = y0 + dy10 + dy21 + dy32,
  sx0 = stats::runif(1, 0, 0.1),
```

```
  sx1 = sample(1:3, 1),
  sx2 = sample(4:6, 1),
  sx3 = sample(2:4, 1),
  sx4 = stats::runif(1, 0, 0.2),
  sy0 = stats::runif(1, 0.5, 1),
  sy1 = sample(-4:-6, 1),
  sy2 = stats::runif(1, -0.5, 0.5),
  sy3 = stats::runif(1, 0.5, 1.5),
  sy4 = stats::runif(1, 0.2, 0.7),
  smx1 = sample(-5:-15, 1),
  smx2 = sample(-5:-15, 1),
  smy1 = stats::runif(1, -1, 1),
  smy2 = stats::runif(1, -1, 1)
)
```

Arguments

`x0`, `y0`	coordinates of the leaf origin
`dx10`, `dy10`, `dx21`, `dy21`, `dx32`, `dy32`, `dx43`, `dy43`	coordinates of the other bezier start & end points
`sx0`, `sx1`, `sx2`, `sx3`, `sx4`	x coordinates of the control points
`sy0`, `sy1`, `sy2`, `sy3`, `sy4`	y coordinates of the control points
`smx1`, `smx2`, `smy1`, `smy2`	x & y coordinates of the mid vein control points

Value

named list of all parameters

Examples

```
gen_leaf_parameters()
```

---

gen_middle_line_slopes

Generate bezier slopes of the line in the middle of the leaf

Description

Generate bezier slopes of the line in the middle of the leaf

Usage

```
gen_middle_line_slopes(leaf_params = gen_leaf_parameters())
```

Arguments

leaf_params	parameters generated by gen_leaf_parameters()

Value

A dataframe containing the coordinates of the two control points of the bezier curve defining the midvein of the leaf.

Examples

```
gen_middle_line_slopes()
```

---

get_one_bezier *Generate a dataframe of one bezier curve*

Description

Generate a dataframe of one bezier curve

Usage

```
get_one_bezier(i, points_df, slopes_df)
```

Arguments

i	number of the bezier
points_df	dataframe generated by gen_benjamini_points() (see example).
slopes_df	dataframe generated by gen_benjamini_slopes() (see example).

Value

A dataframe containing the information for one bezier curve in the format as needed by ggforce::geom_bezier.

Examples

```
set.seed(123)
leaf_params <- gen_leaf_parameters()
points_df <- gen_benjamini_points()
slopes_df <- gen_benjamini_slopes()
df_bezier <- get_one_bezier(1, points_df, slopes_df)
ggplot2::ggplot(df_bezier, ggplot2::aes(x = x, y = y)) + ggforce::geom_bezier()
```

spark_norm *Manipulate the angles of the leaves with a normal distribution*

Description

This function returns a function which itself returns a numerical vector of length of the number of leaves on the branch.

Usage

```
spark_norm(mean = 0, sd = 3)
```

Arguments

mean, sd	Parameters passed to `stats::dnorm()`.

Value

dnorm() function with n_leaves as one of the arguments

Examples

```
spark_norm()
```

spark_weibull *Manipulate the sizes of the leaves with a Weibull distribution*

Description

This function returns a function which itself returns a numerical vector of length `n_leaves`, the number of leaves on the branch. These values serve as relative multiplication factors of the sizes of the leaves of the branch. (The maximum value of this distribution is then normalized to 1.)

Usage

```
spark_weibull(shape = 1.2, scale_factor = 0.5)
```

Arguments

shape, scale_factor	Parameters passed to `stats::dweibull()`.

Value

dweibull() function depending on n_leaves as one of the arguments

Examples

```
#Mark the two consecutive pairs of parentheses:
spark_weibull()(n_leaves = 10)
```

Index