Package 'vayr'

April 15, 2025

Title Extensions for 'ggplot2' to Visualize as You Randomize

Version 1.0.0

Description Position adjustments for 'ggplot2' to implement ``visualize as you randomize'' principles, which can be especially useful when plotting experimental data.

License GPL-2 | file LICENSE

URL https://alexandercoppock.com/vayr/index.html

Depends R (>= 4.1.0)

Imports ggplot2 (>= 3.0.0), packcircles (>= 0.3.7), withr (>= 2.1.1)

Suggests dplyr, estimatr, knitr, patchwork, randomizr, rmarkdown, sessioninfo, testthat, tibble, tidyverse

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Alexander Coppock [aut, cre, cph] (<https://orcid.org/0000-0002-5733-2386>), Elias Hyde [ctb]

Maintainer Alexander Coppock <acoppock@gmail.com>

Repository CRAN

Date/Publication 2025-04-15 20:10:05 UTC

Contents

`patriot_act` *Original and Replication data for the Patriot Act experiment described in Persuasion in Parallel*

Description

Original and Replication data for the Patriot Act experiment described in Persuasion in Parallel

Usage

```
patriot_act
```

Format

`patriot_act`**:**

A data frame with 2062 rows and 4 columns:

sample_label The original study (Chong and Druckman (2011) or the Mechanical Turk replication)

pid_3 Subject partisanship (limited to Republicans and Democracts, including leaners)

T1_content Content of assigned treatment condition: pro-Patriot act statements, anti-Patriot act statements, or a control

PA_support Post-treatment support for the Patriot Act on a 1 to 7 Likert scale

Source

doi:10.7910/DVN/I9GSKI

`position_circlepack` *Arrange over-plotted points with a circle-packing algorithm*

Description

This function uses a circle packing algorithm from the 'packcircles' package to arrange perfectly over-plotted points of varying sizes into a elliptical area.

Usage

```
position_circlepack(density = 1, aspect_ratio = 1)
```

Arguments

density	The density of the circle pack, which defaults to 1 but will have to be adjusted in most cases. The desirable density will depend on both the ranges of the axes and the dimensions of the image. It will also depend on the size scale.
aspect_ratio	An aspect ratio adjustment to compensate for distortion of the circular arrangement, which might occur when plotting if coord_equal() is not used. A wide aspect ratio (eg. 2) would adjust for vertical stretching, whereas a tall aspect ratio (eg. 0.5) would adjust for horizontal stretching. The default aspect ratio of 1 is appropriate when no adjustment is required.

Value

A ggproto object of class PositionCirclePack.

See Also

Other Functions: position_circlepackdodge(), position_jitter_ellipse(), position_jitterdodge_ellipse(), position_sunflower(), position_sunflowerdodge(), sunflower()

Examples

```
library(ggplot2)
library(dplyr)
library(randomizr)
library(tibble)

dat <- data.frame(
  X = c(rep(0, 200)),
  Y = rep(0, 200),
  size = runif(200, 0, 1)
)

ggplot(dat, aes(x = X, y = Y, size = size)) +
  geom_point(position = position_circlepack(density = 0.25, aspect_ratio = 1),
             alpha = 0.25) +
  coord_equal(xlim = c(-1, 1), ylim = c(-1, 1), expand = TRUE) +
  theme(legend.position = "none")

# Applied to a mock experiment with weighted groups

dat <-
  tibble(
    age_group = rep(c("young", "middle", "old"), c(100, 200, 300)),
    treatment = block_ra(age_group, block_m = c(50, 50, 50)),
    latent_outcome =
      case_when(age_group == "young" & treatment == 0 ~ 0.10,
                age_group == "young" & treatment == 1 ~ 0.20,
                age_group == "middle" & treatment == 0 ~ 0.40,
                age_group == "middle" & treatment == 1 ~ 0.45,
                age_group == "old" & treatment == 0 ~ 0.70,
                age_group == "old" & treatment == 1 ~ 0.90),
```

```
    outcome = rbinom(600, size = 1,
                     prob = latent_outcome)
)

dat <-
  dat |>
  mutate(
    treatment_prob =
      case_when(age_group == "young" ~ 50/100,
                age_group == "middle" ~ 50/200,
                age_group == "old" ~ 50/300),
    weights = 1/case_when(treatment == 1 ~ treatment_prob,
                          treatment == 0 ~ 1 - treatment_prob)
  )

ggplot(dat, aes(treatment, outcome, size = weights, color = age_group)) +
  geom_point(alpha = 0.5, position = position_circlepack(density = 0.5))
```

`position_circlepackdodge` *Arrange over-plotted points with a circle-packing algorithm and dodge groups side-to-side*

Description

This function dodges groups and uses a circle packing algorithm from the 'packcircles' package to arrange perfectly over-plotted points of varying sizes into a elliptical area.

Usage

```
position_circlepackdodge(width = 1, density = 1, aspect_ratio = 1)
```

Arguments

`width`	The dodging width, which defaults to 1.
`density`	The density of the circle pack, which defaults to 1 but will have to be adjusted in most cases. The desirable density will depend on both the ranges of the axes and the dimensions of the image. It will also depend on the size scale.
`aspect_ratio`	An aspect ratio adjustment to compensate for distortion of the circular arrangement, which might occur when plotting if coord_equal() is not used. A wide aspect ratio (eg. 2) would adjust for vertical stretching, whereas a tall aspect ratio (eg. 0.5) would adjust for horizontal stretching. The default aspect ratio of 1 is appropriate when no adjustment is required.

Value

A `ggproto` object of class `PositionCirclePackDodge`.

See Also

Other Functions: position_circlepack(), position_jitter_ellipse(), position_jitterdodge_ellipse(), position_sunflower(), position_sunflowerdodge(), sunflower()

Examples

```
library(ggplot2)

dat <- data.frame(
  X = c(rep(0, 200)),
  Y = rep(0, 200),
  size = runif(200, 0, 1),
  id = (rep(c("A", "B"), 100))
)

ggplot(dat, aes(x = X, y = Y, size = size, color = id)) +
 geom_point(position = position_circlepackdodge(width = 1, density = 1, aspect_ratio = 1),
            alpha = 0.25) +
  coord_equal(xlim = c(-1, 1), ylim = c(-1, 1), expand = TRUE) +
  scale_size_continuous(range = c(1, 3)) +
  theme(legend.position = "none")
```

---

position_jitterdodge_ellipse

Jitter points on an ellipse and dodge groups side-to-side

Description

This function dodges groups of points side-to-side and adds elliptical random noise to perfectly over-plotted points. See the position_jitter_ellipse() documentation for more information.

Usage

```
position_jitterdodge_ellipse(
  jitter.width = NULL,
  jitter.height = NULL,
  dodge.width = 1,
  seed = NA
)
```

Arguments

jitter.width, jitter.height	The dimensions of the elliptical field, from which over-plotted points are sampled.
dodge.width	The dodging width, which defaults to 1.
seed	A random seed for reproducibility.

Value

A ggproto object of class PositionJitterdodgeEllipse.

See Also

Other Functions: position_circlepack(), position_circlepackdodge(), position_jitter_ellipse(), position_sunflower(), position_sunflowerdodge(), sunflower()

Examples

```
library(ggplot2)

dat <- data.frame(x = rep(1, 500), y = rep(1, 500),
                  group = sample(LETTERS[1:2], 500, replace = TRUE))

ggplot(dat, aes(x, y, shape = group, color = group)) +
  geom_point(position = position_jitterdodge_ellipse(jitter.width  = 0.5,
                                                     jitter.height =  0.5,
                                                     dodge.width = 1)) +
  coord_cartesian(xlim = c(0, 2), ylim = c(0, 2))
```

---

position_jitter_ellipse

Jitter points on an ellipse to avoid over-plotting

Description

This function adds elliptical random noise to perfectly over-plotted points, offering a pleasing way to visualize many points that represent the same position. In contrast to the position_jitter() function which samples from a rectangular field, the position_jitter_ellipse() function samples from an elliptical field. This function takes algorithmic inspiration from https://stackoverflow.com/questions/5529148/algorithm-calculate-pseudo-random-point-inside-an-ellipse and https://stats.stackexchange.com/questions/120527/simulate-a-uniform-distribution-on-a-disc.

Usage

```
position_jitter_ellipse(width = NULL, height = NULL, seed = NA)
```

Arguments

width, height	The dimensions of the elliptical field, from which over-plotted points are sampled.
seed	A random seed for reproducibility.

Value

A ggproto object of class PositionJitterEllipse.

See Also

Other Functions: position_circlepack(), position_circlepackdodge(), position_jitterdodge_ellipse(), position_sunflower(), position_sunflowerdodge(), sunflower()

Examples

```
library(ggplot2)

dat <- data.frame(x = rep(1, 500), y = rep(1, 500))

# Jitter on an ellipse.
ggplot(dat, aes(x, y)) +
  geom_point(position = position_jitter_ellipse(width = 0.5, height = 0.5)) +
  coord_cartesian(xlim = c(0, 2), ylim = c(0, 2))

# Jitter on a rectangle, for comparison.
ggplot(dat, aes(x, y)) +
  geom_point(position = position_jitter(width = 0.5, height = 0.5)) +
  coord_cartesian(xlim = c(0, 2), ylim = c(0, 2))
```

---

position_sunflower *Arrange over-plotted points in a sunflower pattern*

Description

This function applies the sunflower algorithm, executed by the sunflower() function, as a position adjustment, arranging overlapping points at any given x and y into a sunflower pattern. See the sunflower() documentation for more information.

Usage

```
position_sunflower(density = 1, aspect_ratio = 1)
```

Arguments

density	The pattern density, which defaults to 1 but will have to be adjusted in most cases. The desirable density will depend on both the ranges of the axes and the dimensions of the image.
aspect_ratio	An aspect ratio adjustment to compensate for distortion of the circular arrangement, which might occur when plotting if coord_equal() is not used. A wide aspect ratio (eg. 2) would adjust for vertical stretching, whereas a tall aspect ratio (eg. 0.5) would adjust for horizontal stretching. The default aspect ratio of 1 is appropriate when no adjustment is required.

Value

A ggproto object of class PositionSunflower.

See Also

Other Functions: position_circlepack(), position_circlepackdodge(), position_jitter_ellipse(), position_jitterdodge_ellipse(), position_sunflowerdodge(), sunflower()

Examples

```
library(ggplot2)

# Use the sunflower position function to arrange N points
N <- 100

dat <- data.frame(
  x = rep(1:4, times = N),
  y = rep(1:4, times = N)
)

ggplot(dat, aes(x = x, y = y)) +
  geom_point(size = 1, position = position_sunflower(density = 1, aspect_ratio = 1)) +
  xlim(0, 5) +
  ylim(0, 5) +
  coord_equal()
```

`position_sunflowerdodge` *Arrange over-plotted points in a sunflower pattern and dodge groups side-to-side*

Description

This function applies the sunflower position adjustment alongside the dodge position adjustment, arranging overlapping points per x, y, and group into a sunflower pattern. See the sunflower() documentation for more information.

Usage

```
position_sunflowerdodge(width = 1, density = 1, aspect_ratio = 1)
```

Arguments

`width`	The dodging width, which defaults to 1.
`density`	The pattern density, which defaults to 1 but will have to be adjusted in most cases. The desirable density will depend on both the ranges of the axes and the dimensions of the image.
`aspect_ratio`	An aspect ratio adjustment to compensate for distortion of the circular arrangement, which might occur when plotting if coord_equal() is not used. A wide aspect ratio (eg. 2) would adjust for vertical stretching, whereas a tall aspect ratio (eg. 0.5) would adjust for horizontal stretching. The default aspect ratio of 1 is appropriate when no adjustment is required.

Value

A ggproto object of class PositionSunflowerDodge.

See Also

Other Functions: position_circlepack(), position_circlepackdodge(), position_jitter_ellipse(), position_jitterdodge_ellipse(), position_sunflower(), sunflower()

Examples

```
library(ggplot2)

# Use the sunflower dodge position function to arrange and dodge N points.
N <- 300

dat <- data.frame(
  x = sample(1:2, size = N, replace = TRUE),
  y = sample(1:7, size = N, replace = TRUE),
  type = factor(sample(LETTERS[1:2], N, replace = TRUE))
)

# With coord_equal
ggplot(dat, aes(x, y, color = type, shape = type)) +
 geom_point(position = position_sunflowerdodge(width = 0.5, density = 2, aspect_ratio = 1)) +
  coord_equal()

# Without coord_equal, might want to play with aspect ratio to get a pleasing plot
ggplot(dat, aes(x, y, color = type, shape = type)) +
 geom_point(position = position_sunflowerdodge(width = 0.5, density = 10, aspect_ratio = 1/4))
```

---

sunflower *Distribute points using a sunflower seed algorithm*

Description

This function distributes points in a ellipse via a sunflower seed algorithm as a solution for over-plotting. To implement the algorithm, this function adapts the code from https://stackoverflow.com/questions/28567166/unifo distribute-x-points-inside-a-circle.

Usage

```
sunflower(x = NULL, y = NULL, density, aspect_ratio)
```

Arguments

x, y	The identical coordinates of multiple over-plotted points, as vectors, which will be arranged using a sunflower seed algorithm.
density	The pattern density.
aspect_ratio	An aspect ratio adjustment to compensate for distortion of the circular arrangement, which might occur when plotting if coord_equal() is not used. A wide aspect ratio (eg. 2) would adjust for vertical stretching, whereas a tall aspect ratio (eg. 0.5) would adjust for horizontal stretching. An aspect ratio of 1 is appropriate when no adjustment is required.

Value

A numeric vector of adjusted x or y positions, computed using a sunflower seed algorithm.

See Also

Other Functions: position_circlepack(), position_circlepackdodge(), position_jitter_ellipse(), position_jitterdodge_ellipse(), position_sunflower(), position_sunflowerdodge()

Examples

```
library(ggplot2)
library(dplyr)

# Manually adjust position of N points,
# arranging them per the sunflower algorithm and then dodging groups
N <- 300

dat <- data.frame(
  x = sample(1:2, size = N, replace = TRUE),
  y = sample(1:7, size = N, replace = TRUE),
  type = factor(sample(LETTERS[1:2], N, replace = TRUE))
) |>
  group_by(x, y, type) |>
  mutate(
    x = sunflower(x = x, density = 1, aspect_ratio = 1),
    y = sunflower(y = y, density = 1, aspect_ratio = 1),
    x = if_else(type == "A", x - (1 / 8), x + (1 / 8))
  )

ggplot(dat, aes(x, y, color = type, shape = type)) +
  geom_point() + coord_equal()
```

vayr *Extensions for 'ggplot2' to Visualize as You Randomize*

Description

Position adjustments for 'ggplot2' to implement "visualize as you randomize" principles, which can be especially useful when plotting experimental data.

Details

The 'vayr' package provides 'ggplot2' extensions that foster "visualize as you randomize" principles. These principles should guide the visualization of experimental data. Thus far, the package includes position adjustments to avoid over-plotting, facilitating plotting in "data-space." The 'vayr' paper is here: https://alexandercoppock.com/coppock_2020.pdf.

Author(s)

Maintainer: Alexander Coppock <acoppock@gmail.com> (ORCID) [copyright holder]

Other contributors:

- Elias Hyde <eliasworrallhyde@gmail.com> [contributor]

See Also

Useful links:

- https://alexandercoppock.com/vayr/index.html

Index