Package ‘ggbeeswarm’

August 7, 2017

Type Package
Title Categorical Scatter (Violin Point) Plots
Version 0.6.0
Date 2017-08-03
Description Provides two methods of plotting categorical scatter plots such that the arrangement of points within a category reflects the density of data at that region, and avoids over-plotting.

URL https://github.com/eclarke/ggbeeswarm
BugReports https://github.com/eclarke/ggbeeswarm/issues
License GPL (>= 2)
Depends R (>= 3.0.0), ggplot2 (>= 2.0)
Imports beeswarm, vipor
Suggests gridExtra
RoxygenNote 6.0.1
NeedsCompilation no
Author Erik Clarke [aut, cre], Scott Sherrill-Mix [aut]
Maintainer Erik Clarke <erikclarke@gmail.com>
Repository CRAN
Date/Publication 2017-08-07 13:45:36 UTC

R topics documented:

  geom_beeswarm ................................................. 2
  geom_quasirandom .......................................... 4
  ggbeeswarm .................................................. 6
  position_beeswarm .......................................... 6
  position_quasirandom ....................................... 7

Index 9
**Description**

The beeswarm geom is a convenient means to offset points within categories to reduce overplotting. Uses the beeswarm package

**Usage**

```r
geom_beeswarm(mapping = NULL, data = NULL, priority = c("ascending", "descending", "density", "random", "none"), cex = 1, groupOnX = NULL, dodge.width = 0, stat = "identity", na.rm = FALSE, show.legend = NA, inherit.aes = TRUE, ...)
```

**Arguments**

- `mapping`: Set of aesthetic mappings created by `aes` or `aes_`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.

- `data`: The data to be displayed in this layer. There are three options:
  - If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot`.
  - A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify` for which variables will be created.
  - A function will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data.

- `priority`: Method used to perform point layout (see `swarmx`)

- `cex`: Scaling for adjusting point spacing (see `swarmx`)

- `groupOnX`: if `TRUE` then jitter is added to the x axis and if `FALSE` jitter is added to the y axis. Prior to v0.6.0, the default NULL causes the function to guess which axis is the categorical one based on the number of unique entries in each. This could result in unexpected results when the x variable has few unique values and so in v0.6.0 the default was changed to always jitter on the x axis unless `groupOnX=FALSE`. Also consider `coord_flip`.

- `dodge.width`: Amount by which points from different aesthetic groups will be dodged. This requires that one of the aesthetics is a factor.

- `stat`: The statistical transformation to use on the data for this layer, as a string.

- `na.rm`: If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

- `show.legend`: logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes.
geom_beeswarm

Inherit aesthetics

If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn’t inherit behaviour from the default plot specification, e.g. `borders`.

Other...

Other arguments passed on to `layer`. These are often aesthetics, used to set an aesthetic to a fixed value, like `color = "red"` or `size = 3`. They may also be parameters to the paired geom/stat.

Aesthetics

`geom_point` understands the following aesthetics (required aesthetics are in bold):

- x
- y
- alpha
- colour
- fill
- group
- shape
- size
- stroke

See Also

`geom_quasirandom` an alternative method, `swarmx` how spacing is determined, `geom_point` for regular, unjittered points, `geom_jitter` for jittered points, `geom_boxplot` for another way of looking at the conditional distribution of a variable

Examples

```r
ggplot2::qplot(class, hwy, data = ggplot2::mpg, geom='beeswarm')
# Generate fake data
distro <- data.frame(
  'variable'=rep(c('runif', 'rnorm'), each=100),
  'value'=c(runif(100, min=-3, max=3), rnorm(100))
)
ggplot2::qplot(variable, value, data = distro, geom='beeswarm')
ggplot2::ggplot(distro,aes(variable, value)) +
  geom_beeswarm(priority='density',cex=2.5)
```
Description

The quasirandom geom is a convenient means to offset points within categories to reduce overplotting. Uses the vipor package

Usage

geom_quasirandom(mapping = NULL, data = NULL, width = NULL,
              varwidth = FALSE, bandwidth = 0.5, nbins = NULL,
              method = "quasirandom", groupOnX = NULL, dodge.width = 0,
              stat = "identity", position = "quasirandom", na.rm = FALSE,
              show.legend = NA, inherit.aes = TRUE, ...)

Arguments

mapping Set of aesthetic mappings created by aes or aes_. If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data The data to be displayed in this layer. There are three options:
If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot.
A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify for which variables will be created.
A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data.
width the maximum amount of spread (default: 0.4)
varwidth vary the width by the relative size of each group
bandwidth the bandwidth adjustment to use when calculating density Smaller numbers (< 1) produce a tighter "fit". (default: 0.5)
nbins the number of bins used when calculating density (has little effect with quasirandom/random distribution)
method the method used for distributing points (quasirandom, pseudorandom, smiley or frowney)
groupOnX if TRUE then jitter is added to the x axis and if FALSE jitter is added to the y axis. Prior to v0.6.0, the default NULL causes the function to guess which axis is the categorical one based on the number of unique entries in each. This could result in unexpected results when the x variable has few unique values and so in v0.6.0 the default was changed to always jitter on the x axis unless groupOnX=FALSE. Also consider coord_flip.
geom quasirandom

dodge.width Amount by which points from different aesthetic groups will be dodged. This requires that one of the aesthetics is a factor.

stat The statistical transformation to use on the data for this layer, as a string.

position Position adjustment, either as a string, or the result of a call to a position adjustment function.

na.rm If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.

show.legend logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes.

inherit.aes If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders.

... other arguments passed on to layer. These are often aesthetics, used to set an aesthetic to a fixed value, like color = "red" or size = 3. They may also be parameters to the paired geom/stat.

Aesthetics

geom_point understands the following aesthetics (required aesthetics are in bold):

- x
- y
- alpha
- colour
- fill
- group
- shape
- size
- stroke

See Also

offsetX how spacing is determined, geom_point for regular, unjittered points, geom_jitter for jittered points, geom_boxplot for another way of looking at the conditional distribution of a variable

Examples

```r
ggplot2::qplot(class, hwy, data = ggplot2::mpg, geom='quasirandom')
  # Generate fake data
distro <- data.frame(
  'variable'=rep(c('runif','rnorm'),each=100),
  'value'=c(runif(100, min=-3, max=3), rnorm(100))
)
ggplot2::qplot(variable, value, data = distro, geom = 'quasirandom')
ggplot2::ggplot(distro,aes(variable, value)) + geom_quasirandom(width=0.1)
```
ggbeeswarm extends ggplot2 with violin point/beeswarm plots

Description

This package allows plotting of several groups of one dimensional data as a violin point/beeswarm plot by arranging data points to resemble the underlying distribution. The development version of this package is on http://github.com/eclarke/ggbeeswarm.

Author(s)

Erik Clarke, <ecl@mail.med.upenn.edu>

See Also

position_quasirandom

Examples

```r
ggplot2::ggplot(ggplot2::mpg, aes(class, hwy)) + geom_quasirandom()
# Generate fake data
distro <- data.frame(
  'variable'=rep(c('runif','rnorm'),each=100),
  'value'=c(runif(100, min=-3, max=3), rnorm(100))
)  
ggplot2::ggplot(distro,aes(variable, value)) + geom_quasirandom()
ggplot2::ggplot(distro,aes(variable, value)) + geom_quasirandom(width=.1)
```

position_beeswarm

Violin point-style plots to show overlapping points. x must be discrete.

Description

Violin point-style plots to show overlapping points. x must be discrete.

Usage

```r
position_beeswarm(priority = c("ascending", "descending", "density", "random", "none"), cex = 1, groupOnX = NULL, dodge.width = 0)
```
position_quasirandom

Arguments

- **priority**: Method used to perform point layout (see `swarmx`)
- **cex**: Scaling for adjusting point spacing (see `swarmx`)
- **groupOnX**: if TRUE then jitter is added to the x axis and if FALSE jitter is added to the y axis. Prior to v0.6.0, the default NULL causes the function to guess which axis is the categorical one based on the number of unique entries in each. This could result in unexpected results when the x variable has few unique values and so in v0.6.0 the default was changed to always jitter on the x axis unless groupOnX=FALSE. Also consider `coord_flip`.
- **dodge.width**: Amount by which points from different aesthetic groups will be dodged. This requires that one of the aesthetics is a factor.

See Also

- `position_quasirandom`, `swarmx`

Other position adjustments: `position_quasirandom`

Examples

```r
ggplot2::qplot(class, hwy, data = ggplot2::mpg, position=position_beeswarm())
```

Description

> Violin point-style plots to show overlapping points. x must be discrete.

Usage

```r
position_quasirandom(width = NULL, varwidth = FALSE, bandwidth = 0.5, nbins = NULL, method = "quasirandom", groupOnX = NULL, dodge.width = 0)
```

Arguments

- **width**: the maximum amount of spread (default: 0.4)
- **varwidth**: vary the width by the relative size of each group
- **bandwidth**: the bandwidth adjustment to use when calculating density Smaller numbers (< 1) produce a tighter "fit". (default: 0.5)
- **nbins**: the number of bins used when calculating density (has little effect with quasirandom/random distribution)
position_quasirandom

- **method**: the method used for distributing points (quasirandom, pseudorandom, smiley or frowney)

- **groupOnX**: if TRUE then jitter is added to the x axis and if FALSE jitter is added to the y axis. Prior to v0.6.0, the default NULL causes the function to guess which axis is the categorical one based on the number of unique entries in each. This could result in unexpected results when the x variable has few unique values and so in v0.6.0 the default was changed to always jitter on the x axis unless groupOnX=FALSE. Also consider `coord_flip`.

- **dodge.width**: Amount by which points from different aesthetic groups will be dodged. This requires that one of the aesthetics is a factor.

**See Also**

- `offsetX`

Other position adjustments: `position_beeswarm`

**Examples**

```r
ggplot2::qplot(class, hwy, data = ggplot2::mpg, position=position_quasirandom())
```
Index

aes, 2, 4
aes_, 2, 4

borders, 3, 5

coord_flip, 2, 4, 7, 8

fortify, 2, 4

geom_beeswarm, 2
geom_boxplot, 3, 5
geom_jitter, 3, 5
geom_point, 3, 5
geom_quasirandom, 3, 4

ggbeeswarm, 6

layer, 3, 5

offsetx, 5, 8

position_beeswarm, 6, 8
position_quasirandom, 6, 7, 7

swarmx, 2, 3, 7