Package ‘ggbeeswarm’

April 29, 2023

Type Package
Title Categorical Scatter (Violin Point) Plots
Version 0.7.2
Date 2023-04-28
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
Encoding UTF-8
License GPL (>= 3)
Depends R (>= 3.5.0), ggplot2 (>= 3.3.0)
Imports beeswarm, lifecycle, vipor, cli
Suggests gridExtra
RoxygenNote 7.2.2
NeedsCompilation no
Author Erik Clarke [aut, cre],
Scott Sherrill-Mix [aut],
Charlotte Dawson [aut]
Maintainer Erik Clarke <erikclarke@gmail.com>
Repository CRAN
Date/Publication 2023-04-29 21:40:02 UTC

R topics documented:

geom_beeswarm .................................................. 2
geom_quasirandom .............................................. 4
ggbeeswarm .......................................................... 6
position_beeswarm .............................................. 7
position_quasirandom ......................................... 8
Description
The beeswarm geom is a convenient means to offset points within categories to reduce overplotting.
Uses the beeswarm package

Usage
geom_beeswarm(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  ...,
  method = "swarm",
  cex = 1,
  side = 0L,
  priority = "ascending",
  fast = TRUE,
  dodge.width = NULL,
  corral = "none",
  corral.width = 0.9,
  groupOnX = NULL,
  beeswarmArgs = list(),
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

Arguments
mapping  Set of aesthetic mappings created by 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. A function can be created from a formula (e.g. ~ head(.x, 10)).
stat

The statistical transformation to use on the data for this layer, either as a ggproto Geom subclass or as a string naming the stat stripped of the stat_ prefix (e.g. "count" rather than "stat_count")

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

method

Method for arranging points (see Details below)

cex

Scaling for adjusting point spacing (see beeswarm::swarmx()). Values between 1 (default) and 3 tend to work best.

side

Direction to perform jittering: 0: both directions; 1: to the right or upwards; -1: to the left or downwards.

priority

Method used to perform point layout (see Details below)

fast

Use compiled version of swarm algorithm? This option is ignored for all methods expect "swarm" and "compactswarm".

dodge.width

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

corral

string. Method used to adjust points that would be placed to wide horizontally, default is "none". See details below.

corral.width

numeric. Width of the corral, default is 0.9.

groupOnX

[Deprecated] No longer needed.

beeswarmArgs

[Deprecated] No longer used.

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. It can also be a named logical vector to finely select the aesthetics to display.

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().

Aesthetics

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

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

Learn more about setting these aesthetics in vignette("ggplot2-specs").
See Also

`geom_quasirandom()` an alternative method, `beeswarm:::swarmx()` how spacing is determined, `ggplot2::geom_point()` for regular, unjittered points, `ggplot2::geom_jitter()` for jittered points, `ggplot2::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',size=2.5)
```

Description

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

Usage

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

mapping Set of aesthetic mappings created by `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. A function can be created from a formula (e.g. `~ head(.x, 10)`).

stat The statistical transformation to use on the data for this layer, either as a `ggproto` Geom subclass or as a string naming the stat stripped of the `stat_` prefix (e.g. "count" rather than "stat_count")

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

method the method used for distributing points (quasirandom, pseudorandom, smiley, maxout, frowney, minout, tukey, tukeyDense). See `vipor::offsetSingleGroup()` for the details of each method.

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)

`dodge.width` Amount by which points from different aesthetic groups will be dodged. This requires that one of the aesthetics is a factor. To disable dodging between groups, set this to `NULL`.

groupOnX **[Superseded]** See `orientation`.

orientation The orientation (i.e., which axis to group on) is inferred from the data. This can be overridden by setting `orientation` to either "x" or "y".

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. It can also be a named logical vector to finely select the aesthetics to display.

`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()`.
Aesthetics

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

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

Learn more about setting these aesthetics in vignette("ggplot2-specs").

See Also

viper::offsetSingleGroup() how spacing is determined, ggplot2::geom_point() for regular, unjittered points, ggplot2::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)
```

Description

This package allows plotting of several groups of one dimensional data as a violin point/beeswarm plot in ggplot2 by arranging data points to resemble the underlying distribution. The development version of this package is on https://github.com/eclarke/ggbeeswarm.

Author(s)

Erik Clarke, <erikclarke@gmail.com>
position_beeswarm

See Also

position_quasirandom()

Examples

```
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**  
Arrange points using the \link[beeswarm] package.

Description

Arrange points using the \link[beeswarm] package.

Usage

```
position_beeswarm(
  method = "swarm",
  cex = 1,
  side = 0L,
  priority = "ascending",
  fast = TRUE,
  groupOnX = NULL,
  dodge.width = 0,
  corral = "none",
  corral.width = 0.2
)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>Method for arranging points (see Details below)</td>
</tr>
<tr>
<td>cex</td>
<td>Scaling for adjusting point spacing (see \code{beeswarm::swarmx()}). Values between 1 (default) and 3 tend to work best.</td>
</tr>
<tr>
<td>side</td>
<td>Direction to perform jittering: 0: both directions; 1: to the right or upwards; -1: to the left or downwards.</td>
</tr>
<tr>
<td>priority</td>
<td>Method used to perform point layout (see Details below)</td>
</tr>
<tr>
<td>fast</td>
<td>Use compiled version of swarm algorithm? This option is ignored for all methods expect &quot;swarm&quot; and &quot;compactswarm&quot;.</td>
</tr>
</tbody>
</table>
position_quasirandom

groupOnX [ Deprecated] No longer needed.
dodge.width Amount by which points from different aesthetic groups will be dodged. This requires that one of the aesthetics is a factor.
coral string. Method used to adjust points that would be placed to wide horizontally, default is "none". See details below.
coral.width numeric. Width of the corral, default is 0.9.

Details

**method**: specifies the algorithm used to avoid overlapping points. The default "swarm" method places points in increasing order. If a point would overlap with an existing point, it is shifted sideways (along the group axis) by a minimal amount sufficient to avoid overlap.

While the "swarm" method places points in a predetermined order, the "compactswarm" method uses a greedy strategy to determine which point will be placed next. This often leads to a more tightly-packed layout. The strategy is very simple: on each iteration, a point that can be placed as close as possible to the non-data axis is chosen and placed. If there are two or more equally good points, priority is used to break ties.

The other 3 methods first discretise the values along the data axis, in order to create more efficient packing. The "square" method places points on a square grid, whereas "hex" uses a hexagonal grid. "centre"/"center" uses a square grid to produce a symmetric swarm. The number of break points for discretisation is determined by a combination of the available plotting area and the cex argument.

**priority**: controls the order in which points are placed, which generally has a noticeable effect on the plot appearance. "ascending" gives the 'traditional' beeswarm plot. "descending" is the opposite. "density" prioritizes points with higher local density. "random" places points in a random order. "none" places points in the order provided.

**corral**: By default, swarms from different groups are not prevented from overlapping, i.e. "corral = "none". Thus, datasets that are very large or unevenly distributed may produce ugly overlapping beeswarms. To control runaway points one can use the following methods. "gutter" collects runaway points along the boundary between groups. "wrap" implement periodic boundaries. "random" places runaway points randomly in the region. "omit" omits runaway points.

See Also

geom_beeswarm(), position_quasirandom(), beeswarm::swarmx()

Other position adjustments: offset_beeswarm(), position_quasirandom()
Usage

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

Arguments

- **method**: the method used for distributing points (quasirandom, pseudorandom, smiley, maxout, frowney, minout, tukey, tukeyDense). See `vipor::offsetSingleGroup()` for the details of each method.
- **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)
- **dodge.width**: Amount by which points from different aesthetic groups will be dodged. This requires that one of the aesthetics is a factor. To disable dodging between groups, set this to NULL.
- **orientation**: The orientation (i.e., which axis to group on) is inferred from the data. This can be overridden by setting orientation to either "x" or "y".
- **groupOnX**: [Superseded] See orientation.
- **na.rm**: if FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.

See Also

- `vipor::offsetSingleGroup()`, `geom_quasirandom()`
- Other position adjustments: `offset_beeswarm()`, `position_beeswarm()`
Index

* **position adjustments**
  
  position_beeswarm, 7
  position_quasirandom, 8

aes(), 2, 5

beeswarm::swarmx(), 3, 4, 7, 8
borders(), 3, 5

fortify(), 2, 5

geom_beeswarm, 2
geom_beeswarm(), 8
geom_boxplot(), 6
geom_quasirandom, 4
geom_quasirandom(), 4, 9
ggbeeswarm, 6
ggplot(), 2, 5

ggplot2::geom_boxplot(), 4
ggplot2::geom_jitter(), 4, 6
ggplot2::geom_point(), 4, 6

layer(), 3, 5

offset_beeswarm, 8, 9

position_beeswarm, 7, 9
position_quasirandom, 8, 8
position_quasirandom(), 7, 8

viper::offsetSingleGroup(), 5, 6, 9