Package ‘ggrastr’

June 21, 2020

Type Package
Title Raster Layers for ‘ggplot2’
Version 0.1.9
Description Provides a set of geoms to rasterize only specific layers of the plot while simultaneously keeping all labels and text in vector format. This allows users to keep plots within the reasonable size limit without loosing vector properties of the scale-sensitive information.
License MIT + file LICENSE
Encoding UTF-8
LazyData true
Imports ggplot2 (>= 2.1.0), Cairo (>= 1.5.9), ggbeeswarm
Depends R (>= 3.2.2)
RoxygenNote 7.1.0
Suggests rmarkdown, knitr
VignetteBuilder knitr
URL https://github.com/VPetukhov/ggrastr
BugReports https://github.com/VPetukhov/ggrastr/issues
NeedsCompilation no
Author Viktor Petukhov [aut, cph],
    Evan Biederstedt [cre, aut]
Maintainer Evan Biederstedt <evan.biederstedt@gmail.com>
Repository CRAN
Date/Publication 2020-06-20 22:30:02 UTC

R topics documented:

geom_beeswarm_rast ................................................. 2
geom_boxplot_jitter ................................................. 4
geom_jitter_rast ..................................................... 6
geom_point_rast ..................................................... 8
geom_beeswarm_rast

This geom is similar to geom_beeswarm, but creates a raster layer

Description

This geom is similar to geom_beeswarm, but creates a raster layer

Usage

geom_beeswarm_rast(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "quasirandom",
  priority = c("ascending", "descending", "density", "random", "none"),
  cex = 1,
  groupOnX = NULL,
  dodge.width = 0,
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  raster.width = NULL,
  raster.height = NULL,
  raster.dpi = 300
)

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. A function can be created from a formula (e.g. ~ head(.x, 10)).
geom_beeswarm_rast

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.

priority
Method used to perform point layout (see ggbeeswarm::position_beeswarm)

cex
Scaling for adjusting point spacing (see ggbeeswarm::position_beeswarm)

groupOnX
Should jitter be added to the x axis if TRUE or y axis if FALSE (the default NULL causes the function to guess which axis is the categorical one based on the number of unique entries in each) Refer to see ggbeeswarm::position_beeswarm

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

... 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.

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().

raster.width
Width of the result image (in inches). Default: determinied by the current device parameters.

raster.height
Height of the result image (in inches). Default: determinied by the current device parameters.

raster.dpi
Resolution of the result image.

Value

geom_beeswarm plot with rasterized layer

Examples

library(ggplot2)
library(ggrastr)

ggplot(mtcars) + geom_beeswarm_rast(aes(x = factor(cyl), y = mpg), raster.dpi = 600, cex = 1.5)
This geom is similar to geom_boxplot, but allows to jitter outlier points and to raster points layer.

Usage

geom_boxplot_jitter(
  mapping = NULL,
  data = NULL,
  stat = "boxplot",
  position = "dodge",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,

  ..., 
  outlier.jitter.width = NULL,
  outlier.jitter.height = 0,
  raster = FALSE,
  raster.dpi = 300,
  raster.width = NULL,
  raster.height = NULL
)

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. A function can be created from a formula (e.g. ~ head(.x,10)).
stat Use to override the default connection between geom_boxplot and stat_boxplot.
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. 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().

... 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.

outlier.jitter.width Amount of horizontal jitter. The jitter is added in both positive and negative directions, so the total spread is twice the value specified here. Default: boxplot width.

outlier.jitter.height Amount of horizontal jitter. The jitter is added in both positive and negative directions, so the total spread is twice the value specified here. Default: 0.

raster Should outlier points be rastered?.
raster.dpi Resolution of the rastered image. Ignored if raster == FALSE.
raster.width Width of the result image (in inches). Default: determined by the current device parameters. Ignored if raster == FALSE.
raster.height Height of the result image (in inches). Default: determined by the current device parameters. Ignored if raster == FALSE.

Value

geom_boxplot plot with rasterized layer

Aesthetics

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

• x or y
• lower or xlower
• upper or xupper
• middle or xmiddle
• ymin or xmin
• ymax or xmax
• alpha
• colour
• fill
• group
• linetype
• shape
• size
• weight

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

```r
library(ggplot2)
library(ggrastr)

yvalues = rt(1000, df=3)
xvalues = as.factor(1:1000 %% 2)
ggplot() + geom_boxplot_jitter(aes(y=yvalues, x=xvalues), outlier.jitter.width = 0.1, raster = TRUE)
```

---

**geom_jitter_rast**  
This geom is similar to `geom_jitter`, but creates a raster layer

### Description

This geom is similar to `geom_jitter`, but creates a raster layer

### Usage

```r
geom_jitter_rast(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "jitter",
  width = NULL,
  height = NULL,
  seed = NA,
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  raster.width = NULL,
  raster.height = NULL,
  raster.dpi = 300
)
```

### 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. 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, as a string.

**position**

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

**width**

Amount of vertical and horizontal jitter. The jitter is added in both positive and negative directions, so the total spread is twice the value specified here. Refer to ggplot2::position_jitter.

**height**

Amount of vertical and horizontal jitter. The jitter is added in both positive and negative directions, so the total spread is twice the value specified here. Refer to ggplot2::position_jitter.

**seed**

A random seed to make the jitter reproducible. Refer to ggplot2::position_jitter.

**...**

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.

**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()`.

**raster.width**

Width of the result image (in inches). Default: determined by the current device parameters.

**raster.height**

Height of the result image (in inches). Default: determined by the current device parameters.

**raster.dpi**

Resolution of the result image.

---

### Value

`geom_point_rast` plot with rasterized layer

### 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").

Examples

```r
library(ggplot2)
library(ggrastr)

ggplot(mpg) + geom_jitter_rast(aes(x = factor(cyl), y = hwy), raster.dpi = 600)
```

---

`geom_point_rast`  
This geom is similar to `geom_point`, but creates a raster layer

Description

This geom is similar to `geom_point`, but creates a raster layer

Usage

```r
geom_point_rast(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  raster.width = NULL,
  raster.height = NULL,
  raster.dpi = 300
)
```

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. 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, as a string.

**position**

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

... 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.

**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()`.

**raster.width**

Width of the result image (in inches). Default: determined by the current device parameters.

**raster.height**

Height of the result image (in inches). Default: determined by the current device parameters.

**raster.dpi**

Resolution of the result image.

### Value

```
geom_point_rast
```

### geom_point_rast plot with rasterized layer

### 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")`. 
Examples

```r
library(ggplot2)
library(ggrastr)

ggplot() + geom_point_rast(aes(x=rnorm(1000), y=rnorm(1000)), raster.dpi=600)
```

---

**geom_quasirandom_rast**  *This geom is similar to geom_quasirandom, but creates a raster layer*

Description

This geom is similar to `geom_quasirandom`, but creates a raster layer.

Usage

```r
gem_quasirandom_rast(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "quasirandom",
  width = NULL,
  varwidth = FALSE,
  bandwidth = 0.5,
  nbins = NULL,
  method = "quasirandom",
  groupOnX = NULL,
  dodge.width = 0,
  ...
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  raster.width = NULL,
  raster.height = NULL,
  raster.dpi = 300
)
```

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. 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, as a string.

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

**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`.

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

**...**
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.

**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()`.

**raster.width**
Width of the result image (in inches). Default: determined by the current device parameters.

**raster.height**
Height of the result image (in inches). Default: determined by the current device parameters.

**raster.dpi**
Resolution of the result image.

**Value**

`geom_quasirandom_rast` plot with rasterized layer
Aesthetics

gem_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").

Examples

library(ggplot2)
library(ggrastr)

ggplot(mtcars) + geom_quasirandom_rast(aes(x = factor(cyl), y = mpg), raster.dpi = 600)

geom_tile_rast

This geom is similar to geom_tile, but creates a raster layer

Description

This geom is similar to geom_tile, but creates a raster layer

Usage

geom_tile_rast(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  raster.width = NULL,
  raster.height = NULL,
  raster.dpi = 300
)
Arguments

mapping Set of aesthetic mappings created by \texttt{aes()} or \texttt{aes()}. If specified and \texttt{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 \texttt{NULL}, the default, the data is inherited from the plot data as specified in the call to \texttt{ggplot()}.
A \texttt{data.frame}, or other object, will override the plot data. All objects will be fortified to produce a data frame. See \texttt{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 \texttt{data.frame}, and will be used as the layer data. A function can be created from a formula (e.g. \texttt{~ head(.x,10)}).

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.

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

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

show.legend logical. Should this layer be included in the legends? \texttt{NA}, the default, includes if any aesthetics are mapped. \texttt{FALSE} never includes, and \texttt{TRUE} always includes. It can also be a named logical vector to finely select the aesthetics to display.

inherit.aes If \texttt{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. \texttt{borders()}.

raster.width Width of the result image (in inches). Default: determined by the current device parameters.

raster.height Height of the result image (in inches). Default: determined by the current device parameters.

raster.dpi Resolution of the result image.

Value

\texttt{geom_tile_rast} plot with rasterized layer

Aesthetics

\texttt{geom_tile()} understands the following aesthetics (required aesthetics are in bold):

\begin{itemize}
\item x
\item y
\item alpha
\end{itemize}
• colour
• fill
• group
• height
• linetype
• size
• width

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

Examples

```r
library(ggplot2)
library(ggrastr)

coords <- expand.grid(1:100, 1:100)
coords$Value <- 1 / apply(as.matrix(coords), 1, function(x) sum((x - c(50, 50))^2)^0.01)
ggplot(coords) + geom_tile_rast(aes(x=Var1, y=Var2, fill=Value))
```

---

<table>
<thead>
<tr>
<th>theme_pdf</th>
<th>Pretty theme</th>
</tr>
</thead>
</table>

Description

Pretty theme

Usage

```r
theme_pdf(show.ticks = TRUE, legend.pos = NULL)
```

Arguments

- `show.ticks`: Show x- and y-ticks.
- `legend.pos`: Vector with x and y position of the legend.

Value

ggplot2 with plot ticks and positioned legend

Examples

```r
library(ggplot2)
library(ggrastr)

data = rnorm(100)
colors = (1:100/100)
ggplot() + geom_point(aes(x=data, y=data, color=colors)) + theme_pdf(FALSE, legend.pos=c(1, 1))
```
Index

aes(), 2, 4, 6, 8, 10, 13
aes_(), 2, 4, 6, 8, 10, 13

borders(), 3, 5, 7, 9, 11, 13

coord_flip, 11

fortify(), 2, 4, 6, 8, 11, 13

geom_beeswarm, 2
geom_beeswarm_rast, 2
geom_boxplot, 4
geom_boxplot_jitter, 4
geom_jitter, 6
geom_jitter_rast, 6
geom_point, 8
geom_point_rast, 8
geom_quasirandom, 10
geom_quasirandom_rast, 10
geom_tile, 12
geom_tile_rast, 12

ggplot(), 2, 4, 6, 8, 10, 13

layer(), 3, 5, 7, 9, 11, 13

theme_pdf, 14