Package ‘ggrastr’

March 1, 2021

Type Package
Title Rasterize Layers for ‘ggplot2’
Version 0.2.3
Description Rasterize only specific layers of a ‘ggplot2’ plot while simultaneously keeping all labels and text in vector format. This allows users to keep plots within the reasonable size limit without losing 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, grid, png, ragg
Depends R (>= 3.2.2)
RoxygenNote 7.1.1
Suggests knitr, maps, rmarkdown, sf
VignetteBuilder knitr
URL https://github.com/VPetukhov/ggrastr
BugReports https://github.com/VPetukhov/ggrastr/issues
NeedsCompilation no
Author Viktor Petukhov [aut, cph],
Teun van den Brand [aut],
Evan Biederstedt [cre, aut]
Maintainer Evan Biederstedt <evan.biederstedt@gmail.com>
Repository CRAN
Date/Publication 2021-03-01 14:00:02 UTC

R topics documented:

geom_beeswarm_rast .................................................. 2
geom_boxplot_jitter ................................................... 3
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

```r
gem_beeswarm_rast(
  ..., priority = c("ascending", "descending", "density", "random", "none"),
  cex = 1,
  groupOnX = NULL,
  dodge.width = 0,
  raster.dpi = getOption("ggbeeswarm.default.dpi", 300),
  dev = "cairo",
  scale = 1
)
```

### Arguments

- `...` 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.
- `priority` string Method used to perform point layout (see ggbeeswarm::position_beeswarm).
- `cex` numeric Scaling for adjusting point spacing (see ggbeeswarm::position_beeswarm).
- `groupOnX` boolean Whether jitter be added to the x axis (default=NULL). if TRUE then jitter is added to the x axis and if FALSE jitter is added to the y axis. (The default NULL causes the function to guess which axis is the categorical axis based on the number of unique entries in each). Refer to see ggbeeswarm::position_beeswarm for more details.
- `dodge.width` numeric Amount by which points from different aesthetic groups will be dodged (default=0). This requires that one of the aesthetics is a factor. Refer to see ggbeeswarm::position_beeswarm for more details.
- `raster.dpi` integer Resolution of the rastered image in dots per inch (default=300).
geom_boxplot_jitter

- **dev**
  - string: Specifies the device used, which can be one of: "cairo", "ragg" or "ragg_png" (default="cairo").

- **scale**
  - numeric: Scaling factor to modify the raster object size (default=1). The parameter 'scale=1' results in an object size that is unchanged, 'scale'>1 increase the size, and 'scale'<1 decreases the size. These parameters are passed to 'height' and 'width' of grid::grid.raster(). Please refer to 'rasterise()' and 'grid::grid.raster()' for more details.

**Value**

geom_beeswarm plot with rasterized layer

**Examples**

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

ggplot(mtcars) + geom_beeswarm_rast(aes(x = factor(cyl), y = mpg), raster.dpi = 600, cex = 1.5)
```

---

**Description**

This geom is similar to geom_boxplot, but allows to jitter outlier points and to raster points layer.

**Usage**

```r
gem_boxplot_jitter(
  mapping = NULL,
  data = NULL,
  dev = "cairo",
  stat = "boxplot",
  position = "dodge",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...
  outlier.jitter.width = NULL,
  outlier.jitter.height = 0,
  raster.dpi = getOption("ggrastr.default.dpi", 300),
  scale = 1
)
```
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)`).

**dev**
`string` Specifies the device used, which can be one of: "cairo", "ragg" or "ragg_png" (default="cairo").

**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**
numeric Amount of horizontal jitter (default=NULL). The jitter is added in both positive and negative directions, so the total spread is twice the value specified here. If `NULL`, no jitter performed.

**outlier.jitter.height**
numeric Amount of horizontal jitter (default=0). The jitter is added in both positive and negative directions, so the total spread is twice the value specified here.

**raster.dpi**
`integer` Resolution of the rastered image (default=300). Ignored if `raster == FALSE`.

**scale**
numeric Scaling factor to modify the raster object size (default=1). The parameter `scale=1` results in an object size that is unchanged, `scale>1` increase the size, and `scale<1` decreases the size. These parameters are passed to `height` and `width` of `grid::grid.raster()`. Please refer to `rasterise()` and `grid::grid.raster()` for more details.
geom_jitter_rast

Value

geom_boxplot plot with rasterized layer

Aesthetics

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

- \textit{x} or \textit{y}
- lower or xlower
- upper or xupper
- middle or xmmiddle
- 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

library(ggplot2)
library(ggrastr)

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

\begin{Verbatim}

geom_jitter_rast \hspace{1cm} This geom is similar to geom_jitter, but creates a raster layer

\end{Verbatim}

Description

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

```r
geom_jitter_rast(
  ..., 
  raster.dpi = getOption("ggrastr.default.dpi", 300), 
  dev = "cairo", 
  scale = 1
)
```

Arguments

- `...`: 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.
- `raster.dpi`: integer Resolution of the rastered image in dots per inch (default=300).
- `dev`: string Specifies the device used, which can be one of: "cairo", "ragg" or "ragg_png" (default="cairo").
- `scale`: numeric Scaling factor to modify the raster object size (default=1). The parameter 'scale=1' results in an object size that is unchanged, 'scale'>1 increase the size, and 'scale'<1 decreases the size. These parameters are passed to 'height' and 'width' of `grid::grid.raster()`. Please refer to 'rasterise()' and `grid::grid.raster()` for more details.

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)
```

**Description**

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

**Usage**

```r
gem_point_rast(
  ..., 
  raster.dpi = getOption("ggrastr.default.dpi", 300), 
  dev = "cairo", 
  scale = 1 
)
```

**Arguments**

- `...`: 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.
- `raster.dpi`: integer Resolution of the rastered image in dots per inch (default=300).
- `dev`: string Specifies the device used, which can be one of: "cairo", "ragg" or "ragg_png" (default="cairo").
- `scale`: numeric Scaling factor to modify the raster object size (default=1). The parameter 'scale=1' results in an object size that is unchanged, 'scale'>1 increase the size, and 'scale'<1 decreases the size. These parameters are passed to 'height' and 'width' of grid::grid.raster(). Please refer to 'rasterise()' and 'grid::grid.raster()' for more details.

**Value**

geom_point 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**

```
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
geom_quasirandom_rast(
  ..., 
  width = NULL, 
  varwidth = FALSE, 
  bandwidth = 0.5, 
  nbins = NULL, 
  method = "quasirandom", 
  groupOnX = NULL, 
  dodge.width = 0, 
  raster.dpi = getOption("ggrastr.default.dpi", 300), 
  dev = "cairo", 
  scale = 1  
)
```
Arguments

... 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.

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.
raster.dpi integer Resolution of the rastered image in dots per inch (default=300).
dev string Specifies the device used, which can be one of: "cairo", "ragg" or "ragg_png" (default="cairo").
scale numeric Scaling factor to modify the raster object size (default=1). The parameter 'scale=1' results in an object size that is unchanged, 'scale'>1 increase the size, and 'scale'<1 decreases the size. These parameters are passed to 'height' and 'width' of grid::grid.raster(). Please refer to 'rasterise()' and 'grid::grid.raster()' for more details.

Value

geom_quasirandom 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

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(
  ..., 
  raster.dpi = getOption("ggrastr.default.dpi", 300), 
  dev = "cairo", 
  scale = 1
)

Arguments

... 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.

raster.dpi integer Resolution of the rastered image in dots per inch (default=300).

dev string Specifies the device used, which can be one of: "cairo", "ragg" or "ragg_png" (default="cairo").

scale numeric Scaling factor to modify the raster object size (default=1). The parameter 'scale=1' results in an object size that is unchanged, 'scale'>1 increase the size, and 'scale'<1 decreases the size. These parameters are passed to 'height' and 'width' of grid::grid.raster(). Please refer to 'rasterise()' and 'grid::grid.raster()' for more details.

Value

geom_tile plot with rasterized layer
Aesthetics

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

- x
- y
- alpha
- 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))
```

**geom_violin_rast**  
This geom is similar to `geom_violin`, but creates a raster layer

Description

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

Usage

```r
geom_violin_rast(
  ...,  
  raster.dpi = getOption("ggrastr.default.dpi", 300),  
  dev = "cairo",  
  scale = 1
)
```
Arguments

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`.

`raster.dpi`  integer Resolution of the rastered image in dots per inch (default=300).
`dev` string Specifies the device used, which can be one of: "cairo", "ragg" or "ragg_png" (default="cairo").
`scale` numeric Scaling factor to modify the raster object size (default=1). The parameter `scale=1` results in an object size that is unchanged, `scale>1` increase the size, and `scale<1` decreases the size. These parameters are passed to 'height' and 'width' of `grid::grid.raster()`. Please refer to 'rasterise()' and 'grid::grid.raster()' for more details.

Value

`geom_violin_rast` plot with rasterized layer

Aesthetics

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

- x
- y
- alpha
- colour
- fill
- group
- linetype
- size
- weight

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

Examples

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

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

*Rasterise ggplot layers Takes a ggplot layer as input and renders their graphical output as a raster.*

**Description**

Rasterise ggplot layers Takes a ggplot layer as input and renders their graphical output as a raster.

**Usage**

```r
rasterise(
  layer,
  dpi =getOption("ggrastr.default.dpi"),
  dev = "cairo",
  scale = 1
)
```

**Arguments**

- **layer**
  - A `Layer` object, typically constructed with a call to a `geom_()` or `stat_()` function.
- **dpi**
  - integer Sets the desired resolution in dots per inch (default=NULL).
- **dev**
  - string Specifies the device used, which can be one of: "cairo", "ragg" or "ragg_png" (default="cairo").
- **scale**
  - numeric Scaling factor to modify the raster object size (default=1). The parameter 'scale'=1 results in an object size that is unchanged, 'scale'>1 increase the size, and 'scale'<1 decreases the size. These parameters are passed to 'height' and 'width' of grid::grid.raster(). Please refer to 'rasterise()' and 'grid::grid.raster()' for more details.

**Details**

The default dpi (NULL (i.e. let the device decide)) can conveniently be controlled by setting the option "ggrastr.default.dpi" (e.g. option("ggrastr.default.dpi",30) for drafting).

**Value**

A modified `Layer` object.
Author(s)

Teun van den Brand <t.v.d.brand@nki.nl>

Examples

```r
require(ggplot2)
# `rasterise()` is used to wrap layers
ggplot(pressure, aes(temperature, pressure)) +
  rasterise(geom_line())

# The `dpi` argument controls resolution
ggplot(faithful, aes(eruptions, waiting)) +
  rasterise(geom_point(), dpi = 5)

# The `dev` argument offers a few options for devices
require(ragg)
ggplot(diamonds, aes(carat, depth, z = price)) +
  rasterise(stat_summary_hex(), dev = "ragg")

# The `scale` argument allows you to render a 'big' plot in small window, or vice versa.
ggplot(faithful, aes(eruptions, waiting)) +
  rasterise(geom_point(), scale = 4)
```

---

**theme_pdf**

Pretty theme

Description

Pretty theme

Usage

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

Arguments

- `show.ticks` boolean Whether to show x- and y-ticks (default=TRUE).
- `legend.pos` Vector with x and y position of the legend (default=NULL).

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(), 4
aes_(), 4

borders(), 4
coord_flip, 9

fortify(), 4

geom_beeswarm, 2
geom_beeswarm_rast, 2
geom_boxplot, 3
geom_boxplot_jitter, 3
geom_jitter, 5
geom_jitter_rast, 5
geom_point, 7
geom_point_rast, 7
geom_quasirandom, 8
geom_quasirandom_rast, 8
geom_tile, 10
geom_tile_rast, 10
geom_violin, 11
geom_violin_rast, 11

ggplot(), 4

layer(), 2, 4, 6, 7, 9, 10, 12

rasterise, 13
rasterize(rasterise), 13

theme_pdf, 14