

# Package: ggrastr (via r-universe)

May 22, 2026

**Type** Package

**Title** Rasterize Layers for 'ggplot2'

**Version** 1.0.2

**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

**Imports** ggplot2 (>= 2.1.0), Cairo (>= 1.5.9), ggbeeswarm, grid, png, ragg

**Depends** R (>= 3.2.2)

**RoxygenNote** 7.2.3

**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>

**Config/pak/sysreqs** libcairo2-dev libfontconfig1-dev libfreetype6-dev libfribidi-dev libharfbuzz-dev libjpeg-dev libpng-dev libtiff-dev libwebp-dev

**Repository** <https://fastverse.r-universe.dev>

**Date/Publication** 2023-06-06 22:40:37 UTC

**RemoteUrl** <https://github.com/VPetukhov/ggrastr>

**RemoteRef** HEAD

**RemoteSha** 50ca3e012296ba73a8a421f8f02046d893bf2696

## Contents

geom_beeswarm_rast . . . . .	2
geom_boxplot_jitter . . . . .	3
geom_jitter_rast . . . . .	6
geom_point_rast . . . . .	7
geom_quasirandom_rast . . . . .	8
geom_tile_rast . . . . .	10
geom_violin_rast . . . . .	11
rasterise . . . . .	13
rasterize . . . . .	14
theme_pdf . . . . .	15

<b>Index</b>	<b>17</b>
--------------	-----------

---

geom_beeswarm_rast	<i>This geom is similar to <a href="#">geom_beeswarm</a>, but creates a raster layer</i>
--------------------	------------------------------------------------------------------------------------------

---

## Description

This geom is similar to [geom\\_beeswarm](#), but creates a raster layer

## Usage

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

## Arguments

...	Other arguments passed on to <a href="#">layer()</a> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.
priority	string Method used to perform point layout (see <code>ggbeeswarm::position_beeswarm</code> ).
cex	numeric Scaling for adjusting point spacing (see <code>ggbeeswarm::position_beeswarm</code> )
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 <code>ggbeeswarm::position_beeswarm</code> 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 <code>ggbeeswarm::position_beeswarm</code> for more details.
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 <code>grid::grid.raster()</code> . Please refer to 'rasterise()' and 'grid::grid.raster()' for more details.

**Value**

geom\_beeswarm plot with rasterized layer

**Examples**

```
library(ggplot2)
library(ggtrastr)

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

---

`geom_boxplot_jitter` *This geom is similar to [geom\\_boxplot](#), but allows to jitter outlier points and to raster points layer.*

---

**Description**

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,
  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("ggtrastr.default.dpi", 300),
  scale = 1
)
```

**Arguments**

mapping	Set of aesthetic mappings created by <code>aes()</code> . If specified and <code>inherit.aes = TRUE</code> (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 <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code> . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify()</code> for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code> , and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code> ).
dev	string Specifies the device used, which can be one of: "cairo", "ragg" or "ragg_png" (default="cairo").
stat	string 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 <code>stat_</code> prefix (e.g. "count" rather than "stat_count"). Refer to <code>ggplot2::layer</code> .
position	Position adjustment, either as a string naming the adjustment (e.g. "jitter" to use <code>position_jitter</code> ), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.
na.rm	If <code>FALSE</code> , the default, missing values are removed with a warning. If <code>TRUE</code> , missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? <code>NA</code> , the default, includes if any aesthetics are mapped. <code>FALSE</code> never includes, and <code>TRUE</code> always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If <code>FALSE</code> , 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. <code>borders()</code> .
...	Other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.
outlier.jitter.width	numeric Amount of horizontal jitter (default= <code>NULL</code> ). The jitter is added in both positive and negative directions, so the total spread is twice the value specified here. If <code>NULL</code> , no jitter performed.
outlier.jitter.height	numeric Amount of horizontal jitter (default= <code>0</code> ). 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= <code>300</code> ). Ignored if <code>raster == FALSE</code> .
scale	numeric Scaling factor to modify the raster object size (default= <code>1</code> ). 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\_boxplot plot with rasterized layer

### Aesthetics

geom\_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
- linewidth
- shape
- size
- weight

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

### Examples

```
library(ggplot2)
library(ggtrastr)

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

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

### Arguments

...	Other arguments passed on to <a href="#">layer()</a> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> . 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 <code>grid::grid.raster()</code> . 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

```
library(ggplot2)
library(ggtrastr)

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

```
geom_point_rast(
  ...,
  raster.dpi = getOption("ggtrastr.default.dpi", 300),
  dev = "cairo",
  scale = 1
)
```

### Arguments

...	Other arguments passed on to <a href="#">layer()</a> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> . 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 <code>grid::grid.raster()</code> . 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(ggtrastr)

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

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

**Arguments**

...	Other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.
<code>width</code>	the maximum amount of spread (default: 0.4)
<code>varwidth</code>	vary the width by the relative size of each group
<code>bandwidth</code>	the bandwidth adjustment to use when calculating density. Smaller numbers (< 1) produce a tighter "fit". (default: 0.5)
<code>nbins</code>	the number of bins used when calculating density (has little effect with quasirandom/random distribution)
<code>method</code>	the method used for distributing points (quasirandom, pseudorandom, smiley, maxout, frowney, minout, tukey, tukeyDense). See <code>vipor::offsetSingleGroup()</code> for the details of each method.
<code>groupOnX</code>	<b>[Deprecated]</b> No longer needed.
<code>dodge.width</code>	Amount by which points from different aesthetic groups will be dodged. This requires that one of the aesthetics is a factor.
<code>raster.dpi</code>	integer Resolution of the rastered image in dots per inch (default=300).
<code>dev</code>	string Specifies the device used, which can be one of: "cairo", "ragg" or "ragg_png" (default="cairo").
<code>scale</code>	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 <code>grid::grid.raster()</code> . 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(ggtrastr)

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("ggtrastr.default.dpi", 300),
  dev = "cairo",
  scale = 1
)
```

**Arguments**

...	Other arguments passed on to <a href="#">layer()</a> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.
<code>raster.dpi</code>	integer Resolution of the rastered image in dots per inch (default=300).
<code>dev</code>	string Specifies the device used, which can be one of: "cairo", "ragg" or "ragg_png" (default="cairo").
<code>scale</code>	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 <code>grid::grid.raster()</code> . 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
- linewidth
- width

Note that geom\_raster() ignores colour.

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

## Examples

```
library(ggplot2)
library(ggtrastr)

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

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

**Arguments**

...	Other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.
<code>raster.dpi</code>	integer Resolution of the rastered image in dots per inch (default=300).
<code>dev</code>	string Specifies the device used, which can be one of: "cairo", "ragg" or "ragg_png" (default="cairo").
<code>scale</code>	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 <code>grid::grid.raster()</code> . 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
- linewidth
- weight

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

**Examples**

```
library(ggplot2)
library(ggtrastr)

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

---

rasterise	<i>Rasterise ggplot layers Takes a ggplot object or a layer as input and renders their graphical output as a raster.</i>
-----------	--------------------------------------------------------------------------------------------------------------------------

---

### Description

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

### Usage

```
rasterise(input, ...)

## S3 method for class 'Layer'
rasterise(input, ..., dpi = NULL, dev = "cairo", scale = 1)

## S3 method for class 'list'
rasterise(input, ..., dpi = NULL, dev = "cairo", scale = 1)

## S3 method for class 'ggplot'
rasterise(
  input,
  ...,
  layers = c("Point", "Tile"),
  dpi = NULL,
  dev = "cairo",
  scale = 1
)
```

### Arguments

input	ggplot plot object to rasterize
...	ignored
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", "ragg_png" or "cairo_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.
layers	list of layer types that should be rasterized

### Details

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

**Value**

A modified Layer object.

**Author(s)**

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

**Examples**

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

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

# The `dev` argument offers a few options for devices
require(ragg)
ggplot(diamonds, aes(carat, depth, z = price)) +
  rasterize(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)) +
  rasterize(geom_point(), scale = 4)
```

---

rasterize

*Rasterize ggplot layers Takes a ggplot object or a layer as input and renders their graphical output as a raster.*

---

**Description**

Rasterize ggplot layers Takes a ggplot object or a layer as input and renders their graphical output as a raster.

**Usage**

```
rasterize(input, ...)
```

**Arguments**

input	ggplot plot object to rasterize
...	ignored

**Details**

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

**Value**

A modified Layer object.

**Author(s)**

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

**Examples**

```
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**

```
library(ggplot2)
library(ggtrastr)

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](#)

`borders()`, [4](#)

`fortify()`, [4](#)

`geom_beeswarm`, [2](#)  
`geom_beeswarm_rast`, [2](#)  
`geom_boxplot`, [3](#)  
`geom_boxplot_jitter`, [3](#)  
`geom_jitter`, [6](#)  
`geom_jitter_rast`, [6](#)  
`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`, [14](#)

`theme_pdf`, [15](#)

`vipor::offsetSingleGroup()`, [9](#)