

# Package ‘ggridges’

September 15, 2017

**Type** Package

**Title** Ridgeline Plots in 'ggplot2'

**Version** 0.4.1

**Description** Ridgeline plots provide a convenient way of visualizing changes in distributions over time or space. This package enables the creation of such plots in 'ggplot2'.

**URL** <https://github.com/clauswilke/ggridges>

**Depends** R (>= 3.2), ggplot2 (>= 2.2.0),

**Imports** grid (>= 3.0.0), plyr (>= 1.8.0), scales (>= 0.4.1)

**License** GPL-2 | file LICENSE

**LazyData** true

**Suggests** covr, gridExtra, ggplot2movies, forcats, knitr, tidyverse, testthat, vdiff, viridis

**VignetteBuilder** knitr

**Collate** 'data.R' 'ggridges.R' 'geoms.R' 'geomsv.R' 'geoms-gradient.R' 'scale-cyclical.R' 'stats.R' 'theme.R' 'utils\_ggplot2.R' 'utils.R'

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Author** Claus O. Wilke [cre, aut],  
RStudio [cph] (Copyright for ggplot2 code copied to ggridges)

**Maintainer** Claus O. Wilke <wilke@austin.utexas.edu>

**Repository** CRAN

**Date/Publication** 2017-09-15 07:36:48 UTC

## R topics documented:

Catalan_elections . . . . .	2
geom_density_ridges . . . . .	2
geom_ridgeline . . . . .	5
geom_ridgeline_gradient . . . . .	6

geom_vridgeline . . . . .	8
ggridges . . . . .	9
lincoln_weather . . . . .	9
reduce . . . . .	11
scale_cyclical . . . . .	11
stat_binline . . . . .	12
stat_density_ridges . . . . .	15
theme_ridges . . . . .	16
<b>Index</b>	<b>18</b>

---

Catalan_elections	<i>Results from Catalan regional elections (1980-2015)</i>
-------------------	--

---

### Description

Data from Catalan regional elections for 949 municipalities, from 11 elections spanning the years 1980-2015. The data was obtained and processed from Idescat.cat by Marc Belzunces (Twitter: @marcbeldata).

### Usage

```
Catalan_elections
```

### Format

A tibble with 20764 rows and 4 variables:

Municipality

Year

Option The voter option; either "Indy" or "Unionist"

Percent The percentage of the voters choosing the given option

---

geom_density_ridges	<i>Create ridgeline plot</i>
---------------------	------------------------------

---

### Description

geom\_density\_ridges arranges multiple density plots in a staggered fashion, as in the cover of the famous Joy Division album Unknown Pleasures.

geom\_density\_ridges2 is identical to geom\_density\_ridges except it draws closed polygons rather than ridgelines.

**Usage**

```
geom_density_ridges(mapping = NULL, data = NULL, stat = "density_ridges",
  panel_scaling = TRUE, na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE, ...)

geom_density_ridges2(mapping = NULL, data = NULL, stat = "density_ridges",
  panel_scaling = TRUE, na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE, ...)
```

**Arguments**

mapping	Set of aesthetic mappings created by <code>aes()</code> or <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. 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.
stat	The statistical transformation to use on the data for this layer, as a string.
panel_scaling	If <code>TRUE</code> , the default, relative scaling is calculated separately for each panel. If <code>FALSE</code> , relative scaling is calculated globally.
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.
inherit.aes	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them.
...	other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>color = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.

**Details**

By default, this geom calculates densities from the point data mapped onto the x axis. If density calculation is not wanted, use `stat="identity"` or use `geom_ridgeline`. The difference between `geom_density_ridges` and `geom_ridgeline` is that `geom_density_ridges` will provide automatic scaling of the ridgelines (controlled by the `scale` aesthetic), whereas `geom_ridgeline` will plot the data as is. Note that when you set `stat="identity"`, the `height` aesthetic must be provided.

Note that the default `stat_density_ridges` makes joint density estimation across all datasets. This may not generate the desired result when using faceted plots. As an alternative, you can set `stat = "density"` to use `stat_density`. In this case, it is required to add the aesthetic mapping `height = ..density..` (see examples).

## Aesthetics

Required aesthetics are in bold.

- **x**
- **y**
- **group** Defines the grouping. Not needed if a categorical variable is mapped onto y, but needed otherwise. Will typically be the same variable as is mapped to y.
- **height** The height of each ridgeline at the respective x value. Automatically calculated and provided by `stat_density_ridges` if the default stat is not changed.
- **scale** A scaling factor to scale the height of the ridgelines relative to the spacing between them. A value of 1 indicates that the maximum point of any ridgeline touches the baseline right above, assuming even spacing between baselines.
- **rel\_min\_height** Lines with heights below this cutoff will be removed. The cutoff is measured relative to the overall maximum, so `rel_min_height=0.01` would remove everything that is 1% or less than the highest point among all ridgelines. Default is 0, so nothing is removed.
- **alpha**
- **color, fill, group, alpha, linetype, size**, as in `geom_ridgeline`.

## Examples

```
# set the `rel_min_height` argument to remove tails
ggplot(iris, aes(x = Sepal.Length, y = Species)) +
  geom_density_ridges(rel_min_height = 0.005) +
  scale_y_discrete(expand = c(0.01, 0)) +
  scale_x_continuous(expand = c(0.01, 0)) +
  theme_ridges()

# set the `scale` to determine how much overlap there is among the plots
ggplot(diamonds, aes(x = price, y = cut)) +
  geom_density_ridges(scale = 4) +
  scale_y_discrete(expand=c(0.01, 0)) +
  scale_x_continuous(expand=c(0.01, 0)) +
  theme_ridges()

# the same figure with colors, and using the ggplot2 density stat
ggplot(diamonds, aes(x = price, y = cut, fill = cut, height = ..density..)) +
  geom_density_ridges(scale = 4, stat = "density") +
  scale_y_discrete(expand = c(0.01, 0)) +
  scale_x_continuous(expand = c(0.01, 0)) +
  scale_fill_brewer(palette = 4) +
  theme_ridges() + theme(legend.position = "none")

# use geom_density_ridges2() instead of geom_density_ridges() for solid polygons
ggplot(iris, aes(x = Sepal.Length, y = Species)) +
  geom_density_ridges2() +
  scale_y_discrete(expand = c(0.01, 0)) +
  scale_x_continuous(expand = c(0.01, 0)) +
  theme_ridges()
```

---

geom_ridgeline	<i>Plot a ridgeline (line with filled area underneath)</i>
----------------	--

---

### Description

Plots the sum of the `y` and `height` aesthetics versus `x`, filling the area between `y` and `y + height` with a color. Thus, the data mapped onto `y` and onto `height` must be in the same units. If you want relative scaling of the heights, you can use `geom_density_ridges` with `stat = "identity"`.

### Usage

```
geom_ridgeline(mapping = NULL, data = NULL, stat = "identity",
              position = "identity", na.rm = FALSE, show.legend = NA,
              inherit.aes = TRUE, ...)
```

### Arguments

<code>mapping</code>	Set of aesthetic mappings created by <code>aes()</code> or <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 <code>mapping</code> if there is no plot mapping.
<code>data</code>	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. 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.
<code>stat</code>	The statistical transformation to use on the data for this layer, as a string.
<code>position</code>	Position adjustment, either as a string, or the result of a call to a position adjustment function.
<code>na.rm</code>	If <code>FALSE</code> , the default, missing values are removed with a warning. If <code>TRUE</code> , missing values are silently removed.
<code>show.legend</code>	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.
<code>inherit.aes</code>	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them.
<code>...</code>	other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>color = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired <code>geom/stat</code> .

### Aesthetics

Required aesthetics are in bold.

- **x**
- **y**

- **height** Height of the ridgeline, measured from the respective y value. Assumed to be positive, though this is not required.
- **group** Defines the grouping. Required when the dataset contains multiple distinct ridgelines. Will typically be the same variable as is mapped to y.
- **scale** A scaling factor to scale the height of the ridgelines. A value of 1 indicates that the heights are taken as is. This aesthetic can be used to convert height units into y units.
- **min\_height** A height cutoff on the drawn ridgelines. All values that fall below this cutoff will be removed. The main purpose of this cutoff is to remove long tails right at the baseline level, but other uses are possible. The cutoff is applied before any height scaling is applied via the scale aesthetic. Default is 0, so negative values are removed.
- **color** Color of the ridgeline
- **fill** Fill color of the area under the ridgeline
- **alpha** Transparency level of color and fill
- **group** Grouping, to draw multiple ridgelines from one dataset
- **linetype** Linetype of the ridgeline
- **size** Line thickness

### Examples

```
d <- data.frame(x = rep(1:5, 3), y = c(rep(0, 5), rep(1, 5), rep(3, 5)),
               height = c(0, 1, 3, 4, 0, 1, 2, 3, 5, 4, 0, 5, 4, 4, 1))
ggplot(d, aes(x, y, height = height, group = y)) + geom_ridgeline(fill="lightblue")
```

---

geom\_ridgeline\_gradient

*Plot ridgelines and ridgeline plots with fill gradients along the x axis*

---

### Description

The geoms `geom_ridgeline_gradient` and `geom_density_ridges_gradient` work just like [geom\\_ridgeline](#) and [geom\\_density\\_ridges](#) except that the fill aesthetic can vary along the x axis. Because filling with color gradients is fraught with issues, these geoms should be considered experimental. Don't use them unless you really need to. Note that due to limitations in R's graphics system, transparency (alpha) has to be disabled for gradient fills.

### Usage

```
geom_ridgeline_gradient(mapping = NULL, data = NULL, stat = "identity",
                        position = "identity", na.rm = FALSE, gradient_lwd = 0.5,
                        show.legend = NA, inherit.aes = TRUE, ...)
```

```
geom_density_ridges_gradient(mapping = NULL, data = NULL,
                             stat = "density_ridges", panel_scaling = TRUE, na.rm = TRUE,
                             gradient_lwd = 0.5, show.legend = NA, inherit.aes = TRUE, ...)
```

**Arguments**

mapping	Set of aesthetic mappings created by <code>aes()</code> or <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. 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.
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 <code>FALSE</code> , the default, missing values are removed with a warning. If <code>TRUE</code> , missing values are silently removed.
gradient_lwd	A parameter to needed to remove rendering artifacts inside the rendered gradients. Should ideally be 0, but often needs to be around 0.5 or higher.
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.
inherit.aes	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them.
...	other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>color = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.
panel_scaling	Argument only to <code>geom_density_ridges_gradient</code> . If <code>TRUE</code> , the default, relative scaling is calculated separately for each panel. If <code>FALSE</code> , relative scaling is calculated globally.

**Examples**

```
# Example for `geom_ridgeline_gradient()`
library(viridis)
d <- data.frame(x = rep(1:5, 3) + c(rep(0, 5), rep(0.3, 5), rep(0.6, 5)),
               y = c(rep(0, 5), rep(1, 5), rep(3, 5)),
               height = c(0, 1, 3, 4, 0, 1, 2, 3, 5, 4, 0, 5, 4, 4, 1))
ggplot(d, aes(x, y, height = height, group = y, fill = factor(x+y))) +
  geom_ridgeline_gradient() +
  scale_fill_viridis(discrete = TRUE, direction = -1) +
  theme(legend.position = 'none')

# Example for `geom_density_ridges_gradient()`
ggplot(lincoln_weather, aes(x = `Mean Temperature [F]`, y = `Month`, fill = ..x..)) +
  geom_density_ridges_gradient(scale = 3, rel_min_height = 0.01) +
  scale_x_continuous(expand = c(0.01, 0)) +
  scale_y_discrete(expand = c(0.01, 0)) +
  scale_fill_viridis(name = "Temp. [F]", option = "C") +
  labs(title = 'Temperatures in Lincoln NE in 2016') +
  theme_ridges(font_size = 13, grid = TRUE) + theme(axis.title.y = element_blank())
```

---

geom\_vridgeline      *Plot a vertical ridgeline (ridgeline rotated 90 degrees)*

---

### Description

Plots the sum of the `x` and `width` aesthetics versus `y`, filling the area between `x` and `x + width` with a color. Just like `geom_ridgeline()`, but with `y` and `x` replaced.

### Usage

```
geom_vridgeline(mapping = NULL, data = NULL, stat = "identity",
  position = "identity", na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE, ...)
```

### Arguments

<code>mapping</code>	Set of aesthetic mappings created by <code>aes()</code> or <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 <code>mapping</code> if there is no plot mapping.
<code>data</code>	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. 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.
<code>stat</code>	The statistical transformation to use on the data for this layer, as a string.
<code>position</code>	Position adjustment, either as a string, or the result of a call to a position adjustment function.
<code>na.rm</code>	If <code>FALSE</code> , the default, missing values are removed with a warning. If <code>TRUE</code> , missing values are silently removed.
<code>show.legend</code>	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.
<code>inherit.aes</code>	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them.
<code>...</code>	other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>color = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired <code>geom/stat</code> .

### Aesthetics

Required aesthetics are in bold.

- **x**
- **y**
- **width** Width of the ridgeline, measured from the respective `x` value. Assumed to be positive, though this is not required.



- `group` Defines the grouping. Required when the dataset contains multiple distinct ridgelines. Will typically be the same variable as is mapped to `x`.
- `scale` A scaling factor to scale the widths of the ridgelines. A value of 1 indicates that the widths are taken as is. This aesthetic can be used to convert width units into `x` units.
- `min_width` A width cutoff on the drawn ridgelines. All values that fall below this cutoff will be removed. The main purpose of this cutoff is to remove long tails right at the baseline level, but other uses are possible. The cutoff is applied before any width scaling is applied via the `scale` aesthetic. Default is 0, so negative values are removed.
- `color` Color of the ridgeline
- `fill` Fill color of the area under the ridgeline
- `alpha` Transparency level of color and fill
- `group` Grouping, to draw multiple ridgelines from one dataset
- `linetype` Linetype of the ridgeline
- `size` Line thickness

### Examples

```
d <- data.frame(y = rep(1:5, 3), x = c(rep(0, 5), rep(1, 5), rep(3, 5)),
              width = c(0, 1, 3, 4, 0, 1, 2, 3, 5, 4, 0, 5, 4, 4, 1))
ggplot(d, aes(x, y, width = width, group = x)) + geom_vridgeline(fill="lightblue")

ggplot(iris, aes(x=Species, y=Sepal.Width, width = ..density.., fill=Species)) +
  geom_vridgeline(stat="ydensity", trim=FALSE, alpha = 0.85, scale = 2)
```

---

`ggridges`

*ggridges.*

---

### Description

`ggridges.`

---

`lincoln_weather`

*Weather in Lincoln, Nebraska in 2016.*

---

### Description

A dataset containing weather information from Lincoln, Nebraska, from 2016. Originally downloaded from Weather Underground by Austin Wehrwein, <http://austinwehrwein.com/>. The variables are listed below. Most are self-explanatory. Max, mean, and min measurements are calculated relative to the specific day of measurement.

**Usage**

lincoln\_weather

**Format**

A tibble with 366 rows and 24 variables:

CST Day of the measurement

Max Temperature [F]

Mean Temperature [F]

Min Temperature [F]

Max Dew Point [F]

Mean Dew Point [F]

Min Dewpoint [F]

Max Humidity

Mean Humidity

Min Humidity

Max Sea Level Pressure [In]

Mean Sea Level Pressure [In]

Min Sea Level Pressure [In]

Max Visibility [Miles]

Mean Visibility [Miles]

Min Visibility [Miles]

Max Wind Speed [MPH]

Mean Wind Speed [MPH]

Max Gust Speed [MPH]

Precipitation [In]

CloudCover

Events Specific weather events, such as rain, snow, or fog

WindDir [Degrees]

Month The month in which the measurement was taken

---

reduce	<i>Reduce a list to a single value by iteratively applying a binary function</i>
--------	--

---

### Description

Inspired by `reduce()` from the `purrr` package

### Usage

```
reduce(.x, .f, ..., .init)
```

### Arguments

<code>.x</code>	A list or atomic vector.
<code>.f</code>	A 2-argument function. The function will be passed the accumulated value as the first argument and the "next" value as the second argument.
<code>...</code>	Additional arguments passed on to <code>.f</code> .
<code>.init</code>	If supplied, will be used as the first value to start the accumulation, rather than using <code>x[[1]]</code> . This is useful if you want to ensure that <code>reduce</code> returns a correct value when <code>.x</code> is empty. If missing, and <code>x</code> is empty, will throw an error.

### Author(s)

Jonathon Love [jon@thon.cc](mailto:jon@thon.cc)

---

<code>scale_cyclical</code>	<i>Create a discrete scale that cycles between values</i>
-----------------------------	---

---

### Description

The readability of ridgeline plots can often be improved by alternating between fill colors and other aesthetics. The various cyclical scales make it easy to create plots with this feature, simply map your grouping variable to the respective aesthetic (e.g., `fill`) and then use `scale_fill_cyclical` to define the fill colors between you want to alternate. Note that the cyclical scales do not draw legends by default, because the legends will usually be wrong unless the labels are properly adjusted. To draw legends, set the `guide` argument to "legend", as shown in the examples.

**Usage**

```
scale_colour_cyclical(..., values)

scale_fill_cyclical(..., values)

scale_alpha_cyclical(..., values)

scale_linetype_cyclical(..., values)

scale_size_cyclical(..., values)
```

**Arguments**

...	Common discrete scale parameters: name, breaks, labels, na.value, limits and guide. See <a href="#">discrete_scale</a> for more details.
values	The aesthetic values that the scale should cycle through, e.g. colors if it is a scale for the color or fill aesthetic.

**Examples**

```
# By default, scale_cyclical sets `guide = "none"`, i.e., no legend
# is drawn
ggplot(diamonds, aes(x = price, y = cut, fill = cut)) +
  geom_density_ridges(scale = 4) +
  scale_fill_cyclical(values = c("#3030D0", "#9090F0"))

# However, legends can be turned on by setting `guide = "legend"`
ggplot(diamonds, aes(x = price, y = cut, fill = cut)) +
  geom_density_ridges(scale = 4) +
  scale_fill_cyclical(values = c("#3030D0", "#9090F0"),
                    guide = "legend", name = "Fill colors",
                    labels = c("dark blue", "light blue"))

# Cyclical scales are also available for the various other aesthetics
ggplot(diamonds, aes(x = price, y = cut, fill = cut,
                    color = cut, size = cut,
                    alpha = cut, linetype = cut)) +
  geom_density_ridges(scale = 4, fill = "blue") +
  scale_fill_cyclical(values = c("blue", "green")) +
  scale_color_cyclical(values = c("black", "white")) +
  scale_size_cyclical(values = c(2, 1)) +
  scale_alpha_cyclical(values = c(0.4, 0.8)) +
  scale_linetype_cyclical(values = c(1, 2))
```

**Description**

Works like `stat_bin` except that the output is a ridgeline describing the histogram rather than a set of counts.

**Usage**

```
stat_binline(mapping = NULL, data = NULL, geom = "density_ridges",
  position = "identity", ..., binwidth = NULL, bins = NULL,
  center = NULL, boundary = NULL, breaks = NULL, closed = c("right",
  "left"), pad = TRUE, draw_baseline = TRUE, na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE)
```

**Arguments**

mapping	Set of aesthetic mappings created by <a href="#">aes</a> or <a href="#">aes_</a> . 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 <a href="#">ggplot</a> . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See <a href="#">fortify</a> 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.
geom	Use to override the default connection between <code>geom_histogram/geom_freqpoly</code> and <code>stat_bin</code> .
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
...	other arguments passed on to <a href="#">layer</a> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>color = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.
binwidth	The width of the bins. The default is to use bins that cover the range of the data. You should always override this value, exploring multiple widths to find the best to illustrate the stories in your data. The bin width of a date variable is the number of days in each time; the bin width of a time variable is the number of seconds.
bins	Number of bins. Overridden by <code>binwidth</code> . Defaults to 30
center	The center of one of the bins. Note that if <code>center</code> is above or below the range of the data, things will be shifted by an appropriate number of widths. To center on integers, for example, use <code>width = 1</code> and <code>center = 0</code> , even if <code>0</code> is outside the range of the data. At most one of <code>center</code> and <code>boundary</code> may be specified.
boundary	A boundary between two bins. As with <code>center</code> , things are shifted when <code>boundary</code> is outside the range of the data. For example, to center on integers, use <code>width = 1</code> and <code>boundary = 0.5</code> , even if <code>0.5</code> is outside the range of the data. At most one of <code>center</code> and <code>boundary</code> may be specified.

breaks	Alternatively, you can supply a numeric vector giving the bin boundaries. Overrides binwidth, bins, center, and boundary.
closed	One of "right" or "left" indicating whether right or left edges of bins are included in the bin.
pad	If TRUE, adds empty bins at either end of x. This ensures that the binline always goes back down to 0. Defaults to TRUE.
draw_baseline	If FALSE, removes lines along 0 counts. Defaults to TRUE.
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. <a href="#">borders</a> .

## Examples

```
ggplot(iris, aes(x = Sepal.Length, y = Species, group = Species, fill = Species)) +
  geom_density_ridges(stat = "binline", bins = 20, scale = 2.2) +
  scale_y_discrete(expand = c(0.01, 0)) +
  scale_x_continuous(expand = c(0.01, 0)) +
  theme_ridges()

ggplot(iris, aes(x = Sepal.Length, y = Species, group = Species, fill = Species)) +
  stat_binline(bins = 20, scale = 2.2, draw_baseline = FALSE) +
  scale_y_discrete(expand = c(0.01, 0)) +
  scale_x_continuous(expand = c(0.01, 0)) +
  scale_fill_grey() +
  theme_ridges() + theme(legend.position = 'none')

require(ggplot2movies)
require(viridis)
ggplot(movies[movies$year>1989,], aes(x = length, y = year, fill = factor(year))) +
  stat_binline(scale = 1.9, bins = 40) +
  theme_ridges() + theme(legend.position = "none") +
  scale_x_continuous(limits = c(1, 180), expand = c(0.01, 0)) +
  scale_y_reverse(expand = c(0.01, 0)) +
  scale_fill_viridis(begin = 0.3, discrete = TRUE, option = "B") +
  labs(title = "Movie lengths 1990 - 2005")

count_data <- data.frame(group = rep(letters[1:5], each = 10),
  mean = rep(1:5, each = 10))
count_data$group <- factor(count_data$group, levels = letters[5:1])
count_data$count <- rpois(nrow(count_data), count_data$mean)
ggplot(count_data, aes(x = count, y = group, group = group)) +
  geom_density_ridges2(stat = "binline", aes(fill = group), binwidth = 1, scale = 0.95) +
  geom_text(stat = "bin",
    aes(y = group+0.9*..count../max(..count..),
      label = ifelse(..count..>0, ..count.., "")),
    vjust = 1.2, size = 3, color = "white", binwidth = 1) +
```

```
theme_ridges(grid = FALSE) +
scale_x_continuous(breaks = c(0:12), limits = c(-.5, 13), expand = c(0, 0)) +
scale_y_discrete(expand = c(0.01, 0)) +
scale_fill_cyclical(values = c("#0000B0", "#7070D0")) +
guides(y = "none")
```

---

stat\_density\_ridges     *Stat for density ridgeline plots*

---

## Description

This stat is the default stat used by `geom_density_ridges`. It is very similar to `stat_density`, however there are a few differences. Most importantly, the density bandwidth is chosen across the entire dataset.

## Usage

```
stat_density_ridges(mapping = NULL, data = NULL, geom = "density_ridges",
  position = "identity", na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE, bandwidth = NULL, from = NULL, to = NULL,
  calc_ecdf = FALSE, quantiles = 5, ...)
```

## Arguments

mapping	Set of aesthetic mappings created by <code>aes()</code> or <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. 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.
geom	The geometric object to use to display the data.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
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.
inherit.aes	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them.
bandwidth	Bandwidth used for density calculation. If not provided, is estimated from the data.

from, to	The left and right-most points of the grid at which the density is to be estimated, as in <code>density()</code> . If not provided, there are estimated from the data range and the bandwidth.
calc_ecdf	If TRUE, <code>stat_density_ridges</code> calculates an empirical cumulative distribution function (ecdf) and returns a variable <code>ecdf</code> and a variable <code>quantile</code> . Both can be mapped onto aesthetics via <code>..ecdf..</code> and <code>..quantile..</code> , respectively.
quantiles	Sets the number of quantiles the data should be broken into if <code>calc_ecdf = TRUE</code> . If it is an integer then the data will be cut into that many equal quantiles. If it is a vector of probabilities then the ecdf will cut by them.
...	other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>color = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.

## Examples

```
# Examples of coloring by ecdf or quantiles
library(viridis)
ggplot(iris, aes(x=Sepal.Length, y=Species, fill=factor(..quantile..))) +
  geom_density_ridges_gradient(calc_ecdf = TRUE, quantiles = 5) +
  scale_fill_viridis(discrete = TRUE, name = "Quintiles") + theme_ridges() +
  scale_y_discrete(expand = c(0.01, 0))

ggplot(iris, aes(x=Sepal.Length, y=Species, fill=0.5 - abs(0.5-..ecdf..))) +
  geom_density_ridges_gradient(calc_ecdf = TRUE) +
  scale_fill_viridis(name = "Tail probability", direction = -1) + theme_ridges() +
  scale_y_discrete(expand = c(0.01, 0))

ggplot(iris, aes(x=Sepal.Length, y=Species, fill=factor(..quantile..))) +
  geom_density_ridges_gradient(calc_ecdf = TRUE, quantiles = c(0.05, 0.95)) +
  scale_fill_manual(name = "Probability\nranges",
                   values = c("red", "grey80", "blue")) +
  theme_ridges() + scale_y_discrete(expand = c(0.01, 0))
```

---

theme\_ridges

*A custom theme specifically for use with ridgeline plots*

---

## Description

This theme has some special modifications that make ridgeline plots look better, such as properly aligned y axis labels. It can draw plots with and without background grids (see examples).

## Usage

```
theme_ridges(font_size = 14, font_family = "", line_size = 0.5,
             grid = TRUE, center_axis_labels = FALSE)
```



**Arguments**

font_size	Overall font size. Default is 14.
font_family	Default font family.
line_size	Default line size.
grid	If TRUE (default), a background grid is drawn. If FALSE, background is left empty.
center_axis_labels	If TRUE, axis labels are drawn centered. If FALSE (default), axis labels are drawn right/top-aligned.

**Value**

The theme.

**Examples**

```
# Example with background grid
ggplot(iris, aes(x = Sepal.Length, y = Species, group = Species)) +
  geom_density_ridges(rel_min_height = 0.005) +
  scale_y_discrete(expand = c(0.01, 0)) +
  scale_x_continuous(expand = c(0.01, 0)) +
  theme_ridges()

# Example without background grid
ggplot(iris, aes(x = Sepal.Length, y = Species, group = Species)) +
  geom_density_ridges() +
  scale_y_discrete(expand = c(0.01, 0)) +
  scale_x_continuous(expand = c(0.01, 0)) +
  theme_ridges(grid = FALSE)
```

# Index

## \*Topic **datasets**

- Catalan\_elections, 2
  - geom\_density\_ridges, 2
  - geom\_ridgeline, 5
  - geom\_ridgeline\_gradient, 6
  - geom\_vridgeline, 8
  - lincoln\_weather, 9
  - scale\_cyclical, 11
  - stat\_binline, 12
  - stat\_density\_ridges, 15
- aes, 13
- aes(), 3, 5, 7, 8, 15
- aes\_, 13
- aes\_(), 3, 5, 7, 8, 15
- borders, 14
- Catalan\_elections, 2
- cyclical\_scale (scale\_cyclical), 11
- density(), 16
- discrete\_scale, 12
- fortify, 13
- geom\_density\_ridges, 2, 5, 6, 15
- geom\_density\_ridges2  
(geom\_density\_ridges), 2
- geom\_density\_ridges\_gradient  
(geom\_ridgeline\_gradient), 6
- geom\_ridgeline, 3, 4, 5, 6
- geom\_ridgeline(), 8
- geom\_ridgeline\_gradient, 6
- geom\_vridgeline, 8
- GeomDensityRidges  
(geom\_density\_ridges), 2
- GeomDensityRidges2  
(geom\_density\_ridges), 2
- GeomDensityRidgesGradient  
(geom\_ridgeline\_gradient), 6
- GeomRidgeline (geom\_ridgeline), 5
- GeomRidgelineGradient  
(geom\_ridgeline\_gradient), 6
- GeomVRidgeline (geom\_vridgeline), 8
- ggplot, 13
- ggplot(), 3, 5, 7, 8, 15
- ggridges, 9
- ggridges-package (ggridges), 9
- layer, 13
- layer(), 3, 5, 7, 8, 16
- lincoln\_weather, 9
- reduce, 11
- scale\_alpha\_cyclical (scale\_cyclical), 11
- scale\_color\_cyclical (scale\_cyclical), 11
- scale\_colour\_cyclical (scale\_cyclical), 11
- scale\_cyclical, 11
- scale\_fill\_cyclical (scale\_cyclical), 11
- scale\_linetype\_cyclical  
(scale\_cyclical), 11
- scale\_size\_cyclical (scale\_cyclical), 11
- ScaleCyclical (scale\_cyclical), 11
- stat\_binline, 12
- stat\_density, 3, 15
- stat\_density\_ridges, 3, 4, 15
- StatBinline (stat\_binline), 12
- StatDensityRidges  
(stat\_density\_ridges), 15
- theme\_ridges, 16