# Package 'ggstance'

August 19, 2019

**Title** Horizontal 'ggplot2' Components

**Version** 0.3.3

**Description** A 'ggplot2' extension that provides flipped components: horizontal versions of 'Stats' and 'Geoms', and vertical versions of 'Positions'.

**Depends** R (>= 3.1.0)

**Imports** ggplot2 (>= 3.2.0), plyr, rlang, withr (>= 2.0.0)

**Suggests** Hmisc, testthat, vdiffr (>= 0.3.0)

**License** GPL-3

**LazyData** true

**Encoding** UTF-8

**RoxygenNote** 6.1.1

**Collate** 'flip-aes.R' 'geom-barh.R' 'legend-draw.R' 'geom-boxploth.R' 'geom-colh.R' 'geom-crossbarh.R' 'geom-errorbarh.R' 'geom-histogramh.R' 'geom-linerangeh.R' 'geom-pointrangeh.R' 'geom-violinh.R' 'ggstance.R' 'position-dodgev.R' 'position-dodge2v.R' 'position-jitterdodgev.R' 'position-stackv.R' 'position.R' 'stat-binh.R' 'stat-boxploth.R' 'stat-counth.R' 'stat-summaryh.R' 'stat-xdensity.R'

**NeedsCompilation** no

**Author** Lionel Henry [aut, cre],
Hadley Wickham [aut],
Winston Chang [aut],
RStudio [cph]

**Maintainer** Lionel Henry <lionel@rstudio.com>

**Repository** CRAN

**Date/Publication** 2019-08-19 13:30:03 UTC

# R topics documented:



---

`draw_key` *Horizontal key drawing functions*

---

## Description

Horizontal key drawing functions

## Usage

```
draw_key_hpath(data, params, size)

draw_key_pointrangeh(data, params, size)

draw_key_crossbarh(data, params, size)

draw_key_boxploth(data, params, size)
```

## Arguments

| | |
|---|---|
| `data` | A single row data frame containing the scaled aesthetics to display in this key |
| `params` | A list of additional parameters supplied to the geom. |
| `size` | Width and height of key in mm. |

## Value

A grid grob.

geom_barh *Bars, rectangles with bases on y-axis*

## Description

Horizontal version of `geom_bar()`.

## Usage

```
geom_barh(mapping = NULL, data = NULL, stat = "counth",
  position = "stackv", ..., width = NULL, binwidth = NULL,
  na.rm = FALSE, show.legend = NA, inherit.aes = TRUE)

geom_colh(mapping = NULL, data = NULL, position = "stackv", ...,
  width = NULL, na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE)
```

## 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:<br>If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`.<br>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.<br>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` | Override the default connection between `geom_bar()` and `stat_count()`. |
| `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. |
| `width` | Bar width. By default, set to 90% of the resolution of the data. |
| `binwidth` | `geom_bar()` no longer has a binwidth argument - if you use it you'll get an warning telling to you use `geom_histogram()` instead. |
| `na.rm` | If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed. |
| `show.legend` | logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display. |

inherit.aes — If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders().

## Aesthetics

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

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

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

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

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

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

---

# geom_boxploth *Horizontal box and whiskers plot.*

---

## Description

Horizontal version of geom_boxplot().

## Usage

```
geom_boxploth(mapping = NULL, data = NULL, stat = "boxploth",
  position = "dodge2v", ..., outlier.colour = NULL,
  outlier.color = NULL, outlier.fill = NULL, outlier.shape = 19,
  outlier.size = 1.5, outlier.stroke = 0.5, outlier.alpha = NULL,
  notch = FALSE, notchwidth = 0.5, varwidth = FALSE, na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE)
```

## 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:<br>If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot().<br>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.<br>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. |
| ... | 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.colour, outlier.color, outlier.shape, outlier.size, outlier.stroke | Default aesthetics for outliers. Set to NULL to inherit from the aesthetics used for the box.<br>In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence. |
| outlier.fill | Default aesthetics for outliers. Set to NULL to inherit from the aesthetics used for the box.<br>In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.<br>Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting outlier.shape = NA. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden. |
| outlier.alpha | Default aesthetics for outliers. Set to NULL to inherit from the aesthetics used for the box.<br>In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.<br>Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting outlier.shape = NA. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden. |
| notch | If FALSE (default) make a standard box plot. If TRUE, make a notched box plot. Notches are used to compare groups; if the notches of two boxes do not overlap, this suggests that the medians are significantly different. |

| | |
|---|---|
| notchwidth | For a notched box plot, width of the notch relative to the body (defaults to notchwidth = 0.5). |
| varwidth | If FALSE (default) make a standard box plot. If TRUE, boxes are drawn with widths proportional to the square-roots of the number of observations in the groups (possibly weighted, using the weight aesthetic). |
| na.rm | If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed. |
| show.legend | logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. |
| inherit.aes | If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders(). |

## Aesthetics

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

- y
- xlower
- xupper
- xmiddle
- xmin
- xmax
- alpha
- colour
- fill
- group
- linetype
- shape
- size
- weight

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

## Examples

```
library("ggplot2")

# With ggplot2 we need coord_flip():
ggplot(mpg, aes(class, hwy, fill = factor(cyl))) +
  geom_boxplot() +
  coord_flip()

# With ggstance we use the h-suffixed version:
```

```
ggplot(mpg, aes(hwy, class, fill = factor(cyl))) +
  geom_boxploth()

# With facets ggstance horizontal layers are often the only way of
# having all ggplot features working correctly, for instance free
# scales:
df <- data.frame(
  Group = factor(rep(1:3, each = 4), labels = c("Drug A", "Drug B", "Control")),
  Subject = factor(rep(1:6, each = 2), labels = c("A", "B", "C", "D", "E", "F")),
  Result = rnorm(12)
)

ggplot(df, aes(Result, Subject))+
  geom_boxploth(aes(fill = Group))+
  facet_grid(Group ~ ., scales = "free_y")
```

---

geom_crossbarh *Horizontal intervals: lines, crossbars & errorbars.*

---

## Description

Horizontal versions of geom_linerange(), geom_pointrange(), geom_errorbar() and geom_crossbar().

## Usage

```
geom_crossbarh(mapping = NULL, data = NULL, stat = "identity",
  position = "identity", ..., fatten = 2.5, na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE)

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

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

geom_pointrangeh(mapping = NULL, data = NULL, stat = "identity",
  position = "identity", ..., fatten = 4, na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE)
```

## 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:<br>If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot().<br>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.<br>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. |
| fatten | A multiplicative factor used to increase the size of the middle bar in geom_crossbar() and the middle point in geom_pointrange(). |
| na.rm | If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed. |
| show.legend | logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. |
| inherit.aes | If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders(). |

## Aesthetics

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

- x
- y
- xmin
- xmax
- alpha
- colour
- fill
- group
- linetype
- size

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

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

- y
- xmin
- xmax
- alpha
- colour
- group
- linetype
- size
- width

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

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

- y
- xmin
- xmax
- alpha
- colour
- group
- linetype
- size

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

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

- x
- y
- xmin
- xmax
- alpha
- colour
- fill
- group
- linetype
- shape
- size
- stroke

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

**Different between ggplot2 and ggstance**

‘ggplot2::geom_errorbarh()‘ uses the ‘height‘ aesthetic. The ggstance version uses the ‘width‘ aesthetic. This is for consistency with the direction of the geom and other ggstance functions. You can still supply ‘height‘ for compatibility.

geom_histogramh *Horizontal histograms and frequency polygons.*

## Description

Horizontal version of geom_histogram().

## Usage

```
geom_histogramh(mapping = NULL, data = NULL, stat = "binh",
  position = "stackv", ..., binwidth = NULL, bins = NULL,
  na.rm = FALSE, show.legend = NA, inherit.aes = TRUE)
```

## 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:<br>If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot().<br>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.<br>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_histogram()/geom_freqpoly() and stat_bin(). |
| 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. |
| binwidth | The width of the bins. Can be specified as a numeric value or as a function that calculates width from unscaled x. Here, "unscaled x" refers to the original x values in the data, before application of any scale transformation. When specifying a function along with a grouping structure, the function will be called once per group. The default is to use bins 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.<br>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 binwidth. Defaults to 30. |

| | |
|---|---|
| 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(). |

---

geom_violinh *Horizontal violin plot.*

---

## Description

Horizontal version of geom_violin().

## Usage

```
geom_violinh(mapping = NULL, data = NULL, stat = "xdensity",
  position = "dodgev", ..., draw_quantiles = NULL, trim = TRUE,
  scale = "area", na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE)
```

## 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:<br>If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot().<br>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.<br>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_violin and stat_ydensity. |
| 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. |
| draw_quantiles | If not(NULL) (default), draw horizontal lines at the given quantiles of the density estimate. |

| | |
|---|---|
| trim | If TRUE (default), trim the tails of the violins to the range of the data. If FALSE, don't trim the tails. |
| scale | if "area" (default), all violins have the same area (before trimming the tails). If "count", areas are scaled proportionally to the number of observations. If "width", all violins have the same maximum width. |
| na.rm | If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed. |
| show.legend | logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. |
| inherit.aes | If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders(). |

## Aesthetics

geom_violinh() 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").

---

# hmisc_h *Horizontal versions of summary functions from Hmisc*

---

## Description

These are horizontal versions of the wrappers around functions from **Hmisc** designed to make them easier to use with stat_summaryh. The corresponding vertical versions are hmisc(). See the Hmisc documentation for more details:

- smean.cl.boot
- smean.cl.normal
- smean.sdl
- smedian.hilow

## Usage

```
mean_cl_boot_h(x, ...)

mean_cl_normal_h(x, ...)

mean_sdl_h(x, ...)

median_hilow_h(x, ...)
```

## Arguments

| | |
|---|---|
| x | a numeric vector |
| ... | other arguments passed on to the respective Hmisc function. |

## Value

A data frame with columns x, xmin, and xmax.

## Examples

```
x <- rnorm(100)
mean_cl_boot_h(x)
mean_cl_normal_h(x)
mean_sdl_h(x)
median_hilow_h(x)
```

---

# mean_se_h *Calculate mean and standard error*

---

## Description

For use with stat_summaryh. Corresponding function for vertical geoms is mean_se()

## Usage

```
mean_se_h(x, mult = 1)
```

## Arguments

| | |
|---|---|
| x | numeric vector |
| mult | number of multiples of standard error |

## Value

A data frame with columns x, xmin, and xmax.

## Examples

```
x <- rnorm(100)
mean_se_h(x)
```

---

`position_dodgev` *Vertical Positions*

---

## Description

Vertical versions of `position_dodge()`, `position_jitterdodge()`, `position_fill()`, `position_stack()`,

## Usage

```
position_dodgev(height = NULL, preserve = c("total", "single"))

position_dodge2v(height = NULL, preserve = c("single", "total"),
  padding = 0.1, reverse = TRUE)

position_jitterdodgev(jitter.height = NULL, jitter.width = 0,
  dodge.height = 0.75, seed = NA)

position_stackv(hjust = 1, reverse = FALSE)

position_fillv()
```

## Arguments

| | |
|---|---|
| `height` | Dodging height, when different to the height of the individual elements. This is useful when you want to align narrow geoms with taller geoms. |
| `preserve` | Should dodging preserve the total width of all elements at a position, or the width of a single element? |
| `padding` | Padding between elements at the same position. Elements are shrunk by this proportion to allow space between them. Defaults to 0.1. |
| `reverse` | If `TRUE`, will reverse the default stacking order. This is useful if you're rotating both the plot and legend. |
| `jitter.height` | degree of jitter in y direction. Defaults to 0. |
| `jitter.width` | degree of jitter in x direction. Defaults to 40% of the resolution of the data. |
| `dodge.height` | the amount to dodge in the y direction. Defaults to 0.75, the default `position_dodgev()` height. |
| `seed` | A random seed to make the jitter reproducible. Useful if you need to apply the same jitter twice, e.g., for a point and a corresponding label. The random seed is reset after jittering. If `NA` (the default value), the seed is initialised with a random value; this makes sure that two subsequent calls start with a different seed. Use `NULL` to use the current random seed and also avoid resetting (the behaviour of **ggplot** 2.2.1 and earlier). |

| | |
|---|---|
| hjust | Horizontal adjustment for geoms that have a position (like points or lines), not a dimension (like bars or areas). Set to ‘0‘ to align with the left side, ‘0.5‘ for the middle, and ‘1‘ (the default) for the right side. |

---

stat_binh *Horizontal binning.*

---

## Description

Horizontal version of stat_bin().

## Usage

```
stat_binh(mapping = NULL, data = NULL, geom = "barh",
  position = "stackv", ..., binwidth = NULL, bins = NULL,
  center = NULL, boundary = NULL, closed = c("right", "left"),
  pad = FALSE, na.rm = FALSE, show.legend = NA, inherit.aes = TRUE)
```

## 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:<br>If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot().<br>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.<br>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 | Use to override the default connection between geom_histogram()/geom_freqpoly() and stat_bin(). |
| 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. |
| binwidth | The width of the bins. Can be specified as a numeric value or as a function that calculates width from unscaled x. Here, "unscaled x" refers to the original x values in the data, before application of any scale transformation. When specifying a function along with a grouping structure, the function will be called once per group. The default is to use bins 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 `binwidth`. Defaults to 30.

`center` bin position specifiers. Only one, `center` or `boundary`, may be specified for a single plot. `center` specifies the center of one of the bins. `boundary` specifies the boundary between two bins. Note that if either is above or below the range of the data, things will be shifted by the appropriate integer multiple of `width`. For example, to center on integers use `width = 1` and `center = 0`, even if `0` is outside the range of the data. Alternatively, this same alignment can be specified with `width = 1` and `boundary = 0.5`, even if `0.5` is outside the range of the data.

`boundary` bin position specifiers. Only one, `center` or `boundary`, may be specified for a single plot. `center` specifies the center of one of the bins. `boundary` specifies the boundary between two bins. Note that if either is above or below the range of the data, things will be shifted by the appropriate integer multiple of `width`. For example, to center on integers use `width = 1` and `center = 0`, even if `0` is outside the range of the data. Alternatively, this same alignment can be specified with `width = 1` and `boundary = 0.5`, even if `0.5` is outside the range of the data.

`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 frequency polygons touch 0. Defaults to `FALSE`.

`na.rm` If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

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

`inherit.aes` If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `borders()`.

## Aesthetics

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

- `y`
- `group`
- `x`

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

stat_boxploth *Horizontal boxplot computation.*

## Description

Horizontal version of stat_boxplot().

## Usage

```
stat_boxploth(mapping = NULL, data = NULL, geom = "boxploth",
  position = "dodge2v", ..., coef = 1.5, na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE)
```

## 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:<br>If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot().<br>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.<br>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 | 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. |
| ... | 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. |
| coef | Length of the whiskers as multiple of IQR. Defaults to 1.5. |
| na.rm | If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed. |
| show.legend | logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. |
| inherit.aes | If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders(). |

### Aesthetics

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

- x
- y
- group

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

---

stat_counth *Horizontal counting.*

---

### Description

Horizontal version of stat_count().

### Usage

```
stat_counth(mapping = NULL, data = NULL, geom = "barh",
  position = "stackv", ..., width = NULL, na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE)
```

### 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:<br>If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot().<br>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.<br>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 | Override the default connection between geom_bar() and stat_count(). |
| 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. |
| width | Bar width. By default, set to 90% of the resolution of the data. |
| na.rm | If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed. |

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

inherit.aes — If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders().

## Aesthetics

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

- y
- group
- weight
- x

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

---

# stat_summaryh *Horizontal summary.*

---

## Description

Horizontal version of stat_summary().

## Usage

```
stat_summaryh(mapping = NULL, data = NULL, geom = "pointrangeh",
  position = "identity", ..., fun.data = NULL, fun.x = NULL,
  fun.xmax = NULL, fun.xmin = NULL, fun.args = list(),
  na.rm = FALSE, show.legend = NA, inherit.aes = TRUE)
```

## 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 | Use to override the default connection between geom_histogram()/geom_freqpoly() and stat_bin(). |
| 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. |
| fun.data | A function that is given the complete data and should return a data frame with variables xmin, x, and xmax. |
| fun.xmin, fun.x, fun.xmax | Alternatively, supply three individual functions that are each passed a vector of x's and should return a single number. |
| fun.args | Optional additional arguments passed on to the functions. |
| na.rm | If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed. |
| show.legend | logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. |
| inherit.aes | If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders(). |

## Aesthetics

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

- x
- y
- group

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

---

# stat_xdensity *Density computation on x axis.*

---

## Description

Horizontal version of stat_ydensity().

## Usage

```
stat_xdensity(mapping = NULL, data = NULL, geom = "violinh",
  position = "dodgev", ..., bw = "nrd0", adjust = 1,
  kernel = "gaussian", trim = TRUE, scale = "area", na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE)
```

## 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:<br>If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot().<br>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.<br>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 | Use to override the default connection between geom_violin and stat_ydensity. |
| 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. |
| bw | The smoothing bandwidth to be used. If numeric, the standard deviation of the smoothing kernel. If character, a rule to choose the bandwidth, as listed in stats::bw.nrd(). |
| adjust | A multiplicate bandwidth adjustment. This makes it possible to adjust the bandwidth while still using the a bandwidth estimator. For example, adjust = 1/2 means use half of the default bandwidth. |
| kernel | Kernel. See list of available kernels in density(). |
| trim | If TRUE (default), trim the tails of the violins to the range of the data. If FALSE, don't trim the tails. |
| scale | if "area" (default), all violins have the same area (before trimming the tails). If "count", areas are scaled proportionally to the number of observations. If "width", all violins have the same maximum width. |
| na.rm | If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed. |
| show.legend | logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. |
| inherit.aes | If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders(). |

## Aesthetics

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

- x

- y
- group

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

# Index