

Package ‘ggstance’

July 20, 2018

Title Horizontal 'ggplot2' Components

Version 0.3.1

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 (>= 3.0.0), plyr, rlang, withr (>= 2.0.0)

Suggests Hmisc, testthat, vdiff

License GPL-3

LazyData true

RoxygenNote 6.0.1

Collate 'flip-aes.R' 'geom-barh.R' 'legend-draw.R' 'geom-boxplot.R'
'geom-colh.R' 'geom-crossbarh.R' 'geom-errorbarh.R'
'geom-histogramh.R' 'geom-linrangeh.R' 'geom-pointrangeh.R'
'geom-violinh.R' 'ggstance.R' 'position-dodgev.R'
'position-dodge2v.R' 'position-jitterdodgev.R'
'position-stackv.R' 'position.R' 'stat-bin.R'
'stat-boxplot.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 2018-07-20 10:10:03 UTC

R topics documented:

draw_key 2

geom_barh	3
geom_boxploth	4
geom_crossbarh	6
geom_histogramh	7
geom_violinh	8
hmisc_h	9
mean_se_h	10
position_dodgev	11
stat_binh	12
stat_boxploth	13
stat_counth	14
stat_summaryh	15
stat_xdensity	16

Index	18
--------------	-----------

draw_key	<i>Horizontal key drawing functions</i>
----------	---

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 = "count",
  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 <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. 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.
stat	Override the default connection between <code>geom_bar</code> and <code>stat_count</code> .
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
...	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> .
width	Bar width. By default, set to 90% of the resolution of the data.
binwidth	<code>geom_bar</code> no longer has a <code>binwidth</code> argument - if you use it you'll get a warning telling to you use <code>geom_histogram()</code> instead.
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 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_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:
 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.

`stat` Use to override the default connection between `geom_boxploth` and `stat_boxploth`.

`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 `color = "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.
 In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.

<code>outlier.fill</code>	<p>Default aesthetics for outliers. Set to NULL to inherit from the aesthetics used for the box.</p> <p>In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.</p> <p>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 <code>outlier.shape = NA</code>. 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.</p>
<code>outlier.alpha</code>	<p>Default aesthetics for outliers. Set to NULL to inherit from the aesthetics used for the box.</p> <p>In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.</p> <p>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 <code>outlier.shape = NA</code>. 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.</p>
<code>notch</code>	<p>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.</p>
<code>notchwidth</code>	<p>For a notched box plot, width of the notch relative to the body (default 0.5)</p>
<code>varwidth</code>	<p>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 <code>weight</code> aesthetic).</p>
<code>na.rm</code>	<p>If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.</p>
<code>show.legend</code>	<p>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.</p>
<code>inherit.aes</code>	<p>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. <code>borders()</code>.</p>

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_boxplot(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 <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 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.

<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>...</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> .
<code>fatten</code>	A multiplicative factor used to increase the size of the middle bar in <code>geom_crossbar()</code> and the middle point in <code>geom_pointrange()</code> .
<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. It can also be a named logical vector to finely select the aesthetics to display.
<code>inherit.aes</code>	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> .

 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

<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 mapping 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 `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.

<code>stat</code>	Use to override the default connection between <code>geom_histogram/geom_freqpoly</code> and <code>stat_bin</code> .
<code>position</code>	Position adjustment, either as a string, or the result of a call to a position adjustment function.
<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> .
<code>binwidth</code>	The width of the bins. Can be specified as a numeric value, or a function that calculates width from <code>x</code> . 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.
<code>bins</code>	Number of bins. Overridden by <code>binwidth</code> . Defaults to 30.
<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. It can also be a named logical vector to finely select the aesthetics to display.
<code>inherit.aes</code>	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> .

<code>geom_violinh</code>	<i>Horizontal violin plot.</i>
---------------------------	--------------------------------

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 <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. 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.
stat	Use to override the default connection between <code>geom_violin</code> and <code>stat_ydensity</code> .
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
...	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> .
draw_quantiles	If not (<code>NULL</code>) (default), draw horizontal lines at the given quantiles of the density estimate.
trim	If <code>TRUE</code> (default), trim the tails of the violins to the range of the data. If <code>FALSE</code> , 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 <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> .

 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	<i>Vertical Positions</i>
-----------------	---------------------------

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 = FALSE)

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

position_stackv(hjust = 1, reverse = FALSE)

position_fillv()
```

Arguments

<code>height</code>	Dodging height, when different to the height of the individual elements. This is useful when you want to align narrow geoms with taller geoms.
<code>preserve</code>	Should dodging preserve the total width of all elements at a position, or the width of a single element?
<code>padding</code>	Padding between elements at the same position. Elements are shrunk by this proportion to allow space between them. Defaults to 0.1.
<code>reverse</code>	If TRUE, will reverse the default stacking order. This is useful if you're rotating both the plot and legend.
<code>jitter.height</code>	degree of jitter in y direction. Defaults to 0.
<code>jitter.width</code>	degree of jitter in x direction. Defaults to 40% of the resolution of the data.
<code>dodge.height</code>	the amount to dodge in the y direction. Defaults to 0.75, the default <code>position_dodgev()</code> 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 behavior 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 <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 NULL, the default, the data is inherited from the plot data as specified in the call to ggplot() . A <code>data.frame</code> , 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 <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 layer() . 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. Can be specified as a numeric value, or a function that calculates width from x. 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	The center of one of the bins. Note that if center 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 width = 1 and center = 0, even if 0 is outside the range of the data. At most one of center and boundary may be specified.
boundary	A boundary between two bins. As with center, things are shifted when boundary is outside the range of the data. For example, to center on integers, use width = 1 and boundary = 0.5, even if 0.5 is outside the range of the data. At most one of center and boundary may be specified.
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. <code>borders()</code> .

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 <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. 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.
geom	Use to override the default connection between <code>geom_boxplot</code> and <code>stat_boxplot</code> .
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
...	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> .
coef	Length of the whiskers as multiple of IQR. Defaults to 1.5.
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> .

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 <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. 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.
geom	Override the default connection between <code>geom_bar</code> and <code>stat_count</code> .
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
...	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.
width	Bar width. By default, set to 90% of the resolution of the data.
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> .

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 <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. 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.
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 <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.
fun.data	A function that is given the complete data and should return a data frame with variables <code>xmin</code> , <code>x</code> , and <code>xmax</code> .
fun.xmin, fun.x, fun.xmax	Alternatively, supply three individual functions that are each passed a vector of <code>x</code> 's and should return a single number.
fun.args	Optional additional arguments passed on to the functions.
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> .

stat_xdensity

Density computation on x axis.

Description

Horizontal version of `stat_ydensity()`.

Usage

```
stat_xdensity(mapping = NULL, data = NULL, geom = "violin",
  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 <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. 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.
geom	Use to override the default connection between <code>geom_violin</code> and <code>stat_ydensity</code> .
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
...	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.
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 <code>stats::bw.nrd()</code> .
adjust	A multiplicate bandwidth adjustment. This makes it possible to adjust the bandwidth while still using the a bandwidth estimator. For example, <code>adjust = 1/2</code> means use half of the default bandwidth.
kernel	Kernel. See list of available kernels in <code>density()</code> .
trim	If <code>TRUE</code> (default), trim the tails of the violins to the range of the data. If <code>FALSE</code> , 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 <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> .

Index

`aes()`, [3](#), [4](#), [6](#), [7](#), [9](#), [12](#), [14–17](#)
`aes_()`, [3](#), [4](#), [6](#), [7](#), [9](#), [12](#), [14–17](#)

`borders()`, [4](#), [5](#), [7–9](#), [13–17](#)

`density()`, [17](#)
`draw_key`, [2](#)
`draw_key_boxploth` (`draw_key`), [2](#)
`draw_key_crossbarh` (`draw_key`), [2](#)
`draw_key_hpath` (`draw_key`), [2](#)
`draw_key_pointrangeh` (`draw_key`), [2](#)

`fortify()`, [3](#), [4](#), [7–9](#), [12](#), [14–17](#)

`geom_bar`, [3](#)
`geom_barh`, [3](#)
`geom_boxplot`, [4](#)
`geom_boxploth`, [4](#)
`geom_colh` (`geom_barh`), [3](#)
`geom_crossbar`, [6](#)
`geom_crossbarh`, [6](#)
`geom_errorbar`, [6](#)
`geom_errorbarh` (`geom_crossbarh`), [6](#)
`geom_histogram`, [7](#)
`geom_histogram()`, [3](#)
`geom_histogramh`, [7](#)
`geom_linerange`, [6](#)
`geom_linerangeh` (`geom_crossbarh`), [6](#)
`geom_pointrange`, [6](#)
`geom_pointrangeh` (`geom_crossbarh`), [6](#)
`geom_violin`, [8](#)
`geom_violinh`, [8](#)
`ggplot()`, [3](#), [4](#), [6](#), [7](#), [9](#), [12](#), [14–17](#)

`hmisc`, [9](#)
`hmisc_h`, [9](#)

`layer()`, [3](#), [4](#), [7–9](#), [12](#), [14–17](#)

`mean_cl_boot_h` (`hmisc_h`), [9](#)
`mean_cl_normal_h` (`hmisc_h`), [9](#)

`mean_sdl_h` (`hmisc_h`), [9](#)
`mean_se`, [10](#)
`mean_se_h`, [10](#)
`median_hilow_h` (`hmisc_h`), [9](#)

`position-vertical` (`position_dodgev`), [11](#)
`position_dodge`, [11](#)
`position_dodge2v` (`position_dodgev`), [11](#)
`position_dodgev`, [11](#)
`position_fill`, [11](#)
`position_fillv` (`position_dodgev`), [11](#)
`position_jitterdodge`, [11](#)
`position_jitterdodgev`
 (`position_dodgev`), [11](#)
`position_stack`, [11](#)
`position_stackv` (`position_dodgev`), [11](#)

`smean.cl.boot`, [10](#)
`smean.cl.normal`, [10](#)
`smean.sdl`, [10](#)
`smedian.hilow`, [10](#)
`stat_bin`, [12](#)
`stat_binh`, [12](#)
`stat_boxplot`, [13](#)
`stat_boxploth`, [13](#)
`stat_count`, [14](#)
`stat_counth`, [14](#)
`stat_summary`, [15](#)
`stat_summaryh`, [9](#), [10](#), [15](#)
`stat_xdensity`, [16](#)
`stat_ydensity`, [16](#)
`stats::bw.nrd()`, [17](#)