Package 'ggsignif'

August 8, 2019

Type Package

Title Significance Brackets for 'ggplot2'

Version 0.6.0

Description Enrich your 'ggplots' with group-wise comparisons.
This package provides an easy way to indicate if two groups are significantly different.
Commonly this is shown by a bracket on top connecting the groups of interest which itself is annotated with the level of significance (NS, *, **, ***).
The package provides a single layer (geom_signif()) that takes the groups for comparison and the test (t.test(), wilcox.text() etc.) as arguments and adds the annotation
to the plot.

URL https://github.com/const-ae/ggsignif

License GPL-3

Encoding UTF-8

LazyData true

Imports ggplot2 (>= 2.0.0)

Suggests testthat, knitr, rmarkdown

RoxygenNote 6.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Constantin Ahlmann-Eltze [aut, cre]

Maintainer Constantin Ahlmann-Eltze <artjom31415@googlemail.com>

Repository CRAN

Date/Publication 2019-08-08 12:40:02 UTC

R topics documented:

stat_signif *Create significance layer*

Description

Create significance layer

Usage

```
stat_signif(mapping = NULL, data = NULL, position = "identity",
  na.rm = FALSE, show.legend = NA, inherit.aes = TRUE,
  comparisons = NULL, test = "wilcox.test", test.args = NULL,
  annotations = NULL, map_signif_level = FALSE, y_position = NULL,
  xmin = NULL, xmax = NULL, margin_top = 0.05, step_increase = 0,
  tip_length = 0.03, size = 0.5, textsize = 3.88, family = "",
  vjust = 0, parse = FALSE, manual = FALSE, ...)

geom_signif(mapping = NULL, data = NULL, stat = "signif",
  position = "identity", na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE, comparisons = NULL, test = "wilcox.test",
  test.args = NULL, annotations = NULL, map_signif_level = FALSE,
  y_position = NULL, xmin = NULL, xmax = NULL, margin_top = 0.05,
  step_increase = 0, tip_length = 0.03, size = 0.5,
  textsize = 3.88, family = "", vjust = 0, parse = FALSE,
  manual = FALSE, ...)
```

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)`).

`position` Position adjustment, either as a string, or the result of a call to a position adjustment function.

`na.rm` If `FALSE` (the default), removes missing values with a warning. If `TRUE` silently removes missing values.

`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()`.

`comparisons` A list of length-2 vectors. The entries in the vector are either the names of 2 values on the x-axis or the 2 integers that correspond to the index of the columns of interest

`test` the name of the statistical test that is applied to the values of the 2 columns (e.g. ‘t.test‘, ‘wilcox.test‘ etc.). If you implement a custom test make sure that it returns a list that has an entry called ‘p.value‘.

`test.args` additional arguments for the test method

`annotations` character vector with alternative annotations, if not null test is ignored

`map_signif_level`

boolean value, if the p-value are directly written as annotation or asterisks are used instead. Alternatively one can provide a named numeric vector to create custom mappings from p-values to annotation: For example: c("***"=0.001, "**"=0.01, "*"=0.05) Alternatively, one can provide a function that takes a numeric argument (the p-value) and returns a string

`y_position` numeric vector with the y positions of the brackets

`xmin` numeric vector with the positions of the left sides of the brackets

`xmax` numeric vector with the positions of the right sides of the brackets

`margin_top` numeric vector how much higher that the maximum value that bars start as fraction of total height

`step_increase` numeric vector with the increase in fraction of total height for every additional comparison to minimize overlap.

`tip_length` numeric vector with the fraction of total height that the bar goes down to indicate the precise column

`size` change the width of the lines of the bracket

`textsize` change the size of the text

`family` change the font used for the text

`vjust` move the text up or down relative to the bracket

`parse` If ‘TRUE‘, the labels will be parsed into expressions and displayed as described in ‘?plotmath‘.

`manual` boolean flag that indicates that the parameters are provided with a data.frame. This option is necessary if one wants to plot different annotations per facet.

`...` 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.

`stat` The statistical transformation to use on the data for this layer, as a string.

Examples

```
## Not run:
library(ggplot2)
library(ggsignif)
ggplot(mpg, aes(class, hwy)) +
 geom_boxplot() +
 geom_signif(comparisons = list(c("compact", "pickup"),
                                c("subcompact", "suv")))
ggplot(mpg, aes(class, hwy)) +
 geom_boxplot() +
 geom_signif(comparisons = list(c("compact", "pickup"),
                                c("subcompact", "suv")),
             map_signif_level=function(p)sprintf("p = %.2g", p))

ggplot(mpg, aes(class, hwy)) +
  geom_boxplot() +
  geom_signif(annotations = c("First", "Second"),
              y_position = c(30, 40), xmin=c(4,1), xmax=c(5,3))

## End(Not run)
```

Index