

# Package ‘ggcompare’

July 22, 2025

**Title** Mean Comparison in 'ggplot2'

**Version** 0.0.3

**Description** Add mean comparison annotations to a 'ggplot'.

This package provides an easy way to indicate if two or more groups are significantly different in a 'ggplot'.

Usually you do not need to specify the test method, you only need to tell `stat_compare()` whether you want to perform a parametric test or a nonparametric test, and `stat_compare()` will automatically choose the appropriate test method based on your data.

For comparisons between two groups, the p-value is calculated by t-test (parametric) or Wilcoxon rank sum test (nonparametric). For comparisons among more than two groups, the p-value is calculated by One-way ANOVA (parametric) or Kruskal-Wallis test (nonparametric).

**License** Artistic-2.0

**URL** <https://hmu-wh.github.io/ggcompare/>,  
<https://github.com/HMU-WH/ggcompare/>

**BugReports** <https://github.com/HMU-WH/ggcompare/issues/>

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Depends** R (>= 4.1)

**Imports** ggplot2

**NeedsCompilation** no

**Author** Hao Wang [aut, cre] (ORCID: <<https://orcid.org/0009-0000-9477-9585>>)

**Maintainer** Hao Wang <wanghao8772@gmail.com>

**Repository** CRAN

**Date/Publication** 2024-12-09 06:00:01 UTC

## Contents

<code>geom_bracket</code>	2
<code>stat_compare</code>	4

## Index

8

---

geom_bracket	<i>Add Brackets with Labels to a ggplot</i>
--------------	---

---

**Description**

Add brackets with labeled annotations to a ggplot.

**Usage**

```
geom_bracket(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  arrow = NULL,
  parse = FALSE,
  bracket = TRUE,
  inherit.aes = TRUE
)
```

**Arguments**

- |         |   |
|---------|---|
| mapping | Set of aesthetic mappings created by <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    | <p>The data to be displayed in this layer. There are three options:</p> <p>If <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to <a href="#">ggplot()</a>.</p> <p>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.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).</p>   |
| stat    | <p>The statistical transformation to use on the data for this layer. When using a <code>geom_*()</code> function to construct a layer, the <code>stat</code> argument can be used to override the default coupling between geoms and stats. The <code>stat</code> argument accepts the following:</p> <ul style="list-style-type: none"> <li>• A Stat ggproto subclass, for example <code>StatCount</code>.</li> <li>• A string naming the stat. To give the stat as a string, strip the function name of the <code>stat_</code> prefix. For example, to use <code>stat_count()</code>, give the stat as <code>"count"</code>.</li> <li>• For more information and other ways to specify the stat, see the <a href="#">layer stat</a> documentation.</li> </ul> |

position	<p>A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following:</p> <ul style="list-style-type: none"> <li>• The result of calling a position function, such as <code>position_jitter()</code>. This method allows for passing extra arguments to the position.</li> <li>• A string naming the position adjustment. To give the position as a string, strip the function name of the <code>position_</code> prefix. For example, to use <code>position_jitter()</code>, give the position as "jitter".</li> <li>• For more information and other ways to specify the position, see the <a href="#">layer position</a> documentation.</li> </ul>
...	additional arguments passed on to <a href="#">layer()</a> .
arrow	arrow, the arrow appear at both ends of the brackets, created by <a href="#">grid::arrow()</a> .
parse	logical, whether to parse the labels as expressions and displayed as described in <a href="#">plotmath</a> .
bracket	logical, whether to display the bracket. If FALSE, only the label will be displayed.
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> .

## Value

LayerInstance, a layer that can be added to a ggplot.

## Aesthetics

- required: `xmin`, `xmax`, `ymin`, `ymax`, `label`
- optional: `angle`, `alpha`, `hjust`, `vjust`, `colour`, `family`, `fontsize`, `fontface`, `linetype`, `linewidth`, `lineheight`

## Author(s)

HMU-WH

## Examples

```
library(ggplot2)

ggplot(mpg, aes(class, displ)) +
  geom_boxplot() +
  annotate("bracket", xmin = 2, xmax = 4, ymin = 4.5, ymax = 5, label = "label",
         arrow = grid::arrow(type = "closed", length = unit(0.1, "inches")))
```

stat\_compare

*Add Mean Comparison for Groups to a ggplot***Description**

Add group mean comparisons to a ggplot. The comparisons can be performed using the t-test, Wilcoxon rank-sum test, one-way ANOVA, or Kruskal-Wallis test.

**Usage**

```
stat_compare(
  mapping = NULL,
  data = NULL,
  position = "identity",
  ...,
  nudge = 0,
  start = NULL,
  breaks = NULL,
  labels = NULL,
  cutoff = NULL,
  method = NULL,
  ref_group = NULL,
  tip_length = 0.02,
  parametric = FALSE,
  correction = "none",
  panel_indep = FALSE,
  method_args = NULL,
  comparisons = NULL,
  step_increase = 0.1,
  inherit.aes = TRUE
)
```

**Arguments**

mapping	Set of aesthetic mappings created by <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	<p>The data to be displayed in this layer. There are three options:</p> <p>If <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to <a href="#">ggplot()</a>.</p> <p>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.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).</p>

position	A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following: <ul style="list-style-type: none"> <li>• The result of calling a position function, such as <code>position_jitter()</code>. This method allows for passing extra arguments to the position.</li> <li>• A string naming the position adjustment. To give the position as a string, strip the function name of the <code>position_</code> prefix. For example, to use <code>position_jitter()</code>, give the position as "jitter".</li> <li>• For more information and other ways to specify the position, see the <a href="#">layer position</a> documentation.</li> </ul>
...	additional arguments passed on to <code>geom_bracket()</code> .
nudge	numeric, the nudge of start position in fraction of scale range.
start	numeric, the bracket start position. Defaults to the maximum value of y.
breaks	numeric, the breaks for p-value labels, like <code>c(0, 0.001, 0.01, 0.05, 1)</code> .
labels	character, the labels for p-value breaks, like <code>c("***", "**", "*", "ns")</code> .
cutoff	numeric, the cutoff for p-value, labels above this value will be removed.
method	function, the method for the test; it should support formula interface and return a list with components <code>p.value</code> and <code>method</code> (name).
ref_group	character, the reference group for comparison. other groups will be compared to this group.
tip_length	numeric, the length of the bracket tips in fraction of scale range.
parametric	logical, whether to use parametric test (t-test, One-way ANOVA) or non-parametric test (Wilcoxon rank sum test, Kruskal-Wallis test). Applicable only when method is NULL.
correction	character, the method for p-value adjustment; options include <a href="#">p.adjust.methods</a> with "none" as the default.
panel_indep	logical, whether to correct the p-value only at the panel level. If FALSE, the p-value will be corrected at the layer level.
method_args	list, additional arguments to be passed to the test method.
comparisons	list, a list of comparisons to be made. Each element should contain two groups to be compared.
step_increase	numeric, the step increase in fraction of scale range for every additional comparison, in order to avoid overlapping brackets.
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> .

## Details

Usually you do not need to specify the test method, you only need to tell `stat_compare()` whether you want to perform a parametric test or a nonparametric test, and `stat_compare()` will automatically choose the appropriate test method based on your data. For comparisons between two groups, the p-value is calculated by t-test (parametric) or Wilcoxon rank sum test (nonparametric). For comparisons among more than two groups, the p-value is calculated by One-way ANOVA (parametric) or Kruskal-Wallis test (nonparametric).

**Value**

LayerInstance, a layer that can be added to a ggplot.

**Aesthetics**

- required: x, y

**Computed variables**

- p: p-value of the test.
- q: adjusted p-value of the test.
- label: the label of the bracket.
- method: the method name of the test.
- xmin, xmax, ymin, ymax: the position of the bracket.

**Author(s)**

HMU-WH

**Examples**

```
library(ggplot2)

p <- ggplot(mpg, aes(class, displ, color = class)) +
  geom_boxplot(show.legend = FALSE) +
  theme_test()

# Global comparison: Each x has only one group.
p + stat_compare()
# If you just want to display text, you can set parameters "bracket" to FALSE.
p + stat_compare(bracket = FALSE)
# If you want to display the test method, you can do this.
p + stat_compare(aes(label = after_stat(sprintf("%s: %s", method, label))))

# Comparison between two groups: specify a reference group.
p + stat_compare(ref_group = "minivan")
# If you only want to display the p-value less or equal to 0.01, you can do this.
p + stat_compare(ref_group = "minivan", cutoff = 0.01)
# If you want to display the significance level, you can do this.
p + stat_compare(ref_group = "minivan", breaks = c(0, 0.001, 0.01, 0.05, 1))

# Comparison between two groups: specify the comparison group.
p + stat_compare(tip_length = 0.05,
  step_increase = 0,
  comparisons = list(c("compact", "midsize"), c("pickup", "suv")),
  arrow = grid::arrow(type = "closed", length = unit(0.1, "inches")))
# Yeah, this supports adding arrows.

# Within-group (grouped by the x-axis) population comparison.
ggplot(mpg, aes(drv, displ, fill = class)) +
```

```
geom_boxplot() +
stat_compare() +
stat_compare(aes(group = drv), nudge = 0.1, color = "gray") + # add global comparison
theme_test()

# Better adaptation to faceting.
ggplot(mpg, aes(drv, displ)) +
  geom_boxplot() +
  stat_compare(comparisons = combn(unique(mpg$drv), 2, simplify = FALSE)) +
  facet_grid(cols = vars(class), scales = "free") +
  theme_test()

# P-value correction
p <- ggplot(mpg, aes(class, displ)) +
  geom_boxplot() +
  facet_grid(cols = vars(cyl), scales = "free") +
  theme_test()

# Layer-level P-value correction
p + stat_compare(ref_group = 1, correction = "fdr")
# Panel-level P-value correction
p + stat_compare(ref_group = 1, correction = "fdr", panel_indep = TRUE)
```

# Index

`aes()`, [2](#), [4](#)

`borders()`, [3](#), [5](#)

`fortify()`, [2](#), [4](#)

`geom_bracket`, [2](#)

`geom_bracket()`, [5](#)

`ggplot()`, [2](#), [4](#)

`grid::arrow()`, [3](#)

layer position, [3](#), [5](#)

layer stat, [2](#)

`layer()`, [3](#)

`p.adjust.methods`, [5](#)

`plotmath`, [3](#)

`stat_compare`, [4](#)