Package 'ggstatsplot'

October 6, 2020

Type Package

Title 'ggplot2' Based Plots with Statistical Details

Version 0.6.1

Maintainer Indrajeet Patil <patilindrajeet.science@gmail.com>

Description Extension of 'ggplot2', 'ggstatsplot' creates graphics with details from statistical tests included in the plots themselves. It provides an easier API to generate information-rich plots for statistical analysis of continuous (violin plots, scatterplots, histograms, dot plots, dot-and-whisker plots) or categorical (pie and bar charts) data. Currently, it supports only the most common types of statistical tests: parametric, nonparametric, robust, and Bayesian versions of t-test/ANOVA, correlation analyses, contingency table analysis, meta-analysis, and regression analyses.

License GPL-3 | file LICENSE

URL https://indrajeetpatil.github.io/ggstatsplot/, https://github.com/IndrajeetPatil/ggstatsplot

BugReports https://github.com/IndrajeetPatil/ggstatsplot/issues

Depends R (>= 3.6.0)

Imports broomExtra, cowplot, dplyr, effectsize, ggcorrplot, ggExtra, ggplot2, ggrepel, ggsignif, insight, ipmisc (>= 4.0.0), pairwiseComparisons (>= 3.0.0), paletteer, parameters, purrr, rlang, stats, statsExpressions (>= 0.5.1), tidyr

Suggests forcats, knitr, MASS, rmarkdown, spelling, testthat, tibble, WRS2

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Author Indrajeet Patil [cre, aut, cph]
(<https://orcid.org/0000-0003-1995-6531>)

Repository CRAN

Date/Publication 2020-10-06 11:20:03 UTC

R topics documented:

`ggstatsplot-package` *ggstatsplot: 'ggplot2' Based Plots with Statistical Details*

Description

ggstatsplot is an extension of ggplot2 package for creating graphics with details from statistical tests included in the plots themselves and targeted primarily at behavioral sciences community to provide a one-line code to produce information-rich plots. In a typical exploratory data analysis workflow, data visualization and statistical modeling are two different phases: visualization informs modeling, and modeling in its turn can suggest a different visualization method, and so on and so forth. The central idea of ggstatsplot is simple: combine these two phases into one in the form of graphics with statistical details, which makes data exploration simpler and faster. Currently, it supports only the most common types of statistical tests (parametric, non-parametric, Bayes Factor, and robust versions of *t*-test/anova, correlation, regression analyses, contingency tables analyses, and meta-analysis).

Details

ggstatsplot

The main functions are-

- ggbetweenstats function to produce information-rich comparison plot *between* different groups or conditions with ggplot2 and details from the statistical tests in the subtitle.
- ggwithinstats function to produce information-rich comparison plot *within* different groups or conditions with ggplot2 and details from the statistical tests in the subtitle.
- ggscatterstats function to produce ggplot2 scatterplots along with a marginal histograms/boxplots/density plots from ggExtra and details from the statistical tests in the subtitle.
- ggpiestats function to produce pie chart with details from the statistical tests in the subtitle.
- ggbarstats function to produce stacked bar chart with details from the statistical tests in the subtitle.
- gghistostats function to produce histogram for a single variable with results from one sample test displayed in the subtitle.
- ggdotplotstats function to produce Cleveland-style dot plots/charts for a single variable with labels and results from one sample test displayed in the subtitle.
- ggcorrmat function to visualize the correlation matrix.
- ggcoefstats function to visualize results from regression analyses.
- combine_plots2 helper function to combine multiple ggstatsplot plots using cowplot::plot_grid() with a combination of title, caption, and annotation label.

For more documentation, see the dedicated Website.

Author(s)

Maintainer: Indrajeet Patil <patilindrajeet.science@gmail.com> (ORCID) [copyright holder]

See Also

Useful links:

- https://indrajeetpatil.github.io/ggstatsplot/

- https://github.com/IndrajeetPatil/ggstatsplot
- Report bugs at https://github.com/IndrajeetPatil/ggstatsplot/issues

---

bugs_long *Tidy version of the "Bugs" dataset.*

Description

Tidy version of the "Bugs" dataset.

Usage

```
bugs_long
```

Format

A data frame with 372 rows and 6 variables

- subject. Dummy identity number for each participant.
- gender. Participant's gender (Female, Male).
- region. Region of the world the participant was from.
- education. Level of education.
- condition. Condition of the experiment the participant gave rating for (**LDLF**: low freighteningness and low disgustingness; **LFHD**: low freighteningness and high disgustingness; **HFHD**: high freighteningness and low disgustingness; **HFHD**: high freighteningness and high disgustingness).
- desire. The desire to kill an arthropod was indicated on a scale from 0 to 10.

Details

This data set, "Bugs", provides the extent to which men and women want to kill arthropods that vary in freighteningness (low, high) and disgustingness (low, high). Each participant rates their attitudes towards all anthropods. Subset of the data reported by Ryan et al. (2013).

Source

https://www.sciencedirect.com/science/article/pii/S0747563213000277

Examples

```
dim(bugs_long)
head(bugs_long)
dplyr::glimpse(bugs_long)
```

bugs_wide *Wide-format version of the "Bugs" dataset.*

Description

Wide-format version of the "Bugs" dataset.

Usage

```
bugs_wide
```

Format

A data frame with 93 rows and 6 variables

- subject. Dummy identity number for each participant.
- gender. Participant's gender (Female, Male).
- region. Region of the world the participant was from.
- education. Level of education.
- ldlf,ldhf,hdlf,hdhf.The desire to kill an arthropod was indicated on a scale from 0 to 10 in each condition of the experiment (**LDLF**: low freighteningness and low disgustingness; **LFHD**: low freighteningness and high disgustingness; **HFHD**: high freighteningness and low disgustingness; **HFHD**: high freighteningness and high disgustingness).

Details

This data set, "Bugs", provides the extent to which men and women want to kill arthropods that vary in freighteningness (low, high) and disgustingness (low, high). Each participant rates their attitudes towards all anthropods. Subset of the data reported by Ryan et al. (2013).

Source

https://www.sciencedirect.com/science/article/pii/S0747563213000277

Examples

```
dim(bugs_wide)
head(bugs_wide)
dplyr::glimpse(bugs_wide)
```

combine_plots *Combining and arranging multiple plots in a grid*

Description

Wrapper around plot_grid that will return a plotgrid along with a combination of title, caption, and annotation label

Usage

```
combine_plots(
  ...,
  title.text = NULL,
  title.color = "black",
  title.size = 16,
  title.vjust = 0.5,
  title.hjust = 0.5,
  title.fontface = "bold",
  caption.text = NULL,
  caption.color = "black",
  caption.size = 10,
  caption.vjust = 0.5,
  caption.hjust = 0.5,
  caption.fontface = "plain",
  sub.text = NULL,
  sub.color = "black",
  sub.size = 12,
  sub.vjust = 0.5,
  sub.hjust = 0.5,
  sub.fontface = "plain",
  sub.x = 0.5,
  sub.y = 0.5,
  sub.vpadding = ggplot2::unit(1, "lines"),
  sub.angle = 0,
  sub.lineheight = 0.9,
  title.rel.heights = c(0.1, 1.2),
  caption.rel.heights = c(1.2, 0.1),
  title.caption.rel.heights = c(0.1, 1.2, 0.1)
)
```

Arguments

...	Arguments passed on to cowplot::plot_grid
	plotlist (optional) List of plots to display. Alternatively, the plots can be provided individually as the first n arguments of the function plot_grid (see examples).

`align` (optional) Specifies whether graphs in the grid should be horizontally ("h") or vertically ("v") aligned. Options are "none" (default), "hv" (align in both directions), "h", and "v".

`axis` (optional) Specifies whether graphs should be aligned by the left ("l"), right ("r"), top ("t"), or bottom ("b") margins. Options are "none" (default), or a string of any combination of l, r, t, and b in any order (e.g. "tblr" or "rlbt" for aligning all margins). Must be specified if any of the graphs are complex (e.g. faceted) and alignment is specified and desired. See `align_plots()` for details.

`nrow` (optional) Number of rows in the plot grid.

`ncol` (optional) Number of columns in the plot grid.

`rel_widths` (optional) Numerical vector of relative columns widths. For example, in a two-column grid, `rel_widths = c(2,1)` would make the first column twice as wide as the second column.

`rel_heights` (optional) Numerical vector of relative rows heights. Works just as `rel_widths` does, but for rows rather than columns.

`labels` (optional) List of labels to be added to the plots. You can also set `labels="AUTO"` to auto-generate upper-case labels or `labels="auto"` to auto-generate lower-case labels.

`label_size` (optional) Numerical value indicating the label size. Default is 14.

`label_fontfamily` (optional) Font family of the plot labels. If not provided, is taken from the current theme.

`label_fontface` (optional) Font face of the plot labels. Default is "bold".

`label_colour` (optional) Color of the plot labels. If not provided, is taken from the current theme.

`label_x` (optional) Single value or vector of x positions for plot labels, relative to each subplot. Defaults to 0 for all labels. (Each label is placed all the way to the left of each plot.)

`label_y` (optional) Single value or vector of y positions for plot labels, relative to each subplot. Defaults to 1 for all labels. (Each label is placed all the way to the top of each plot.)

`hjust` Adjusts the horizontal position of each label. More negative values move the label further to the right on the plot canvas. Can be a single value (applied to all labels) or a vector of values (one for each label). Default is -0.5.

`vjust` Adjusts the vertical position of each label. More positive values move the label further down on the plot canvas. Can be a single value (applied to all labels) or a vector of values (one for each label). Default is 1.5.

`scale` Individual number or vector of numbers greater than 0. Enables you to scale the size of all or select plots. Usually it's preferable to set margins instead of using `scale`, but `scale` can sometimes be more powerful.

`greedy` (optional) How should margins be adjusted during alignment. See `align_plots()` for details.

`byrow` Logical value indicating if the plots should be arrange by row (default) or by column.

`cols` Deprecated. Use `ncol`.

rows Deprecated. Use nrow.

title.text String or plotmath expression to be drawn as title for the *combined plot*.

title.color Text color for title.

title.size Point size of title text.

title.vjust Vertical justification for title. Default = 0.5 (centered on y). 0 = baseline at y, 1 = ascender at y.

title.hjust Horizontal justification for title. Default = 0.5 (centered on x). 0 = flush-left at x, 1 = flush-right.

title.fontface The font face ("plain", "bold" (default), "italic", "bold.italic") for title.

caption.text String or plotmath expression to be drawn as the caption for the *combined plot*.

caption.color Text color for caption.

caption.size Point size of title text.

caption.vjust Vertical justification for caption. Default = 0.5 (centered on y). 0 = baseline at y, 1 = ascender at y.

caption.hjust Horizontal justification for caption. Default = 0.5 (centered on x). 0 = flush-left at x, 1 = flush-right.

caption.fontface
The font face ("plain" (default), "bold", "italic", "bold.italic") for caption.

sub.text The label with which the *combined plot* should be annotated. Can be a plotmath expression.

sub.color Text color for annotation label (Default: "black").

sub.size Point size of annotation text (Default: 12).

sub.vjust Vertical justification for annotation label (Default: 0.5).

sub.hjust Horizontal justification for annotation label (Default: 0.5).

sub.fontface The font face ("plain" (default), "bold", "italic", "bold.italic") for the annotation label.

sub.x The x position of annotation label (Default: 0.5).

sub.y The y position of annotation label (Default: 0.5).

sub.vpadding Vertical padding. The total vertical space added to the label, given in grid units. By default, this is added equally above and below the label. However, by changing the y and vjust parameters, this can be changed (Default: ggplot2::unit(1,"lines")).

sub.angle Angle at which annotation label is to be drawn (Default: 0).

sub.lineheight Line height of annotation label.

title.rel.heights
Numerical vector of relative columns heights while combining (title, plot).

caption.rel.heights
Numerical vector of relative columns heights while combining (plot, caption).

title.caption.rel.heights
Numerical vector of relative columns heights while combining (title, plot, caption).

Value

Combined plot with title and/or caption and/or annotation label

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/combine_plots.html

Examples

```
# loading the necessary libraries
library(ggplot2)

# preparing the first plot
p1 <-
  ggplot2::ggplot(
    data = subset(iris, iris$Species == "setosa"),
    aes(x = Sepal.Length, y = Sepal.Width)
  ) +
  geom_point() +
  labs(title = "setosa")

# preparing the second plot
p2 <-
  ggplot2::ggplot(
    data = subset(iris, iris$Species == "versicolor"),
    aes(x = Sepal.Length, y = Sepal.Width)
  ) +
  geom_point() +
  labs(title = "versicolor")

# combining the plot with a title and a caption
combine_plots(
  p1,
  p2,
  labels = c("(a)", "(b)"),
  title.text = "Dataset: Iris Flower dataset",
  caption.text = "Note: Only two species of flower are displayed",
  title.color = "red",
  caption.color = "blue"
)
```

---

combine_plots2 *Simpler way to combine and arrange multiple plots in a grid*

Description

Wrapper around `cowplot::plot_grid` that will return a plotgrid along with a combination of title, caption, and annotation label. This is a simpler version of the `combine_plots` function in this package.

Usage

```
combine_plots2(
  plotlist,
  plotgrid.args = list(),
  title.text = NULL,
  title.args = list(size = 16, fontface = "bold"),
  caption.text = NULL,
  caption.args = list(size = 10),
  sub.text = NULL,
  sub.args = list(size = 12),
  title.rel.heights = c(0.1, 1.2),
  caption.rel.heights = c(1.2, 0.1),
  title.caption.rel.heights = c(0.1, 1.2, 0.1),
  ...
)
```

Arguments

`plotlist`	A list of plots to display.
`plotgrid.args`	A list of additional arguments to `cowplot::plot_grid`.
`title.text`	String or plotmath expression to be drawn as title for the *combined plot*.
`title.args, caption.args, sub.args`	A list of additional arguments provided to `title`, `caption` and `sub`, resp.
`caption.text`	String or plotmath expression to be drawn as the caption for the *combined plot*.
`sub.text`	The label with which the *combined plot* should be annotated. Can be a plotmath expression.
`title.rel.heights`	Numerical vector of relative columns heights while combining (title, plot).
`caption.rel.heights`	Numerical vector of relative columns heights while combining (plot, caption).
`title.caption.rel.heights`	Numerical vector of relative columns heights while combining (title, plot, caption).
`...`	Currently ignored.

Value

Combined plot with title and/or caption and/or annotation label

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/combine_plots.html

Examples

```
# loading the necessary libraries
library(ggplot2)

# preparing the first plot
p1 <-
  ggplot2::ggplot(
    data = subset(iris, iris$Species == "setosa"),
    aes(x = Sepal.Length, y = Sepal.Width)
  ) +
  geom_point() +
  labs(title = "setosa")

# preparing the second plot
p2 <-
  ggplot2::ggplot(
    data = subset(iris, iris$Species == "versicolor"),
    aes(x = Sepal.Length, y = Sepal.Width)
  ) +
  geom_point() +
  labs(title = "versicolor")

# combining the plot with a title and a caption
combine_plots2(
  plotlist = list(p1, p2),
  plotlist.args = list(labels = c("(a)", "(b)")),
  title.text = "Dataset: Iris Flower dataset",
  caption.text = "Note: Only two species of flower are displayed",
  title.args = list(color = "red"),
  caption.args = list(color = "blue")
)
```

---

ggbarstats *Bar (column) charts with statistical tests*

Description

Bar charts for categorical data with statistical details included in the plot as a subtitle.

Usage

```
ggbarstats(
  data,
  main,
  condition,
  counts = NULL,
  ratio = NULL,
  paired = FALSE,
  results.subtitle = TRUE,
```

```
  sample.size.label = TRUE,
  label = "percentage",
  label.args = list(alpha = 1, fill = "white"),
  conf.level = 0.95,
  nboot = 100L,
  k = 2L,
  proportion.test = TRUE,
  perc.k = 0,
  bf.message = TRUE,
  sampling.plan = "indepMulti",
  fixed.margin = "rows",
  prior.concentration = 1,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  legend.title = NULL,
  xlab = NULL,
  ylab = NULL,
  ggtheme = ggplot2::theme_bw(),
  ggstatsplot.layer = TRUE,
  package = "RColorBrewer",
  palette = "Dark2",
  ggplot.component = NULL,
  output = "plot",
  x = NULL,
  y = NULL,
  ...
)
```

Arguments

`data`	A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will **not** be accepted.
`main`	The variable to use as the **rows** in the contingency table.
`condition`	The variable to use as the **columns** in the contingency table. Default is `NULL`. If `NULL`, one-sample proportion test (a goodness of fit test) will be run for the `x` variable. Otherwise an appropriate association test will be run. This argument can not be `NULL` for `ggbarstats` function.
`counts`	A string naming a variable in data containing counts, or `NULL` if each row represents a single observation (Default).
`ratio`	A vector of proportions: the expected proportions for the proportion test (should sum to 1). Default is `NULL`, which means the null is equal theoretical proportions across the levels of the nominal variable. This means if there are two levels this will be `ratio = c(0.5,0.5)` or if there are four levels this will be `ratio = c(0.25,0.25,0.25,0.25)`, etc.
`paired`	Logical indicating whether data came from a within-subjects or repeated measures design study (Default: `FALSE`). If `TRUE`, McNemar's test subtitle will be returned. If `FALSE`, Pearson's chi-square test will be returned.

`results.subtitle`
: Decides whether the results of statistical tests are to be displayed as a subtitle (Default: `TRUE`). If set to `FALSE`, only the plot will be returned.

`sample.size.label`
: Logical that decides whether sample size information should be displayed for each level of the grouping variable `y` (Default: `TRUE`).

`label`
: Character decides what information needs to be displayed on the label in each pie slice. Possible options are `"percentage"` (default), `"counts"`, `"both"`.

`label.args`
: Additional aesthetic arguments that will be passed to `geom_label`.

`conf.level`
: Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (`0.95`).

`nboot`
: Number of bootstrap samples for computing confidence interval for the effect size (Default: `100`).

`k`
: Number of digits after decimal point (should be an integer) (Default: `k = 2`).

`proportion.test`
: Decides whether proportion test for `main` variable is to be carried out for each level of `y` (Default: `TRUE`).

`perc.k`
: Numeric that decides number of decimal places for percentage labels (Default: `0`).

`bf.message`
: Logical that decides whether to display Bayes Factor in favor of the *null* hypothesis. This argument is relevant only **for parametric test** (Default: `TRUE`).

`sampling.plan`
: Character describing the sampling plan. Possible options are `"indepMulti"` (independent multinomial; default), `"poisson"`, `"jointMulti"` (joint multinomial), `"hypergeom"` (hypergeometric). For more, see `?BayesFactor::contingencyTableBF()`.

`fixed.margin`
: For the independent multinomial sampling plan, which margin is fixed (`"rows"` or `"cols"`). Defaults to `"rows"`.

`prior.concentration`
: Specifies the prior concentration parameter, set to 1 by default. It indexes the expected deviation from the null hypothesis under the alternative, and corresponds to Gunel and Dickey's (1974) `"a"` parameter.

`title`
: The text for the plot title.

`subtitle`
: The text for the plot subtitle. Will work only if `results.subtitle = FALSE`.

`caption`
: The text for the plot caption.

`legend.title`
: Title text for the legend.

`xlab`
: Custom text for the `x` axis label (Default: `NULL`, which will cause the `x` axis label to be the `x` variable).

`ylab`
: Custom text for the `y` axis label (Default: `NULL`).

`ggtheme`
: A function, `ggplot2` theme name. Default value is `ggplot2::theme_bw()`. Any of the `ggplot2` themes, or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`, etc.).

`ggstatsplot.layer`
: Logical that decides whether `theme_ggstatsplot` theme elements are to be displayed along with the selected `ggtheme` (Default: `TRUE`). `theme_ggstatsplot` is an opinionated theme layer that override some aspects of the selected `ggtheme`.

package	Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running `View(paletteer::palettes_d_names)`.
palette	Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running `View(paletteer::palettes_d_names)`.
ggplot.component	A ggplot component to be added to the plot prepared by ggstatsplot. This argument is primarily helpful for grouped_ variant of the current function. Default is NULL. The argument should be entered as a function.
output	Character that describes what is to be returned: can be "plot" (default) or "subtitle" or "caption". Setting this to "subtitle" will return the expression containing statistical results. If you have set results.subtitle = FALSE, then this will return a NULL. Setting this to "caption" will return the expression containing details about Bayes Factor analysis, but valid only when type = "parametric" and bf.message = TRUE, otherwise this will return a NULL. For functions ggpiestats and ggbarstats, setting output = "proptest" will return a dataframe containing results from proportion tests.
x	The variable to use as the **rows** in the contingency table.
y	The variable to use as the **columns** in the contingency table. Default is NULL. If NULL, one-sample proportion test (a goodness of fit test) will be run for the x variable. Otherwise an appropriate association test will be run. This argument can not be NULL for ggbarstats function.
...	Currently ignored.

Value

Unlike a number of statistical softwares, ggstatsplot doesn't provide the option for Yates' correction for the Pearson's chi-squared statistic. This is due to compelling amount of Monte-Carlo simulation research which suggests that the Yates' correction is overly conservative, even in small sample sizes. As such it is recommended that it should not ever be applied in practice (Camilli & Hopkins, 1978, 1979; Feinberg, 1980; Larntz, 1978; Thompson, 1988).

For more about how the effect size measures and their confidence intervals are computed, see ?rcompanion::cohenG, ?rcompanion::cramerV, and ?rcompanion::cramerVFit.

See Also

grouped_ggbarstats, ggpiestats, grouped_ggpiestats

Examples

```
# for reproducibility
set.seed(123)

# association test (or contingency table analysis)
ggstatsplot::ggbarstats(
  data = mtcars,
  x = vs,
  y = cyl
)
```

`ggbetweenstats` *Box/Violin plots for group or condition comparisons in between-subjects designs.*

Description

A combination of box and violin plots along with jittered data points for between-subjects designs with statistical details included in the plot as a subtitle.

Usage

```
ggbetweenstats(
  data,
  x,
  y,
  plot.type = "boxviolin",
  type = "parametric",
  pairwise.comparisons = TRUE,
  pairwise.display = "significant",
  p.adjust.method = "holm",
  effsize.type = "unbiased",
  partial = TRUE,
  bf.prior = 0.707,
  bf.message = TRUE,
  results.subtitle = TRUE,
  xlab = NULL,
  ylab = NULL,
  caption = NULL,
  title = NULL,
  subtitle = NULL,
  sample.size.label = TRUE,
  k = 2L,
  var.equal = FALSE,
  conf.level = 0.95,
  nboot = 100,
  tr = 0.1,
  mean.plotting = TRUE,
  mean.ci = FALSE,
  mean.point.args = list(size = 5, color = "darkred"),
  mean.label.args = list(size = 3),
  notch = FALSE,
  notchwidth = 0.5,
  outlier.tagging = FALSE,
  outlier.label = NULL,
  outlier.coef = 1.5,
  outlier.shape = 19,
  outlier.color = "black",
```

```
  outlier.label.args = list(size = 3),
  outlier.point.args = list(),
 point.args = list(position = ggplot2::position_jitterdodge(dodge.width = 0.6), alpha
    = 0.4, size = 3, stroke = 0),
  violin.args = list(width = 0.5, alpha = 0.2),
  ggsignif.args = list(textsize = 3, tip_length = 0.01),
  ggtheme = ggplot2::theme_bw(),
  ggstatsplot.layer = TRUE,
  package = "RColorBrewer",
  palette = "Dark2",
  ggplot.component = NULL,
  output = "plot",
  ...
)
```

Arguments

`data`	A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will **not** be accepted.
`x`	The grouping variable from the dataframe `data`.
`y`	The response (a.k.a. outcome or dependent) variable from the dataframe `data`.
`plot.type`	Character describing the *type* of plot. Currently supported plots are `"box"` (for pure boxplots), `"violin"` (for pure violin plots), and `"boxviolin"` (for a combination of box and violin plots; default).
`type`	Type of statistic expected (`"parametric"` or `"nonparametric"` or `"robust"` or `"bayes"`).Corresponding abbreviations are also accepted: `"p"` (for parametric), `"np"` (nonparametric), `"r"` (robust), or `"bf"`resp.
`pairwise.comparisons`	Logical that decides whether pairwise comparisons are to be displayed (default: `FALSE`). Please note that only **significant** comparisons will be shown by default. To change this behavior, select appropriate option with `pairwise.display` argument. The pairwise comparison dataframes are prepared using the `pairwiseComparisons::pairwise_` function. For more details about pairwise comparisons, see the documentation for that function.
`pairwise.display`	Decides which pairwise comparisons to display. Available options are `"significant"` (abbreviation accepted: `"s"`) or `"non-significant"` (abbreviation accepted: `"ns"`) or `"everything"/"all"`. The default is `"significant"`. You can use this argument to make sure that your plot is not uber-cluttered when you have multiple groups being compared and scores of pairwise comparisons being displayed.
`p.adjust.method`	Adjustment method for *p*-values for multiple comparisons. Possible methods are: `"holm"` (default), `"hochberg"`, `"hommel"`, `"bonferroni"`, `"BH"`, `"BY"`, `"fdr"`, `"none"`.
`effsize.type`	Type of effect size needed for *parametric* tests. The argument can be `"biased"` (equivalent to `"d"` for Cohen's *d* for **t-test**; `"partial_eta"` for partial eta-

squared for **anova**) or "unbiased" (equivalent to "g" Hedge's g for **t-test**; "partial_omega" for partial omega-squared for **anova**)).

partial — If TRUE, return partial indices.

bf.prior — A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors.

bf.message — Logical that decides whether to display Bayes Factor in favor of the *null* hypothesis. This argument is relevant only **for parametric test** (Default: TRUE).

results.subtitle — Decides whether the results of statistical tests are to be displayed as a subtitle (Default: TRUE). If set to FALSE, only the plot will be returned.

xlab, ylab — Labels for x and y axis variables. If NULL (default), variable names for x and y will be used.

caption — The text for the plot caption.

title — The text for the plot title.

subtitle — The text for the plot subtitle. Will work only if results.subtitle = FALSE.

sample.size.label — Logical that decides whether sample size information should be displayed for each level of the grouping variable x (Default: TRUE).

k — Number of digits after decimal point (should be an integer) (Default: k = 2).

var.equal — a logical variable indicating whether to treat the variances in the samples as equal. If TRUE, then a simple F test for the equality of means in a one-way analysis of variance is performed. If FALSE, an approximate method of Welch (1951) is used, which generalizes the commonly known 2-sample Welch test to the case of arbitrarily many samples.

conf.level — Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (0.95).

nboot — Number of bootstrap samples for computing confidence interval for the effect size (Default: 100).

tr — Trim level for the mean when carrying out robust tests. If you get error stating "Standard error cannot be computed because of Winsorized variance of 0 (e.g., due to ties). Try to decrease the trimming level.", try to play around with the value of tr, which is by default set to 0.1. Lowering the value might help.

mean.plotting — Logical that decides whether mean is to be highlighted and its value to be displayed (Default: TRUE).

mean.ci — Logical that decides whether 95% confidence interval for mean is to be displayed (Default: FALSE).

mean.point.args, mean.label.args — A list of additional aesthetic arguments to be passed to ggplot2::geom_point and ggrepel::geom_label_repel geoms involved mean value plotting.

notch — A logical. If FALSE (default), a standard box plot will be displayed. If TRUE, a notched box plot will be used. Notches are used to compare groups; if the notches of two boxes do not overlap, this suggests that the medians are significantly different. In a notched box plot, the notches extend 1.58 * IQR / sqrt(n). This gives a roughly 95% confidence interval for comparing medians. IQR: Inter-Quartile Range.

notchwidth For a notched box plot, width of the notch relative to the body (default 0.5).

outlier.tagging
Decides whether outliers should be tagged (Default: FALSE).

outlier.label Label to put on the outliers that have been tagged. This **can't** be the same as x argument.

outlier.coef Coefficient for outlier detection using Tukey's method. With Tukey's method, outliers are below (1st Quartile) or above (3rd Quartile) outlier.coef times the Inter-Quartile Range (IQR) (Default: 1.5).

outlier.shape 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.color Default aesthetics for outliers (Default: "black").

outlier.point.args, outlier.label.args
A list of additional aesthetic arguments to be passed to ggplot2::geom_point and ggrepel::geom_label_repel geoms involved outlier value plotting.

point.args A list of additional aesthetic arguments to be passed to the geom_point displaying the raw data.

violin.args A list of additional aesthetic arguments to be passed to the geom_violin.

ggsignif.args A list of additional aesthetic arguments to be passed to ggsignif::geom_signif.

ggtheme A function, ggplot2 theme name. Default value is ggplot2::theme_bw(). Any of the ggplot2 themes, or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.).

ggstatsplot.layer
Logical that decides whether theme_ggstatsplot theme elements are to be displayed along with the selected ggtheme (Default: TRUE). theme_ggstatsplot is an opinionated theme layer that override some aspects of the selected ggtheme.

package, palette
Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running View(paletteer::palettes_d_names).

ggplot.component
A ggplot component to be added to the plot prepared by ggstatsplot. This argument is primarily helpful for grouped_ variant of the current function. Default is NULL. The argument should be entered as a function.

output Character that describes what is to be returned: can be "plot" (default) or "subtitle" or "caption". Setting this to "subtitle" will return the expression containing statistical results. If you have set results.subtitle = FALSE, then this will return a NULL. Setting this to "caption" will return the expression containing details about Bayes Factor analysis, but valid only when type = "parametric" and bf.message = TRUE, otherwise this will return a NULL. For functions ggpiestats and ggbarstats, setting output = "proptest" will return a dataframe containing results from proportion tests.

... Currently ignored.

Details

For parametric tests, Welch's ANOVA/*t*-test are used as a default (i.e., `var.equal = FALSE`).

If robust tests are selected, following tests are used is .

- ANOVA: one-way ANOVA on trimmed means (see `?WRS2::t1way`)
- *t*-test: Yuen's test for trimmed means (see `?WRS2::yuen`)

For more about how the effect size measures (for nonparametric tests) and their confidence intervals are computed, see `?rcompanion::wilcoxonR`.

For repeated measures designs, use `ggwithinstats`.

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggbetweenstats.html

See Also

grouped_ggbetweenstats, ggwithinstats, grouped_ggwithinstats

Examples

```
# to get reproducible results from bootstrapping
set.seed(123)
library(ggstatsplot)

# simple function call with the defaults
ggstatsplot::ggbetweenstats(
  data = mtcars,
  x = am,
  y = mpg,
  title = "Fuel efficiency by type of car transmission",
  caption = "Transmission (0 = automatic, 1 = manual)"
)

# more detailed function call
ggstatsplot::ggbetweenstats(
  data = datasets::morley,
  x = Expt,
  y = Speed,
  type = "nonparametric",
  plot.type = "box",
  xlab = "The experiment number",
  ylab = "Speed-of-light measurement",
  pairwise.comparisons = TRUE,
  p.adjust.method = "fdr",
  outlier.tagging = TRUE,
  outlier.label = Run,
  ggtheme = ggplot2::theme_grey(),
  ggstatsplot.layer = FALSE
)
```

ggcoefstats *Dot-and-whisker plots for regression analyses*

Description

Dot-and-whisker plots for regression analyses

Usage

```
ggcoefstats(
  x,
  output = "plot",
  statistic = NULL,
  conf.int = TRUE,
  conf.level = 0.95,
  k = 2L,
  exclude.intercept = TRUE,
  exponentiate = FALSE,
  effsize = "eta",
  partial = TRUE,
  meta.analytic.effect = FALSE,
  meta.type = "parametric",
  bf.message = TRUE,
  sort = "none",
  xlab = "regression coefficient",
  ylab = "term",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  only.significant = FALSE,
  point.args = list(size = 3, color = "blue"),
  errorbar.args = list(height = 0),
  vline = TRUE,
  vline.args = list(size = 1, linetype = "dashed"),
  stats.labels = TRUE,
  stats.label.color = NULL,
  stats.label.args = list(size = 3, direction = "y"),
  package = "RColorBrewer",
  palette = "Dark2",
  ggtheme = ggplot2::theme_bw(),
  ggstatsplot.layer = TRUE,
  ...
)
```

Arguments

x	A model object to be tidied, or a tidy data frame containing results from a regression model. Function internally uses parameters::model_parameters or broom::tidy to get a tidy dataframe. If a data frame is used, it *must* contain columns named term (names of predictors) and estimate (corresponding estimates of coefficients or other quantities of interest).
output	Character describing the expected output from this function: "plot" (visualization of regression coefficients) or "tidy" (tidy dataframe of results from broom::tidy) or "glance" (object from broom::glance) or "augment" (object from broom::augment).
statistic	Which statistic is to be displayed (either "t" or "f"or "z" or "chi") in the label. This is relevant if the x argument is a *dataframe*.
conf.int	Logical. Decides whether to display confidence intervals as error bars (Default: TRUE).
conf.level	Numeric deciding level of confidence intervals (Default: 0.95). For MCMC model objects (Stan, JAGS, etc.), this will be probability level for CI.
k	Number of digits after decimal point (should be an integer) (Default: k = 2L).
exclude.intercept	Logical that decides whether the intercept should be excluded from the plot (Default: TRUE).
exponentiate	If TRUE, the x-axis will be logarithmic (Default: FALSE). Note that exponents for the coefficient estimates and associated standard errors plus confidence intervals are computed by the underlying tidying packages (broom/parameters) and not done by ggcoefstats. So this might not work if the underlying packages don't support exponentiation.
effsize	Character describing the effect size to be displayed: "eta" (default) or "omega". This argument is relevant only for models objects of class aov, anova, aovlist, "Gam", and "manova".
partial	Logical that decides if partial eta-squared or partial omega-squared are returned (Default: TRUE). If FALSE, eta-squared or omega-squared will be returned.
meta.analytic.effect	Logical that decides whether subtitle for meta-analysis via linear (mixed-effects) models (default: FALSE). If TRUE, input to argument subtitle will be ignored. This will be mostly relevant if a data frame with estimates and their standard errors is entered.
meta.type	Type of statistics used to carry out random-effects meta-analysis. If "parametric" (default), metafor::rma function will be used. If "robust", metaplus::metaplus function will be used. If "bayes", metaBMA::meta_random function will be used.
bf.message	Logical that decides whether results from running a Bayesian meta-analysis assuming that the effect size *d* varies across studies with standard deviation *t* (i.e., a random-effects analysis) should be displayed in caption. Defaults to TRUE.
sort	If "none" (default) do not sort, "ascending" sort by increasing coefficient value, or "descending" sort by decreasing coefficient value.

xlab, ylab	Labels for x- and y- axis variables, respectively (Defaults: "regression coefficient" and "term").
title	The text for the plot title.
subtitle	The text for the plot subtitle. The input to this argument will be ignored if meta.analytic.effect is set to TRUE.
caption	Text to display as caption. This argument is relevant only when output = "caption".
only.significant	If TRUE, only stats labels for significant effects is shown (Default: FALSE). This can be helpful when a large number of regression coefficients are to be displayed in a single plot. Relevant only when the output is a plot.
point.args	Additional arguments that will be passed to ggplot2::geom_point geom. Please see documentation for that function to know more about these arguments.
errorbar.args	Additional arguments that will be passed to ggplot2::geom_errorbarh geom. Please see documentation for that function to know more about these arguments.
vline	Decides whether to display a vertical line (Default: "TRUE").
vline.args	Additional arguments that will be passed to ggplot2::geom_vline geom. Please see documentation for that function to know more about these arguments.
stats.labels	Logical. Decides whether the statistic and *p*-values for each coefficient are to be attached to each dot as a text label using ggrepel (Default: TRUE).
stats.label.color	Color for the labels. If set to NULL, colors will be chosen from the specified package (Default: "RColorBrewer") and palette (Default: "Dark2").
stats.label.args	Additional arguments that will be passed to ggrepel::geom_label_repel geom. Please see documentation for that function to know more about these arguments.
package	Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running View(paletteer::palettes_d_names).
palette	Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running View(paletteer::palettes_d_names).
ggtheme	A function, ggplot2 theme name. Default value is ggplot2::theme_bw(). Any of the ggplot2 themes, or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.).
ggstatsplot.layer	Logical that decides whether theme_ggstatsplot theme elements are to be displayed along with the selected ggtheme (Default: TRUE). theme_ggstatsplot is an opinionated theme layer that override some aspects of the selected ggtheme.
...	Additional arguments to tidying method. For more, see ?parameters::model_parameters and broom::tidy.

Value

Plot with the regression coefficients' point estimates as dots with confidence interval whiskers and other statistical details included as labels.

Note

All rows of regression estimates where either of the following quantities is NA will be removed if labels are requested: `estimate`, `statistic`, `p.value`.

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggcoefstats.html

Examples

```
# for reproducibility
set.seed(123)

# -------------- with model object ------------------------------------

# model object
mod <- lm(formula = mpg ~ cyl * am, data = mtcars)

# to get a plot
ggstatsplot::ggcoefstats(x = mod, output = "plot")

# to get a tidy dataframe
ggstatsplot::ggcoefstats(x = mod, output = "tidy")

# to get a glance summary
ggstatsplot::ggcoefstats(x = mod, output = "glance")

# to get augmented dataframe
ggstatsplot::ggcoefstats(x = mod, output = "augment")

# -------------- with custom dataframe ---------------------------------

# creating a dataframe
df <-
  structure(
    list(
      term = structure(
        c(3L, 4L, 1L, 2L, 5L),
        .Label = c(
          "Africa",
          "Americas", "Asia", "Europe", "Oceania"
        ),
        class = "factor"
      ),
      estimate = c(
        0.382047603321706,
        0.780783111514665,
        0.425607573765058,
        0.558365541235078,
        0.956473848429961
      ),
```

```
      std.error = c(
        0.0465576338644502,
        0.0330218199731529,
        0.0362834986178494,
        0.0480571500648261,
        0.062215818388157
      ),
      statistic = c(
        8.20590677855356,
        23.6444603038067,
        11.7300588415607,
        11.6187818146078,
        15.3734833553524
      ),
      conf.low = c(
        0.290515146096969,
        0.715841986960399,
        0.354354575031406,
        0.46379116008131,
        0.827446138277154
      ),
      conf.high = c(
        0.473580060546444,
        0.845724236068931,
        0.496860572498711,
        0.652939922388847,
        1.08550155858277
      ),
      p.value = c(
        3.28679518728519e-15,
        4.04778497135963e-75,
        7.59757330804449e-29,
        5.45155840151592e-26,
        2.99171217913312e-13
      ),
      df.error = c(
        394L, 358L, 622L,
        298L, 22L
      )
    ),
    row.names = c(NA, -5L),
    class = c(
      "tbl_df",
      "tbl", "data.frame"
    )
  )

# plotting the dataframe
ggstatsplot::ggcoefstats(
  x = df,
  statistic = "t",
  meta.analytic.effect = TRUE,
  k = 3
```

```
)
```

`ggcorrmat` *Visualization of a correlation matrix*

Description

Visualization of a correlation matrix

Usage

```
ggcorrmat(
  data,
  cor.vars = NULL,
  cor.vars.names = NULL,
  output = "plot",
  matrix.type = "full",
  matrix.method = "square",
  type = "parametric",
  beta = 0.1,
  k = 2L,
  sig.level = 0.05,
  conf.level = 0.95,
  bf.prior = 0.707,
  p.adjust.method = "none",
  pch = "cross",
  ggcorrplot.args = list(outline.color = "black"),
  package = "RColorBrewer",
  palette = "Dark2",
  colors = c("#E69F00", "white", "#009E73"),
  ggtheme = ggplot2::theme_bw(),
  ggstatsplot.layer = TRUE,
  ggplot.component = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  messages = TRUE,
  ...
)
```

Arguments

`data`	Dataframe from which variables specified are preferentially to be taken.
`cor.vars`	List of variables for which the correlation matrix is to be computed and visualized. If `NULL` (default), all numeric variables from `data` will be used.

cor.vars.names	Optional list of names to be used for cor.vars. The names should be entered in the same order.
output	Character that decides expected output from this function. If "plot", the visualization matrix will be returned. If "dataframe" (or literally anything other than "plot"), a dataframe containing all details from statistical analyses (e.g., correlation coefficients, statistic values, *p*-values, no. of observations, etc.) will be returned.
matrix.type	Character, "full" (default), "upper" or "lower", display full matrix, lower triangular or upper triangular matrix.
matrix.method	The visualization method of correlation matrix to be used. Allowed values are "square" (default) or "circle".
type	Type of association between paired samples required (""parametric": Pearson's product moment correlation coefficient" or ""nonparametric": Spearman's rho" or ""robust": percentage bend correlation coefficient" or ""bayes": Bayes Factor for Pearson's *r*"). Corresponding abbreviations are also accepted: "p" (for parametric/pearson), "np" (nonparametric/spearman), "r" (robust), "bf" (for bayes factor), resp.
beta	bending constant (Default: 0.1). For more, see WRS2::pbcor().
k	Number of digits after decimal point (should be an integer) (Default: k = 2L).
sig.level	Significance level (Default: 0.05). If the *p*-value in *p*-value matrix is bigger than sig.level, then the corresponding correlation coefficient is regarded as insignificant and flagged as such in the plot. Relevant only when output = "plot".
conf.level	Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (0.95).
bf.prior	A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors.
p.adjust.method	What adjustment for multiple tests should be used? ("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"). See stats::p.adjust for details about why to use "holm" rather than "bonferroni"). Default is "none". If adjusted *p*-values are displayed in the visualization of correlation matrix, the **adjusted** *p*-values will be used for the **upper** triangle, while **unadjusted** *p*-values will be used for the **lower** triangle of the matrix.
pch	Decides the point shape to be used for insignificant correlation coefficients (only valid when insig = "pch"). Default: pch = "cross".
ggcorrplot.args	A list of additional (mostly aesthetic) arguments that will be passed to ggcorrplot::ggcorrplot function. The list should avoid any of the following arguments since they are already internally being used by ggstatsplot: corr, method, p.mat, sig.level, ggtheme, colors, matrix.type, lab, pch, legend.title, digits.
package	Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running View(paletteer::palettes_d_names).
palette	Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running View(paletteer::palettes_d_names).

colors	A vector of 3 colors for low, mid, and high correlation values. If set to NULL, manual specification of colors will be turned off and 3 colors from the specified palette from package will be selected.
ggtheme	A function, ggplot2 theme name. Default value is ggplot2::theme_bw(). Any of the ggplot2 themes, or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.).
ggstatsplot.layer	Logical that decides whether theme_ggstatsplot theme elements are to be displayed along with the selected ggtheme (Default: TRUE). theme_ggstatsplot is an opinionated theme layer that override some aspects of the selected ggtheme.
ggplot.component	A ggplot component to be added to the plot prepared by ggstatsplot. This argument is primarily helpful for grouped_ variant of the current function. Default is NULL. The argument should be entered as a function.
title	The text for the plot title.
subtitle	The text for the plot subtitle. Will work only if results.subtitle = FALSE.
caption	The text for the plot caption.
messages	Decides whether messages references, notes, and warnings are to be displayed (Default: TRUE).
...	Currently ignored.

Value

Correlation matrix plot or a dataframe containing results from pairwise correlation tests. The package internally uses ggcorrplot::ggcorrplot for creating the visualization matrix, while the correlation analysis is carried out using the correlation::correlation function.

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggcorrmat.html

See Also

grouped_ggcorrmat ggscatterstats grouped_ggscatterstats

Examples

```
# for reproducibility
set.seed(123)

# if `cor.vars` not specified, all numeric variables used
ggstatsplot::ggcorrmat(iris)

# to get the correlalogram
# note that the function will run even if the vector with variable names is
# not of same length as the number of variables
```

```
ggstatsplot::ggcorrmat(
  data = ggplot2::msleep,
  type = "robust",
  cor.vars = sleep_total:bodywt,
  cor.vars.names = c("total sleep", "REM sleep"),
  matrix.type = "lower"
)

# to get the correlation analyses results in a dataframe
ggstatsplot::ggcorrmat(
  data = ggplot2::msleep,
  cor.vars = sleep_total:bodywt,
  output = "dataframe"
)
```

---

ggdotplotstats *Dot plot/chart for labeled numeric data.*

Description

A dot chart (as described by William S. Cleveland) with statistical details from one-sample test included in the plot as a subtitle.

Usage

```
ggdotplotstats(
  data,
  x,
  y,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  type = "parametric",
  test.value = 0,
  bf.prior = 0.707,
  bf.message = TRUE,
  effsize.type = "g",
  conf.level = 0.95,
  nboot = 100,
  k = 2,
  results.subtitle = TRUE,
  point.args = list(color = "black", size = 3, shape = 16),
  test.k = 0,
  test.value.line = FALSE,
  test.value.line.args = list(size = 1),
```

```
  test.value.label.args = list(size = 3),
  centrality.parameter = "mean",
  centrality.k = 2,
  centrality.line.args = list(color = "blue", size = 1),
  centrality.label.args = list(color = "blue", size = 3),
  ggplot.component = NULL,
  ggtheme = ggplot2::theme_bw(),
  ggstatsplot.layer = TRUE,
  output = "plot",
  ...
)
```

Arguments

`data`	A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will **not** be accepted.
`x`	A numeric variable from the dataframe `data`.
`y`	Label or grouping variable.
`xlab`	Labels for `x` and `y` axis variables. If `NULL` (default), variable names for `x` and `y` will be used.
`ylab`	Labels for `x`- and `y`- axis variables, respectively (Defaults: `"regression coefficient"` and `"term"`).
`title`	The text for the plot title.
`subtitle`	The text for the plot subtitle. Will work only if `results.subtitle = FALSE`.
`caption`	The text for the plot caption.
`type`	Type of statistic expected (`"parametric"` or `"nonparametric"` or `"robust"` or `"bayes"`).Corresponding abbreviations are also accepted: `"p"` (for parametric), `"np"` (nonparametric), `"r"` (robust), or `"bf"`resp.
`test.value`	A number specifying the value of the null hypothesis (Default: `0`).
`bf.prior`	A number between `0.5` and `2` (default `0.707`), the prior width to use in calculating Bayes factors.
`bf.message`	Logical that decides whether to display Bayes Factor in favor of the *null* hypothesis. This argument is relevant only **for parametric test** (Default: `TRUE`).
`effsize.type`	Type of effect size needed for *parametric* tests. The argument can be `"d"` (for Cohen's *d*) or `"g"` (for Hedge's *g*).
`conf.level`	Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (`0.95`).
`nboot`	Number of bootstrap samples for computing confidence interval for the effect size (Default: `100`).
`k`	Number of digits after decimal point (should be an integer) (Default: `k = 2`).
`results.subtitle`	Decides whether the results of statistical tests are to be displayed as a subtitle (Default: `TRUE`). If set to `FALSE`, only the plot will be returned.
`point.args`	A list of additional aesthetic arguments passed to `geom_point`.

`test.k` Integer denoting the number of decimal places expected for `test.value` label. (Default: `0`).

`test.value.line`
Logical that decides whether a line corresponding to the `test.value` should be superimposed on the plot.

`test.value.line.args`
A list of additional aesthetic arguments to be passed to the `geom_line` used to display the lines corresponding to the centrality parameter and test value.

`test.value.label.args`
A list of additional aesthetic arguments to be passed to the `geom_label` used to display the label corresponding to the centrality parameter and test value.

`centrality.parameter`
Decides *which* measure of central tendency (`"mean"` or `"median"`) is to be displayed as a vertical line. To not show any parameter, set this to `"none"`.

`centrality.k` Integer denoting the number of decimal places expected for centrality parameter label. (Default: `2`).

`centrality.line.args`
A list of additional aesthetic arguments to be passed to the `geom_line` used to display the lines corresponding to the centrality parameter and test value.

`centrality.label.args`
A list of additional aesthetic arguments to be passed to the `geom_label` used to display the label corresponding to the centrality parameter and test value.

`ggplot.component`
A `ggplot` component to be added to the plot prepared by `ggstatsplot`. This argument is primarily helpful for `grouped_` variant of the current function. Default is `NULL`. The argument should be entered as a function.

`ggtheme` A function, `ggplot2` theme name. Default value is `ggplot2::theme_bw()`. Any of the `ggplot2` themes, or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`, etc.).

`ggstatsplot.layer`
Logical that decides whether `theme_ggstatsplot` theme elements are to be displayed along with the selected `ggtheme` (Default: `TRUE`). `theme_ggstatsplot` is an opinionated theme layer that override some aspects of the selected `ggtheme`.

`output` Character that describes what is to be returned: can be `"plot"` (default) or `"subtitle"` or `"caption"`. Setting this to `"subtitle"` will return the expression containing statistical results. If you have set `results.subtitle = FALSE`, then this will return a `NULL`. Setting this to `"caption"` will return the expression containing details about Bayes Factor analysis, but valid only when `type = "parametric"` and `bf.message = TRUE`, otherwise this will return a `NULL`. For functions `ggpiestats` and `ggbarstats`, setting `output = "proptest"` will return a dataframe containing results from proportion tests.

`...` Currently ignored.

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggdotplotstats.html

See Also

grouped_gghistostats, gghistostats, grouped_ggdotplotstats

Examples

```
# for reproducibility
set.seed(123)

# plot
ggdotplotstats(
  data = ggplot2::mpg,
  x = cty,
  y = manufacturer,
  conf.level = 0.99,
  test.value = 15,
  test.value.line = TRUE,
  test.line.labeller = TRUE,
  centrality.parameter = "median",
  centrality.k = 0,
  title = "Fuel economy data",
  xlab = "city miles per gallon",
  caption = substitute(
    paste(italic("Source"), ": EPA dataset on http://fueleconomy.gov")
  )
)
```

---

gghistostats *Histogram for distribution of a numeric variable*

Description

Histogram with statistical details from one-sample test included in the plot as a subtitle.

Usage

```
gghistostats(
  data,
  x,
  binwidth = NULL,
  bar.measure = "count",
  xlab = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  type = "parametric",
  test.value = 0,
```

```
  bf.prior = 0.707,
  bf.message = TRUE,
  effsize.type = "g",
  conf.level = 0.95,
  nboot = 100,
  k = 2L,
  ggtheme = ggplot2::theme_bw(),
  ggstatsplot.layer = TRUE,
  bar.fill = "grey50",
  results.subtitle = TRUE,
  test.k = 0,
  test.value.line = FALSE,
  test.value.line.args = list(size = 1),
  test.value.label.args = list(size = 3),
  centrality.parameter = "mean",
  centrality.k = 2,
  centrality.line.args = list(size = 1, color = "blue"),
  centrality.label.args = list(color = "blue", size = 3),
  normal.curve = FALSE,
  normal.curve.args = list(size = 3),
  ggplot.component = NULL,
  output = "plot",
  ...
)
```

Arguments

data	A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will **not** be accepted.
x	A numeric variable from the dataframe data.
binwidth	The width of the histogram bins. Can be specified as a numeric value, or a function that calculates width from x. The default is to use the max(x) -min(x) / sqrt(N). You should always check this value and explore multiple widths to find the best to illustrate the stories in your data.
bar.measure	Character describing what value needs to be represented as height in the bar chart. This can either be "count", which shows number of points in bin, or "density", which density of points in bin, scaled to integrate to 1, or "proportion", which shows relative frequencies of observations in each bin, or "mix", which shows *both* count and proportion in the same plot.
xlab	Labels for x and y axis variables. If NULL (default), variable names for x and y will be used.
title	The text for the plot title.
subtitle	The text for the plot subtitle. Will work only if results.subtitle = FALSE.
caption	The text for the plot caption.
type	Type of statistic expected ("parametric" or "nonparametric" or "robust" or "bayes").Corresponding abbreviations are also accepted: "p" (for parametric), "np" (nonparametric), "r" (robust), or "bf"resp.

`test.value` A number specifying the value of the null hypothesis (Default: `0`).

`bf.prior` A number between `0.5` and `2` (default `0.707`), the prior width to use in calculating Bayes factors.

`bf.message` Logical that decides whether to display Bayes Factor in favor of the *null* hypothesis. This argument is relevant only **for parametric test** (Default: `TRUE`).

`effsize.type` Type of effect size needed for *parametric* tests. The argument can be `"d"` (for Cohen's *d*) or `"g"` (for Hedge's *g*).

`conf.level` Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (`0.95`).

`nboot` Number of bootstrap samples for computing confidence interval for the effect size (Default: `100`).

`k` Number of digits after decimal point (should be an integer) (Default: `k = 2`).

`ggtheme` A function, `ggplot2` theme name. Default value is `ggplot2::theme_bw()`. Any of the `ggplot2` themes, or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`, etc.).

`ggstatsplot.layer` Logical that decides whether `theme_ggstatsplot` theme elements are to be displayed along with the selected `ggtheme` (Default: `TRUE`). `theme_ggstatsplot` is an opinionated theme layer that override some aspects of the selected `ggtheme`.

`bar.fill` Character input that decides which color will uniformly fill all the bars in the histogram (Default: `"grey50"`).

`results.subtitle` Decides whether the results of statistical tests are to be displayed as a subtitle (Default: `TRUE`). If set to `FALSE`, only the plot will be returned.

`test.k` Integer denoting the number of decimal places expected for `test.value` label. (Default: `0`).

`test.value.line` Logical that decides whether a line corresponding to the `test.value` should be superimposed on the plot.

`test.value.line.args` A list of additional aesthetic arguments to be passed to the `geom_line` used to display the lines corresponding to the centrality parameter and test value.

`test.value.label.args` A list of additional aesthetic arguments to be passed to the `geom_label` used to display the label corresponding to the centrality parameter and test value.

`centrality.parameter` Decides *which* measure of central tendency (`"mean"` or `"median"`) is to be displayed as a vertical line. To not show any parameter, set this to `"none"`.

`centrality.k` Integer denoting the number of decimal places expected for centrality parameter label. (Default: `2`).

`centrality.line.args` A list of additional aesthetic arguments to be passed to the `geom_line` used to display the lines corresponding to the centrality parameter and test value.

centrality.label.args
: A list of additional aesthetic arguments to be passed to the `geom_label` used to display the label corresponding to the centrality parameter and test value.

normal.curve
: A logical value that decides whether to super-impose a normal curve using `stats::dnorm(mean(x),sd(x))`. Default is `FALSE`.

normal.curve.args
: A list of additional aesthetic arguments to be passed to the normal curve.

ggplot.component
: A `ggplot` component to be added to the plot prepared by `ggstatsplot`. This argument is primarily helpful for `grouped_` variant of the current function. Default is `NULL`. The argument should be entered as a function.

output
: Character that describes what is to be returned: can be `"plot"` (default) or `"subtitle"` or `"caption"`. Setting this to `"subtitle"` will return the expression containing statistical results. If you have set `results.subtitle = FALSE`, then this will return a `NULL`. Setting this to `"caption"` will return the expression containing details about Bayes Factor analysis, but valid only when `type = "parametric"` and `bf.message = TRUE`, otherwise this will return a `NULL`. For functions `ggpiestats` and `ggbarstats`, setting `output = "proptest"` will return a dataframe containing results from proportion tests.

...
: Currently ignored.

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/gghistostats.html

See Also

grouped_gghistostats, ggdotplotstats, grouped_ggdotplotstats

Examples

```
# most basic function call with the defaults
# this is the **only** function where data argument can be `NULL`
ggstatsplot::gghistostats(
  data = ToothGrowth,
  x = len,
  xlab = "Tooth length",
  centrality.parameter = "median"
)

# a detailed function call
ggstatsplot::gghistostats(
  data = iris,
  x = Sepal.Length,
  bar.measure = "mix",
  type = "p",
  caption = substitute(paste(italic("Note"), ": Iris dataset by Fisher.")),
  bf.prior = 0.8,
  test.value = 3,
```

```
  test.value.line = TRUE,
  binwidth = 0.10,
  bar.fill = "grey50"
)
```

---

ggpiestats *Pie charts with statistical tests*

Description

Pie charts for categorical data with statistical details included in the plot as a subtitle.

Usage

```
ggpiestats(
  data,
  main,
  condition = NULL,
  counts = NULL,
  ratio = NULL,
  paired = FALSE,
  results.subtitle = TRUE,
  label = "percentage",
  label.args = list(direction = "both"),
  label.repel = FALSE,
  conf.level = 0.95,
  nboot = 100L,
  k = 2L,
  proportion.test = TRUE,
  perc.k = 0,
  bf.message = TRUE,
  sampling.plan = "indepMulti",
  fixed.margin = "rows",
  prior.concentration = 1,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  legend.title = NULL,
  ggtheme = ggplot2::theme_bw(),
  ggstatsplot.layer = TRUE,
  package = "RColorBrewer",
  palette = "Dark2",
  ggplot.component = NULL,
  output = "plot",
  x = NULL,
  y = NULL,
  ...
)
```

Arguments

data	A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will **not** be accepted.
counts	A string naming a variable in data containing counts, or NULL if each row represents a single observation (Default).
ratio	A vector of proportions: the expected proportions for the proportion test (should sum to 1). Default is NULL, which means the null is equal theoretical proportions across the levels of the nominal variable. This means if there are two levels this will be ratio = c(0.5,0.5) or if there are four levels this will be ratio = c(0.25,0.25,0.25,0.25), etc.
paired	Logical indicating whether data came from a within-subjects or repeated measures design study (Default: FALSE). If TRUE, McNemar's test subtitle will be returned. If FALSE, Pearson's chi-square test will be returned.
results.subtitle	Decides whether the results of statistical tests are to be displayed as a subtitle (Default: TRUE). If set to FALSE, only the plot will be returned.
label	Character decides what information needs to be displayed on the label in each pie slice. Possible options are "percentage" (default), "counts", "both".
label.args	Additional aesthetic arguments that will be passed to geom_label.
label.repel	Whether labels should be repelled using ggrepel package. This can be helpful in case the labels are overlapping.
conf.level	Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (0.95).
nboot	Number of bootstrap samples for computing confidence interval for the effect size (Default: 100).
k	Number of digits after decimal point (should be an integer) (Default: k = 2).
proportion.test	Decides whether proportion test for main variable is to be carried out for each level of condition (Default: TRUE).
perc.k	Numeric that decides number of decimal places for percentage labels (Default: 0).
bf.message	Logical that decides whether to display Bayes Factor in favor of the *null* hypothesis. This argument is relevant only **for parametric test** (Default: TRUE).
sampling.plan	Character describing the sampling plan. Possible options are "indepMulti" (independent multinomial; default), "poisson", "jointMulti" (joint multinomial), "hypergeom" (hypergeometric). For more, see ?BayesFactor::contingencyTableBF().
fixed.margin	For the independent multinomial sampling plan, which margin is fixed ("rows" or "cols"). Defaults to "rows".
prior.concentration	Specifies the prior concentration parameter, set to 1 by default. It indexes the expected deviation from the null hypothesis under the alternative, and corresponds to Gunel and Dickey's (1974) "a" parameter.
title	The text for the plot title.

subtitle	The text for the plot subtitle. Will work only if results.subtitle = FALSE.
caption	The text for the plot caption.
legend.title	Title text for the legend.
ggtheme	A function, ggplot2 theme name. Default value is ggplot2::theme_bw(). Any of the ggplot2 themes, or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.).
ggstatsplot.layer	Logical that decides whether theme_ggstatsplot theme elements are to be displayed along with the selected ggtheme (Default: TRUE). theme_ggstatsplot is an opinionated theme layer that override some aspects of the selected ggtheme.
package	Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running View(paletteer::palettes_d_names).
palette	Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running View(paletteer::palettes_d_names).
ggplot.component	A ggplot component to be added to the plot prepared by ggstatsplot. This argument is primarily helpful for grouped_ variant of the current function. Default is NULL. The argument should be entered as a function.
output	Character that describes what is to be returned: can be "plot" (default) or "subtitle" or "caption". Setting this to "subtitle" will return the expression containing statistical results. If you have set results.subtitle = FALSE, then this will return a NULL. Setting this to "caption" will return the expression containing details about Bayes Factor analysis, but valid only when type = "parametric" and bf.message = TRUE, otherwise this will return a NULL. For functions ggpiestats and ggbarstats, setting output = "proptest" will return a dataframe containing results from proportion tests.
x, main	The variable to use as the **rows** in the contingency table.
y, condition	The variable to use as the **columns** in the contingency table. Default is NULL. If NULL, one-sample proportion test (a goodness of fit test) will be run for the x variable. Otherwise an appropriate association test will be run. This argument can not be NULL for ggbarstats function.
...	Currently ignored.

Value

Unlike a number of statistical softwares, ggstatsplot doesn't provide the option for Yates' correction for the Pearson's chi-squared statistic. This is due to compelling amount of Monte-Carlo simulation research which suggests that the Yates' correction is overly conservative, even in small sample sizes. As such it is recommended that it should not ever be applied in practice (Camilli & Hopkins, 1978, 1979; Feinberg, 1980; Larntz, 1978; Thompson, 1988).

For more about how the effect size measures and their confidence intervals are computed, see ?rcompanion::cohenG, ?rcompanion::cramerV, and ?rcompanion::cramerVFit.

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggpiestats.html

See Also

grouped_ggpiestats, ggbarstats, grouped_ggbarstats

Examples

```
# for reproducibility
set.seed(123)

# one sample goodness of fit proportion test
ggstatsplot::ggpiestats(ggplot2::msleep, vore)

# association test (or contingency table analysis)
ggstatsplot::ggpiestats(
  data = mtcars,
  x = vs,
  y = cyl,
  legend.title = "Engine"
)
```

---

ggscatterstats *Scatterplot with marginal distributions and statistical results*

Description

Scatterplots from ggplot2 combined with marginal histograms/boxplots/density plots with statistical details added as a subtitle.

Usage

```
ggscatterstats(
  data,
  x,
  y,
  type = "parametric",
  conf.level = 0.95,
  bf.prior = 0.707,
  bf.message = TRUE,
  label.var = NULL,
  label.expression = NULL,
  point.label.args = list(size = 3),
  formula = y ~ x,
  smooth.line.args = list(size = 1.5, color = "blue"),
  method = "lm",
  method.args = list(),
  point.args = list(size = 3, alpha = 0.4),
  point.width.jitter = 0,
  point.height.jitter = 0,
  marginal = TRUE,
```

```
  marginal.type = "histogram",
  margins = "both",
  marginal.size = 5,
  xfill = "#009E73",
  yfill = "#D55E00",
  xparams = list(fill = xfill),
  yparams = list(fill = yfill),
  centrality.parameter = "none",
  centrality.label.args = list(size = 3),
  vline.args = list(color = xfill, size = 1, linetype = "dashed"),
  hline.args = list(color = yfill, size = 1, linetype = "dashed"),
  results.subtitle = TRUE,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  beta = 0.1,
  k = 2L,
  ggtheme = ggplot2::theme_bw(),
  ggstatsplot.layer = TRUE,
  ggplot.component = NULL,
  output = "plot",
  ...
)
```

Arguments

`data`	A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will **not** be accepted.
`x`	The column in `data` containing the explanatory variable to be plotted on the `x`-axis. Can be entered either as a character string (e.g., `"x"`) or as a bare expression (e.g, `x`).
`y`	The column in `data` containing the response (outcome) variable to be plotted on the `y`-axis. Can be entered either as a character string (e.g., `"y"`) or as a bare expression (e.g, `y`).
`type`	Type of association between paired samples required (`""parametric"`: Pearson's product moment correlation coefficient" or `""nonparametric"`: Spearman's rho" or `""robust"`: percentage bend correlation coefficient" or `""bayes"`: Bayes Factor for Pearson's *r*"). Corresponding abbreviations are also accepted: `"p"` (for parametric/pearson), `"np"` (nonparametric/spearman), `"r"` (robust), `"bf"` (for bayes factor), resp.
`conf.level`	Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (`0.95`).
`bf.prior`	A number between `0.5` and `2` (default `0.707`), the prior width to use in calculating Bayes factors.
`bf.message`	Logical that decides whether to display Bayes Factor in favor of the *null* hypothesis. This argument is relevant only **for parametric test** (Default: `TRUE`).

`label.var` Variable to use for points labels. Can be entered either as a character string (e.g., `"var1"`) or as a bare expression (e.g, `var1`).

`label.expression`
An expression evaluating to a logical vector that determines the subset of data points to label. This argument can be entered either as a character string (e.g., `"y < 4 & z < 20"`) or as a bare expression (e.g., `y < 4 & z < 20`).

`point.label.args`
A list of additional aesthetic arguments to be passed to `ggrepel::geom_label_repel` geom used to display the labels.

`formula` Formula to use in smoothing function, eg. `y ~ x`, `y ~ poly(x,2)`, `y ~ log(x)`. `NULL` by default, in which case `method = NULL` implies `formula = y ~ x` when there are fewer than 1,000 observations and `formula = y ~ s(x,bs = "cs")` otherwise.

`smooth.line.args`
A list of additional aesthetic arguments to be passed to `ggplot2::geom_smooth` geom used to display the regression line.

`method` Smoothing method (function) to use, accepts either `NULL` or a character vector, e.g. `"lm"`, `"glm"`, `"gam"`, `"loess"` or a function, e.g. `MASS::rlm` or `mgcv::gam`, `stats::lm`, or `stats::loess`. `"auto"` is also accepted for backwards compatibility. It is equivalent to `NULL`.

For `method = NULL` the smoothing method is chosen based on the size of the largest group (across all panels). `stats::loess()` is used for less than 1,000 observations; otherwise `mgcv::gam()` is used with `formula = y ~ s(x,bs = "cs")` with `method = "REML"`. Somewhat anecdotally, `loess` gives a better appearance, but is $O(N^2)$ in memory, so does not work for larger datasets.

If you have fewer than 1,000 observations but want to use the same `gam()` model that `method = NULL` would use, then set method = "gam", formula = y ~ s(x, bs = "cs").

`method.args` List of additional arguments passed on to the modelling function defined by `method`.

`point.args` A list of additional aesthetic arguments to be passed to `ggplot2::geom_point` geom used to display the raw data points.

`point.width.jitter, point.height.jitter`
Degree of jitter in `x` and `y` direction, respectively. Defaults to `0` (0%) of the resolution of the data. Note that the jitter should not be specified in the `point.args` because this information will be passed to two different geoms: one displaying the points and the other displaying the labels for these points.

`marginal` Decides whether `ggExtra::ggMarginal()` plots will be displayed; the default is `TRUE`.

`marginal.type` Type of marginal distribution to be plotted on the axes (`"histogram"`, `"boxplot"`, `"density"`, `"violin"`, `"densigram"`).

`margins` Along which margins to show the plots. One of: [both, x, y].

`marginal.size` Integer describing the relative size of the marginal plots compared to the main plot. A size of 5 means that the main plot is 5x wider and 5x taller than the marginal plots.

`xfill, yfill` Character describing color fill for `x` and `y` axes marginal distributions (default: `"#009E73"` (for `x`) and `"#D55E00"` (for `y`)). The same colors will also be used for the lines denoting centrality parameters if `centrality.parameter` argument is set to `TRUE`. Note that the defaults are colorblind-friendly.

`xparams` List of extra parameters to use only for the marginal plot along the `x` axis.

`yparams` List of extra parameters to use only for the marginal plot along the `y` axis.

`centrality.parameter`
Decides *which* measure of central tendency (`"mean"` or `"median"`) is to be displayed as vertical (for `x`) and horizontal (for `y`) lines. Note that mean values corresponds to arithmetic mean and not geometric mean.

`centrality.label.args`
A list of additional aesthetic arguments to be passed to the `geom_label` used to display the label corresponding to the centrality parameter and test value.

`vline.args, hline.args`
A list of additional aesthetic arguments to be passed to `ggplot2::geom_vline` and `ggplot2::geom_hline` geoms used to display the centrality parameter labels on vertical and horizontal lines.

`results.subtitle`
Decides whether the results of statistical tests are to be displayed as a subtitle (Default: `TRUE`). If set to `FALSE`, only the plot will be returned.

`xlab` Labels for `x` and `y` axis variables. If `NULL` (default), variable names for `x` and `y` will be used.

`ylab` Labels for `x` and `y` axis variables. If `NULL` (default), variable names for `x` and `y` will be used.

`title` The text for the plot title.

`subtitle` The text for the plot subtitle. Will work only if `results.subtitle = FALSE`.

`caption` The text for the plot caption.

`beta` bending constant (Default: `0.1`). For more, see `WRS2::pbcor()`.

`k` Number of digits after decimal point (should be an integer) (Default: `k = 2L`).

`ggtheme` A function, `ggplot2` theme name. Default value is `ggplot2::theme_bw()`. Any of the `ggplot2` themes, or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`, etc.).

`ggstatsplot.layer`
Logical that decides whether `theme_ggstatsplot` theme elements are to be displayed along with the selected `ggtheme` (Default: `TRUE`). `theme_ggstatsplot` is an opinionated theme layer that override some aspects of the selected `ggtheme`.

`ggplot.component`
A `ggplot` component to be added to the plot prepared by `ggstatsplot`. This argument is primarily helpful for `grouped_` variant of the current function. Default is `NULL`. The argument should be entered as a function.

`output` Character that describes what is to be returned: can be `"plot"` (default) or `"subtitle"` or `"caption"`. Setting this to `"subtitle"` will return the expression containing statistical results. If you have set `results.subtitle = FALSE`,

then this will return a NULL. Setting this to "caption" will return the expression containing details about Bayes Factor analysis, but valid only when type = "parametric" and bf.message = TRUE, otherwise this will return a NULL. For functions ggpiestats and ggbarstats, setting output = "proptest" will return a dataframe containing results from proportion tests.

... Currently ignored.

Note

- If you set marginal = TRUE, the resulting plot can't be further modified with ggplot2 functions since it is no longer a ggplot object. In case you want a ggplot object, set marginal = FALSE. Also have a look at the ggplot.component argument.
- The plot uses ggrepel::geom_label_repel to attempt to keep labels from over-lapping to the largest degree possible. As a consequence plot times will slow down massively (and the plot file will grow in size) if you have a lot of labels that overlap.

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggscatterstats.html

See Also

grouped_ggscatterstats, ggcorrmat, grouped_ggcorrmat

Examples

```
# to get reproducible results from bootstrapping
set.seed(123)
library(ggstatsplot)

# creating dataframe with rownames converted to a new column
mtcars_new <- as_tibble(mtcars, rownames = "car")

# simple function call with the defaults
ggstatsplot::ggscatterstats(
  data = mtcars_new,
  x = wt,
  y = mpg,
  type = "np",
  label.var = car,
  label.expression = wt < 4 & mpg < 20,
  centrality.parameter = "median"
)
```

`ggwithinstats` *Box/Violin plots for group or condition comparisons in within-subjects (or repeated measures) designs.*

Description

A combination of box and violin plots along with raw (unjittered) data points for within-subjects designs with statistical details included in the plot as a subtitle.

Usage

```
ggwithinstats(
  data,
  x,
  y,
  type = "parametric",
  pairwise.comparisons = TRUE,
  pairwise.display = "significant",
  p.adjust.method = "holm",
  effsize.type = "unbiased",
  partial = TRUE,
  bf.prior = 0.707,
  bf.message = TRUE,
  sphericity.correction = TRUE,
  results.subtitle = TRUE,
  xlab = NULL,
  ylab = NULL,
  caption = NULL,
  title = NULL,
  subtitle = NULL,
  sample.size.label = TRUE,
  k = 2,
  conf.level = 0.95,
  nboot = 100,
  tr = 0.1,
  mean.plotting = TRUE,
  mean.ci = FALSE,
  mean.point.args = list(size = 5, color = "darkred"),
  mean.label.args = list(size = 3),
  point.path = TRUE,
  point.path.args = list(alpha = 0.5, linetype = "dashed"),
  mean.path = TRUE,
  mean.path.args = list(color = "red", size = 1, alpha = 0.5),
  notch = FALSE,
  notchwidth = 0.5,
  outlier.tagging = FALSE,
  outlier.label = NULL,
```

```
  outlier.coef = 1.5,
  outlier.label.args = list(),
  outlier.point.args = list(),
  violin.args = list(width = 0.5, alpha = 0.2),
  ggsignif.args = list(textsize = 3, tip_length = 0.01),
  ggtheme = ggplot2::theme_bw(),
  ggstatsplot.layer = TRUE,
  package = "RColorBrewer",
  palette = "Dark2",
  ggplot.component = NULL,
  output = "plot",
  ...
)
```

Arguments

`data`	A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will **not** be accepted.
`x`	The grouping variable from the dataframe `data`.
`y`	The response (a.k.a. outcome or dependent) variable from the dataframe `data`.
`type`	Type of statistic expected (`"parametric"` or `"nonparametric"` or `"robust"` or `"bayes"`).Corresponding abbreviations are also accepted: `"p"` (for parametric), `"np"` (nonparametric), `"r"` (robust), or `"bf"`resp.
`pairwise.comparisons`	Logical that decides whether pairwise comparisons are to be displayed (default: `FALSE`). Please note that only **significant** comparisons will be shown by default. To change this behavior, select appropriate option with `pairwise.display` argument. The pairwise comparison dataframes are prepared using the `pairwiseComparisons::pairwise_` function. For more details about pairwise comparisons, see the documentation for that function.
`pairwise.display`	Decides which pairwise comparisons to display. Available options are `"significant"` (abbreviation accepted: `"s"`) or `"non-significant"` (abbreviation accepted: `"ns"`) or `"everything"/"all"`. The default is `"significant"`. You can use this argument to make sure that your plot is not uber-cluttered when you have multiple groups being compared and scores of pairwise comparisons being displayed.
`p.adjust.method`	Adjustment method for *p*-values for multiple comparisons. Possible methods are: `"holm"` (default), `"hochberg"`, `"hommel"`, `"bonferroni"`, `"BH"`, `"BY"`, `"fdr"`, `"none"`.
`effsize.type`	Type of effect size needed for *parametric* tests. The argument can be `"biased"` (equivalent to `"d"` for Cohen's *d* for **t-test**; `"partial_eta"` for partial eta-squared for **anova**) or `"unbiased"` (equivalent to `"g"` Hedge's *g* for **t-test**; `"partial_omega"` for partial omega-squared for **anova**)).
`partial`	If `TRUE`, return partial indices.

`bf.prior` A number between `0.5` and `2` (default `0.707`), the prior width to use in calculating Bayes factors.

`bf.message` Logical that decides whether to display Bayes Factor in favor of the *null* hypothesis. This argument is relevant only **for parametric test** (Default: `TRUE`).

`sphericity.correction`
Logical that decides whether to apply correction to account for violation of sphericity in a repeated measures design ANOVA (Default: `TRUE`).

`results.subtitle`
Decides whether the results of statistical tests are to be displayed as a subtitle (Default: `TRUE`). If set to `FALSE`, only the plot will be returned.

`xlab` Labels for `x` and `y` axis variables. If `NULL` (default), variable names for `x` and `y` will be used.

`ylab` Labels for `x` and `y` axis variables. If `NULL` (default), variable names for `x` and `y` will be used.

`caption` The text for the plot caption.

`title` The text for the plot title.

`subtitle` The text for the plot subtitle. Will work only if `results.subtitle = FALSE`.

`sample.size.label`
Logical that decides whether sample size information should be displayed for each level of the grouping variable `x` (Default: `TRUE`).

`k` Number of digits after decimal point (should be an integer) (Default: `k = 2`).

`conf.level` Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (`0.95`).

`nboot` Number of bootstrap samples for computing confidence interval for the effect size (Default: `100`).

`tr` Trim level for the mean when carrying out `robust` tests. If you get error stating "Standard error cannot be computed because of Winsorized variance of 0 (e.g., due to ties). Try to decrease the trimming level.", try to play around with the value of `tr`, which is by default set to `0.1`. Lowering the value might help.

`mean.plotting` Logical that decides whether mean is to be highlighted and its value to be displayed (Default: `TRUE`).

`mean.ci` Logical that decides whether 95% confidence interval for mean is to be displayed (Default: `FALSE`).

`mean.point.args`
A list of additional aesthetic arguments to be passed to `ggplot2::geom_point` and `ggrepel::geom_label_repel` geoms involved mean value plotting.

`mean.label.args`
A list of additional aesthetic arguments to be passed to `ggplot2::geom_point` and `ggrepel::geom_label_repel` geoms involved mean value plotting.

`point.path, mean.path`
Logical that decides whether individual data points and means, respectively, should be connected using `geom_path`. Both default to `TRUE`. Note that `point.path` argument is relevant only when there are two groups (i.e., in case of a *t*-test). In case of large number of data points, it is advisable to set `point.path = FALSE` as these lines can overwhelm the plot.

`mean.path.args, point.path.args`
: A list of additional aesthetic arguments passed on to `geom_path` connecting raw data points and mean points.

`notch`
: A logical. If `FALSE` (default), a standard box plot will be displayed. If `TRUE`, a notched box plot will be used. Notches are used to compare groups; if the notches of two boxes do not overlap, this suggests that the medians are significantly different. In a notched box plot, the notches extend `1.58 * IQR / sqrt(n)`. This gives a roughly 95% confidence interval for comparing medians. IQR: Inter-Quartile Range.

`notchwidth`
: For a notched box plot, width of the notch relative to the body (default `0.5`).

`outlier.tagging`
: Decides whether outliers should be tagged (Default: `FALSE`).

`outlier.label`
: Label to put on the outliers that have been tagged. This **can't** be the same as `x` argument.

`outlier.coef`
: Coefficient for outlier detection using Tukey's method. With Tukey's method, outliers are below (1st Quartile) or above (3rd Quartile) `outlier.coef` times the Inter-Quartile Range (IQR) (Default: `1.5`).

`outlier.label.args`
: A list of additional aesthetic arguments to be passed to `ggplot2::geom_point` and `ggrepel::geom_label_repel` geoms involved outlier value plotting.

`outlier.point.args`
: A list of additional aesthetic arguments to be passed to `ggplot2::geom_point` and `ggrepel::geom_label_repel` geoms involved outlier value plotting.

`violin.args`
: A list of additional aesthetic arguments to be passed to the `geom_violin`.

`ggsignif.args`
: A list of additional aesthetic arguments to be passed to `ggsignif::geom_signif`.

`ggtheme`
: A function, `ggplot2` theme name. Default value is `ggplot2::theme_bw()`. Any of the `ggplot2` themes, or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`, etc.).

`ggstatsplot.layer`
: Logical that decides whether `theme_ggstatsplot` theme elements are to be displayed along with the selected `ggtheme` (Default: `TRUE`). `theme_ggstatsplot` is an opinionated theme layer that override some aspects of the selected `ggtheme`.

`package`
: Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running `View(paletteer::palettes_d_names)`.

`palette`
: Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running `View(paletteer::palettes_d_names)`.

`ggplot.component`
: A `ggplot` component to be added to the plot prepared by `ggstatsplot`. This argument is primarily helpful for `grouped_` variant of the current function. Default is `NULL`. The argument should be entered as a function.

`output`
: Character that describes what is to be returned: can be `"plot"` (default) or `"subtitle"` or `"caption"`. Setting this to `"subtitle"` will return the expression containing statistical results. If you have set `results.subtitle = FALSE`,

then this will return a NULL. Setting this to "caption" will return the expression containing details about Bayes Factor analysis, but valid only when type = "parametric" and bf.message = TRUE, otherwise this will return a NULL. For functions ggpiestats and ggbarstats, setting output = "proptest" will return a dataframe containing results from proportion tests.

... Currently ignored.

Details

For more about how the effect size measures (for nonparametric tests) and their confidence intervals are computed, see ?rcompanion::wilcoxonPairedR.

For independent measures designs, use ggbetweenstats.

See Also

grouped_ggbetweenstats, ggbetweenstats, grouped_ggwithinstats

Examples

```
# setup
set.seed(123)
library(ggstatsplot)

# two groups (*t*-test)
ggstatsplot::ggwithinstats(
  data = VR_dilemma,
  x = modality,
  y = score,
  xlab = "Presentation modality",
  ylab = "Proportion of utilitarian decisions"
)

# more than two groups (anova)
library(WRS2)

ggstatsplot::ggwithinstats(
  data = as_tibble(WineTasting),
  x = Wine,
  y = Taste,
  type = "np",
  conf.level = 0.99,
  pairwise.comparisons = TRUE,
  outlier.tagging = TRUE,
  outlier.label = Taster
)
```

`grouped_ggbarstats` *Grouped bar (column) charts with statistical tests*

Description

Helper function for `ggstatsplot::ggbarstats` to apply this function across multiple levels of a given factor and combining the resulting plots using `ggstatsplot::combine_plots`.

Usage

```
grouped_ggbarstats(
  data,
  main,
  condition,
  counts = NULL,
  grouping.var,
  title.prefix = NULL,
  output = "plot",
  x = NULL,
  y = NULL,
  ...,
  plotgrid.args = list(),
  title.text = NULL,
  title.args = list(size = 16, fontface = "bold"),
  caption.text = NULL,
  caption.args = list(size = 10),
  sub.text = NULL,
  sub.args = list(size = 12)
)
```

Arguments

`data`	A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will **not** be accepted.
`main`	The variable to use as the **rows** in the contingency table.
`condition`	The variable to use as the **columns** in the contingency table. Default is `NULL`. If `NULL`, one-sample proportion test (a goodness of fit test) will be run for the `x` variable. Otherwise an appropriate association test will be run. This argument can not be `NULL` for `ggbarstats` function.
`counts`	A string naming a variable in data containing counts, or `NULL` if each row represents a single observation (Default).
`grouping.var`	A single grouping variable (can be entered either as a bare name `x` or as a string `"x"`).
`title.prefix`	Character string specifying the prefix text for the fixed plot title (name of each factor level) (Default: `NULL`). If `NULL`, the variable name entered for `grouping.var` will be used.

output — Character that describes what is to be returned: can be "plot" (default) or "subtitle" or "caption". Setting this to "subtitle" will return the expression containing statistical results. If you have set results.subtitle = FALSE, then this will return a NULL. Setting this to "caption" will return the expression containing details about Bayes Factor analysis, but valid only when type = "parametric" and bf.message = TRUE, otherwise this will return a NULL. For functions ggpiestats and ggbarstats, setting output = "proptest" will return a dataframe containing results from proportion tests.

x — The variable to use as the **rows** in the contingency table.

y — The variable to use as the **columns** in the contingency table. Default is NULL. If NULL, one-sample proportion test (a goodness of fit test) will be run for the x variable. Otherwise an appropriate association test will be run. This argument can not be NULL for ggbarstats function.

... — Arguments passed on to ggbarstats

xlab Custom text for the x axis label (Default: NULL, which will cause the x axis label to be the x variable).

ylab Custom text for the y axis label (Default: NULL).

proportion.test Decides whether proportion test for main variable is to be carried out for each level of y (Default: TRUE).

label Character decides what information needs to be displayed on the label in each pie slice. Possible options are "percentage" (default), "counts", "both".

sample.size.label Logical that decides whether sample size information should be displayed for each level of the grouping variable y (Default: TRUE).

ratio A vector of proportions: the expected proportions for the proportion test (should sum to 1). Default is NULL, which means the null is equal theoretical proportions across the levels of the nominal variable. This means if there are two levels this will be ratio = c(0.5,0.5) or if there are four levels this will be ratio = c(0.25,0.25,0.25,0.25), etc.

paired Logical indicating whether data came from a within-subjects or repeated measures design study (Default: FALSE). If TRUE, McNemar's test subtitle will be returned. If FALSE, Pearson's chi-square test will be returned.

results.subtitle Decides whether the results of statistical tests are to be displayed as a subtitle (Default: TRUE). If set to FALSE, only the plot will be returned.

label.args Additional aesthetic arguments that will be passed to geom_label.

conf.level Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (0.95).

nboot Number of bootstrap samples for computing confidence interval for the effect size (Default: 100).

k Number of digits after decimal point (should be an integer) (Default: k = 2).

perc.k Numeric that decides number of decimal places for percentage labels (Default: 0).

bf.message Logical that decides whether to display Bayes Factor in favor of the *null* hypothesis. This argument is relevant only **for parametric test** (Default: TRUE).

sampling.plan Character describing the sampling plan. Possible options are "indepMulti" (independent multinomial; default), "poisson", "jointMulti" (joint multinomial), "hypergeom" (hypergeometric). For more, see ?BayesFactor::contingencyTa

fixed.margin For the independent multinomial sampling plan, which margin is fixed ("rows" or "cols"). Defaults to "rows".

prior.concentration Specifies the prior concentration parameter, set to 1 by default. It indexes the expected deviation from the null hypothesis under the alternative, and corresponds to Gunel and Dickey's (1974) "a" parameter.

subtitle The text for the plot subtitle. Will work only if results.subtitle = FALSE.

caption The text for the plot caption.

legend.title Title text for the legend.

ggtheme A function, ggplot2 theme name. Default value is ggplot2::theme_bw(). Any of the ggplot2 themes, or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.).

ggstatsplot.layer Logical that decides whether theme_ggstatsplot theme elements are to be displayed along with the selected ggtheme (Default: TRUE). theme_ggstatsplot is an opinionated theme layer that override some aspects of the selected ggtheme.

package Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running View(paletteer::palettes_d_nam

palette Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running View(paletteer::palettes_d_nam

ggplot.component A ggplot component to be added to the plot prepared by ggstatsplot. This argument is primarily helpful for grouped_ variant of the current function. Default is NULL. The argument should be entered as a function.

plotgrid.args A list of additional arguments to cowplot::plot_grid.

title.text String or plotmath expression to be drawn as title for the *combined plot*.

title.args A list of additional arguments provided to title, caption and sub, resp.

caption.text String or plotmath expression to be drawn as the caption for the *combined plot*.

caption.args A list of additional arguments provided to title, caption and sub, resp.

sub.text The label with which the *combined plot* should be annotated. Can be a plotmath expression.

sub.args A list of additional arguments provided to title, caption and sub, resp.

Value

Unlike a number of statistical softwares, ggstatsplot doesn't provide the option for Yates' correction for the Pearson's chi-squared statistic. This is due to compelling amount of Monte-Carlo simulation research which suggests that the Yates' correction is overly conservative, even in small

sample sizes. As such it is recommended that it should not ever be applied in practice (Camilli & Hopkins, 1978, 1979; Feinberg, 1980; Larntz, 1978; Thompson, 1988).

For more about how the effect size measures and their confidence intervals are computed, see `?rcompanion::cohenG`, `?rcompanion::cramerV`, and `?rcompanion::cramerVFit`.

See Also

ggbarstats, ggpiestats, grouped_ggpiestats

Examples

```
# for reproducibility
set.seed(123)

# let's create a smaller dataframe
diamonds_short <- ggplot2::diamonds %>%
  dplyr::filter(.data = ., cut %in% c("Very Good", "Ideal")) %>%
  dplyr::filter(.data = ., clarity %in% c("SI1", "SI2", "VS1", "VS2")) %>%
  dplyr::sample_frac(tbl = ., size = 0.05)

# plot
ggstatsplot::grouped_ggbarstats(
  data = diamonds_short,
  x = color,
  y = clarity,
  grouping.var = cut,
  title.prefix = "Quality",
  bar.label = "both",
  plotgrid.args = list(nrow = 2)
)
```

---

grouped_ggbetweenstats

Violin plots for group or condition comparisons in between-subjects designs repeated across all levels of a grouping variable.

Description

A combined plot of comparison plot created for levels of a grouping variable.

Usage

```
grouped_ggbetweenstats(
  data,
  x,
  y,
  grouping.var,
```

```
  outlier.label = NULL,
  title.prefix = NULL,
  output = "plot",
  ...,
  plotgrid.args = list(),
  title.text = NULL,
  title.args = list(size = 16, fontface = "bold"),
  caption.text = NULL,
  caption.args = list(size = 10),
  sub.text = NULL,
  sub.args = list(size = 12)
)
```

Arguments

`data`	A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will **not** be accepted.
`x`	The grouping variable from the dataframe `data`.
`y`	The response (a.k.a. outcome or dependent) variable from the dataframe `data`.
`grouping.var`	A single grouping variable (can be entered either as a bare name `x` or as a string `"x"`).
`outlier.label`	Label to put on the outliers that have been tagged. This **can't** be the same as `x` argument.
`title.prefix`	Character string specifying the prefix text for the fixed plot title (name of each factor level) (Default: `NULL`). If `NULL`, the variable name entered for `grouping.var` will be used.
`output`	Character that describes what is to be returned: can be `"plot"` (default) or `"subtitle"` or `"caption"`. Setting this to `"subtitle"` will return the expression containing statistical results. If you have set `results.subtitle = FALSE`, then this will return a `NULL`. Setting this to `"caption"` will return the expression containing details about Bayes Factor analysis, but valid only when `type = "parametric"` and `bf.message = TRUE`, otherwise this will return a `NULL`. For functions `ggpiestats` and `ggbarstats`, setting `output = "proptest"` will return a dataframe containing results from proportion tests.
`...`	Arguments passed on to `ggbetweenstats` `plot.type` Character describing the *type* of plot. Currently supported plots are `"box"` (for pure boxplots), `"violin"` (for pure violin plots), and `"boxviolin"` (for a combination of box and violin plots; default). `xlab` Labels for `x` and `y` axis variables. If `NULL` (default), variable names for `x` and `y` will be used. `ylab` Labels for `x` and `y` axis variables. If `NULL` (default), variable names for `x` and `y` will be used. `pairwise.comparisons` Logical that decides whether pairwise comparisons are to be displayed (default: `FALSE`). Please note that only **significant** comparisons will be shown by default. To change this behavior, select appropriate option with `pairwise.display` argument. The pairwise comparison

dataframes are prepared using the `pairwiseComparisons::pairwise_comparisons` function. For more details about pairwise comparisons, see the documentation for that function.

`p.adjust.method` Adjustment method for *p*-values for multiple comparisons. Possible methods are: `"holm"` (default), `"hochberg"`, `"hommel"`, `"bonferroni"`, `"BH"`, `"BY"`, `"fdr"`, `"none"`.

`pairwise.display` Decides which pairwise comparisons to display. Available options are `"significant"` (abbreviation accepted: `"s"`) or `"non-significant"` (abbreviation accepted: `"ns"`) or `"everything"/"all"`. The default is `"significant"`. You can use this argument to make sure that your plot is not uber-cluttered when you have multiple groups being compared and scores of pairwise comparisons being displayed.

`bf.prior` A number between `0.5` and `2` (default `0.707`), the prior width to use in calculating Bayes factors.

`bf.message` Logical that decides whether to display Bayes Factor in favor of the *null* hypothesis. This argument is relevant only **for parametric test** (Default: `TRUE`).

`results.subtitle` Decides whether the results of statistical tests are to be displayed as a subtitle (Default: `TRUE`). If set to `FALSE`, only the plot will be returned.

`subtitle` The text for the plot subtitle. Will work only if `results.subtitle = FALSE`.

`caption` The text for the plot caption.

`sample.size.label` Logical that decides whether sample size information should be displayed for each level of the grouping variable `x` (Default: `TRUE`).

`notch` A logical. If `FALSE` (default), a standard box plot will be displayed. If `TRUE`, a notched box plot will be used. Notches are used to compare groups; if the notches of two boxes do not overlap, this suggests that the medians are significantly different. In a notched box plot, the notches extend `1.58 * IQR / sqrt(n)`. This gives a roughly 95% confidence interval for comparing medians. IQR: Inter-Quartile Range.

`notchwidth` For a notched box plot, width of the notch relative to the body (default `0.5`).

`outlier.color` Default aesthetics for outliers (Default: `"black"`).

`outlier.tagging` Decides whether outliers should be tagged (Default: `FALSE`).

`outlier.shape` 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.point.args` A list of additional aesthetic arguments to be passed to `ggplot2::geom_point` and `ggrepel::geom_label_repel` geoms involved outlier value plotting.

`outlier.label.args` A list of additional aesthetic arguments to be passed to `ggplot2::geom_point` and `ggrepel::geom_label_repel` geoms involved outlier value plotting.

`outlier.coef` Coefficient for outlier detection using Tukey's method. With Tukey's method, outliers are below (1st Quartile) or above (3rd Quartile) `outlier.coef` times the Inter-Quartile Range (IQR) (Default: `1.5`).

`mean.plotting` Logical that decides whether mean is to be highlighted and its value to be displayed (Default: `TRUE`).

`mean.ci` Logical that decides whether 95% confidence interval for mean is to be displayed (Default: `FALSE`).

`point.args` A list of additional aesthetic arguments to be passed to the `geom_point` displaying the raw data.

`violin.args` A list of additional aesthetic arguments to be passed to the `geom_violin`.

`ggplot.component` A `ggplot` component to be added to the plot prepared by `ggstatsplot`. This argument is primarily helpful for `grouped_` variant of the current function. Default is `NULL`. The argument should be entered as a function.

`package` Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running `View(paletteer::palettes_d_nam`

`palette` Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running `View(paletteer::palettes_d_nam`

`mean.point.args` A list of additional aesthetic arguments to be passed to `ggplot2::geom_point` and `ggrepel::geom_label_repel` geoms involved mean value plotting.

`mean.label.args` A list of additional aesthetic arguments to be passed to `ggplot2::geom_point` and `ggrepel::geom_label_repel` geoms involved mean value plotting.

`ggsignif.args` A list of additional aesthetic arguments to be passed to `ggsignif::geom_signif`.

`ggtheme` A function, `ggplot2` theme name. Default value is `ggplot2::theme_bw()`. Any of the `ggplot2` themes, or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`, etc.).

`ggstatsplot.layer` Logical that decides whether `theme_ggstatsplot` theme elements are to be displayed along with the selected `ggtheme` (Default: `TRUE`). `theme_ggstatsplot` is an opinionated theme layer that override some aspects of the selected `ggtheme`.

`effsize.type` Type of effect size needed for *parametric* tests. The argument can be `"biased"` (equivalent to `"d"` for Cohen's *d* for **t-test**; `"partial_eta"` for partial eta-squared for **anova**) or `"unbiased"` (equivalent to `"g"` Hedge's *g* for **t-test**; `"partial_omega"` for partial omega-squared for **anova**)).

`partial` If `TRUE`, return partial indices.

`k` Number of digits after decimal point (should be an integer) (Default: `k = 2`).

`var.equal` a logical variable indicating whether to treat the variances in the samples as equal. If `TRUE`, then a simple F test for the equality of means in a one-way analysis of variance is performed. If `FALSE`, an approximate method of Welch (1951) is used, which generalizes the commonly known 2-sample Welch test to the case of arbitrarily many samples.

`conf.level` Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (`0.95`).

`type` Type of statistic expected (`"parametric"` or `"nonparametric"` or `"robust"` or `"bayes"`).Corresponding abbreviations are also accepted: `"p"` (for parametric), `"np"` (nonparametric), `"r"` (robust), or `"bf"`resp.

	`nboot` Number of bootstrap samples for computing confidence interval for the effect size (Default: `100`).
	`tr` Trim level for the mean when carrying out `robust` tests. If you get error stating "Standard error cannot be computed because of Winsorized variance of 0 (e.g., due to ties). Try to decrease the trimming level.", try to play around with the value of `tr`, which is by default set to `0.1`. Lowering the value might help.
`plotgrid.args`	A list of additional arguments to `cowplot::plot_grid`.
`title.text`	String or plotmath expression to be drawn as title for the *combined plot*.
`title.args`	A list of additional arguments provided to `title`, `caption` and `sub`, resp.
`caption.text`	String or plotmath expression to be drawn as the caption for the *combined plot*.
`caption.args`	A list of additional arguments provided to `title`, `caption` and `sub`, resp.
`sub.text`	The label with which the *combined plot* should be annotated. Can be a plotmath expression.
`sub.args`	A list of additional arguments provided to `title`, `caption` and `sub`, resp.

Details

For parametric tests, Welch's ANOVA/*t*-test are used as a default (i.e., `var.equal = FALSE`).

If robust tests are selected, following tests are used is .

- ANOVA: one-way ANOVA on trimmed means (see `?WRS2::t1way`)
- *t*-test: Yuen's test for trimmed means (see `?WRS2::yuen`)

For more about how the effect size measures (for nonparametric tests) and their confidence intervals are computed, see `?rcompanion::wilcoxonR`.

For repeated measures designs, use `ggwithinstats`.

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggbetweenstats.html

See Also

`ggbetweenstats`, `ggwithinstats`, `grouped_ggwithinstats`

Examples

```
# to get reproducible results from bootstrapping
set.seed(123)

# the most basic function call
ggstatsplot::grouped_ggbetweenstats(
  data = dplyr::filter(ggplot2::mpg, drv != "4"),
  x = year,
  y = hwy,
  grouping.var = drv,
```

```
  conf.level = 0.99
)

# modifying individual plots using `ggplot.component` argument
ggstatsplot::grouped_ggbetweenstats(
  data = dplyr::filter(
    ggstatsplot::movies_long,
    genre %in% c("Action", "Comedy"),
    mpaa %in% c("R", "PG")
  ),
  x = genre,
  y = rating,
  grouping.var = mpaa,
  results.subtitle = FALSE,
  ggplot.component = ggplot2::scale_y_continuous(
    breaks = seq(1, 9, 1),
    limits = (c(1, 9))
  ),
  messages = FALSE
)
```

`grouped_ggcorrmat` *Visualization of a correlalogram (or correlation matrix) for all levels of a grouping variable*

Description

Helper function for `ggstatsplot::ggcorrmat` to apply this function across multiple levels of a given factor and combining the resulting plots using `ggstatsplot::combine_plots`.

Usage

```
grouped_ggcorrmat(
  data,
  cor.vars = NULL,
  cor.vars.names = NULL,
  grouping.var,
  title.prefix = NULL,
  output = "plot",
  ...,
  plotgrid.args = list(),
  title.text = NULL,
  title.args = list(size = 16, fontface = "bold"),
  caption.text = NULL,
  caption.args = list(size = 10),
  sub.text = NULL,
  sub.args = list(size = 12)
)
```

Arguments

`data`	Dataframe from which variables specified are preferentially to be taken.
`cor.vars`	List of variables for which the correlation matrix is to be computed and visualized. If `NULL` (default), all numeric variables from `data` will be used.
`cor.vars.names`	Optional list of names to be used for `cor.vars`. The names should be entered in the same order.
`grouping.var`	A single grouping variable (can be entered either as a bare name `x` or as a string `"x"`).
`title.prefix`	Character string specifying the prefix text for the fixed plot title (name of each factor level) (Default: `NULL`). If `NULL`, the variable name entered for `grouping.var` will be used.
`output`	Character that decides expected output from this function. If `"plot"`, the visualization matrix will be returned. If `"dataframe"` (or literally anything other than `"plot"`), a dataframe containing all details from statistical analyses (e.g., correlation coefficients, statistic values, *p*-values, no. of observations, etc.) will be returned.
`...`	Arguments passed on to ggcorrmat

`matrix.type` Character, `"full"` (default), `"upper"` or `"lower"`, display full matrix, lower triangular or upper triangular matrix.

`matrix.method` The visualization method of correlation matrix to be used. Allowed values are `"square"` (default) or `"circle"`.

`sig.level` Significance level (Default: `0.05`). If the *p*-value in *p*-value matrix is bigger than `sig.level`, then the corresponding correlation coefficient is regarded as insignificant and flagged as such in the plot. Relevant only when `output = "plot"`.

`p.adjust.method` What adjustment for multiple tests should be used? (`"holm"`, `"hochberg"`, `"hommel"`, `"bonferroni"`, `"BH"`, `"BY"`, `"fdr"`, `"none"`). See `stats::p.adjust` for details about why to use `"holm"` rather than `"bonferroni"`). Default is `"none"`. If adjusted *p*-values are displayed in the visualization of correlation matrix, the **adjusted** *p*-values will be used for the **upper** triangle, while **unadjusted** *p*-values will be used for the **lower** triangle of the matrix.

`colors` A vector of 3 colors for low, mid, and high correlation values. If set to `NULL`, manual specification of colors will be turned off and 3 colors from the specified `palette` from `package` will be selected.

`pch` Decides the point shape to be used for insignificant correlation coefficients (only valid when `insig = "pch"`). Default: `pch = "cross"`.

`ggcorrplot.args` A list of additional (mostly aesthetic) arguments that will be passed to `ggcorrplot::ggcorrplot` function. The list should avoid any of the following arguments since they are already internally being used by `ggstatsplot`: `corr`, `method`, `p.mat`, `sig.level`, `ggtheme`, `colors`, `matrix.type`, `lab`, `pch`, `legend.title`, `digits`.

`messages` Decides whether messages references, notes, and warnings are to be displayed (Default: `TRUE`).

`type` Type of association between paired samples required (`""parametric"`: Pearson's product moment correlation coefficient" or `""nonparametric"`: Spearman's rho" or `""robust"`: percentage bend correlation coefficient" or `""bayes"`: Bayes Factor for Pearson's *r*"). Corresponding abbreviations are also accepted: `"p"` (for parametric/pearson), `"np"` (nonparametric/spearman), `"r"` (robust), `"bf"` (for bayes factor), resp.

`beta` bending constant (Default: `0.1`). For more, see `WRS2::pbcor()`.

`k` Number of digits after decimal point (should be an integer) (Default: `k = 2L`).

`conf.level` Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (`0.95`).

`bf.prior` A number between `0.5` and `2` (default `0.707`), the prior width to use in calculating Bayes factors.

`package` Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running `View(paletteer::palettes_d_nam`

`palette` Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running `View(paletteer::palettes_d_nam`

`ggtheme` A function, `ggplot2` theme name. Default value is `ggplot2::theme_bw()`. Any of the `ggplot2` themes, or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`, etc.).

`ggstatsplot.layer` Logical that decides whether `theme_ggstatsplot` theme elements are to be displayed along with the selected `ggtheme` (Default: `TRUE`). `theme_ggstatsplot` is an opinionated theme layer that override some aspects of the selected `ggtheme`.

`ggplot.component` A `ggplot` component to be added to the plot prepared by `ggstatsplot`. This argument is primarily helpful for `grouped_` variant of the current function. Default is `NULL`. The argument should be entered as a function.

`subtitle` The text for the plot subtitle. Will work only if `results.subtitle = FALSE`.

`caption` The text for the plot caption.

`plotgrid.args` A list of additional arguments to `cowplot::plot_grid`.

`title.text` String or plotmath expression to be drawn as title for the *combined plot*.

`title.args` A list of additional arguments provided to `title`, `caption` and `sub`, resp.

`caption.text` String or plotmath expression to be drawn as the caption for the *combined plot*.

`caption.args` A list of additional arguments provided to `title`, `caption` and `sub`, resp.

`sub.text` The label with which the *combined plot* should be annotated. Can be a plotmath expression.

`sub.args` A list of additional arguments provided to `title`, `caption` and `sub`, resp.

Value

Correlation matrix plot or a dataframe containing results from pairwise correlation tests. The package internally uses `ggcorrplot::ggcorrplot` for creating the visualization matrix, while the correlation analysis is carried out using the `correlation::correlation` function.

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggcorrmat.html

See Also

ggcorrmat, ggscatterstats, grouped_ggscatterstats

Examples

```
# for reproducibility
set.seed(123)

# for plot
ggstatsplot::grouped_ggcorrmat(
  data = iris,
  grouping.var = Species,
  type = "robust",
  p.adjust.method = "holm"
)

# for dataframe
ggstatsplot::grouped_ggcorrmat(
  data = ggplot2::msleep,
  grouping.var = vore,
  type = "bayes",
  output = "dataframe"
)
```

---

grouped_ggdotplotstats

Grouped histograms for distribution of a labeled numeric variable

---

Description

Helper function for `ggstatsplot::ggdotplotstats` to apply this function across multiple levels of a given factor and combining the resulting plots using `ggstatsplot::combine_plots`.

Usage

```
grouped_ggdotplotstats(
  data,
  x,
  y,
  grouping.var,
  title.prefix = NULL,
  output = "plot",
```

```
  ...,
  plotgrid.args = list(),
  title.text = NULL,
  title.args = list(size = 16, fontface = "bold"),
  caption.text = NULL,
  caption.args = list(size = 10),
  sub.text = NULL,
  sub.args = list(size = 12)
)
```

Arguments

`data`	A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will **not** be accepted.
`x`	A numeric variable from the dataframe `data`.
`y`	Label or grouping variable.
`grouping.var`	A single grouping variable (can be entered either as a bare name `x` or as a string `"x"`).
`title.prefix`	Character string specifying the prefix text for the fixed plot title (name of each factor level) (Default: `NULL`). If `NULL`, the variable name entered for `grouping.var` will be used.
`output`	Character that describes what is to be returned: can be `"plot"` (default) or `"subtitle"` or `"caption"`. Setting this to `"subtitle"` will return the expression containing statistical results. If you have set `results.subtitle = FALSE`, then this will return a `NULL`. Setting this to `"caption"` will return the expression containing details about Bayes Factor analysis, but valid only when `type = "parametric"` and `bf.message = TRUE`, otherwise this will return a `NULL`. For functions `ggpiestats` and `ggbarstats`, setting `output = "proptest"` will return a dataframe containing results from proportion tests.
`...`	Arguments passed on to `ggdotplotstats` `point.args` A list of additional aesthetic arguments passed to `geom_point`. `test.value` A number specifying the value of the null hypothesis (Default: `0`). `test.k` Integer denoting the number of decimal places expected for `test.value` label. (Default: `0`). `test.value.line` Logical that decides whether a line corresponding to the `test.value` should be superimposed on the plot. `test.value.line.args` A list of additional aesthetic arguments to be passed to the `geom_line` used to display the lines corresponding to the centrality parameter and test value. `test.value.label.args` A list of additional aesthetic arguments to be passed to the `geom_label` used to display the label corresponding to the centrality parameter and test value. `centrality.parameter` Decides *which* measure of central tendency (`"mean"` or `"median"`) is to be displayed as a vertical line. To not show any parameter, set this to `"none"`.

centrality.k Integer denoting the number of decimal places expected for centrality parameter label. (Default: 2).

centrality.line.args A list of additional aesthetic arguments to be passed to the geom_line used to display the lines corresponding to the centrality parameter and test value.

centrality.label.args A list of additional aesthetic arguments to be passed to the geom_label used to display the label corresponding to the centrality parameter and test value.

xlab Labels for x and y axis variables. If NULL (default), variable names for x and y will be used.

subtitle The text for the plot subtitle. Will work only if results.subtitle = FALSE.

caption The text for the plot caption.

type Type of statistic expected ("parametric" or "nonparametric" or "robust" or "bayes").Corresponding abbreviations are also accepted: "p" (for parametric), "np" (nonparametric), "r" (robust), or "bf"resp.

bf.prior A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors.

bf.message Logical that decides whether to display Bayes Factor in favor of the *null* hypothesis. This argument is relevant only **for parametric test** (Default: TRUE).

effsize.type Type of effect size needed for *parametric* tests. The argument can be "d" (for Cohen's *d*) or "g" (for Hedge's *g*).

conf.level Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (0.95).

nboot Number of bootstrap samples for computing confidence interval for the effect size (Default: 100).

k Number of digits after decimal point (should be an integer) (Default: k = 2).

results.subtitle Decides whether the results of statistical tests are to be displayed as a subtitle (Default: TRUE). If set to FALSE, only the plot will be returned.

ggplot.component A ggplot component to be added to the plot prepared by ggstatsplot. This argument is primarily helpful for grouped_ variant of the current function. Default is NULL. The argument should be entered as a function.

ggtheme A function, ggplot2 theme name. Default value is ggplot2::theme_bw(). Any of the ggplot2 themes, or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.).

ggstatsplot.layer Logical that decides whether theme_ggstatsplot theme elements are to be displayed along with the selected ggtheme (Default: TRUE). theme_ggstatsplot is an opinionated theme layer that override some aspects of the selected ggtheme.

ylab Labels for x- and y- axis variables, respectively (Defaults: "regression coefficient" and "term").

plotgrid.args A list of additional arguments to cowplot::plot_grid.

`title.text`	String or plotmath expression to be drawn as title for the *combined plot*.
`title.args`	A list of additional arguments provided to `title`, `caption` and `sub`, resp.
`caption.text`	String or plotmath expression to be drawn as the caption for the *combined plot*.
`caption.args`	A list of additional arguments provided to `title`, `caption` and `sub`, resp.
`sub.text`	The label with which the *combined plot* should be annotated. Can be a plotmath expression.
`sub.args`	A list of additional arguments provided to `title`, `caption` and `sub`, resp.

Author(s)

Indrajeet Patil

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggdotplotstats.html

See Also

grouped_gghistostats, ggdotplotstats, gghistostats

Examples

```
# for reproducibility
set.seed(123)

# removing factor level with very few no. of observations
df <- dplyr::filter(.data = ggplot2::mpg, cyl %in% c("4", "6", "8"))

# plot
ggstatsplot::grouped_ggdotplotstats(
  data = df,
  x = "cty",
  y = "manufacturer",
  grouping.var = "cyl",
  test.value = 15.5,
  title.prefix = "cylinder count",
  ggplot.component = ggplot2::scale_x_continuous(
    sec.axis = ggplot2::dup_axis(),
    limits = c(12, 24),
    breaks = seq(12, 24, 2)
  ),
  messages = FALSE
)
```

grouped_gghistostats *Grouped histograms for distribution of a numeric variable*

Description

Helper function for ggstatsplot::gghistostats to apply this function across multiple levels of a given factor and combining the resulting plots using ggstatsplot::combine_plots.

Usage

```
grouped_gghistostats(
  data,
  x,
  grouping.var,
  binwidth = NULL,
  title.prefix = NULL,
  output = "plot",
  ...,
  plotgrid.args = list(),
  title.text = NULL,
  title.args = list(size = 16, fontface = "bold"),
  caption.text = NULL,
  caption.args = list(size = 10),
  sub.text = NULL,
  sub.args = list(size = 12)
)
```

Arguments

data	A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will **not** be accepted.
x	A numeric variable from the dataframe data.
grouping.var	A single grouping variable (can be entered either as a bare name x or as a string "x").
binwidth	The width of the histogram bins. Can be specified as a numeric value, or a function that calculates width from x. The default is to use the max(x) -min(x) / sqrt(N). You should always check this value and explore multiple widths to find the best to illustrate the stories in your data.
title.prefix	Character string specifying the prefix text for the fixed plot title (name of each factor level) (Default: NULL). If NULL, the variable name entered for grouping.var will be used.
output	Character that describes what is to be returned: can be "plot" (default) or "subtitle" or "caption". Setting this to "subtitle" will return the expression containing statistical results. If you have set results.subtitle = FALSE, then this will return a NULL. Setting this to "caption" will return the expression containing details about Bayes Factor analysis, but valid only when type =

"parametric" and bf.message = TRUE, otherwise this will return a NULL. For functions ggpiestats and ggbarstats, setting output = "proptest" will return a dataframe containing results from proportion tests.

... Arguments passed on to gghistostats

bar.measure Character describing what value needs to be represented as height in the bar chart. This can either be "count", which shows number of points in bin, or "density", which density of points in bin, scaled to integrate to 1, or "proportion", which shows relative frequencies of observations in each bin, or "mix", which shows *both* count and proportion in the same plot.

normal.curve A logical value that decides whether to super-impose a normal curve using stats::dnorm(mean(x),sd(x)). Default is FALSE.

normal.curve.args A list of additional aesthetic arguments to be passed to the normal curve.

bar.fill Character input that decides which color will uniformly fill all the bars in the histogram (Default: "grey50").

type Type of statistic expected ("parametric" or "nonparametric" or "robust" or "bayes").Corresponding abbreviations are also accepted: "p" (for parametric), "np" (nonparametric), "r" (robust), or "bf"resp.

test.value A number specifying the value of the null hypothesis (Default: 0).

bf.prior A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors.

effsize.type Type of effect size needed for *parametric* tests. The argument can be "d" (for Cohen's *d*) or "g" (for Hedge's *g*).

conf.level Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (0.95).

nboot Number of bootstrap samples for computing confidence interval for the effect size (Default: 100).

k Number of digits after decimal point (should be an integer) (Default: k = 2).

test.k Integer denoting the number of decimal places expected for test.value label. (Default: 0).

test.value.line Logical that decides whether a line corresponding to the test.value should be superimposed on the plot.

test.value.line.args A list of additional aesthetic arguments to be passed to the geom_line used to display the lines corresponding to the centrality parameter and test value.

test.value.label.args A list of additional aesthetic arguments to be passed to the geom_label used to display the label corresponding to the centrality parameter and test value.

centrality.parameter Decides *which* measure of central tendency ("mean" or "median") is to be displayed as a vertical line. To not show any parameter, set this to "none".

centrality.k Integer denoting the number of decimal places expected for centrality parameter label. (Default: 2).

centrality.line.args A list of additional aesthetic arguments to be passed to the geom_line used to display the lines corresponding to the centrality parameter and test value.

centrality.label.args A list of additional aesthetic arguments to be passed to the geom_label used to display the label corresponding to the centrality parameter and test value.

xlab Labels for x and y axis variables. If NULL (default), variable names for x and y will be used.

subtitle The text for the plot subtitle. Will work only if results.subtitle = FALSE.

caption The text for the plot caption.

bf.message Logical that decides whether to display Bayes Factor in favor of the *null* hypothesis. This argument is relevant only **for parametric test** (Default: TRUE).

ggtheme A function, ggplot2 theme name. Default value is ggplot2::theme_bw(). Any of the ggplot2 themes, or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.).

ggstatsplot.layer Logical that decides whether theme_ggstatsplot theme elements are to be displayed along with the selected ggtheme (Default: TRUE). theme_ggstatsplot is an opinionated theme layer that override some aspects of the selected ggtheme.

results.subtitle Decides whether the results of statistical tests are to be displayed as a subtitle (Default: TRUE). If set to FALSE, only the plot will be returned.

ggplot.component A ggplot component to be added to the plot prepared by ggstatsplot. This argument is primarily helpful for grouped_ variant of the current function. Default is NULL. The argument should be entered as a function.

plotgrid.args A list of additional arguments to cowplot::plot_grid.

title.text String or plotmath expression to be drawn as title for the *combined plot*.

title.args A list of additional arguments provided to title, caption and sub, resp.

caption.text String or plotmath expression to be drawn as the caption for the *combined plot*.

caption.args A list of additional arguments provided to title, caption and sub, resp.

sub.text The label with which the *combined plot* should be annotated. Can be a plotmath expression.

sub.args A list of additional arguments provided to title, caption and sub, resp.

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/gghistostats.html

See Also

gghistostats, ggdotplotstats, grouped_ggdotplotstats

Examples

```
# for reproducibility
set.seed(123)

# plot
ggstatsplot::grouped_gghistostats(
  data = iris,
  x = Sepal.Length,
  test.value = 5,
  grouping.var = Species,
  bar.fill = "orange",
  ggplot.component = list(
    ggplot2::scale_x_continuous(breaks = seq(3, 9, 1), limits = (c(3, 9))),
    ggplot2::scale_y_continuous(breaks = seq(0, 25, 5), limits = (c(0, 25)))
  ),
  messages = FALSE,
  plotgrid.args = list(nrow = 1, labels = c("(i)", "(ii)", "(iii)")),
)
```

---

grouped_ggpiestats *Grouped pie charts with statistical tests*

Description

Helper function for `ggstatsplot::ggpiestats` to apply this function across multiple levels of a given factor and combining the resulting plots using `ggstatsplot::combine_plots`.

Usage

```
grouped_ggpiestats(
  data,
  main,
  condition = NULL,
  counts = NULL,
  grouping.var,
  title.prefix = NULL,
  output = "plot",
  x = NULL,
  y = NULL,
  ...,
  plotgrid.args = list(),
  title.text = NULL,
  title.args = list(size = 16, fontface = "bold"),
  caption.text = NULL,
  caption.args = list(size = 10),
  sub.text = NULL,
```

```
  sub.args = list(size = 12)
)
```

Arguments

data	A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will **not** be accepted.
main	The variable to use as the **rows** in the contingency table.
condition	The variable to use as the **columns** in the contingency table. Default is NULL. If NULL, one-sample proportion test (a goodness of fit test) will be run for the x variable. Otherwise an appropriate association test will be run. This argument can not be NULL for ggbarstats function.
counts	A string naming a variable in data containing counts, or NULL if each row represents a single observation (Default).
grouping.var	A single grouping variable (can be entered either as a bare name x or as a string "x").
title.prefix	Character string specifying the prefix text for the fixed plot title (name of each factor level) (Default: NULL). If NULL, the variable name entered for grouping.var will be used.
output	Character that describes what is to be returned: can be "plot" (default) or "subtitle" or "caption". Setting this to "subtitle" will return the expression containing statistical results. If you have set results.subtitle = FALSE, then this will return a NULL. Setting this to "caption" will return the expression containing details about Bayes Factor analysis, but valid only when type = "parametric" and bf.message = TRUE, otherwise this will return a NULL. For functions ggpiestats and ggbarstats, setting output = "proptest" will return a dataframe containing results from proportion tests.
x	The variable to use as the **rows** in the contingency table.
y	The variable to use as the **columns** in the contingency table. Default is NULL. If NULL, one-sample proportion test (a goodness of fit test) will be run for the x variable. Otherwise an appropriate association test will be run. This argument can not be NULL for ggbarstats function.
...	Arguments passed on to ggpiestats proportion.test Decides whether proportion test for main variable is to be carried out for each level of condition (Default: TRUE). perc.k Numeric that decides number of decimal places for percentage labels (Default: 0). label Character decides what information needs to be displayed on the label in each pie slice. Possible options are "percentage" (default), "counts", "both". label.args Additional aesthetic arguments that will be passed to geom_label. label.repel Whether labels should be repelled using ggrepel package. This can be helpful in case the labels are overlapping. legend.title Title text for the legend.

`results.subtitle` Decides whether the results of statistical tests are to be displayed as a subtitle (Default: `TRUE`). If set to `FALSE`, only the plot will be returned.

`conf.level` Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (`0.95`).

`nboot` Number of bootstrap samples for computing confidence interval for the effect size (Default: `100`).

`k` Number of digits after decimal point (should be an integer) (Default: `k = 2`).

`bf.message` Logical that decides whether to display Bayes Factor in favor of the *null* hypothesis. This argument is relevant only **for parametric test** (Default: `TRUE`).

`subtitle` The text for the plot subtitle. Will work only if `results.subtitle = FALSE`.

`caption` The text for the plot caption.

`ggtheme` A function, `ggplot2` theme name. Default value is `ggplot2::theme_bw()`. Any of the `ggplot2` themes, or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`, etc.).

`ggstatsplot.layer` Logical that decides whether `theme_ggstatsplot` theme elements are to be displayed along with the selected `ggtheme` (Default: `TRUE`). `theme_ggstatsplot` is an opinionated theme layer that override some aspects of the selected `ggtheme`.

`package` Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running `View(paletteer::palettes_d_nam`

`palette` Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running `View(paletteer::palettes_d_nam`

`ggplot.component` A `ggplot` component to be added to the plot prepared by `ggstatsplot`. This argument is primarily helpful for `grouped_` variant of the current function. Default is `NULL`. The argument should be entered as a function.

`ratio` A vector of proportions: the expected proportions for the proportion test (should sum to 1). Default is `NULL`, which means the null is equal theoretical proportions across the levels of the nominal variable. This means if there are two levels this will be `ratio = c(0.5,0.5)` or if there are four levels this will be `ratio = c(0.25,0.25,0.25,0.25)`, etc.

`sampling.plan` Character describing the sampling plan. Possible options are `"indepMulti"` (independent multinomial; default), `"poisson"`, `"jointMulti"` (joint multinomial), `"hypergeom"` (hypergeometric). For more, see `?BayesFactor::contingencyTa`

`fixed.margin` For the independent multinomial sampling plan, which margin is fixed (`"rows"` or `"cols"`). Defaults to `"rows"`.

`prior.concentration` Specifies the prior concentration parameter, set to `1` by default. It indexes the expected deviation from the null hypothesis under the alternative, and corresponds to Gunel and Dickey's (1974) `"a"` parameter.

`paired` Logical indicating whether data came from a within-subjects or repeated measures design study (Default: `FALSE`). If `TRUE`, McNemar's test subtitle will be returned. If `FALSE`, Pearson's chi-square test will be returned.

plotgrid.args	A list of additional arguments to cowplot::plot_grid.
title.text	String or plotmath expression to be drawn as title for the *combined plot*.
title.args	A list of additional arguments provided to title, caption and sub, resp.
caption.text	String or plotmath expression to be drawn as the caption for the *combined plot*.
caption.args	A list of additional arguments provided to title, caption and sub, resp.
sub.text	The label with which the *combined plot* should be annotated. Can be a plotmath expression.
sub.args	A list of additional arguments provided to title, caption and sub, resp.

Value

Unlike a number of statistical softwares, ggstatsplot doesn't provide the option for Yates' correction for the Pearson's chi-squared statistic. This is due to compelling amount of Monte-Carlo simulation research which suggests that the Yates' correction is overly conservative, even in small sample sizes. As such it is recommended that it should not ever be applied in practice (Camilli & Hopkins, 1978, 1979; Feinberg, 1980; Larntz, 1978; Thompson, 1988).

For more about how the effect size measures and their confidence intervals are computed, see ?rcompanion::cohenG, ?rcompanion::cramerV, and ?rcompanion::cramerVFit.

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggpiestats.html

See Also

ggbarstats, ggpiestats, grouped_ggbarstats

Examples

```
# grouped one-sample proportion tests
ggstatsplot::grouped_ggpiestats(
  data = mtcars,
  grouping.var = am,
  x = cyl
)

# the following will take slightly more amount of time
# for reproducibility
set.seed(123)

# let's create a smaller dataframe
diamonds_short <- ggplot2::diamonds %>%
  dplyr::filter(.data = ., cut %in% c("Fair", "Very Good", "Ideal")) %>%
  dplyr::sample_frac(tbl = ., size = 0.10)

# plot
ggstatsplot::grouped_ggpiestats(
  data = diamonds_short,
```

```
  x = color,
  y = clarity,
  grouping.var = cut,
  nboot = 20,
  sampling.plan = "poisson",
  title.prefix = "Quality",
  slice.label = "both",
  messages = FALSE,
  perc.k = 1,
  plotgrid.args = list(nrow = 3)
)
```

`grouped_ggscatterstats` *Scatterplot with marginal distributions for all levels of a grouping variable*

Description

Grouped scatterplots from `ggplot2` combined with marginal histograms/boxplots/density plots with statistical details added as a subtitle.

Usage

```
grouped_ggscatterstats(
  data,
  x,
  y,
  grouping.var,
  label.var = NULL,
  label.expression = NULL,
  title.prefix = NULL,
  output = "plot",
  ...,
  plotgrid.args = list(),
  title.text = NULL,
  title.args = list(size = 16, fontface = "bold"),
  caption.text = NULL,
  caption.args = list(size = 10),
  sub.text = NULL,
  sub.args = list(size = 12)
)
```

Arguments

`data`	A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will **not** be accepted.

`x` The column in `data` containing the explanatory variable to be plotted on the x-axis. Can be entered either as a character string (e.g., `"x"`) or as a bare expression (e.g, `x`).

`y` The column in `data` containing the response (outcome) variable to be plotted on the y-axis. Can be entered either as a character string (e.g., `"y"`) or as a bare expression (e.g, `y`).

`grouping.var` A single grouping variable (can be entered either as a bare name `x` or as a string `"x"`).

`label.var` Variable to use for points labels. Can be entered either as a character string (e.g., `"var1"`) or as a bare expression (e.g, `var1`).

`label.expression`
An expression evaluating to a logical vector that determines the subset of data points to label. This argument can be entered either as a character string (e.g., `"y < 4 & z < 20"`) or as a bare expression (e.g., `y < 4 & z < 20`).

`title.prefix` Character string specifying the prefix text for the fixed plot title (name of each factor level) (Default: `NULL`). If `NULL`, the variable name entered for `grouping.var` will be used.

`output` Character that describes what is to be returned: can be `"plot"` (default) or `"subtitle"` or `"caption"`. Setting this to `"subtitle"` will return the expression containing statistical results. If you have set `results.subtitle = FALSE`, then this will return a `NULL`. Setting this to `"caption"` will return the expression containing details about Bayes Factor analysis, but valid only when `type = "parametric"` and `bf.message = TRUE`, otherwise this will return a `NULL`. For functions `ggpiestats` and `ggbarstats`, setting `output = "proptest"` will return a dataframe containing results from proportion tests.

`...` Arguments passed on to ggscatterstats

`point.label.args` A list of additional aesthetic arguments to be passed to `ggrepel::geom_label_repel` geom used to display the labels.

`smooth.line.args` A list of additional aesthetic arguments to be passed to `ggplot2::geom_smooth` geom used to display the regression line.

`point.args` A list of additional aesthetic arguments to be passed to `ggplot2::geom_point` geom used to display the raw data points.

`marginal` Decides whether `ggExtra::ggMarginal()` plots will be displayed; the default is `TRUE`.

`point.width.jitter` Degree of jitter in `x` and `y` direction, respectively. Defaults to `0` (0%) of the resolution of the data. Note that the jitter should not be specified in the `point.args` because this information will be passed to two different geoms: one displaying the points and the other displaying the labels for these points.

`point.height.jitter` Degree of jitter in `x` and `y` direction, respectively. Defaults to `0` (0%) of the resolution of the data. Note that the jitter should not be specified in the `point.args` because this information will be passed to two different geoms: one displaying the points and the other displaying the labels for these points.

`marginal.type` Type of marginal distribution to be plotted on the axes (`"histogram"`, `"boxplot"`, `"density"`, `"violin"`, `"densigram"`).

`marginal.size` Integer describing the relative size of the marginal plots compared to the main plot. A size of `5` means that the main plot is 5x wider and 5x taller than the marginal plots.

`xfill` Character describing color fill for `x` and `y` axes marginal distributions (default: `"#009E73"` (for `x`) and `"#D55E00"` (for `y`)). The same colors will also be used for the lines denoting centrality parameters if `centrality.parameter` argument is set to `TRUE`. Note that the defaults are colorblind-friendly.

`yfill` Character describing color fill for `x` and `y` axes marginal distributions (default: `"#009E73"` (for `x`) and `"#D55E00"` (for `y`)). The same colors will also be used for the lines denoting centrality parameters if `centrality.parameter` argument is set to `TRUE`. Note that the defaults are colorblind-friendly.

`centrality.parameter` Decides *which* measure of central tendency (`"mean"` or `"median"`) is to be displayed as vertical (for `x`) and horizontal (for `y`) lines. Note that mean values corresponds to arithmetic mean and not geometric mean.

`vline.args` A list of additional aesthetic arguments to be passed to `ggplot2::geom_vline` and `ggplot2::geom_hline` geoms used to display the centrality parameter labels on vertical and horizontal lines.

`hline.args` A list of additional aesthetic arguments to be passed to `ggplot2::geom_vline` and `ggplot2::geom_hline` geoms used to display the centrality parameter labels on vertical and horizontal lines.

`type` Type of association between paired samples required (`""parametric"`: Pearson's product moment correlation coefficient" or `""nonparametric"`: Spearman's rho" or `""robust"`: percentage bend correlation coefficient" or `""bayes"`: Bayes Factor for Pearson's *r*"). Corresponding abbreviations are also accepted: `"p"` (for parametric/pearson), `"np"` (nonparametric/spearman), `"r"` (robust), `"bf"` (for bayes factor), resp.

`conf.level` Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (`0.95`).

`bf.prior` A number between `0.5` and `2` (default `0.707`), the prior width to use in calculating Bayes factors.

`beta` bending constant (Default: `0.1`). For more, see `WRS2::pbcor()`.

`k` Number of digits after decimal point (should be an integer) (Default: `k = 2L`).

`formula` Formula to use in smoothing function, eg. `y ~ x`, `y ~ poly(x,2)`, `y ~ log(x)`. `NULL` by default, in which case `method = NULL` implies `formula = y ~ x` when there are fewer than 1,000 observations and `formula = y ~ s(x,bs = "cs")` otherwise.

`method` Smoothing method (function) to use, accepts either `NULL` or a character vector, e.g. `"lm"`, `"glm"`, `"gam"`, `"loess"` or a function, e.g. `MASS::rlm` or `mgcv::gam`, `stats::lm`, or `stats::loess`. `"auto"` is also accepted for backwards compatibility. It is equivalent to `NULL`.

For `method = NULL` the smoothing method is chosen based on the size of the largest group (across all panels). `stats::loess()` is used for less than 1,000 observations; otherwise `mgcv::gam()` is used with `formula = y ~ s(x,bs = "cs")` with `method = "REML"`. Somewhat anecdotally, `loess` gives a better appearance, but is $O(N^2)$ in memory, so does not work for larger datasets.

If you have fewer than 1,000 observations but want to use the same `gam()` model that `method = NULL` would use, then set method = "gam", formula = y ~ s(x, bs = "cs").

`method.args` List of additional arguments passed on to the modelling function defined by `method`.

`ggtheme` A function, `ggplot2` theme name. Default value is `ggplot2::theme_bw()`. Any of the `ggplot2` themes, or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`, etc.).

`ggstatsplot.layer` Logical that decides whether `theme_ggstatsplot` theme elements are to be displayed along with the selected `ggtheme` (Default: `TRUE`). `theme_ggstatsplot` is an opinionated theme layer that override some aspects of the selected `ggtheme`.

`bf.message` Logical that decides whether to display Bayes Factor in favor of the *null* hypothesis. This argument is relevant only **for parametric test** (Default: `TRUE`).

`results.subtitle` Decides whether the results of statistical tests are to be displayed as a subtitle (Default: `TRUE`). If set to `FALSE`, only the plot will be returned.

`xlab` Labels for `x` and `y` axis variables. If `NULL` (default), variable names for `x` and `y` will be used.

`ylab` Labels for `x` and `y` axis variables. If `NULL` (default), variable names for `x` and `y` will be used.

`subtitle` The text for the plot subtitle. Will work only if `results.subtitle = FALSE`.

`caption` The text for the plot caption.

`ggplot.component` A `ggplot` component to be added to the plot prepared by `ggstatsplot`. This argument is primarily helpful for `grouped_` variant of the current function. Default is `NULL`. The argument should be entered as a function.

`margins` Along which margins to show the plots. One of: [both, x, y].

`xparams` List of extra parameters to use only for the marginal plot along the x axis.

`yparams` List of extra parameters to use only for the marginal plot along the y axis.

`centrality.label.args` A list of additional aesthetic arguments to be passed to the `geom_label` used to display the label corresponding to the centrality parameter and test value.

`plotgrid.args` A list of additional arguments to `cowplot::plot_grid`.

`title.text` String or plotmath expression to be drawn as title for the *combined plot*.

`title.args` A list of additional arguments provided to `title`, `caption` and `sub`, resp.

`caption.text` String or plotmath expression to be drawn as the caption for the *combined plot*.

`caption.args` A list of additional arguments provided to `title`, `caption` and `sub`, resp.

`sub.text` The label with which the *combined plot* should be annotated. Can be a plotmath expression.

`sub.args` A list of additional arguments provided to `title`, `caption` and `sub`, resp.

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggscatterstats.html

See Also

ggscatterstats, ggcorrmat, grouped_ggcorrmat

Examples

```
# to ensure reproducibility
set.seed(123)
library(ggstatsplot)

# basic function call
ggstatsplot::grouped_ggscatterstats(
  data = dplyr::filter(movies_long, genre == "Comedy" | genre == "Drama"),
  x = length,
  y = rating,
  method = "lm",
  formula = y ~ x + I(x^3),
  grouping.var = genre
)

# using labeling
# (also show how to modify basic plot from within function call)
grouped_ggscatterstats(
  data = dplyr::filter(ggplot2::mpg, cyl != 5),
  x = displ,
  y = hwy,
  grouping.var = cyl,
  title.prefix = "Cylinder count",
  type = "robust",
  label.var = manufacturer,
  label.expression = hwy > 25 & displ > 2.5,
  ggplot.component = ggplot2::scale_y_continuous(sec.axis = ggplot2::dup_axis()),
  messages = FALSE
)

# labeling without expression

ggstatsplot::grouped_ggscatterstats(
  data = dplyr::filter(
    .data = movies_long,
    rating == 7,
    genre %in% c("Drama", "Comedy")
  ),
  x = budget,
  y = length,
  grouping.var = genre,
  bf.message = FALSE,
  label.var = "title",
  marginal = FALSE,
```

```
  title.prefix = "Genre",
  caption.text = "All movies have IMDB rating equal to 7."
)
```

`grouped_ggwithinstats` *Violin plots for group or condition comparisons in within-subjects designs repeated across all levels of a grouping variable.*

Description

A combined plot of comparison plot created for levels of a grouping variable.

Usage

```
grouped_ggwithinstats(
  data,
  x,
  y,
  grouping.var,
  outlier.label = NULL,
  title.prefix = NULL,
  output = "plot",
  ...,
  plotgrid.args = list(),
  title.text = NULL,
  title.args = list(size = 16, fontface = "bold"),
  caption.text = NULL,
  caption.args = list(size = 10),
  sub.text = NULL,
  sub.args = list(size = 12)
)
```

Arguments

`data`	A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will **not** be accepted.
`x`	The grouping variable from the dataframe `data`.
`y`	The response (a.k.a. outcome or dependent) variable from the dataframe `data`.
`grouping.var`	A single grouping variable (can be entered either as a bare name `x` or as a string `"x"`).
`outlier.label`	Label to put on the outliers that have been tagged. This **can't** be the same as `x` argument.
`title.prefix`	Character string specifying the prefix text for the fixed plot title (name of each factor level) (Default: `NULL`). If `NULL`, the variable name entered for `grouping.var` will be used.

output Character that describes what is to be returned: can be "plot" (default) or "subtitle" or "caption". Setting this to "subtitle" will return the expression containing statistical results. If you have set results.subtitle = FALSE, then this will return a NULL. Setting this to "caption" will return the expression containing details about Bayes Factor analysis, but valid only when type = "parametric" and bf.message = TRUE, otherwise this will return a NULL. For functions ggpiestats and ggbarstats, setting output = "proptest" will return a dataframe containing results from proportion tests.

... Arguments passed on to ggwithinstats

point.path Logical that decides whether individual data points and means, respectively, should be connected using geom_path. Both default to TRUE. Note that point.path argument is relevant only when there are two groups (i.e., in case of a *t*-test). In case of large number of data points, it is advisable to set point.path = FALSE as these lines can overwhelm the plot.

mean.path Logical that decides whether individual data points and means, respectively, should be connected using geom_path. Both default to TRUE. Note that point.path argument is relevant only when there are two groups (i.e., in case of a *t*-test). In case of large number of data points, it is advisable to set point.path = FALSE as these lines can overwhelm the plot.

mean.path.args A list of additional aesthetic arguments passed on to geom_path connecting raw data points and mean points.

point.path.args A list of additional aesthetic arguments passed on to geom_path connecting raw data points and mean points.

type Type of statistic expected ("parametric" or "nonparametric" or "robust" or "bayes").Corresponding abbreviations are also accepted: "p" (for parametric), "np" (nonparametric), "r" (robust), or "bf"resp.

pairwise.comparisons Logical that decides whether pairwise comparisons are to be displayed (default: FALSE). Please note that only **significant** comparisons will be shown by default. To change this behavior, select appropriate option with pairwise.display argument. The pairwise comparison dataframes are prepared using the pairwiseComparisons::pairwise_comparisons function. For more details about pairwise comparisons, see the documentation for that function.

pairwise.display Decides which pairwise comparisons to display. Available options are "significant" (abbreviation accepted: "s") or "non-significant" (abbreviation accepted: "ns") or "everything"/"all". The default is "significant". You can use this argument to make sure that your plot is not uber-cluttered when you have multiple groups being compared and scores of pairwise comparisons being displayed.

p.adjust.method Adjustment method for *p*-values for multiple comparisons. Possible methods are: "holm" (default), "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".

effsize.type Type of effect size needed for *parametric* tests. The argument can be "biased" (equivalent to "d" for Cohen's *d* for **t-test**; "partial_eta" for partial eta-squared for **anova**) or "unbiased" (equivalent to "g" Hedge's *g* for **t-test**; "partial_omega" for partial omega-squared for **anova**)).

partial If TRUE, return partial indices.

bf.prior A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors.

bf.message Logical that decides whether to display Bayes Factor in favor of the *null* hypothesis. This argument is relevant only **for parametric test** (Default: TRUE).

results.subtitle Decides whether the results of statistical tests are to be displayed as a subtitle (Default: TRUE). If set to FALSE, only the plot will be returned.

xlab Labels for x and y axis variables. If NULL (default), variable names for x and y will be used.

ylab Labels for x and y axis variables. If NULL (default), variable names for x and y will be used.

caption The text for the plot caption.

subtitle The text for the plot subtitle. Will work only if results.subtitle = FALSE.

sample.size.label Logical that decides whether sample size information should be displayed for each level of the grouping variable x (Default: TRUE).

k Number of digits after decimal point (should be an integer) (Default: k = 2).

conf.level Scalar between 0 and 1. If unspecified, the defaults return 95% lower and upper confidence intervals (0.95).

nboot Number of bootstrap samples for computing confidence interval for the effect size (Default: 100).

tr Trim level for the mean when carrying out robust tests. If you get error stating "Standard error cannot be computed because of Winsorized variance of 0 (e.g., due to ties). Try to decrease the trimming level.", try to play around with the value of tr, which is by default set to 0.1. Lowering the value might help.

mean.plotting Logical that decides whether mean is to be highlighted and its value to be displayed (Default: TRUE).

mean.ci Logical that decides whether 95% confidence interval for mean is to be displayed (Default: FALSE).

mean.point.args A list of additional aesthetic arguments to be passed to ggplot2::geom_point and ggrepel::geom_label_repel geoms involved mean value plotting.

mean.label.args A list of additional aesthetic arguments to be passed to ggplot2::geom_point and ggrepel::geom_label_repel geoms involved mean value plotting.

notch A logical. If FALSE (default), a standard box plot will be displayed. If TRUE, a notched box plot will be used. Notches are used to compare groups; if the notches of two boxes do not overlap, this suggests that the medians are significantly different. In a notched box plot, the notches extend 1.58 * IQR / sqrt(n). This gives a roughly 95% confidence interval for comparing medians. IQR: Inter-Quartile Range.

notchwidth For a notched box plot, width of the notch relative to the body (default 0.5).

outlier.tagging Decides whether outliers should be tagged (Default: FALSE).

outlier.coef Coefficient for outlier detection using Tukey's method. With Tukey's method, outliers are below (1st Quartile) or above (3rd Quartile) outlier.coef times the Inter-Quartile Range (IQR) (Default: 1.5).

`outlier.label.args` A list of additional aesthetic arguments to be passed to `ggplot2::geom_point` and `ggrepel::geom_label_repel` geoms involved outlier value plotting.

`outlier.point.args` A list of additional aesthetic arguments to be passed to `ggplot2::geom_point` and `ggrepel::geom_label_repel` geoms involved outlier value plotting.

`violin.args` A list of additional aesthetic arguments to be passed to the `geom_violin`.

`ggsignif.args` A list of additional aesthetic arguments to be passed to `ggsignif::geom_signif`.

`ggtheme` A function, `ggplot2` theme name. Default value is `ggplot2::theme_bw()`. Any of the `ggplot2` themes, or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`, etc.).

`ggstatsplot.layer` Logical that decides whether `theme_ggstatsplot` theme elements are to be displayed along with the selected `ggtheme` (Default: `TRUE`). `theme_ggstatsplot` is an opinionated theme layer that override some aspects of the selected `ggtheme`.

`package` Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running `View(paletteer::palettes_d_nam`

`palette` Name of the package from which the given palette is to be extracted. The available palettes and packages can be checked by running `View(paletteer::palettes_d_nam`

`ggplot.component` A `ggplot` component to be added to the plot prepared by `ggstatsplot`. This argument is primarily helpful for `grouped_` variant of the current function. Default is `NULL`. The argument should be entered as a function.

`sphericity.correction` Logical that decides whether to apply correction to account for violation of sphericity in a repeated measures design ANOVA (Default: `TRUE`).

`plotgrid.args` A list of additional arguments to `cowplot::plot_grid`.

`title.text` String or plotmath expression to be drawn as title for the *combined plot*.

`title.args` A list of additional arguments provided to `title`, `caption` and `sub`, resp.

`caption.text` String or plotmath expression to be drawn as the caption for the *combined plot*.

`caption.args` A list of additional arguments provided to `title`, `caption` and `sub`, resp.

`sub.text` The label with which the *combined plot* should be annotated. Can be a plotmath expression.

`sub.args` A list of additional arguments provided to `title`, `caption` and `sub`, resp.

Details

For more about how the effect size measures (for nonparametric tests) and their confidence intervals are computed, see `?rcompanion::wilcoxonPairedR`.

For independent measures designs, use `ggbetweenstats`.

See Also

`ggwithinstats`, `ggbetweenstats`, `grouped_ggbetweenstats`

Examples

```
# to get reproducible results from bootstrapping
set.seed(123)
library(ggstatsplot)

# the most basic function call
ggstatsplot::grouped_ggwithinstats(
  data = VR_dilemma,
  x = modality,
  y = score,
  grouping.var = order,
  ggplot.component = ggplot2::scale_y_continuous(
    breaks = seq(0, 1, 0.1),
    limits = c(0, 1)
  ),
  messages = TRUE
)
```

---

iris_long *Edgar Anderson's Iris Data in long format.*

Description

Edgar Anderson's Iris Data in long format.

Usage

```
iris_long
```

Format

A data frame with 600 rows and 5 variables

- id. Dummy identity number for each flower (150 flowers in total).
- Species. The species are *Iris setosa*, *versicolor*, and *virginica*.
- condition. Factor giving a detailed description of the attribute (Four levels: `"Petal.Length"`, `"Petal.Width"`, `"Sepal.Length"`, `"Sepal.Width"`).
- attribute. What attribute is being measured (`"Sepal"` or `"Pepal"`).
- measure. What aspect of the attribute is being measured (`"Length"` or `"Width"`).
- value. Value of the measurement.

Details

This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica.

This is a modified dataset from `datasets` package.

Examples

```
dim(iris_long)
head(iris_long)
dplyr::glimpse(iris_long)
```

`movies_long` *Movie information and user ratings from IMDB.com (long format).*

Description

Movie information and user ratings from IMDB.com (long format).

Usage

```
movies_long
```

Format

A data frame with 1,579 rows and 8 variables

- title. Title of the movie.
- year. Year of release.
- budget. Total budget (if known) in US dollars
- length. Length in minutes.
- rating. Average IMDB user rating.
- votes. Number of IMDB users who rated this movie.
- mpaa. MPAA rating.
- genre. Different genres of movies (action, animation, comedy, drama, documentary, romance, short).

Details

Modified dataset from `ggplot2movies` package.

The internet movie database, https://imdb.com/, is a website devoted to collecting movie data supplied by studios and fans. It claims to be the biggest movie database on the web and is run by amazon.

Movies were are identical to those selected for inclusion in movies_wide but this dataset has been constructed such that every movie appears in one and only one genre category.

Source

https://CRAN.R-project.org/package=ggplot2movies

Examples

```
dim(movies_long)
head(movies_long)
dplyr::glimpse(movies_long)
```

`movies_wide` *Movie information and user ratings from IMDB.com (wide format).*

Description

Movie information and user ratings from IMDB.com (wide format).

Usage

```
movies_wide
```

Format

A data frame with 1,579 rows and 13 variables

- title. Title of the movie.
- year. Year of release.
- budget. Total budget in millions of US dollars
- length. Length in minutes.
- rating. Average IMDB user rating.
- votes. Number of IMDB users who rated this movie.
- mpaa. MPAA rating.
- action, animation, comedy, drama, documentary, romance, short. Binary variables representing if movie was classified as belonging to that genre.
- NumGenre. The number of different genres a film was classified in an integer between one and four

Details

Modified dataset from `ggplot2movies` package.

The internet movie database, https://imdb.com/, is a website devoted to collecting movie data supplied by studios and fans. It claims to be the biggest movie database on the web and is run by amazon.

Movies were selected for inclusion if they had a known length and had been rated by at least one imdb user. Small categories such as documentaries and NC-17 movies were removed.

Source

https://CRAN.R-project.org/package=ggplot2movies

Examples

```
dim(movies_wide)
head(movies_wide)
dplyr::glimpse(movies_wide)
```

theme_ggstatsplot *Default theme used in all* ggstatsplot *package plots*

Description

Common theme used across all plots generated in ggstatsplot and *assumed* by the author to be aesthetically pleasing to the user/reader.

Usage

```
theme_ggstatsplot(ggtheme = ggplot2::theme_bw(), ggstatsplot.layer = TRUE)

theme_corrmat()

theme_pie(ggtheme = ggplot2::theme_bw(), ggstatsplot.layer = TRUE)
```

Arguments

ggtheme	A function, ggplot2 theme name. Default value is ggplot2::theme_bw(). Any of the ggplot2 themes, or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.).
ggstatsplot.layer	Logical that decides whether theme_ggstatsplot theme elements are to be displayed along with the selected ggtheme (Default: TRUE). theme_ggstatsplot is an opinionated theme layer that override some aspects of the selected ggtheme.

Value

A ggplot2 object with the theme_ggstatsplot theme overlaid.

References

https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/theme_ggstatsplot.html

Titanic_full *Titanic dataset.*

Description

Titanic dataset.

Usage

```
Titanic_full
```

Format

A data frame with 2201 rows and 5 variables

- id. Dummy identity number for each person.
- Class. 1st, 2nd, 3rd, Crew.
- Sex. Male, Female.
- Age. Child, Adult.
- Survived. No, Yes.

Details

This data set provides information on the fate of passengers on the fatal maiden voyage of the ocean liner 'Titanic', summarized according to economic status (class), sex, age and survival.

This is a modified dataset from `datasets` package.

Examples

```
dim(Titanic_full)
head(Titanic_full)
dplyr::glimpse(Titanic_full)
```

VR_dilemma *Virtual reality moral dilemmas.*

Description

Virtual reality moral dilemmas.

Usage

```
VR_dilemma
```

Format

A data frame with 68 rows and 4 variables

- id. Dummy identity number for each participant.
- order. The order in which the participants completed the two sessions: `"text_first"` (`0`) or `"text_second"` (`1`).
- modality. Describes how the moral dilemmas were presented to the participants: either in text format (`"text"`) or in Virtual Reality (`"vr"`).
- score. Proportion of "utilitarian" decisions. In other words, of the 4 decisions, how many affirmative were responses. Range: 0 (all utilitarian) - 1 (none utilitarian).

Details

Dataset from a study where participants completed identical moral dilemmas in two different sessions held on separate days: in one session, they read text description of the scenario, while in another session they completed the same scenarios in Virtual Reality (videos: https://www.youtube.com/watch?v=ebdU3HhhYs8). The study investigated if there was a discrepancy between how people judged the same scenarios while reading them in text versus experiencing them in virtual reality.

Source

https://psyarxiv.com/ry3ap/

Examples

```
dim(VR_dilemma)
head(VR_dilemma)
dplyr::glimpse(VR_dilemma)
```

Index